



矩阵计算

李宇峰

liyf@nju.edu.cn

人工智能学院



7. Vandermonde矩阵与 Fourier矩阵

在工程应用中经常会遇到每行元素组成一个等比序列的特殊矩阵，这就是Vandermonde矩阵。Vandermonde矩阵的形式是：

也可以称为等比序列

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{N-1} \end{bmatrix}_{n \times N}$$

A^T 也是Vandermonde矩阵，当 A 的 n 个参数 x_1, x_2, \dots, x_n 都不相同时，Vandermonde矩阵是行满秩的。

Vandermonde矩阵的行列式

*Vandermonde*矩阵的行列式，从阶为2, 3开始

$$\begin{aligned}\det A_3 &= \begin{vmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{vmatrix} \\&= \begin{vmatrix} 1 & x_1 & x_1^2 \\ 0 & x_2 - x_1 & x_2^2 - x_1^2 \\ 0 & x_3 - x_1 & x_3^2 - x_1^2 \end{vmatrix} \\&= \begin{vmatrix} x_2 - x_1 & x_2^2 - x_1^2 \\ x_3 - x_1 & x_3^2 - x_1^2 \end{vmatrix} \\&= (x_2 - x_1)(x_3 - x_2)(x_3 - x_1).\end{aligned}$$

大胆猜测：

$$\det A_n = \prod_{1 \leq j < i \leq n} (x_i - x_j)$$

数学归纳法

Vandermonde矩阵的行列式

证明：假设 k 阶Vandermonde行列式为

$$\det A_k = \prod_{1 \leq j < i \leq k} (x_i - x_j)$$

我们要证明 $k+1$ 阶Vandermonde行列式也有相同的形式。将 A_{k+1} 的最末列替换为 $1, x, x^2, \dots, x^k$ ，

设此矩阵的行列式为 $f(x)$, 亦即

$$f(x) = \begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^k \\ 1 & x & x^2 & \cdots & x^k \end{vmatrix}.$$

由最末列的余因子展开式可知 f 为变数 x 的 k 次多项式。若矩阵有相同的两列，其行列式等于零，故

$f(x_i) = 0$, $i = 1, 2, \dots, k$, 也就是说 x_i , $i = 1, 2, \dots, k$, 为多项式 $f(x)$ 的 k 个根, $f(x)$ 可表示为

$$f(x) = \alpha(x - x_1)(x - x_2) \cdots (x - x_k)$$

上式中 α 为非零常数，剩下的问题是决定 α 。

Vandermonde矩阵的行列式

由 $f(x)$ 的余因子展开式可得到 x^k 的系数，也就是 α 的值，

$$f(x_{k+1}) = \det A_{k+1} \quad \checkmark$$

注意，上面的行列式即为 $\det A_k$ 。根据归纳法的假设，就有

$$\begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{k-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^{k-1} \end{vmatrix} \quad \text{行} \quad | \leftarrow \text{列} \quad x^k.$$

还有一个不能错过的事实： $f(x_{k+1}) = \det A_{k+1}$ 。将 $x = x_{k+1}$ 代入上式，

$$\det A_{k+1} = \left[\prod_{1 \leq j < i \leq k} (x_i - x_j) \right] (x_{k+1} - x_1)(x_{k+1} - x_2) \cdots (x_{k+1} - x_k).$$

整理等号右端，得到

$$\det A_{k+1} = \prod_{1 \leq j < i \leq k+1} (x_i - x_j),$$

故证明所求。

Vandermonde矩阵

n个资料点

Vandermonde矩阵常见于数值分析的内插(interpolation)问题。给出 n个资料点 (x_i, y_i) , $i = 1, 2, \dots, n$, 求 $(n - 1)$ 次多项式

$$p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0$$

满足

$$p(x_1) = a_0 + a_1x_1 + \dots + a_{n-1}x_1^{n-1} = y_1$$

$$p(x_2) = a_0 + a_1x_2 + \dots + a_{n-1}x_2^{n-1} = y_2$$

⋮

$$p(x_n) = a_0 + a_1x_n + \dots + a_{n-1}x_n^{n-1} = y_n.$$

将上面的线性方程组写为矩阵形式 $A\mathbf{a} = \mathbf{y}$, 其中 A 为 $n \times n$ 阶 Vandermonde 矩阵。内插问题就是要解出系数向量 $\mathbf{a} = [a_0 \ a_1 \ \dots \ a_{n-1}]^T$ 。
如果 n 个参数 x_1, x_2, \dots, x_n 彼此相异, 推知 $\det A_n \neq 0$, A 是可逆的, 方程式必定存在唯一解。

Vandermonde矩阵

例2.1(扩展Prony方法) 在谐波恢复的扩展Prony方法中，信号模型假定是一组 p 个指数函数的叠加，这组指数函数有任意的幅值、相位、频率和阻尼因子。于是离散时间的数学模型是

$$\widehat{x_n} = \sum_{i=0}^p b_i z_i^n, \quad n = 0, 1, 2, \dots, N-1 \quad (2.1)$$

被用作拟合观测数据 x_1, x_2, \dots, x_{N-1} 的数学模型。通常假定 b_i 和 z_i 为复数，并且 $b_i = A_i \exp(j\theta_i)$ 。

$$b_i = A_i \exp(j\theta_i), \quad z_i = \exp[(\alpha_i + j2\pi f_i)\Delta t]$$

其中 A_i 是幅值、 θ_i 是相位(rad)、 α_i 为阻尼因子、 f_i 为振荡频率(Hz)、 Δt 代表采样间隔(s)。

Vandermonde矩阵

(2.1)式的矩阵形式是 $\Phi \mathbf{b} = \hat{\mathbf{x}}$, 这里 $\mathbf{b} = (b_1, b_2, \dots, b_p)^T$, $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{N-1})^T$, 而 Φ 是一复Vandermonde矩阵

$$\Phi = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ z_1 & z_2 & z_3 & \cdots & z_p \\ z_1^2 & z_2^2 & z_3^2 & \cdots & z_p^2 \\ \vdots & \vdots & \vdots & & \vdots \\ z_1^{N-1} & z_2^{N-1} & z_3^{N-1} & \cdots & z_p^{N-1} \end{bmatrix}$$

使平方误差 $\varepsilon = \sum_{n=0}^{N-1} |x_n - \hat{x}_n|^2$ 最小, 便得到最小二乘解

$$\mathbf{b} = (\Phi^H \Phi)^{-1} \Phi^H \hat{\mathbf{x}}$$

Vandermonde矩阵

容易证明，上式中的 $\Phi^H \Phi$ 的计算可以大大简化，使得无须作
*Vandermonde*矩阵的乘法运算，就能直接利用

$$\Phi^H \Phi = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1p} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2p} \\ \vdots & \vdots & & \vdots \\ \gamma_{p1} & \gamma_{p2} & \cdots & \gamma_{pp} \end{bmatrix}$$

计算出 $\Phi^H \Phi$ 其中

$$\gamma_{ij} = \frac{(z_i^* z_j)^N - 1}{z_i^* z_j - 1}$$

Applications: Reed-Solomon Codes (1960)

In the original construction of Reed & Solomon(1960), the message $\underline{\mathbf{x}} = (x_1, \dots, x_k) \in \underline{\mathbf{F}}^k$ is mapped to the polynomial \mathbf{p}_x with

$$\mathbf{p}_x(a) = \sum_{i=1}^k x_i a^{i-1}.$$

As described above, the codeword is then obtained by evaluating p at n different points a_1, \dots, a_n of the field \mathbf{F} . Thus the classical encoding function $C: \underline{\mathbf{F}}^k \rightarrow \underline{\mathbf{F}}^n$ for the Reed-Solomon code is defined as follows:

$$C(\underline{\mathbf{x}}) = (\mathbf{p}_x(a_1), \dots, \mathbf{p}_x(a_n)).$$

This function C is a linear mapping, that is, it satisfies $C(\underline{\mathbf{x}}) = \mathbf{x} \cdot \mathbf{A}$ for the following $(k \times n)$ -matrix A with elements from \mathbf{F} :

$$A = \begin{bmatrix} 1 & \dots & 1 \\ a_1 & \dots & a_n \\ a_1^2 & \dots & a_n^2 \\ \vdots & \ddots & \vdots \\ a_n^{k-1} & \dots & a_n^{k-1} \end{bmatrix}$$

This matrix is the transpose of a Vandermonde matrix over \mathbf{F} . In other words, the Reed-Solomon code is a linear code, and in the classical encoding procedure, its generator matrix is A .

Applications: Reed-Solomon Codes (1960)

- Reed-Solomon Codes are widely used in media data storage, such as CD and DVD
- Reed-Solomon Codes are used in nearly all two-dimensional bar codes, such as QR code.

Fourier 矩阵

离散时间信号 x_0, x_1, \dots, x_{N-1} 的 Fourier 变换 称为信号的离散 Fourier 变换 (DFT) 或者频谱，定义为

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi nk/N} = \sum_{n=0}^{N-1} x_n \omega^{nk},$$
$$k = 0, 1, \dots, N - 1$$

式中 $\omega = e^{-j2\pi/N}$ 。离散 Fourier 变换的矩阵表示形式为

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{N-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

或简记为 $\bar{x} = Fx$

Fourier 矩阵

$$\bar{x} = Fx$$

式中 $x = (x_0, x_1, \dots, x_{N-1})^T$ 和 $\bar{x} = (X_0, X_1, \dots, X_{N-1})^T$ 分别是离散时间信号向量和频谱向量，而

$$F = \omega^{-\frac{1}{2}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{N-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

称为 *Fourier* 矩阵，其 (i, k) 元素为 $F(i, k) = \omega^{(i-1)(k-1)}$ 。

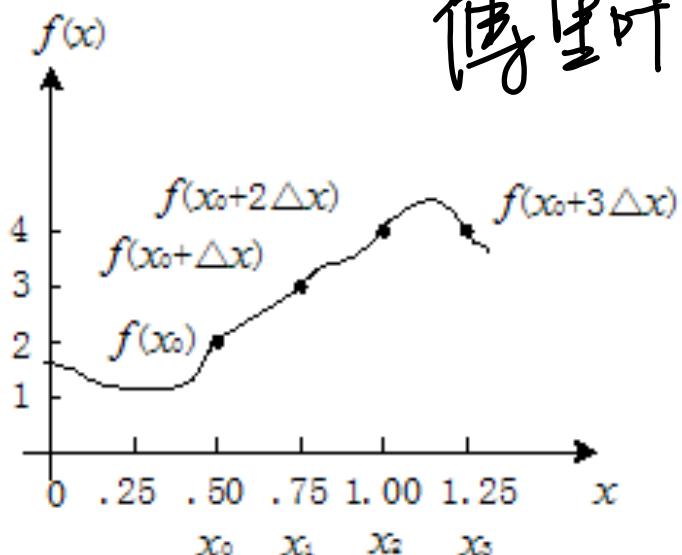
显然，*Fourier* 矩阵是 *Vandermonde* 矩阵，具有特殊结构：每一行和每一列的元素都分别组成各自的等比数列。



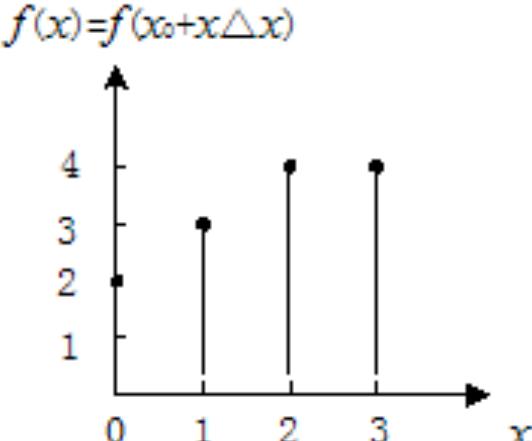
例 设一函数如图 (a) 所示, 如果将此函数在自变量 $x_0 = 0.5, x_1 = 0.75, x_2 = 1.00, x_3 = 1.25$ 取样。

并重新定义为图 (b) 离散函数, 求其傅里叶变换。

傅里叶变换



(a)



(b)



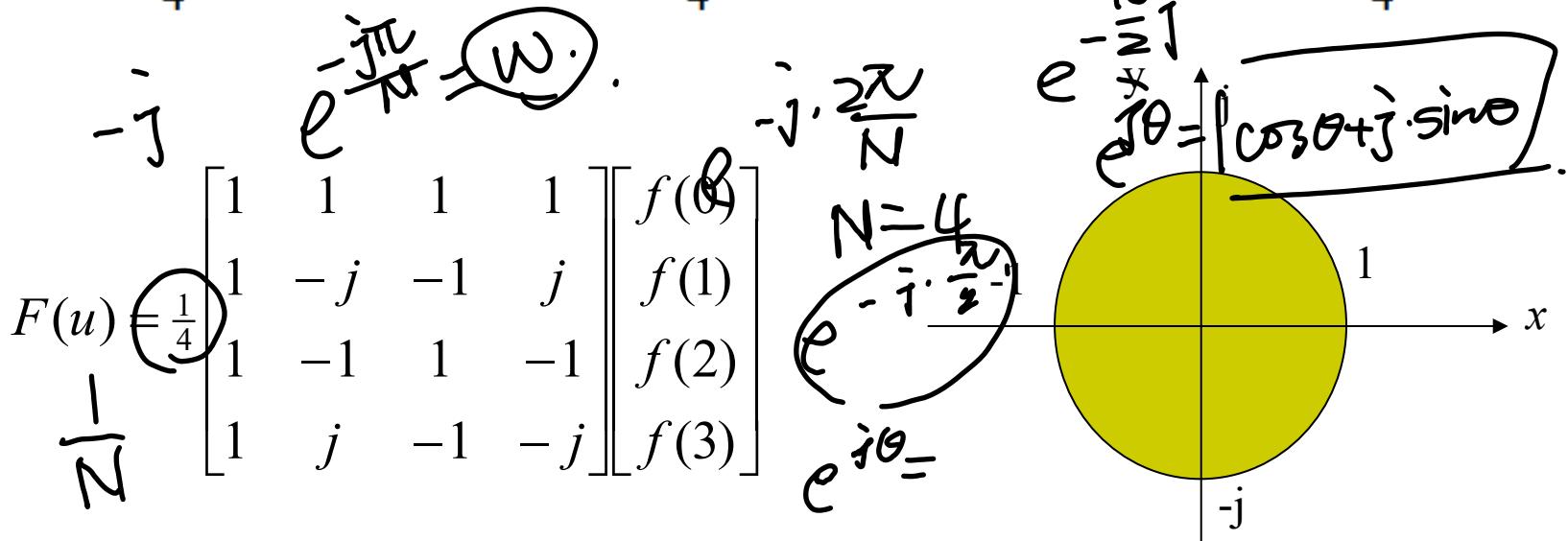
$$F(0) = \frac{1}{4} \sum_{x=0}^3 f(x) \exp\{0\}$$

$$= (1/4)[f(0) + f(1) + f(2) + f(3)] = (1/4)[2 + 3 + 4 + 4] = 3.25$$

$$F(1) = \frac{1}{4} \sum_{x=0}^3 f(x) \exp\{-j2\pi x/4\} = \frac{1}{4} [2e^0 + 3e^{-j\pi/2} + 4e^{-j\pi} + 4e^{-j3\pi/2}] = \frac{1}{4} [-2 + j]$$

$$F(2) = \frac{1}{4} \sum_{x=0}^3 f(x) \exp\{-j4\pi x/N\} = \frac{1}{4} [2e^0 + 3e^{-j\pi} + 4e^{-j2\pi} + 4e^{-j3\pi}] = -\frac{1}{4} [1 + j0]$$

$$F(3) = \frac{1}{4} \sum_{x=0}^3 f(x) \exp\{-j6\pi x/4\} = \frac{1}{4} [2e^0 + 3e^{-j3\pi/2} + 4e^{-j3\pi} + 4e^{-j9\pi/2}] = -\frac{1}{4} [2 + j]$$



Fourier 矩阵

同时有 $\bar{x} = Fx$ 可以看出，一个 离散时间信号向量的离散 Fourier 变换可以用矩阵 F 表示，因此称 F 为 Fourier 矩阵。根据定义可以验证 $FF^H = F^H F = NI_n$ 。 F 是一个 $Vandermonde$ 矩阵，它是非奇异的，于是由 $F^H F = NI_n$ 可知 Fourier 矩阵的逆矩阵是

$$F^{-1} = \frac{1}{N} F^H = \frac{1}{N} F^*,$$

共轭.

$$e^{-\frac{jw}{N} \cdot (nk)}.$$

Fourier 矩阵

因此 $x = F^{-1}\bar{x} = \frac{1}{N}F^*\bar{x}$

或写成 $\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega^* & \cdots & (\omega^{N-1})^* \\ \vdots & \vdots & & \vdots \\ 1 & (\omega^{N-1})^* & \cdots & (\omega^{(N-1)(N-1)})^* \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{bmatrix}$

即有

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j2\pi nk/N}, \quad n = 0, 1, \dots, N-1$$

这恰好是离散 Fourier 逆变换的公式。

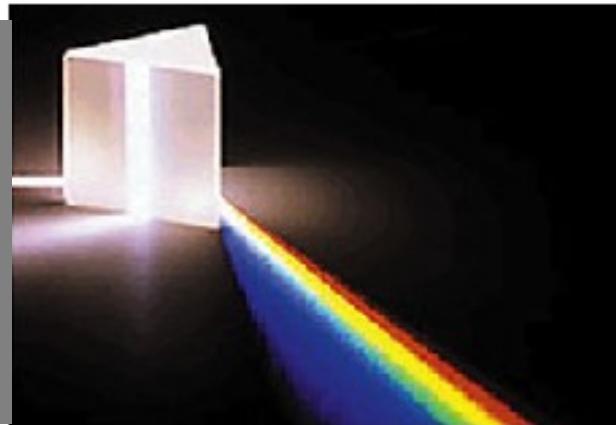


傅里叶变换的直观意义

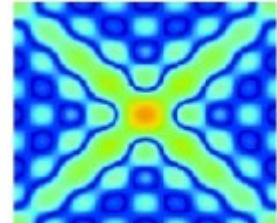
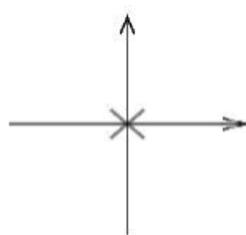
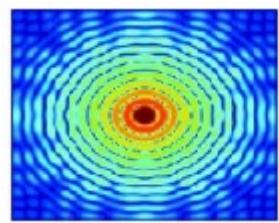
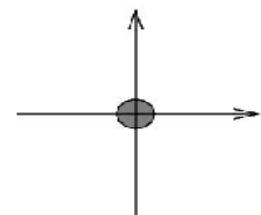
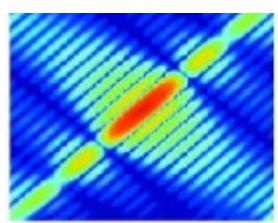
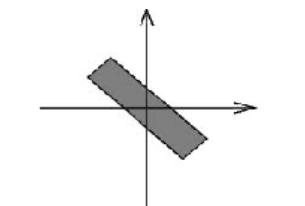
傅里叶变换好比一个玻璃棱镜

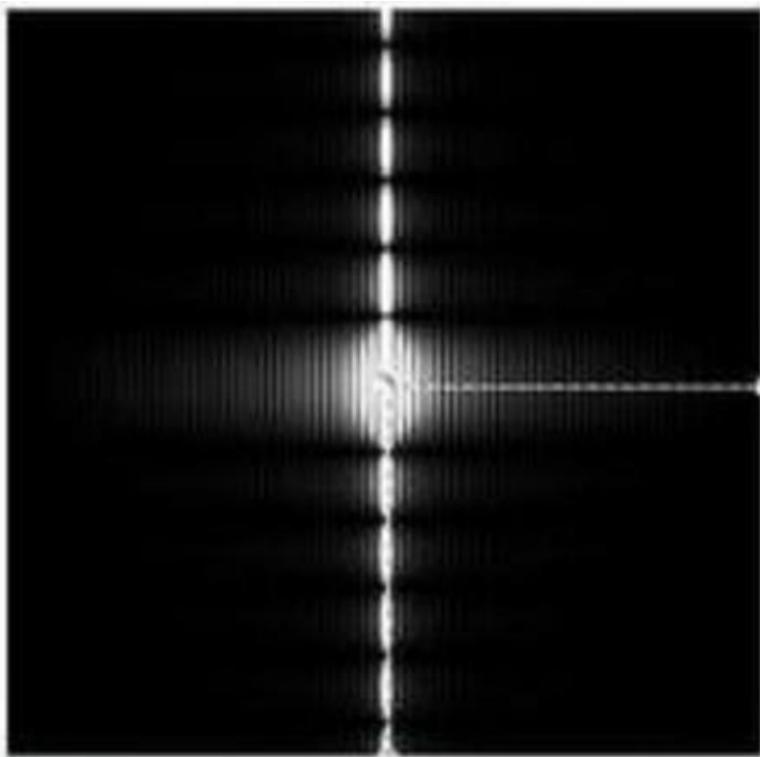
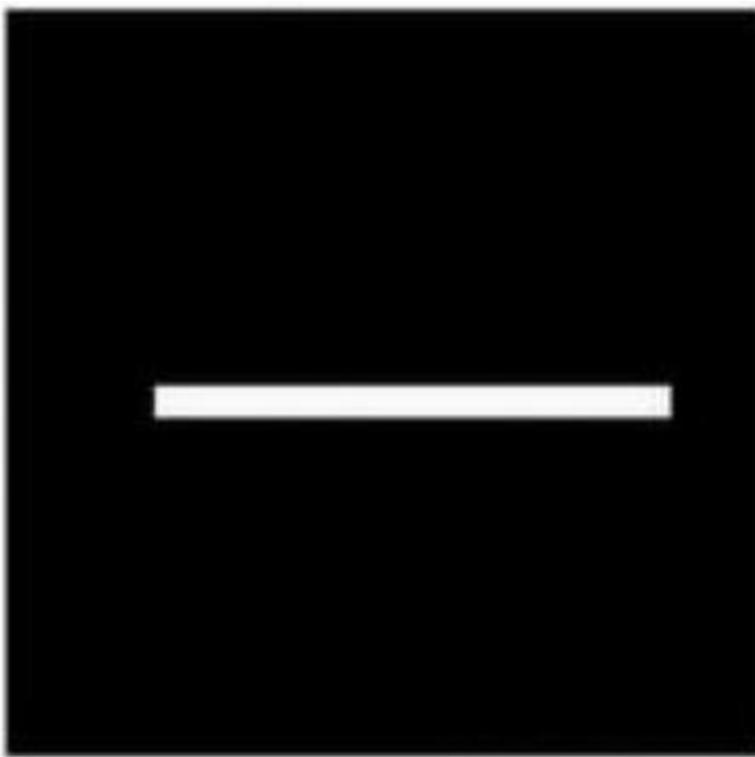
棱镜是可以将光分成不同颜色的物理仪器，每个成分的颜色由波长决定。

傅里叶变换可看做是“数学中的棱镜”，将函数基于频率分成不同的成分。

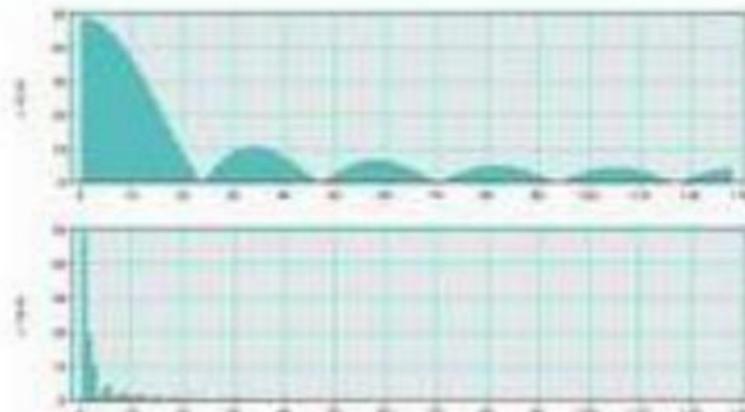


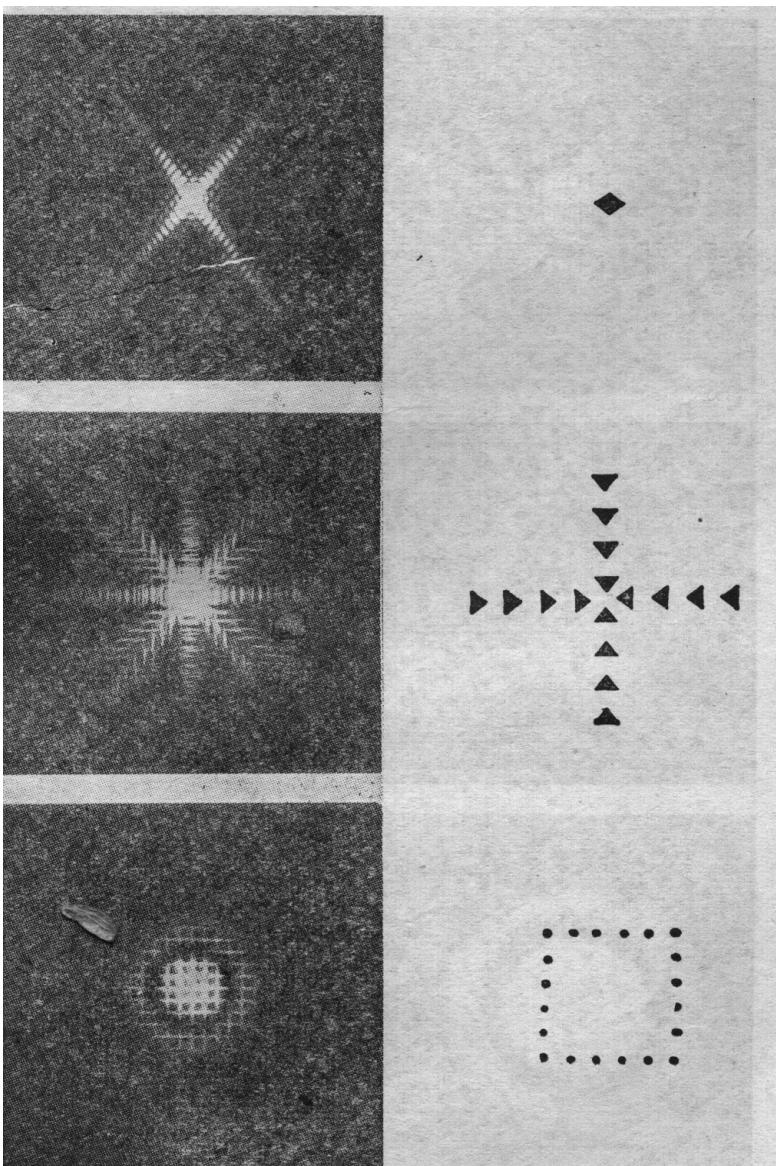
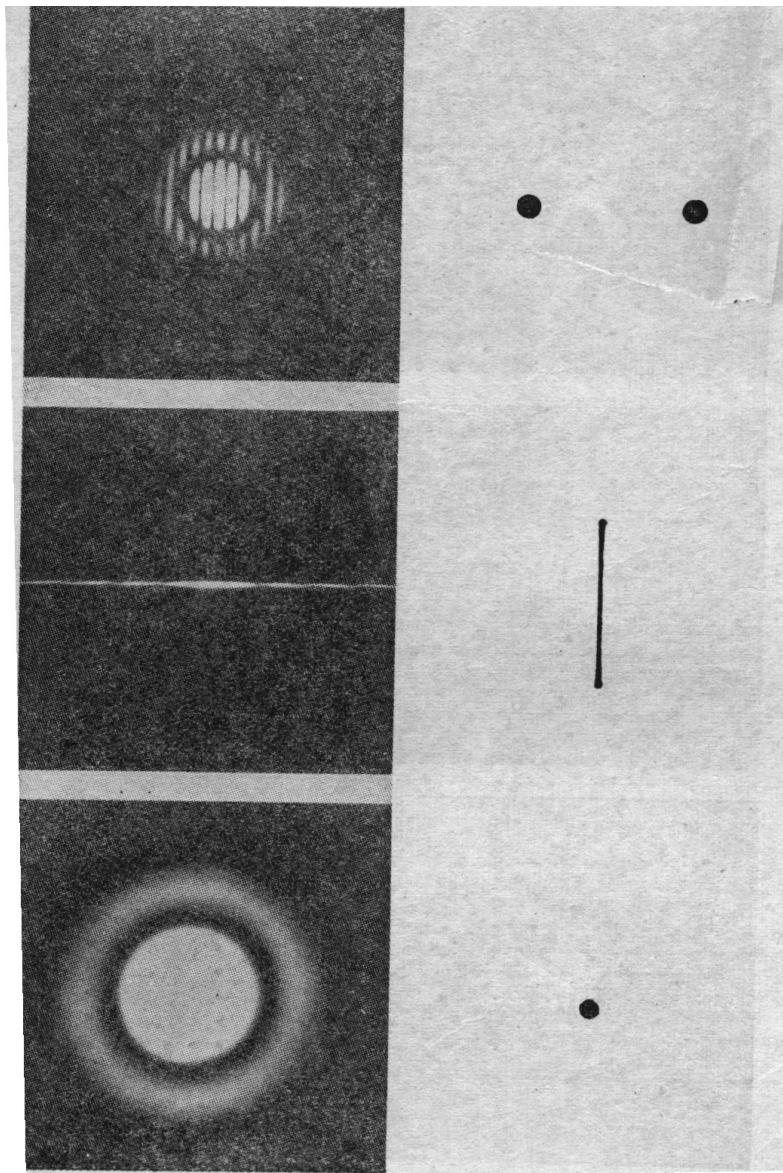
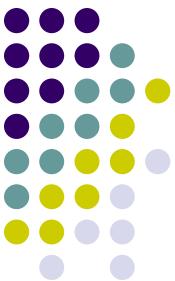
一些图像的傅里叶变换



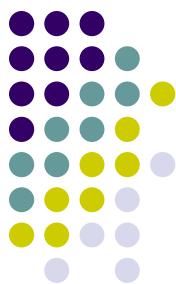


在垂直方向上需要更多的频率分量，所以它的波峰比较宽，而水平方向上的波峰比较窄



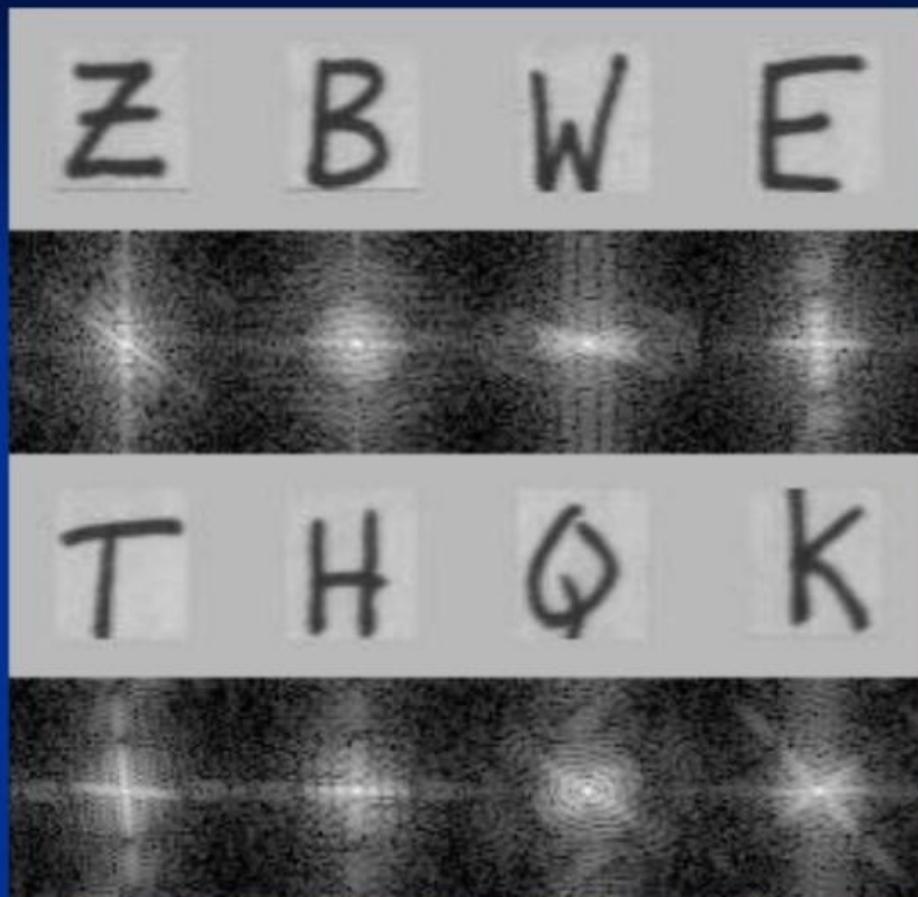


有趣的幅度谱

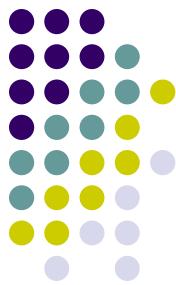


幅度谱和相位谱

- 从幅度谱中我们可以看出明亮线和原始图像中对应的轮廓线是垂直的。如果原始图像中有圆形区域那么幅度谱中也呈圆形分布

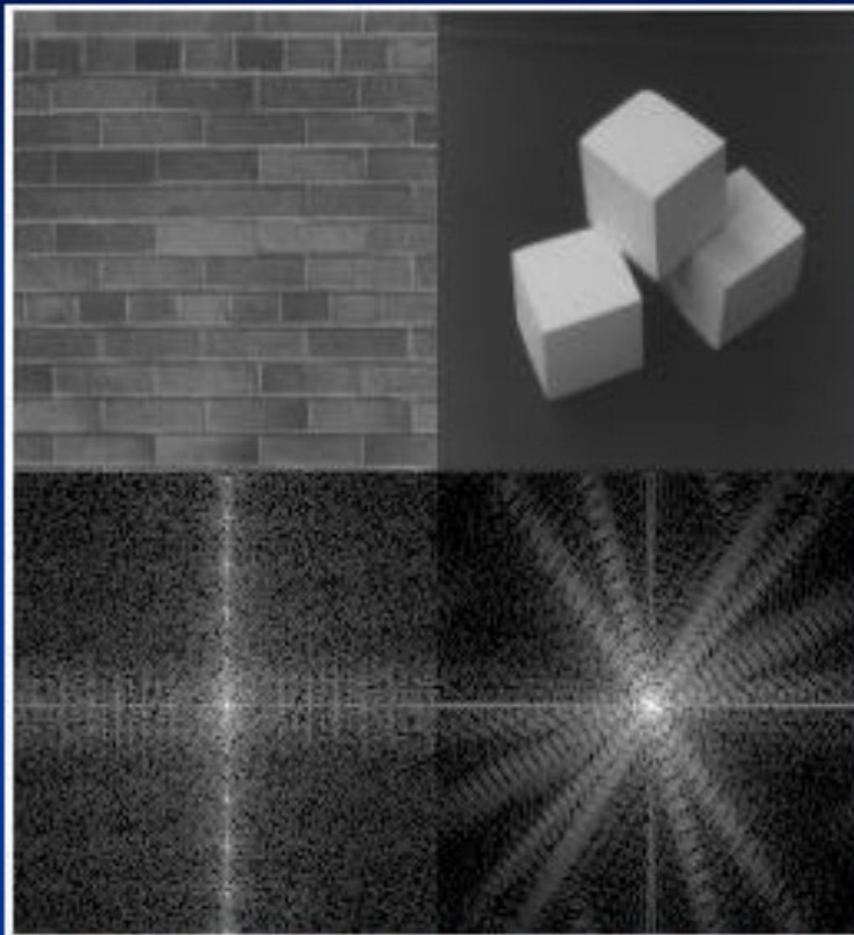


有趣的幅度谱



幅度谱和相位谱

- 从幅度谱中我们可以看出明亮线反映出原始图像的灰度级变化，这正是图像的轮廓边

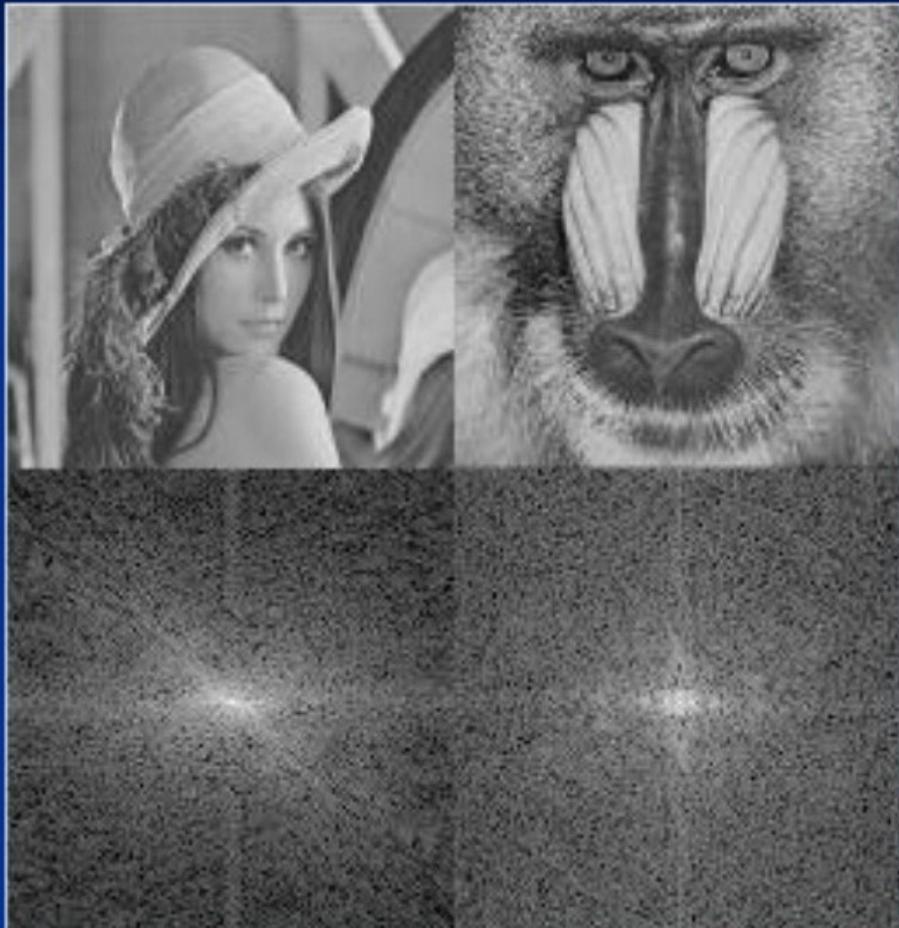


有趣的幅度谱



幅度谱和相位谱

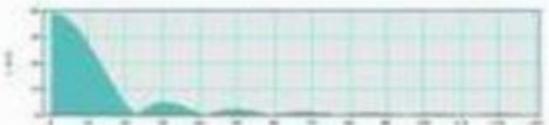
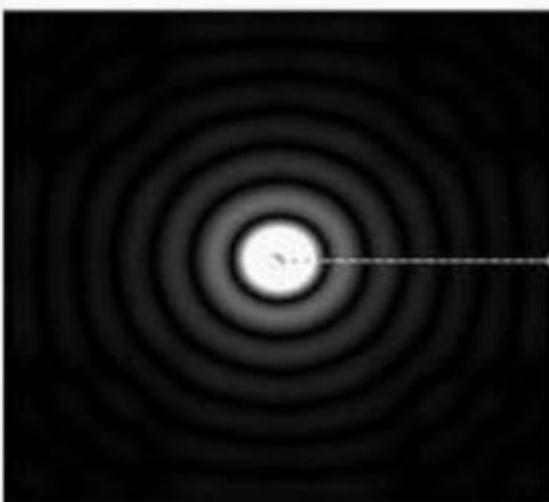
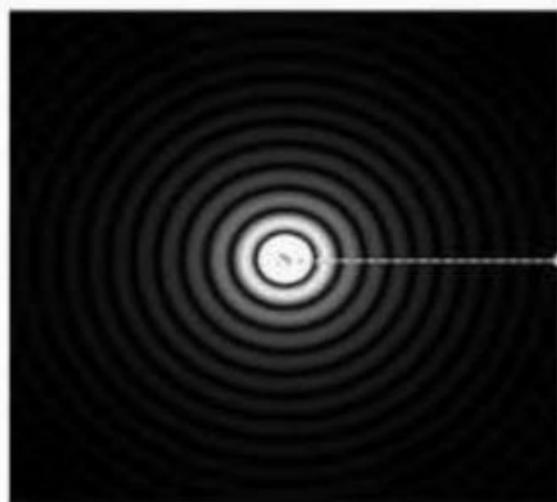
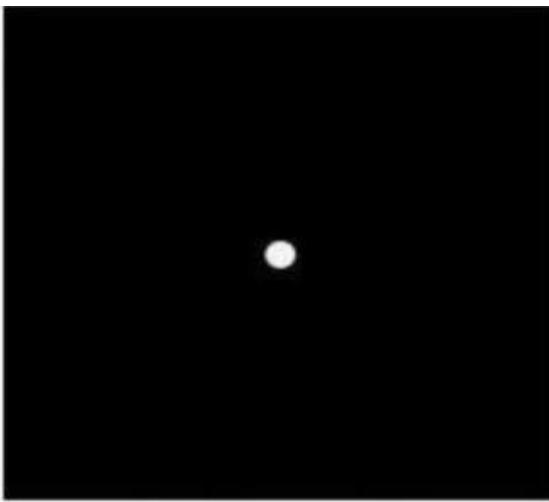
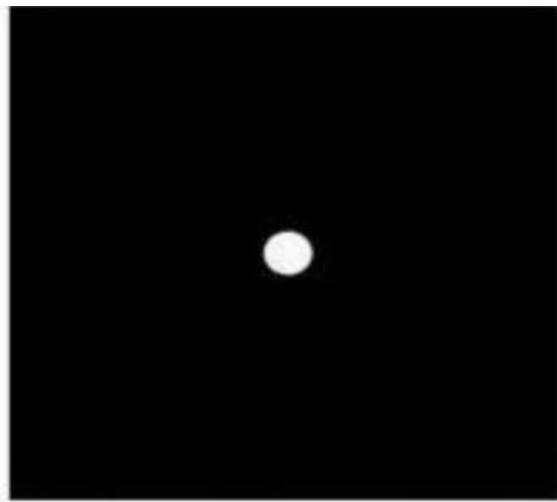
- 这些图像没有特定的结构，左上角到右下角有一条斜线，它可能是由帽子和头发之间的边线产生的
- 两个图像都存在一些小边界





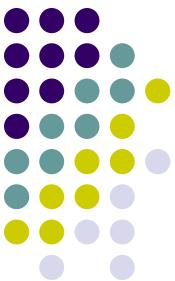
频谱图中暗的点数更多，那么实际图像是比较柔和的，反之，如果频谱图中亮的点数多，那么实际图像一定是尖锐的，边界分明且边界两边像素差异较大的



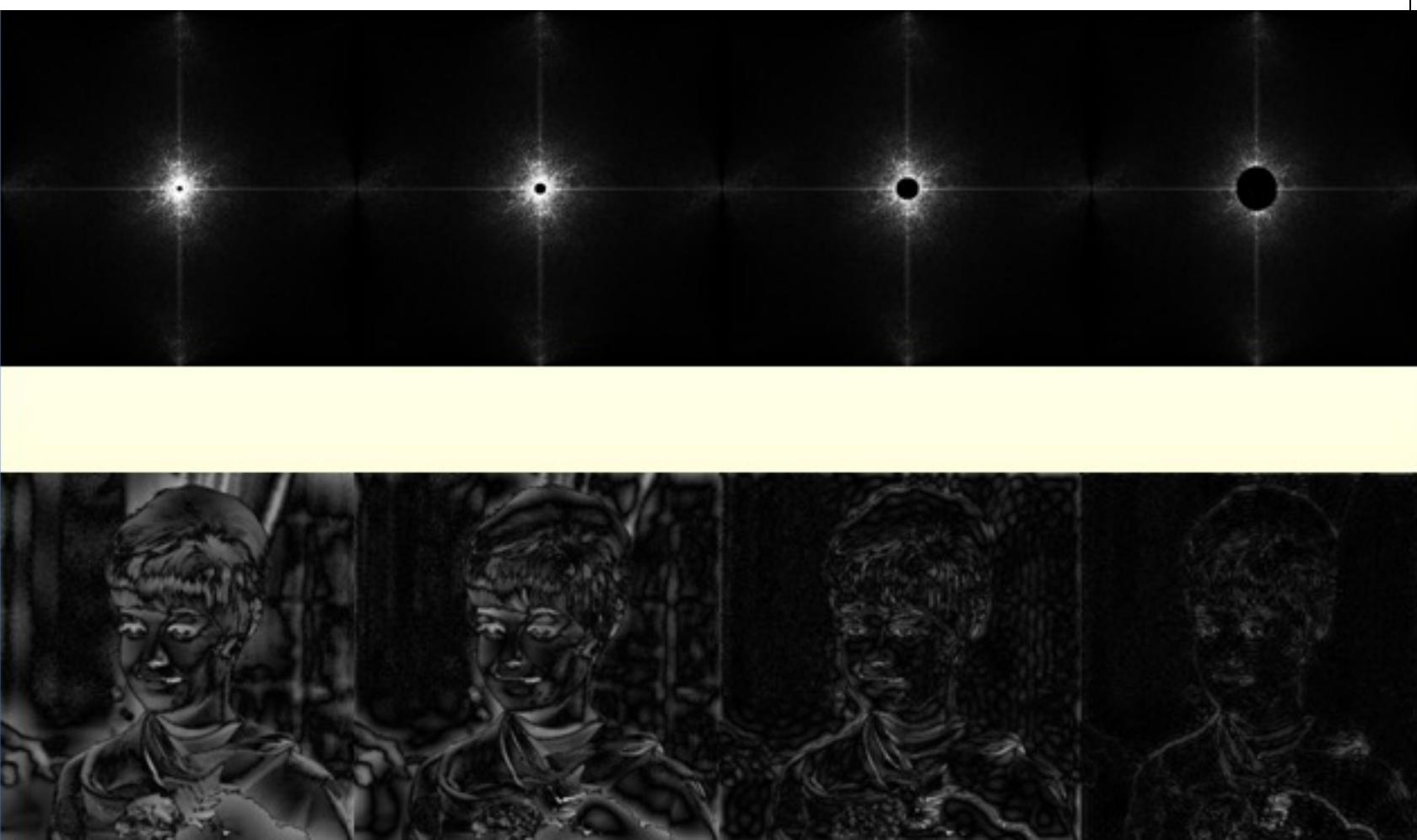


可以看到大圆经傅立叶变换之后，其圆环小；而小圆经傅立叶变换之后，其圆环反而大。

因为越尖锐变换越剧烈的信号总包含着更多的频率成分。



高频部分（距离中心较远区域）越多，图像细节越丰富。反之图像越平滑，显得模糊



高频部分（距离中心较远区域）对应于图像的细节（边缘，线条等）。低频部分（中心部分）对应于图像的整体信息。



傅里叶变换的作用

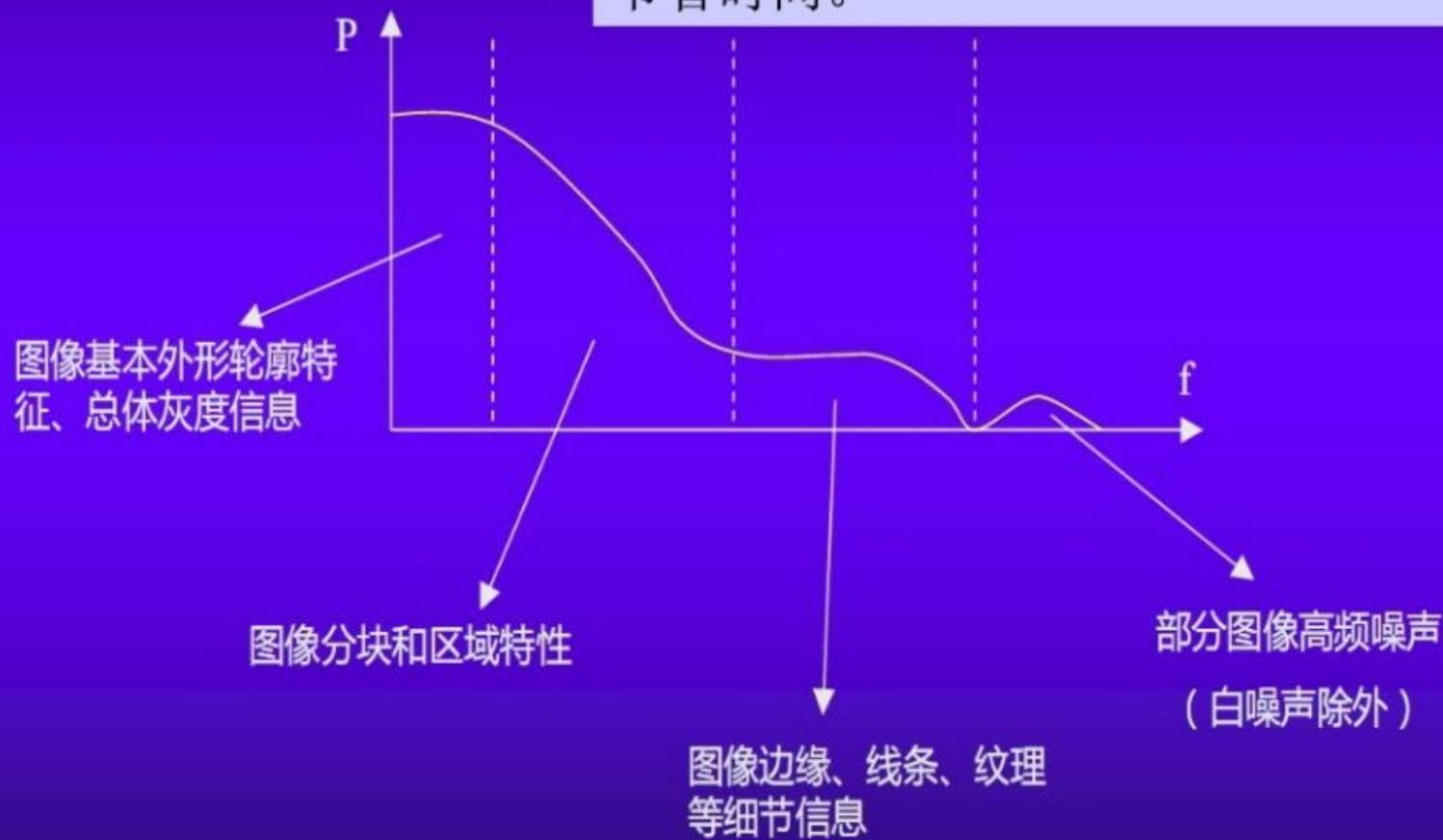
- * 傅里叶变换将信号分成不同频率成份。类似光学中的分色棱镜把白光按波长(频率)分成不同颜色，称数学棱镜。
- * 傅里叶变换的成份： 直流分量和交流分量
- * 信号变化的快慢与频率域的频率有关。噪声、边缘、跳跃部分代表图像的高频分量；背景区域和慢变部分代表图像的低频分量

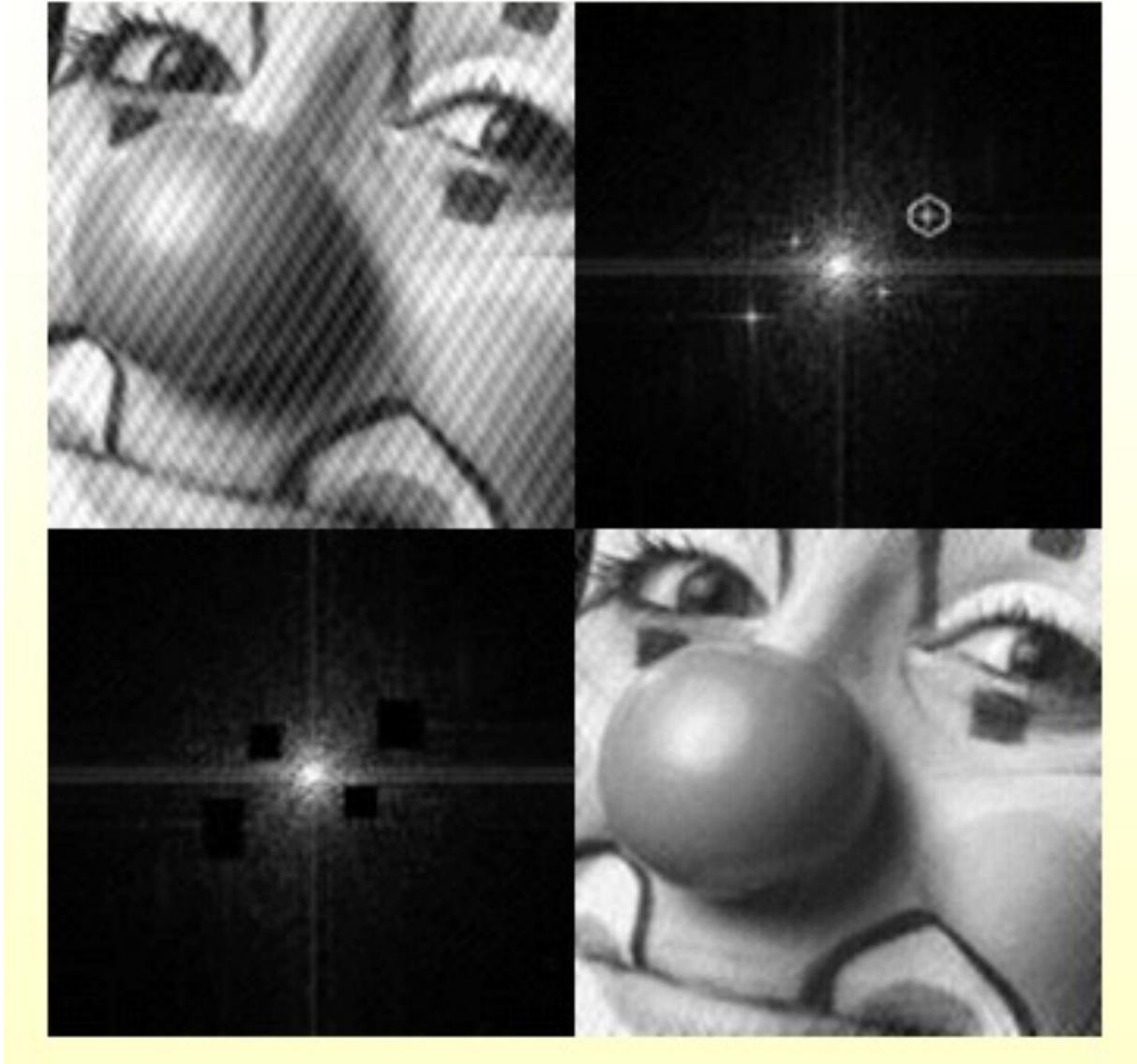


为什么要做图像变换？

图像信号的频域模型

◆ 变换后的图象，大部分**能量**都分布于**低频谱段**，这对以后图象的**压缩、传输**都比较有利。使得运算次数减少，节省时间。





通过去除图像频域谱的噪声点，增强图像



8. Hadamard 矩阵

*Hadamard*矩阵在通信、信息论和信号处理中都有重要应用。

$H_n \in R^{n \times n}$ 称为 *Hadamard* 矩阵是指该矩阵的所有元素取+1或者取-1，且满足

$$H_n H_n^T = H_n^T H_n = nI_n$$

*Hadamard*矩阵具有以下基本性质：

- (1) 只有当 $n=2$ 或者 n 是 4 的整数倍时，*Hadamard*矩阵才存在；
- (2) $\frac{1}{\sqrt{n}} H_n$ 为 标准正交矩阵，即满足

$$\left(\frac{1}{\sqrt{n}} H_n\right)^T \left(\frac{1}{\sqrt{n}} H_n\right) = \left(\frac{1}{\sqrt{n}} H_n\right) \left(\frac{1}{\sqrt{n}} H_n\right)^T = I$$

8. Hadamard 矩阵

(3) $n \times n$ 的 *Hadamard* 矩阵 H_n 的行列式 $\det(H_n) = \pm n^{\frac{n}{2}}$

可以看出，用 -1 乘 *Hadamard* 矩阵的任一行或者任一列，得到的结果仍然是一个 *Hadamard* 矩阵。于是得到第 1 列和第 1 行所有元素为 $+1$ 的 *Hadamard* 矩阵，这称为 规范化 *Hadamard* 矩阵。

令 $n = 2^k, k = 1, 2, \dots$ ，则 规范化的 *Hadamard* 矩阵 具有通用的构造公式 (Sylvester's Construction)：

$$\bar{H}_{n/2} = \begin{bmatrix} \bar{H}_{n/2} & \bar{H}_{n/2} \\ \bar{H}_{n/2} & -\bar{H}_{n/2} \end{bmatrix} \quad \text{其中 } \bar{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Sylvester 构造矩阵的性质： \bar{H}_n 是对称的； $\text{trace}(\bar{H}_n) = 0$ ；除了第一行和第一列，其他行和列的正负元素数相等。

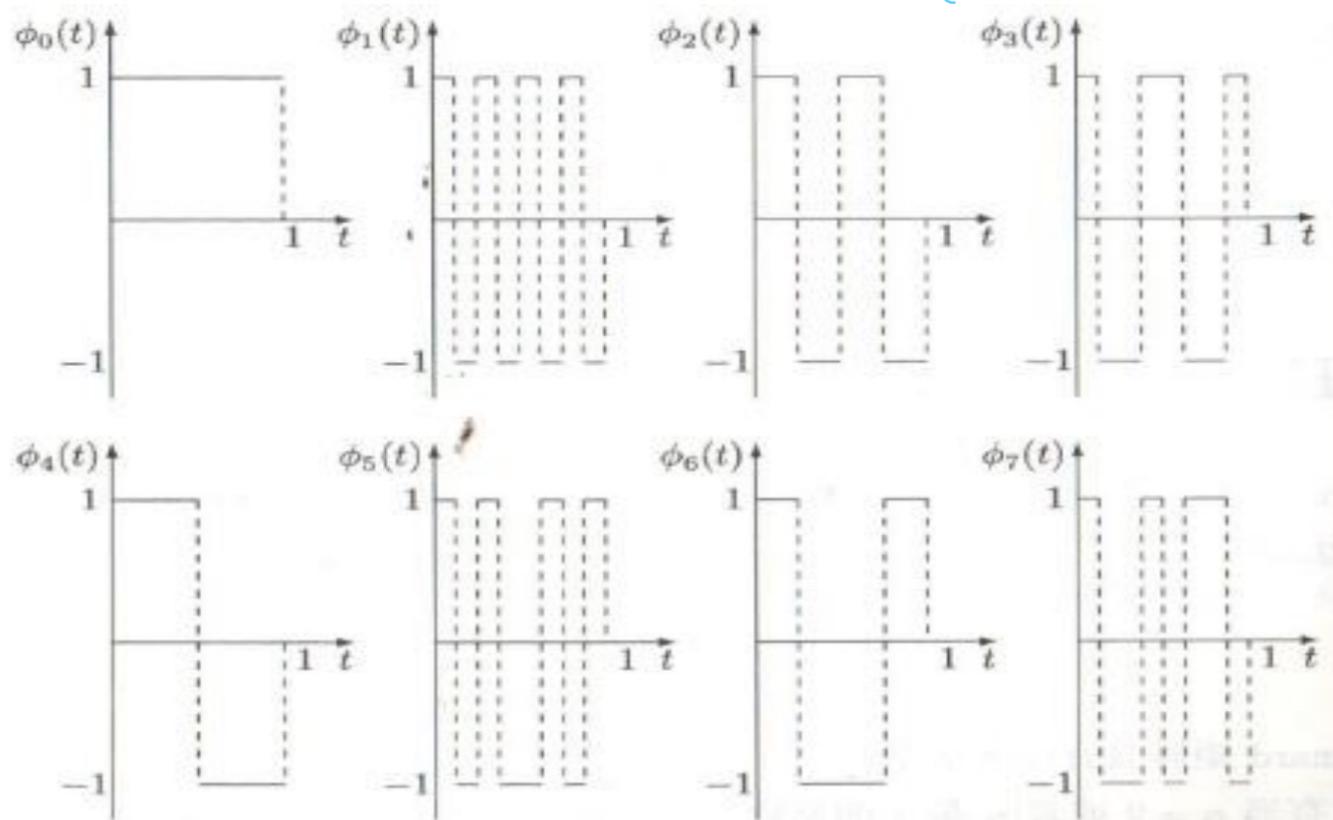
8. Hadamard 矩阵

例2.2 当 $n = 2^3 = 8$ 时，Hadamard矩阵为

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

8. Hadamard 矩阵

Hadamard矩阵为 H_8 的1~8行的波形函数 $\phi_0(t), \phi_1(t) \dots, \phi_7(t)$



Hadamard 矩阵每行的波形

8. Hadamard 矩阵

由上图容易看出， H_8 的8个矩形脉冲波形函数
 $\phi_0(t), \phi_1(t) \dots, \phi_7(t)$ 相互正交，即

$$\int_0^1 \phi_i(t) \phi_j(t) dt = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

使用Hadamard矩阵 H 的线性变换 $Y = HX$ 称为矩阵 X 的
Hadamard变换。

8. Hadamard 矩阵

由于Hadamard矩阵是正交矩阵，并且元素只取+1或-1，故Hadamard变换是唯一只使用加法和减法的正交变换。

Hadamard变换的典型应用有：

- (1) 在移动通信中用作编码，得到的码称为Hadamard码（或称Walsh-Hadamard码）
- (2) Hadamard矩阵的行向量可以用来仿真码分多址中各个用户的扩频波形向量。
- (3) 视频编码（比如H. 264），量子计算

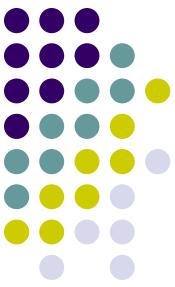
8. Hadamard 矩阵的应用

傅里叶变换提供了一个通用的数学变换。但它有一个假定：作用在连续图像上，而且结果可以取复数。

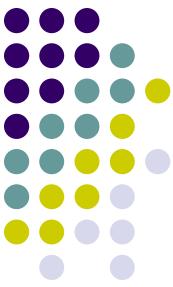
现实图像通常是离散的，另外复数计算对于计算机不好表示。

因此，需要专门了解离散图像上的变换。

背景知识



- 最简单的做法：线性变换
- 一维离散线性变换



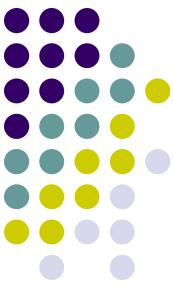
一维离散线性变换

假定图像表示为一个向量 x

线性变换定义：

$$y = Tx$$

Nx1的输出向量 Nx1的输入向量
核矩阵：大小NxN



一维离散线性变换

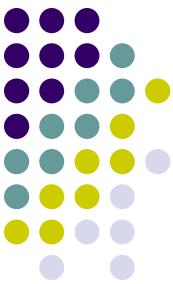
- 例子

平面坐标系中的向量旋转变换

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

2x1的输出向量 核矩阵: 大小2x2 2x1的输入向量

问题



在图像处理中，核矩阵要求满足非奇异性。
(为什么？)

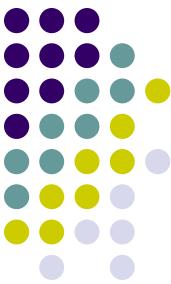
逆线性变换：

$$x = T^{-1}y$$

如果T奇异，那么T不可逆，则线性变换后，
图像无法复原



问题



- 可逆矩阵太多。什么样的可逆矩阵更实用？
- 两大类
 - 正交矩阵=》正交变换
 - 酉矩阵=》酉变换

正交矩阵

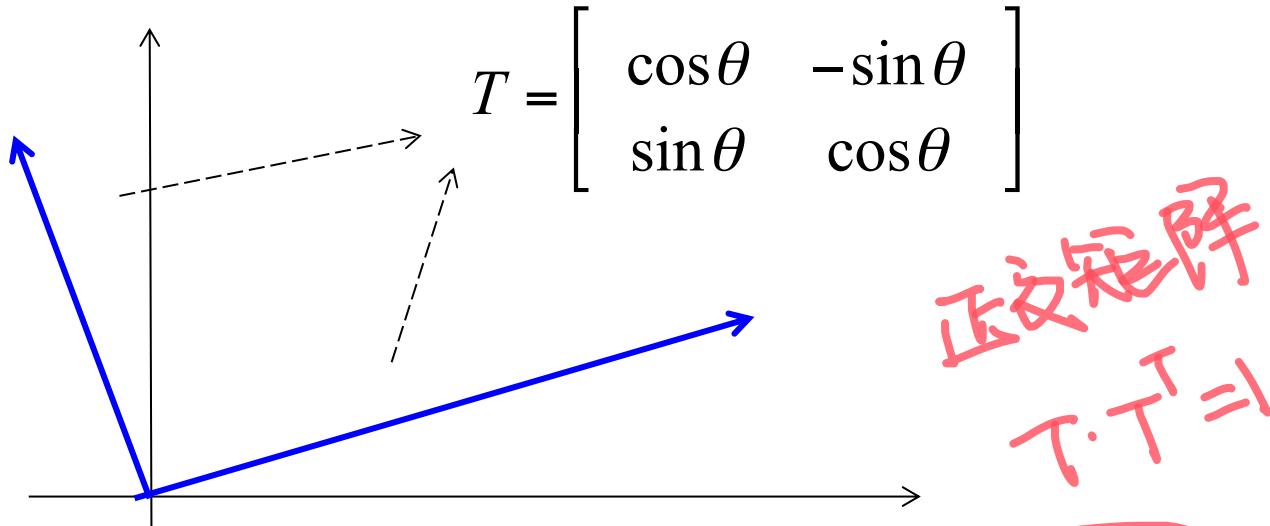


正交矩阵定义

$$T^{-1} = T', TT' = I \rightarrow \text{单位阵}$$

T' 表示T矩阵的转置

正交矩阵的例子



相当于将输入图像投影到一个N维正交空间。T的每一行对应于N维空间的正交基

请课后证明T为正交矩阵



正交矩阵的好处

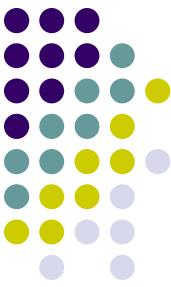
- 线性变换

$$y = Tx$$

- 逆变换

$$x = T^{-1}y = T'y$$

- 不涉及求逆计算，减少了计算开销



酉矩阵

酉矩阵

$$T^{-1} = (T^*)', T(T^*)' = I \rightarrow \text{单位阵}$$

~~T*表示矩阵T的共轭矩阵 (如果T为实数矩阵，
T*=T；如果T为复数矩阵，则T*将T矩阵的虚
数部分取相反数)~~

相当于将输入图像投影到一个N维复数正交空
间。T的每一行对应于N维复数空间的正交基



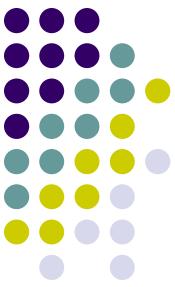
傅里叶变换与酉变换

- 离散傅里叶变换矩阵是一个酉矩阵

$$\tilde{F}_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} f_i \exp(-j2\pi k \frac{i}{N})$$

$$F = Wf, W_{i,k} = \frac{1}{\sqrt{N}} \exp(-j2\pi k \frac{i}{N})$$

- 怎么证明？



举例证明

- 当N=4时

$$T_4 = 0.5 \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$



证明过程

- 证明：一维离散傅立叶变换是酉变换。

举例 $N = 4$:

$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \quad T_4^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$
$$T_4 T_4^{**} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$
$$= \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

因此通常为了归一化，傅立叶变换时乘以系数 $\frac{1}{\sqrt{N}}$

问题



- 之前傅里叶变换都采用实数运算
- 但是对于计算机而言，整数（最好是采用1, 0, -1这样类型的整数）更方便计算。
- 能不能设计这样的变换？
 - 方波型变换



离散沃尔什变换 (Walsh)

基本想法：核矩阵中只有+1和-1元素，
要求 $N=2^p$ ，是对称的可分离的酉矩阵。

在 $N = 2^p$ 时， 定义一维离散沃尔什变换为

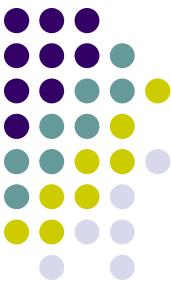
$$W(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \prod_{i=0}^{p-1} (-1)^{b_i(x)b_{p-1-i}(u)}$$

常数项也可取 $\sqrt{\frac{1}{N}}$

其中 $b_i(x)$ 是 x 的二进制表示的第 i 位值。

如 $p = 3$, $N = 2^p = 8$, $x = 6$ (二进制110)时,

$$b_0(6) = 0, \quad b_1(6) = 1, \quad b_2(6) = 1.$$



N=2,4,8时的b值

N值	N=2 p=1		N=4 p=2				N=8 p=3							
x值	0	1	0	1	2	3	0	1	2	3	4	5	6	7
x二进制	0	1	00	01	10	11	00 0	0 0	01 0	01 1	10 0	1 0	11 0	11 1
b ₀ (x)	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b ₁ (x)			0	0	1	1	0	0	1	1	0	0	1	1
b ₂ (x)							0	0	0	0	1	1	1	1



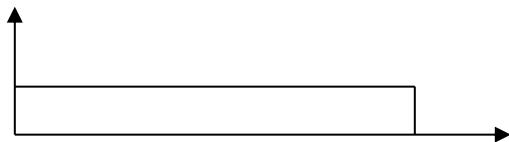
N=2,4,8时的沃尔什变换核

N	N=2 p=1		N=4 p=2				N=8 p=3							
x u \	0	1	0	1	2	3	0	1	2	3	4	5	6	7
0	+	+	+	+	+	+	+	+	+	+	+	+	+	+
1	+	-	+	+	-	-	+	+	+	+	-	-	-	-
2			+	-	+	-	+	+	-	-	+	+	-	-
3			+	-	-	+	+	+	-	-	-	-	+	+
4							+	-	+	-	+	-	+	-
5							+	-	+	-	-	+	-	+
6							+	-	-	+	+	-	-	+
7							+	-	-	+	-	+	+	-

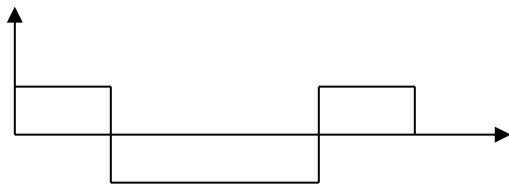
图例



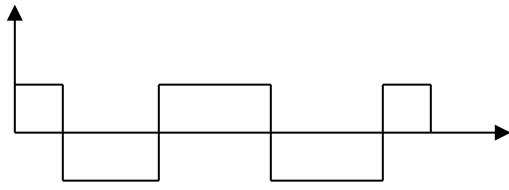
$u=0$



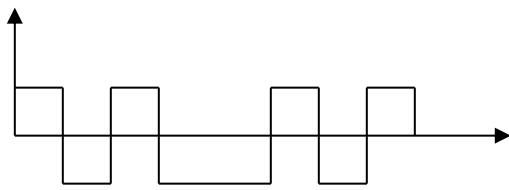
$u=3$



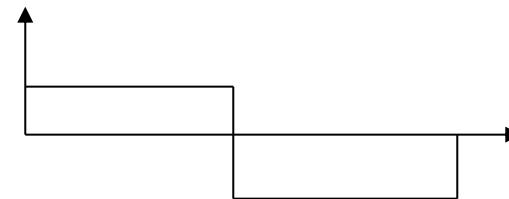
$u=6$



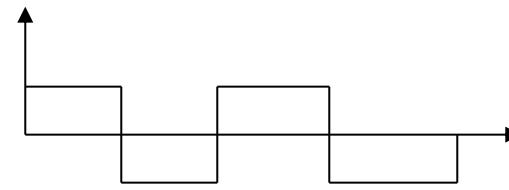
$u=5$



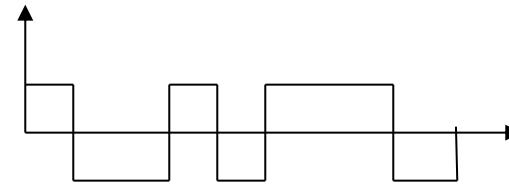
$u=1$



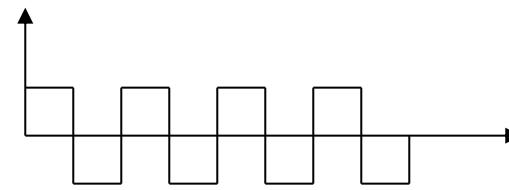
$u=2$

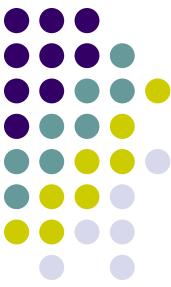


$u=7$



$u=4$





问题

● 计算N=4时沃尔什变换

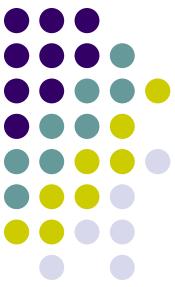
解：通过定义得，

$$W(0) = \frac{1}{4} \sum_{x=0}^3 \left[f(x) \prod_{i=0}^1 (-1)^{b_i(x)b_{p-1-i}(0)} \right] = \frac{1}{4} [f(0) + f(1) + f(2) + f(3)]$$

$$W(1) = \frac{1}{4} \sum_{x=0}^3 \left[f(x) \prod_{i=0}^1 (-1)^{b_i(x)b_{p-1-i}(1)} \right] = \frac{1}{4} [f(0) + f(1) - f(2) - f(3)]$$

$$W(2) = \frac{1}{4} \sum_{x=0}^3 \left[f(x) \prod_{i=0}^1 (-1)^{b_i(x)b_{p-1-i}(2)} \right] = \frac{1}{4} [f(0) - f(1) + f(2) - f(3)]$$

$$W(3) = \frac{1}{4} \sum_{x=0}^3 \left[f(x) \prod_{i=0}^1 (-1)^{b_i(x)b_{p-1-i}(3)} \right] = \frac{1}{4} [f(0) - f(1) - f(2) + f(3)]$$



二维离散沃尔什变换

• 公式

$$\begin{aligned} W(u, v) &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \prod_{i=0}^{p-1} (-1)^{b_i(x)b_{p-1-i}(u) + b_i(y)b_{p-1-i}(v)} \\ &= \frac{1}{N} \left[\frac{1}{N} \sum_{x=0}^{N-1} f(x, y) \prod_{i=0}^{p-1} (-1)^{b_i(x)b_{p-1-i}(u)} \right] \prod_{i=0}^{p-1} (-1)^{b_i(y)b_{p-1-i}(v)} \\ &= \frac{1}{N^2} GfG \end{aligned}$$

问题



- 计算下列数字图像信号矩阵的DWT。

$$(1) f_1 = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad (2) f_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$



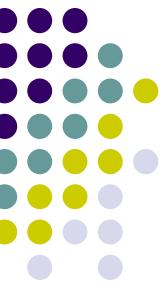
问题

- 计算下列数字图像信号矩阵的DWT。

$$(1) f_1 = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad (2) f_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Solution: $F_1 = \frac{1}{N^2} W f_1 W = \frac{1}{4^2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$

$$= \begin{bmatrix} 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \frac{1}{N^2} \cdot W f_1 W$$



解: $F_2 = \frac{1}{N^2} W f_2 W$

$$= \frac{1}{4^2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

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

点评



- 沃尔什变换具有某种能量集中。
- 原始图像中灰度级越靠近均匀分布，经变换后的图像越集中于矩阵的边角上。对于这样的图像，沃尔什变换可以用于压缩图像信息，而且压缩比率比傅立叶变换大。
- 缺点：图像的规格要求是2的整数幂



哈达玛 (Hadamard) 变换

- 哈达玛变换本质上是一种特殊排序的沃尔什变换；
- 其与沃尔什变换的最大区别是变换核矩阵行的次序不同；
- 这样得到的好处是：哈达玛变换的变换矩阵具有简单的递推关系，即高阶的变换矩阵可以用低阶转换矩阵构成。



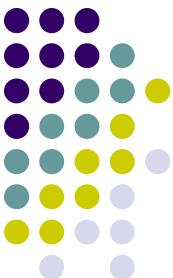
哈达玛 (Hadamard) 变换

- 这样得到的好处是：哈达玛变换的变换矩阵具有简单的递推关系，即高阶的变换矩阵可以用低阶转换矩阵构成。

一维离散哈达玛变换定义为：

$$H(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) (-1)^{\sum_{i=0}^{p-1} b_i(x)b_i(u)}$$

其中 $N = 2^p$, $x, u = 0, 1, 2, \dots, N - 1$ 。



递推公式

$$\text{当 } N = 2 \text{ 时, } H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\text{当 } N > 2 \text{ 时, } H_N = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix}$$

当 $N = 4$ 时,

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

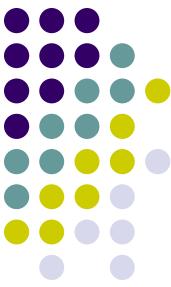
$\boxed{W = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}}$

问题

- 计算下列图像的哈达玛变换。



$$(1) f_1 = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad (2) f_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$



问题

- 计算下列图像的哈达玛变换。

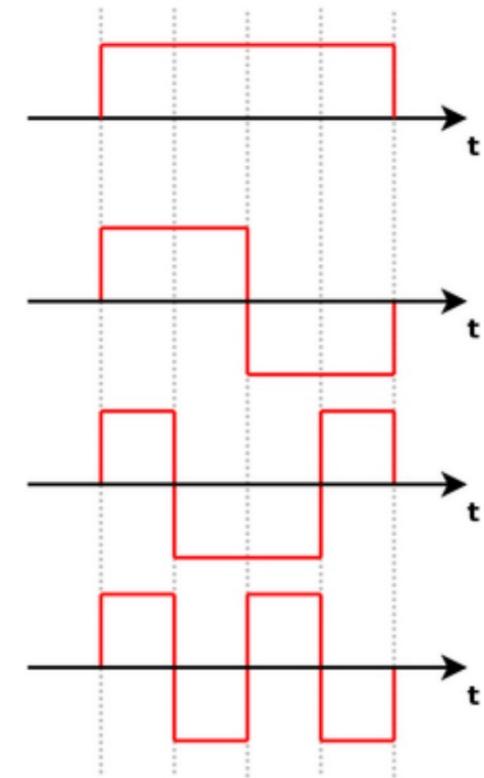
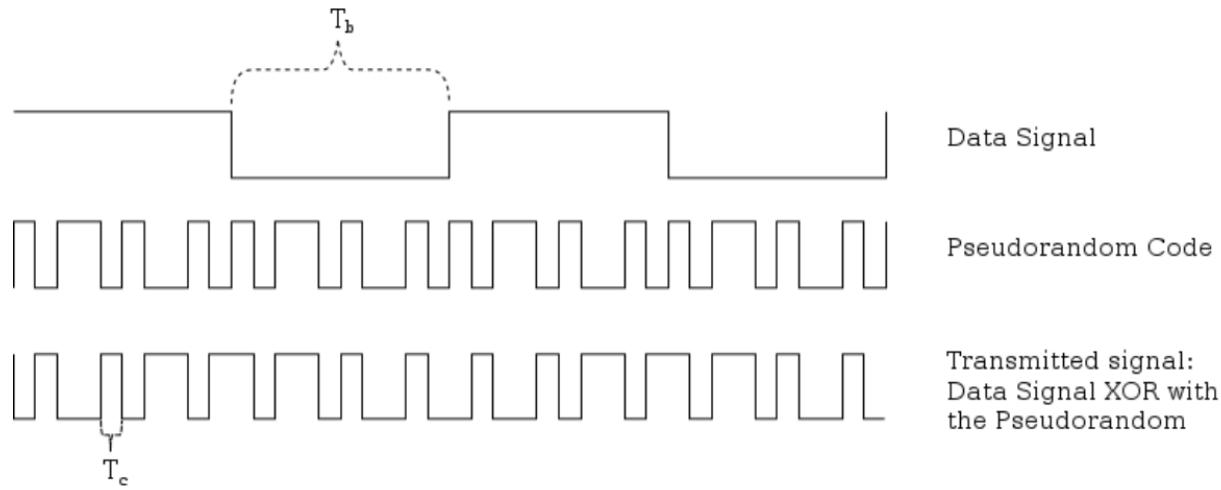
$$(1) f_1 = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad (2) f_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\text{解: } F_1 = \frac{1}{4} H f_1 H = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad F_2 = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

8. Hadamard 矩阵

例：DS-CDMA signals



Four mutually
orthogonal signals

9. Toeplitz 矩阵

任何一条对角线的元素取相同值的特殊矩阵

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n} \\ a_1 & a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_2 & a_1 & a_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & a_{-1} \\ a_n & a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix} = [a_{i-j}]_{i,j=0}^n$$

称为 *Toeplitz* 矩阵。显然一个 $(n+1) \times (n+1)$ 的 *Toeplitz* 矩阵由其第一行元素 $a_0, a_{-1}, \dots, a_{-n}$ 和第一列元素 a_0, a_1, \dots, a_n 完全确定。

9. Toeplitz 矩阵

最常见的是对称Toeplitz矩阵 $A = [a_{|i-j|}]_{i,j=0}^n$, 即其元素还满足对称关系 $a_{-i} = a_i (i = 1, 2, \dots, n)$ 。对称Toeplitz矩阵仅由其第1行元素就可以完全描述。因此常将 $(n+1) \times (n+1)$ 对称Toeplitz矩阵简单记为 $A = Teop[a_0, a_1, \dots, a_n]$ 。

Toeplitz矩阵具有如下性质：

- (1) Toeplitz矩阵的线性组合仍然是Toeplitz矩阵；
- (2) 若Toeplitz矩阵 A 的元素 $a_{ij} = a_{|i-j|}$, 则称 A 为对称Toeplitz矩阵；
- (3) Toeplitz矩阵 A 的转置仍然为Toeplitz矩阵；
- (4) Toeplitz矩阵的元素相对于交叉对角线对称。

9. Toeplitz 矩阵

在时间序列分析与功率谱估计中，经常运用 Autoregressive (AR) Model，需要求解 Yule-Walker 方程

$$\begin{bmatrix} R(0) & R(-1) & \dots & R(-m+1) \\ R(1) & R(0) & \dots & R(-m+2) \\ \vdots & \vdots & \ddots & \vdots \\ R(m-1) & R(m-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(m) \end{bmatrix}$$

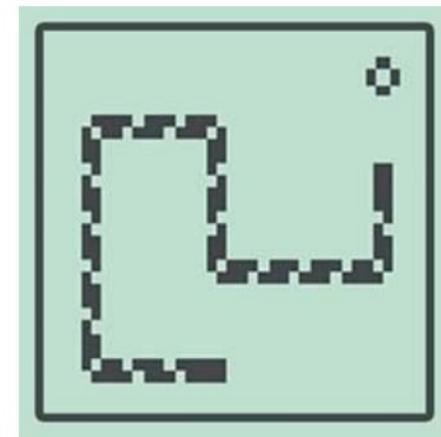
其中 $R(-i) = E\{x^*(n)x(n-i)\}$ 是观测信号 $x(n)$ 的自相关函数， a_i 表示 AR Model 的第 i 个系数。

解这类特殊的线性方程组可用 Levinson-Durbin recursion, FFT 等快速算法。

一种特殊的Toeplitz矩阵 -- 循环矩阵

- 循环矩阵是一种特殊形式的Toeplitz矩阵，它的行向量的每个元素都是前一个行向量各元素依次右移一个位置得到的结果。
- 形式为

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_n \\ c_n & c_1 & c_2 & & c_{n-1} \\ c_{n-1} & c_n & c_1 & & c_{n-2} \\ \vdots & & & \ddots & \vdots \\ c_2 & c_3 & c_4 & \dots & c_1 \end{bmatrix}$$



的 $n \times n$ 矩阵 C 就是循环矩阵 (Circulant Matrix)

一种特殊的Toeplitz矩阵 -- 循环矩阵

- 任何循环矩阵均能被Fourier矩阵对角化形式为

$$C = F_n^{-1} \cdot \text{diag}(F_n c) \cdot F_n$$

$$F_n^{-1} \text{diag}(F_n c) \cdot F_n$$

- 广泛应用于数字信号处理
- 现代4G无线通信的基础: OFDM

Orthogonal Frequency Division Multiplexing (OFDM)

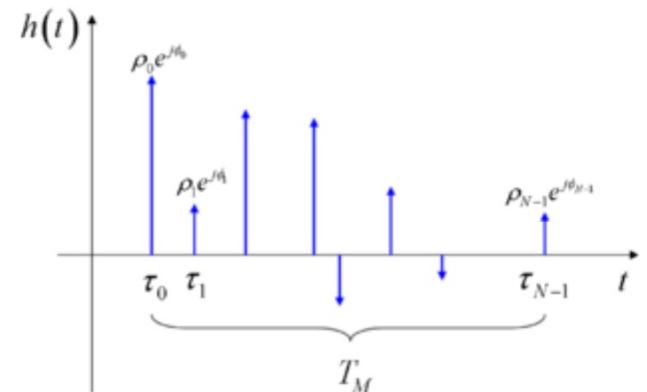
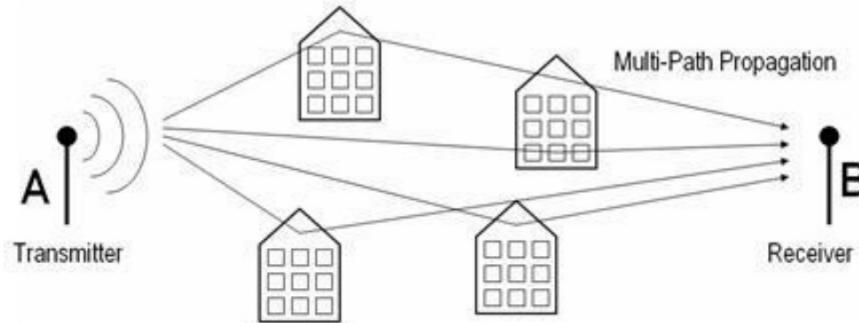
Multipath propagation:

$$y[n] = x[n] * h[n] + v[n]$$

$$= \sum_{l=0}^{\mu} h[l] \cdot x[n-l] + v[n]$$

$$= h[0] \cdot x[n] + h[1] \cdot x[n-1] + \dots + h[\mu] \cdot x[n-\mu] +$$

$v[n]$



OFDM

$y_{-1} \dots y_{-\mu}$ are discarded

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} h_\mu & h_{\mu-1} & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_\mu & \cdots & h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_\mu & h_{\mu-1} & \cdots & h_0 \end{bmatrix} \cdot \begin{bmatrix} x_{-\mu} \\ \vdots \\ x_{-1} \\ x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} + \begin{bmatrix} v_0 \\ \vdots \\ v_{N-2} \\ v_{N-1} \end{bmatrix},$$

$$\tilde{\mathbf{y}} = \mathbf{H}\mathbf{x} + \nu.$$



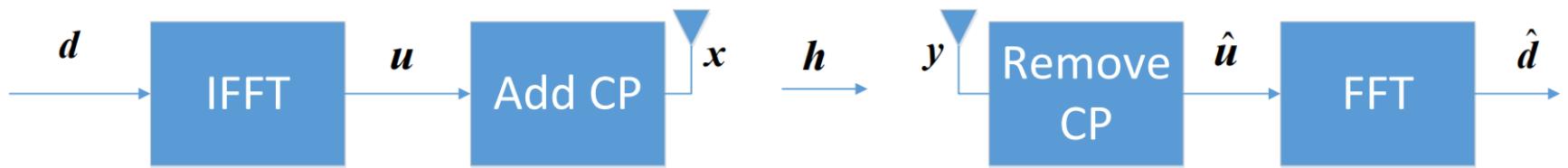
$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} h_0 & 0 & \cdots & h_\mu & h_{\mu-1} & \cdots & h_1 \\ h_1 & h_0 & \cdots & 0 & h_\mu & \cdots & h_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ h_\mu & h_{\mu-1} & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{\mu-1} & \cdots & h_0 & 0 \\ 0 & \cdots & 0 & h_\mu & \cdots & h_1 & h_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-2} \\ x_{N-1} \end{bmatrix} + \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-2} \\ v_{N-1} \end{bmatrix},$$

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \nu.$$

The last μ symbols of $x[n]$ correspond to the cyclic prefix: $x_{-1} = x_{N-1}, x_{-2} = x_{N-2}, \dots, x_{-\mu} = x_{N-\mu}$.

OFDM

- OFDM系统的传输示意图:



$$\mathbf{u} = \mathbf{F}^{-1}\mathbf{d}$$

$$\mathbf{y} = \mathbf{Hx}$$

$$\hat{\mathbf{u}} = \tilde{\mathbf{H}}\mathbf{u}$$

$$\tilde{\mathbf{H}} = \mathbf{F}^{-1}\Lambda\mathbf{F}$$

$$\hat{\mathbf{d}} = \mathbf{F}\hat{\mathbf{u}}$$

$$\hat{\mathbf{d}} = \Lambda\mathbf{d}$$

特殊矩阵

- Hermitian matrix
- 置换与互换矩阵
- 广义置换与选择矩阵
- 正交矩阵与酉矩阵
- 三角矩阵
- Vandermonde matrix
 - Fourier matrix
- Hadamard matrix
- Toeplitz matrix
 - Circulant matrix
- 应用
 - DS-CDMA
 - Yule-Walker方程
 - OFDM