图像正交变换

杭州电子科技大学　李黎

主要内容

- 离散傅立叶变换
  - 一维和二维傅立叶变换定义
  - 快速傅立叶变换
  - 应用
- 离散余旋DCT变换
  - 一维和二维DCT变换定义
  - DCT变换编码
- 离散K-L变换
- 沃尔什和哈达玛变换
- 数字图像的正交基表示

图像信号的正交变换-离散傅立叶变换

离散傅里叶变换 Discrete Fourier Transform

DFT是重要的变换

1.分析有限长序列的有用工具。

2.在信号处理的理论上有重要意义。

3.在运算方法上起核心作用，谱分析、卷积、相关都可以通DFT在计算机上实现。

图像信号的正交变换-离散傅立叶变换

一维离散傅立叶变换：

定义：设{f(n)|n=0,...,N-1}为一维信号的N个采样值，其离散傅立叶变换及其逆变换分别为：

$$F(u)=\frac{1}{N}\sum_{k=0}^{N-1}f(k)e^{-j2\pi\frac{uk}{N}},u=0,1,\ldots,N-1$$

$$f(k)=\sum_{u=0}^{N-1}F(u)e^{j2\pi\frac{uk}{N}},k=0,1,\ldots,N-1$$

图像信号的正交变换-离散傅立叶变换

当u取不同值后，有：

$$F(0)=\frac{1}{N}[f(0)+f(1)+\ldots+f(N-1)]$$

$$F(1)=\frac{1}{N}[f(0)+f(1)e^{-j2\pi\frac{1}{N}}+\ldots+f(N-1)e^{-j2\pi\frac{N-1}{N}}]$$

$$\vdots$$

$$F(N-1)=\frac{1}{N}[f(0)+f(1)e^{-j2\pi\frac{(N-1)}{N}}+\ldots+f(N-1)e^{-j2\pi\frac{(N-1)(N-1)}{N}}]$$

图像信号的正交变换-离散傅立叶变换

$$
\begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & & 1 \\ 1 & e^{-j2\pi\frac{1}{N}} & & ..e^{-j2\pi\frac{N-1}{N}} \\ & & \vdots & \\ 1 & e^{-j2\pi\frac{N-1}{N}} & & ..e^{-j2\pi\frac{(N-1)(N-1)}{N}} \end{bmatrix} \times \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix}
$$

$$
F = H \times f
$$

图像信号的正交变换-离散傅立叶变换

二维离散傅立叶变换：

定义：设{f(x,y)|x=0,..,N-1, y=0,..,M-1}为二维图像信号其离散傅立叶变换及其逆变换分别为：

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

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

图像信号的正交变换-离散傅立叶变换

二维图像信号的频谱：

$$F(u,v) = |F(u,v)| \exp[j\varphi(u,v)] = R(u,v) + jI(u,v)$$

相位谱：

$$\varphi(u,v) = tg^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

幅度谱：

$$|F(u,v)| = [R(u,v)^2 + I(u,v)^2]^{1/2}$$

图像信号的正交变换-离散傅立叶变换

二维离散傅立叶变换性质：

- 线性、位移、尺度、卷积、相关
- 变换的可分离性

  可以分解为两个一维离散傅立叶变换

  运算过程，先行后列，先列后行

- 旋转不变性

  图像在空间域旋转一定角度，其频谱在频域旋转同样角度

- 加法定理 $$f(x)+g(x,y) \Leftrightarrow F(u,v)+G(u,v)$$
- 相似性定理 $$f(ax,,by) \Leftrightarrow \frac{1}{|ab|}F(\frac{u}{a},\frac{v}{b})$$
- 位移定理 $$f(x-a,y-b) \Leftrightarrow e^{-j2\pi(au+bv)}F(u,v)$$
- 卷积定理 $$f(x,y)\times g(x,y) \Leftrightarrow F(u,v)G(u,v)$$

- 可分离乘积

$$f(x)\times g(y) \Leftrightarrow F(u)G(v)$$

- 微分

$$(\frac{\partial}{\partial x})^m(\frac{\partial}{\partial y})^n f(x,y) \Leftrightarrow (j2\pi u)^m(j2\pi v)^n F(u,v)$$

- 旋转

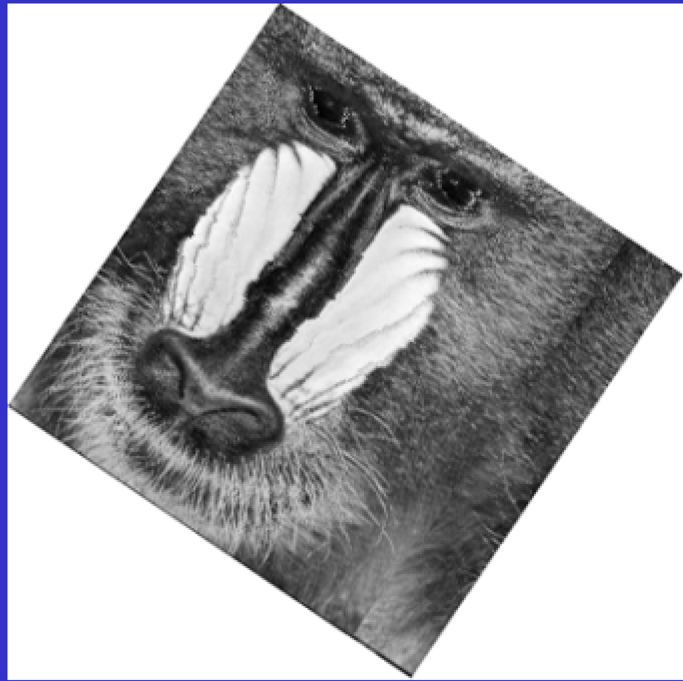
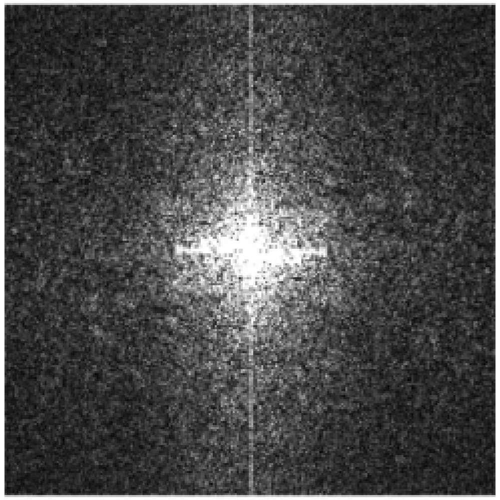
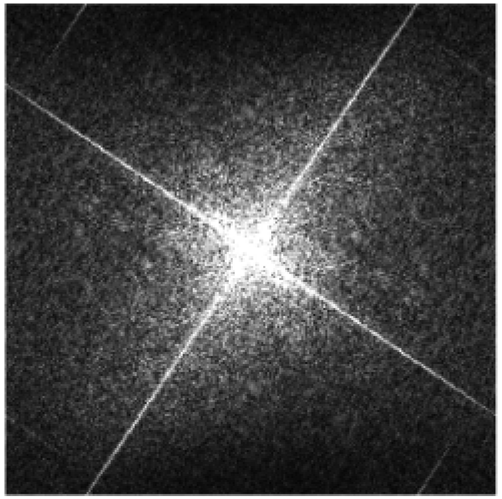
$$f(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$$
$$\Leftrightarrow F(u\cos\theta + v\sin\theta, -u\sin\theta + v\cos\theta)$$

- 拉普拉斯变换

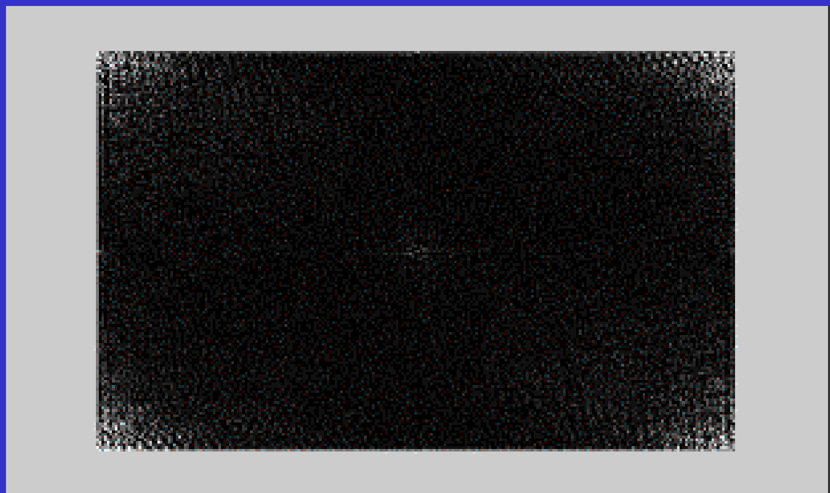
$$\nabla^2 f(x,y) = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} f(x,y)$$
$$\Leftrightarrow -4\pi^2(u^2+v^2)F(u,v)$$

- Rayleigh定理

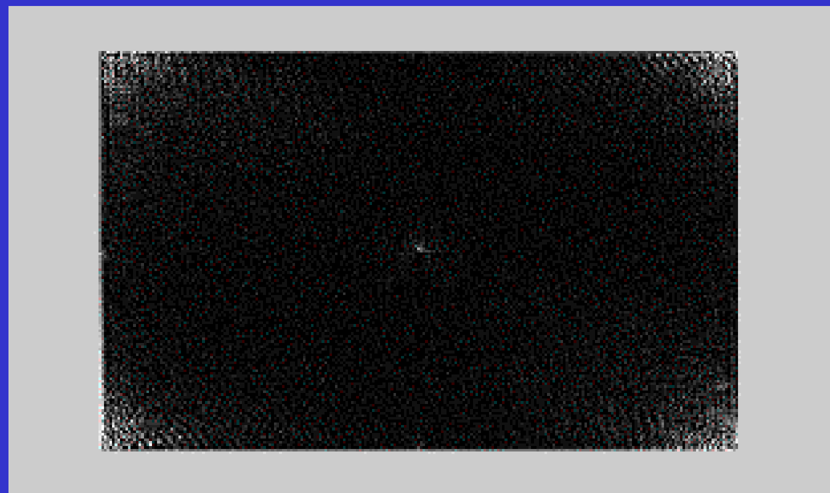
$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left|f(x,y)\right|^2 dxdy = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left|F(u,v)\right|^2 dudv$$

原始图像

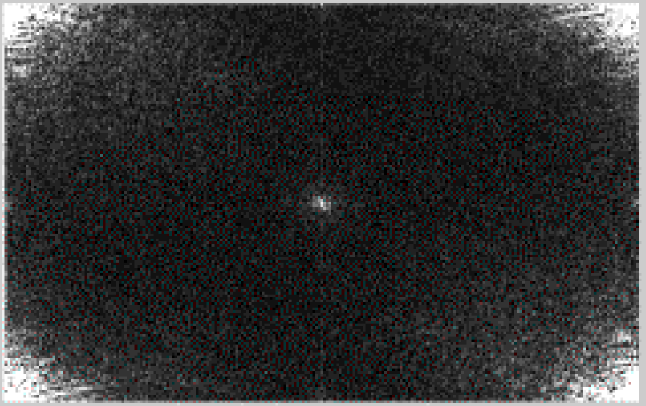


实部频谱

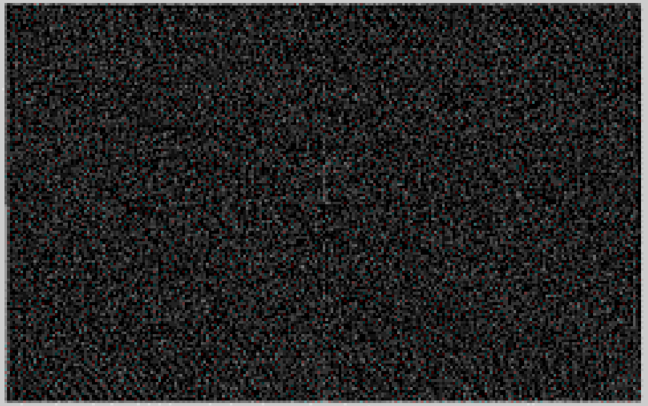


虚部频谱

原始图像

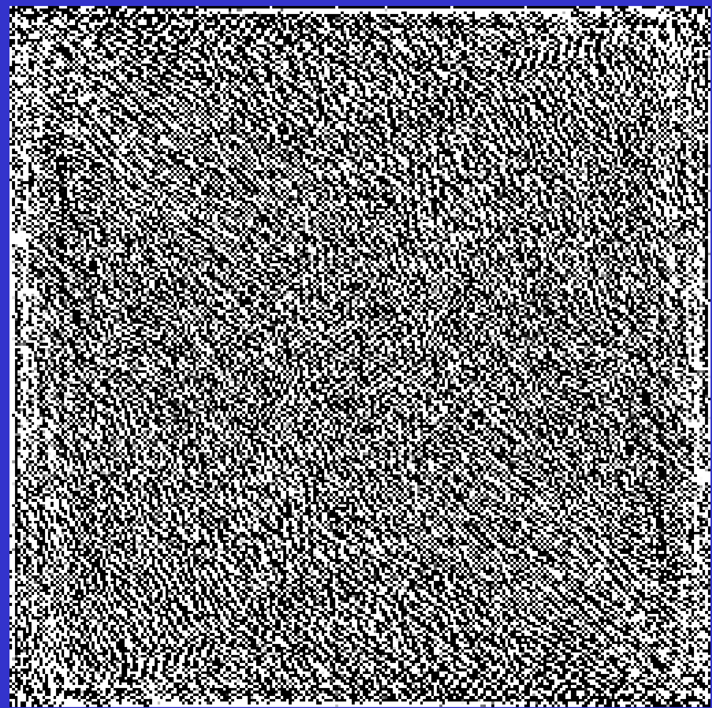


幅度频谱



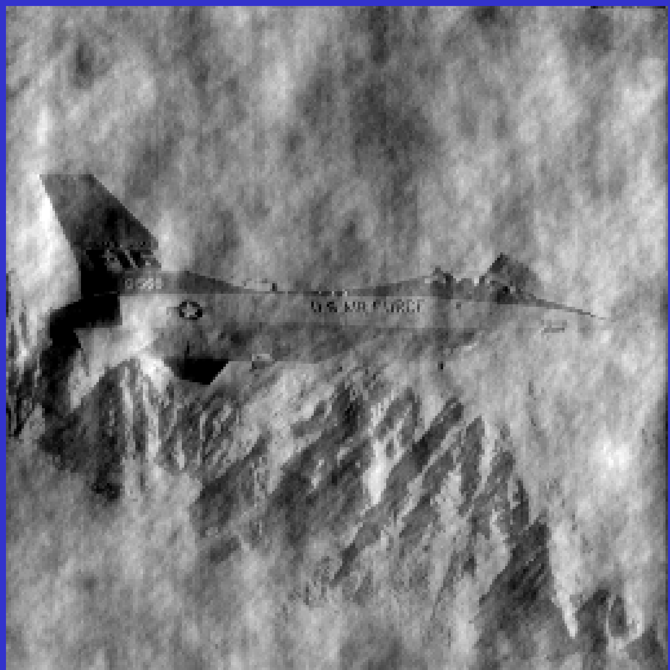
相位频谱

幅值谱 (DFT of Lena)



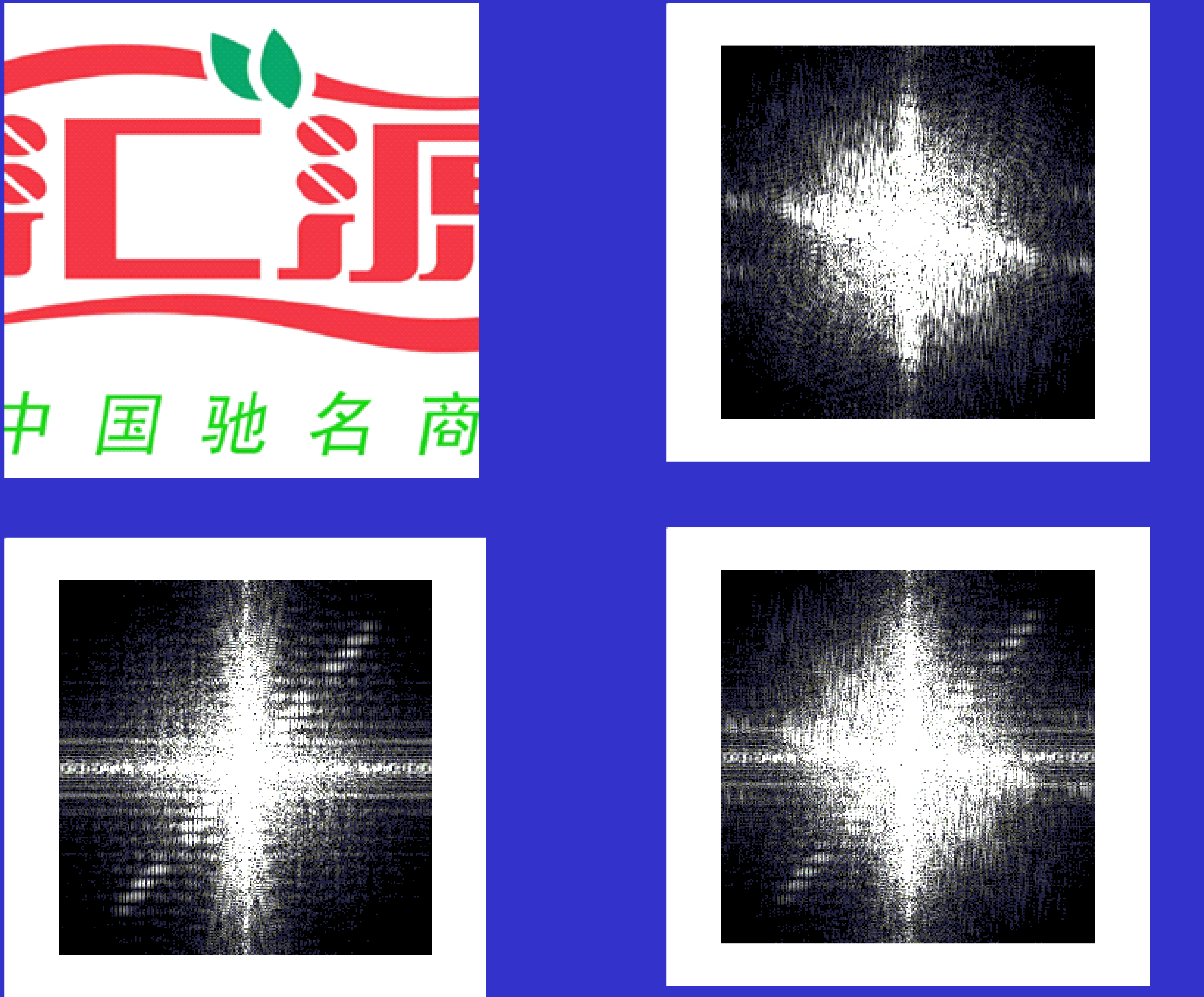
将两幅图的幅度谱和相位分别合成

合成结果



MATLAB程序

- mat000=imread('2.tif');
- mat011=fftshift(fft2(mat000(:,:,1)));
- % mat011=fftshift(fft2(mat000(:,:,2)));
- % mat011=fftshift(fft2(mat000(:,:,3)));
- mat=abs(mat011);
- mat=double(mat);
- figure;imshow((mat./30),gray(256))



图像信号的正交变换-离散傅立叶变换

DFT的计算工作量

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk}, \qquad k=0,1,\cdots,N-1$$

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-nk}, \qquad n=0,1,\cdots,N-1$$

两者的差别仅在指数的符号和因子1/N.

图像信号的正交变换-离散傅立叶变换

一个X(k)的值的工作量，如X(1)

$$X(1)=x(0)W_N^0+x(1)W_N^1+x(2)W_N^2+\cdots+x(N-1)W_N^{N-1}$$

通常x(n)和W_N^{nk}都是复数，所以计算一个X(k)的值需要N次复数乘法运算，和$N-1$次复数加法运算。那么，所有的X(k)就要N^2次复数乘法运算，N(N-1)次复数加法运算。当N很大时，运算量将是惊人的，如N=1024，则要完成1048576 次(一百多万次)运算。这样，难以做到实时处理。

改进的途径

1. W_N^{nk}的对称性和周期性

对称性：$(W_N^{nk})^* = W_N^{-nk}$，

周期性：$W_N^{nk} = W_N^{(n+N)k} = W_N^{n(k+N)}$；

得：

$$W_N^{n(N-k)} = W_N^{(N-n)k} = W_N^{-nk} (\because W_N^{Nk} = W_N^{Nn} = e^{-2\pi k(n)} = 1),$$

$$W_N^{N/2} = -1 (\because W_N^{N/2} = e^{-j\pi} = -1),$$

$$W_N^{(k+N/2)} = -W_N^k.$$

改进的途径

将原函数分为奇数项和偶数项，通过不断的一个奇数一个偶数的相加（减），最终得到需要的结果。也就是说FFT是将复杂的运算变成两个数相加（减）的简单运算的重复。从而降低运算次数,提高运算速度.1965年,库利(cooley)和图基(Tukey)首先提出FFT算法.对于N点DFT,仅需 $(N/2)\log_2 N$ 次复数乘法运算.例如 $N=1024=2^{10}$ 时，需要 $(1024/2)\log_2 2^{10}$ $=512*10=5120$次。5120/1048576=4.88% ，速度提高20倍

改进的途径

2、快速Fourier变换的推导

令： $w_N^{\mu x} = \exp(-j\frac{2\pi\mu x}{N})$

则： $F(\mu) = \frac{1}{N}\sum_{x=0}^{N-1} f(x) w_N^{\mu x}$

$$= \frac{1}{2}[\frac{2}{N}\sum_{x=0}^{N/2-1} f(2x) w_N^{2\mu x} + \frac{2}{N}\sum_{x=1}^{N/2-1} f(2x+1) w_N^{\mu(2x+1)}]$$

$$\underline{\underline{M \triangleq \frac{N}{2}}}\ \frac{1}{2}[\frac{1}{M}\sum_{x=0}^{M-1} f(2x) w_M^{\mu x} + \frac{1}{M}\sum_{x=1}^{M-1} f(2x+1) w_M^{\mu x} w_N^{\mu}]$$

$$= \frac{1}{2}[F_e(\mu) + w_N^{\mu} F_o(\mu)]$$

（分成奇数项和偶数项之和）

$0 \le \mu \le M$

改进的途径

$$F(\mu+M)=\frac{1}{2}\left[F_e(\mu+M)+w_N^{\mu+M}F_o(\mu+M)\right]$$

$$=\frac{1}{2}\left[F_e(\mu)+w_N^{\mu+M}F_o(\mu)\right]$$

$$w_N^{\mu+M}=w_N^{\mu}\cdot w_N^{M}=w_N^{\mu}\cdot\exp(-j\tfrac{2\pi M}{N})$$

$$=w_N^{\mu}\cdot\exp(-j\pi)=-w_N^{\mu}$$

$$\therefore F(\mu+M)=\frac{1}{2}\left[F_e(\mu)-w_N^{\mu}F_o(\mu)\right]$$

改进的途径

例：设对一个函数进行快速Fourier变换，函数为：

$$f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7$$

分成偶数、奇数为：

$$f_0, f_2, f_4, f_6 \quad | \quad f_1, f_3, f_5, f_7$$

$$f_0, f_4 \quad | \quad f_2, f_6 \quad \rightarrow \quad f_1, f_5 \quad | \quad f_3, f_7$$

改进的途径

$f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7$

偶数区

f_0, f_4, f_2, f_6

$$F^{(0)}(0) = \tfrac{1}{2}\cdot\left[f_0 + w_2^0 f_4\right]$$
$$F^{(0)}(1) = \tfrac{1}{2}\cdot\left[f_0 - w_2^0 f_4\right]$$
$$F^{(2)}(0) = \tfrac{1}{2}\cdot\left[f_2 + w_2^0 f_6\right]$$
$$F^{(2)}(1) = \tfrac{1}{2}\cdot\left[f_2 - w_2^0 f_6\right]$$

$$F^{(0)}(0) = \tfrac{1}{2}\cdot\left[F^{(0)}(0) + w_4^0 F^{(2)}(0)\right]$$
$$F^{(0)}(1) = \tfrac{1}{2}\cdot\left[F^{(0)}(1) + w_4^1 F^{(2)}(1)\right]$$
$$F^{(0)}(2) = \tfrac{1}{2}\cdot\left[F^{(0)}(0) - w_4^0 F^{(2)}(0)\right]$$
$$F^{(0)}(3) = \tfrac{1}{2}\cdot\left[F^{(0)}(1) - w_4^1 F^{(2)}(1)\right]$$

基数区

f_1, f_5, f_3, f_7

$$F^{(1)}(0) = \tfrac{1}{2}\cdot\left[f_1 + w_2^0 f_5\right]$$
$$F^{(1)}(1) = \tfrac{1}{2}\cdot\left[f_1 - w_2^0 f_5\right]$$
$$F^{(3)}(0) = \tfrac{1}{2}\cdot\left[f_3 + w_2^0 f_7\right]$$
$$F^{(3)}(1) = \tfrac{1}{2}\cdot\left[f_3 - w_2^0 f_7\right]$$

$$F^{(1)}(0) = \tfrac{1}{2}\cdot\left[F^{(1)}(0) + w_4^0 F^{(3)}(0)\right]$$
$$F^{(1)}(1) = \tfrac{1}{2}\cdot\left[F^{(1)}(1) + w_4^1 F^{(3)}(1)\right]$$
$$F^{(1)}(2) = \tfrac{1}{2}\cdot\left[F^{(1)}(0) - w_4^0 F^{(3)}(0)\right]$$
$$F^{(1)}(3) = \tfrac{1}{2}\cdot\left[F^{(1)}(1) - w_4^1 F^{(3)}(1)\right]$$

改进的途径

$$\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}$$

$$F(0)=\tfrac{1}{2}\cdot\left[F^{(0)}(0)+w_8^0F^{(1)}(0)\right]$$
$$F(1)=\tfrac{1}{2}\cdot\left[F^{(0)}(1)+w_8^1F^{(1)}(1)\right]$$
$$F(2)=\tfrac{1}{2}\cdot\left[F^{(0)}(2)+w_8^2F^{(1)}(2)\right]$$
$$F(3)=\tfrac{1}{2}\cdot\left[F^{(0)}(3)+w_8^3F^{(1)}(3)\right]$$
$$F(4)=\tfrac{1}{2}\cdot\left[F^{(0)}(0)-w_8^0F^{(1)}(0)\right]$$
$$F(5)=\tfrac{1}{2}\cdot\left[F^{(0)}(1)-w_8^1F^{(1)}(1)\right]$$
$$F(6)=\tfrac{1}{2}\cdot\left[F^{(0)}(2)-w_8^2F^{(1)}(2)\right]$$
$$F(7)=\tfrac{1}{2}\cdot\left[F^{(0)}(3)-w_8^3F^{(1)}(3)\right]$$

改进的途径

3、二维快速Fourier变换：

因为2维DFT可以看成是两次的1维DFT变换，即：

$$F(\mu,\upsilon)=f_{行}\{f_{列}[f(x,y)]\}$$

所以二维快速Fourier变换实际上是对其进行了2次的一维FFT变换。

改进的途径

4、稍微变动FFT程序和参数可实现IFFT

$$X(k) = DFT[x(n)] = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

$$x(n) = IDFT[X(k)] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}$$

比较两式可知，只要DFT的每个系数 W_N^{nk} 换成 W_N^{-nk}，最后再乘以常数1/N就可以得到IDFT的快速算法--IFFT。

另外，可以将常数1/N分配到每级运算中，

$\because \dfrac{1}{N}=\dfrac{1}{2^L}=(\dfrac{1}{2})^L$，也就是每级蝶形运算均乘以1/2。

5、不改(FFT)的程序直接实现IFFT

$$\because [W_N^{-nk}]^* = W_N^{nk}, [A \cdot B]^* = A^* \cdot B^*$$

$$\therefore x^*(n) = [\frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-nk}]^*$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X^*(k) W_N^{nk}$$

因此，$x(n)$ $= \frac{1}{N}\left[\sum_{k=0}^{N-1} X^*(k) W_N^{nk}\right]^*$

$$= \frac{1}{N}\left\{DFT\left[X^*(k)\right]\right\}^*$$

改进的途径

这就是说，先将X(k)取共轭，即将X(k)的虚部乘-1，直接利用FFT程序计算DFT；然后再取一次共轭；最后再乘1/N，即得x(n)。所以FFT，IFFT可用一个子程序。

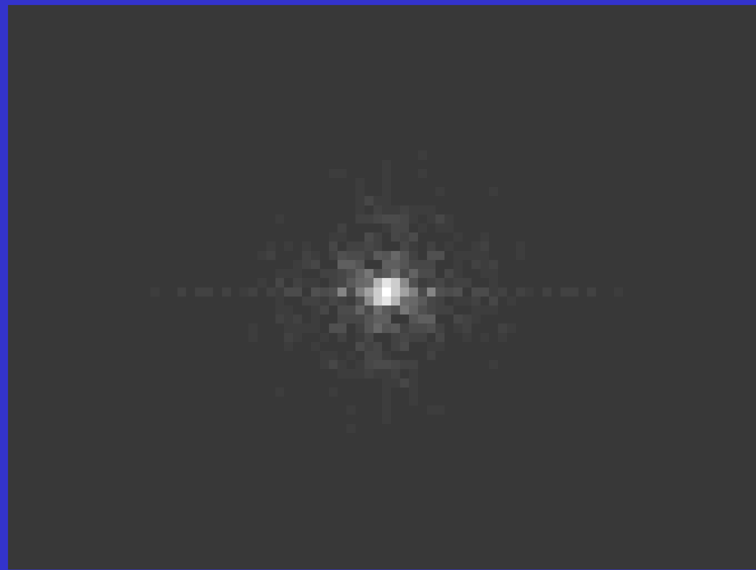
二维Fourier变换的应用

1、Fourier变换在图像滤波中的应用

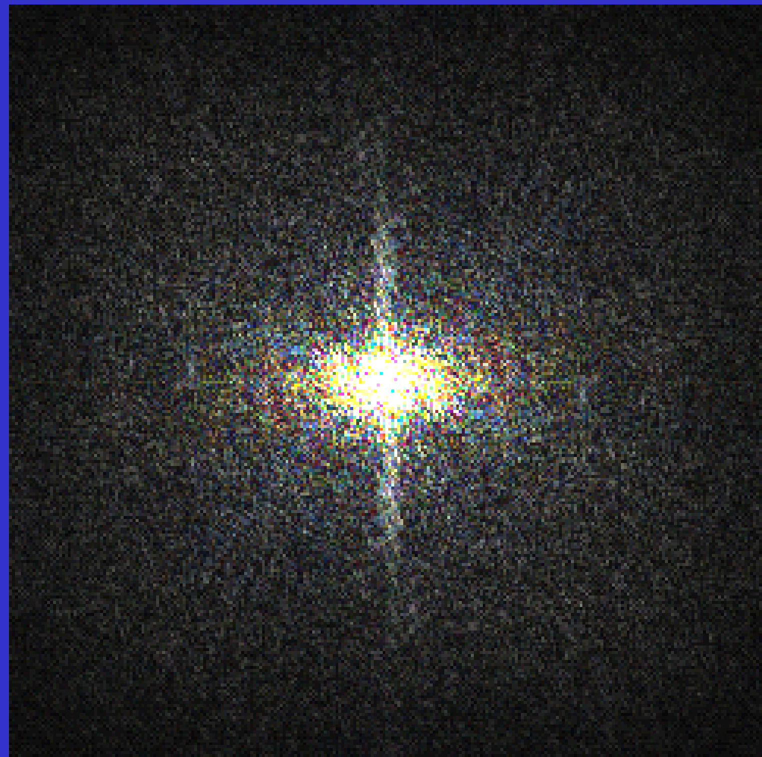
首先，我们来看Fourier变换后的图像，中间部分为低频部分，越靠外边频率越高。

因此，我们可以在Fourier变换图中，选择所需要的高频或是低频滤波。

Fourier 变换示意图

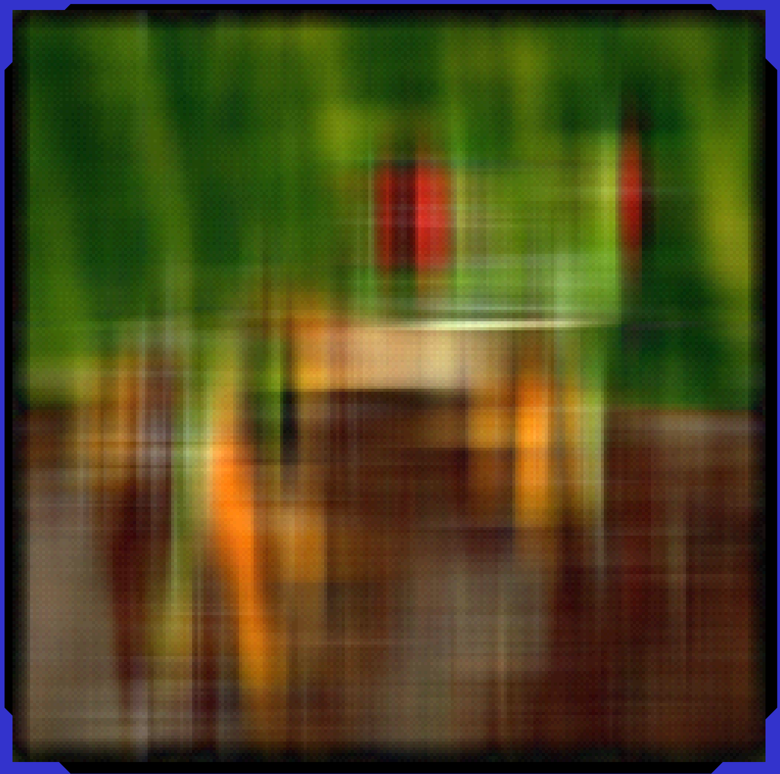


Fourier变换的频率特性



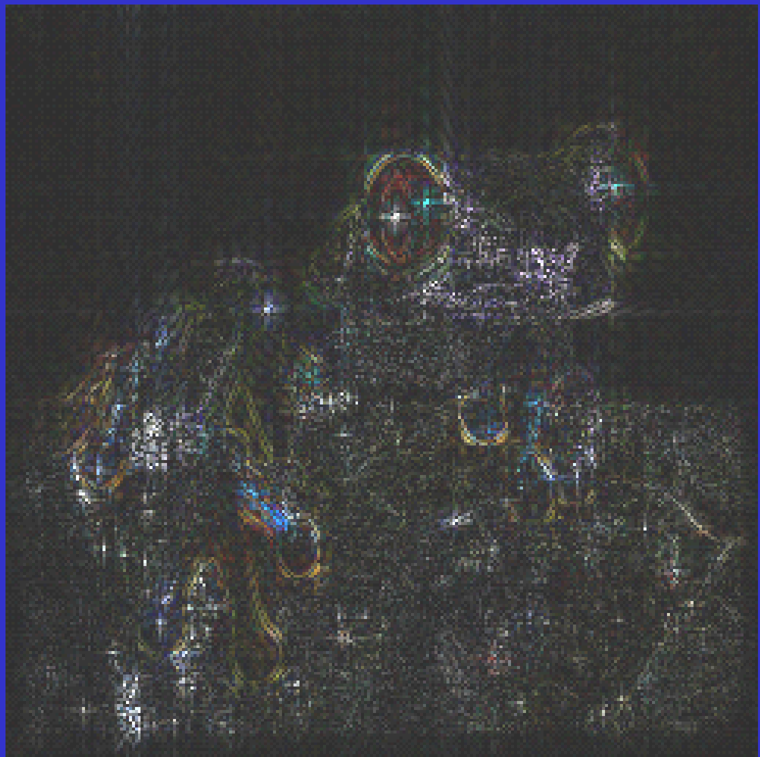
Fourier变换的低通滤波

低频通过，低频反映景物概貌



Fourier变换的高通滤波

高频通过，高频反映细节。 .

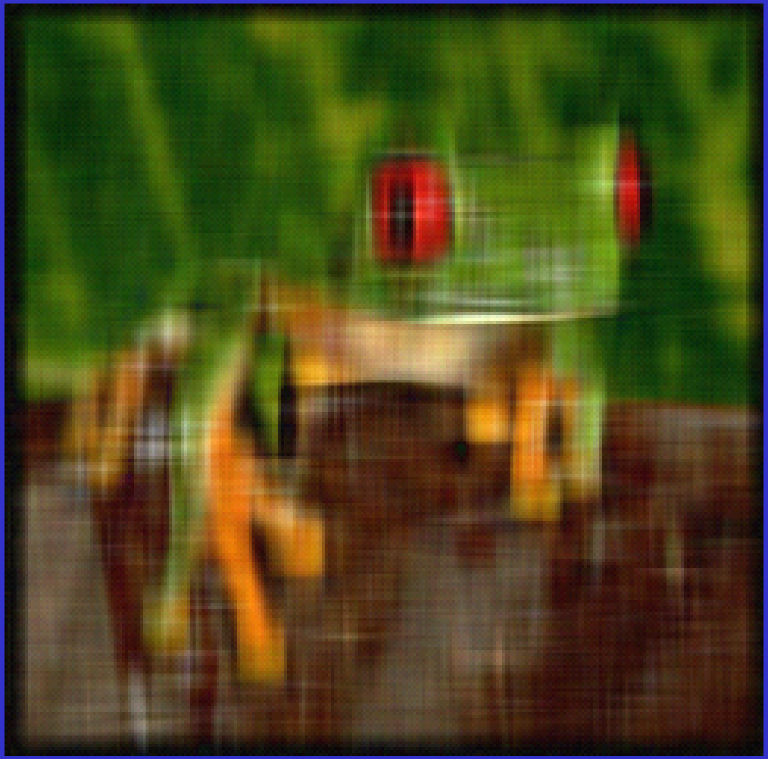


Fourier变换的应用

2、Fourier变换在图像压缩中的应用

变换系数刚好表现的是各个频率点上的幅值。在小波变换没有提出时，用来进行压缩编码。考虑到高频反映细节、低频反映景物概貌的特性。往往认为可将高频系数置为0，骗过人眼。

Fourier变换的压缩原理



压缩率为：3.3：1

Fourier变换的压缩原理

压缩率为：16.1：1

二维Fourier变换的应用

3、 Fourier变换在卷积中的应用：

从前面的图像处理算法中知道，如果抽象来看，其实都可以认为是图像信息经过了滤波器的滤波（如：平滑滤波、锐化滤波等 ）。 如果滤波器的结构比较复杂时，直接进行时域中的卷积运算是不可思议的。

二维Fourier变换的应用

$$f(i,j) \longrightarrow \boxed{G(S)} \longrightarrow f_g(i,j)$$

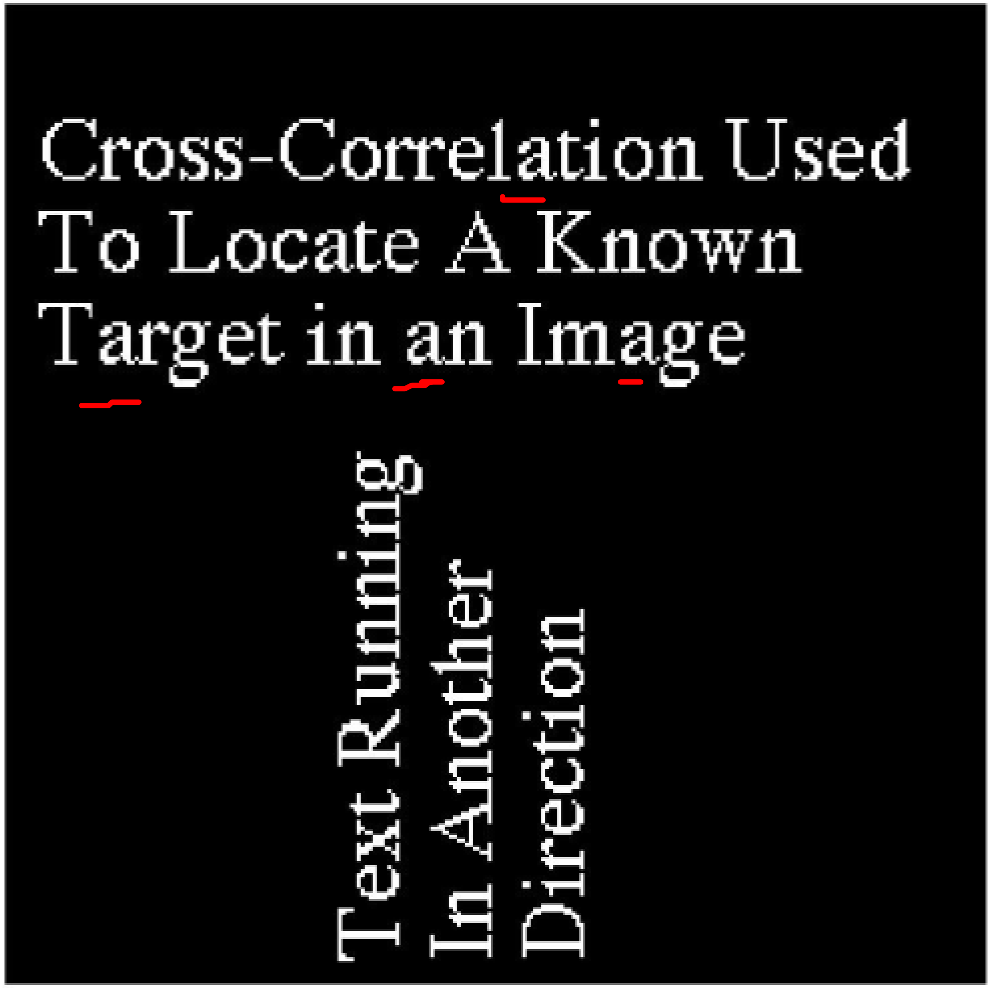
$$f_g = g * f$$

$$F_g(\mu,\upsilon) = G(\mu,\upsilon) \cdot F(\mu,\upsilon)$$

$$f_g = FFT^{-1}(F_g)$$

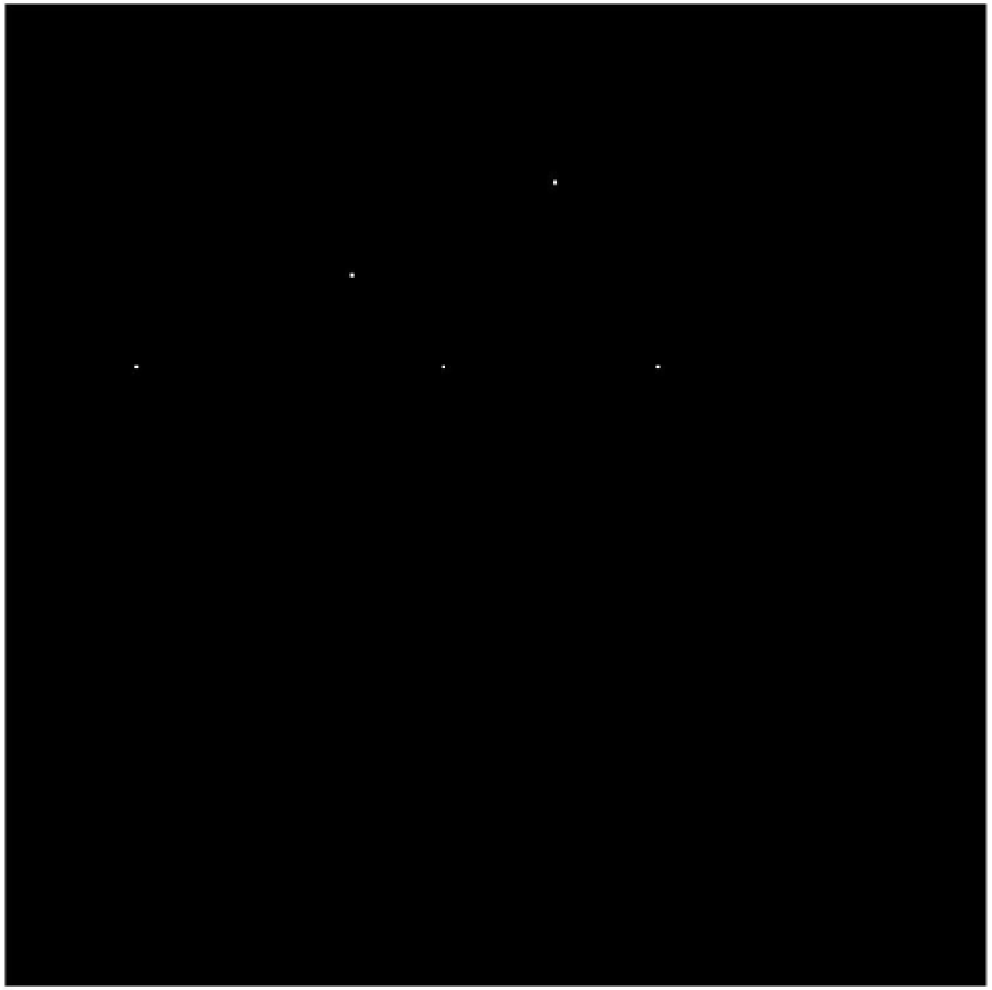
二维Fourier变换的应用

- 对图像定位的应用
- 以待定位的目标为模板在待识别图像上滑动，利用卷积的性质


Cross-Correlation Used
To Locate A Known
Target in an Image
Text Running
In Another
Direction

a


Cross Correlation Used
To Locate A Known
Target in an Image



图像信号的正交变换-离散余弦变换（DCT）

问题的提出：

Fourier变换的一个最大的问题是：它的参数都是复数，在数据的描述上相当于实数的两倍。为此，我们希望有一种能够达到相同功能但数据量又不大的变换。

在此期望下，产生了DCT变换。

图像信号的正交变换－离散余弦变换（DCT）

DCT变换的应用：

余弦变换实际上是傅立叶变换的实数部分。余弦变换主要用于图像的压缩，如目前的国际压缩标准的JPEG格式中就用到了DCT变换。具体的做法与DFT 相似。给高频系数大间隔量化，低频部分小间隔量化。

图像信号的正交变换-离散余旋变换

一维离散余旋变换：

由于实偶函数的傅立叶变换只含有余旋项，因此构造了一种实数域的变换—DCT变换。

DCT变换的基本思想是将一个实函数对称延拓成一个实偶函数，实偶函数的傅立叶变换也必然是实偶函数。

$$g(x)=\begin{cases} f(x-\frac{1}{2}), & x=\frac{1}{2},\ldots,\frac{1}{2}+(N-1) \\ f(-x+\frac{1}{2}), & x=-\frac{1}{2},\ldots,-\frac{1}{2}-(N-1) \end{cases}$$

图像信号的正交变换-离散余旋变换

一维离散余旋变换：

$$F(u)=C(u)\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f(x)\cos\frac{(2x+1)u\pi}{2N}$$

$$f(x)=\sqrt{\frac{2}{N}}\sum_{u=0}^{N-1}C(u)F(ux)\cos\frac{(2x+1)u\pi}{2N}$$

$$C(u)=\begin{cases}1/\sqrt{2}, u=0\\ 1, otherwise\end{cases}$$

余旋变换基函数

$$
\begin{bmatrix}
1 & \cdots & 1 \\
\cos(\pi/2N) & & \ldots\cos((2N-1)\pi/2N) \\
 & \vdots & \\
\cos((N-1)\pi/2N & & \ldots\cos((N-1)(2N-1)\pi/2N)
\end{bmatrix}
$$

$$F = G \times f$$

图像信号的正交变换-离散余旋变换

二维离散余旋变换：

$$F(u,v)=C(v)C(u)\sqrt{\frac{2}{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cos\frac{(2x+1)u\pi}{2M}\cos\frac{(2y+1)u\pi}{2N}$$

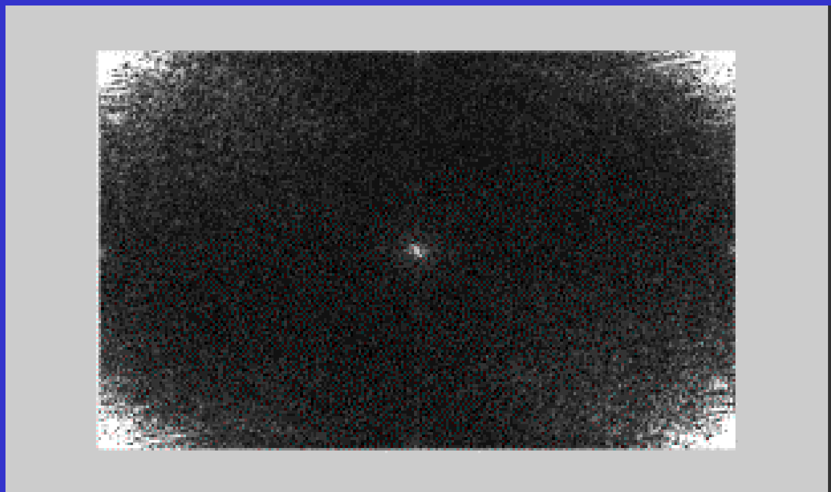
$$f(x,y)=\sqrt{\frac{2}{MN}}\sum_{u=0}^{M}\sum_{v=0}^{N-1}C(u)C(v)F(u)\cos\frac{(2x+1)u\pi}{2M}\cos\frac{(2y+1)u\pi}{2N}$$

$$F=G\times f\times G^{T}$$

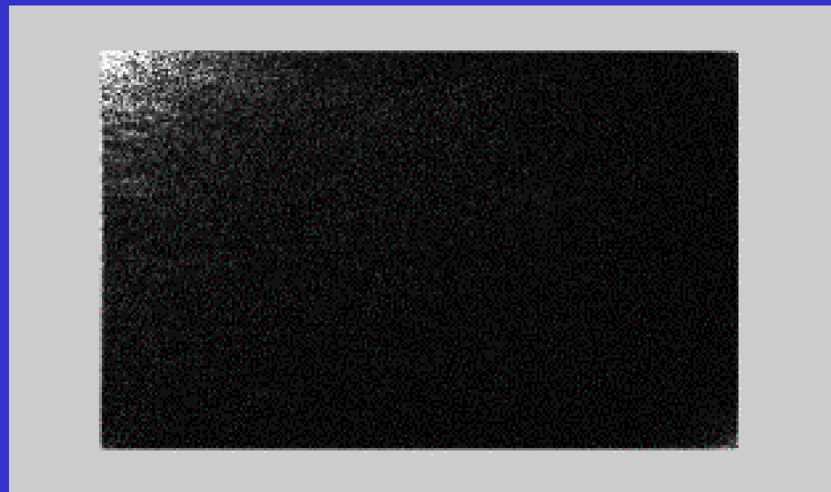
图像信号的正

二维离散余旋信号的频谱：

图像处理中DCT 之后的矩阵大的幅度系数都集中在左上角，这个就是能量集中的意思，先量化再zigzag扫描以后再编码的时候因为后面都是零，码长会比较短，有利于数据压缩，而FFT做不到这点，FFT是有cos又有sin的，编码码长会很长，所以数据压缩一般是DCT。



FFT 图像频谱

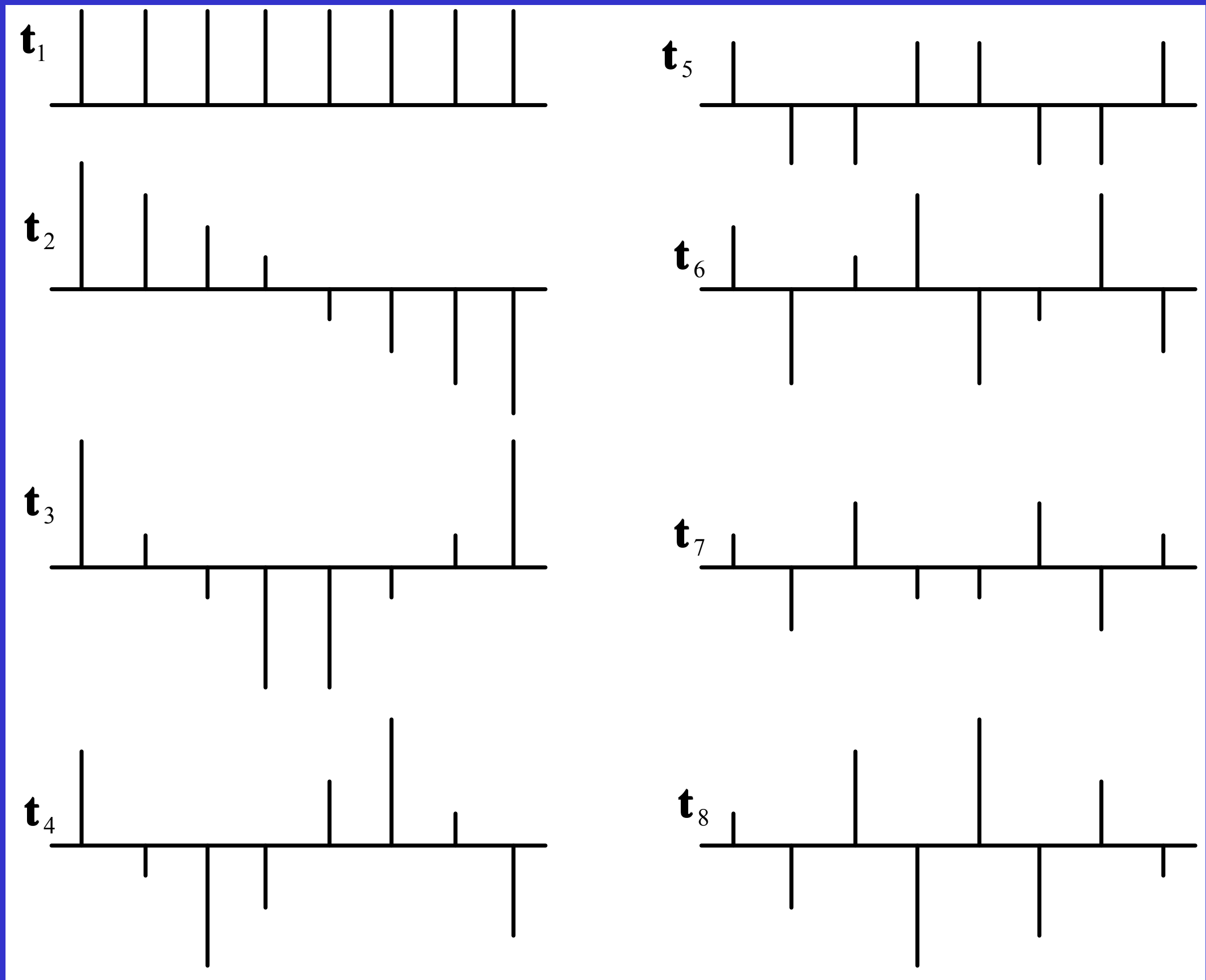


DCT图像频谱

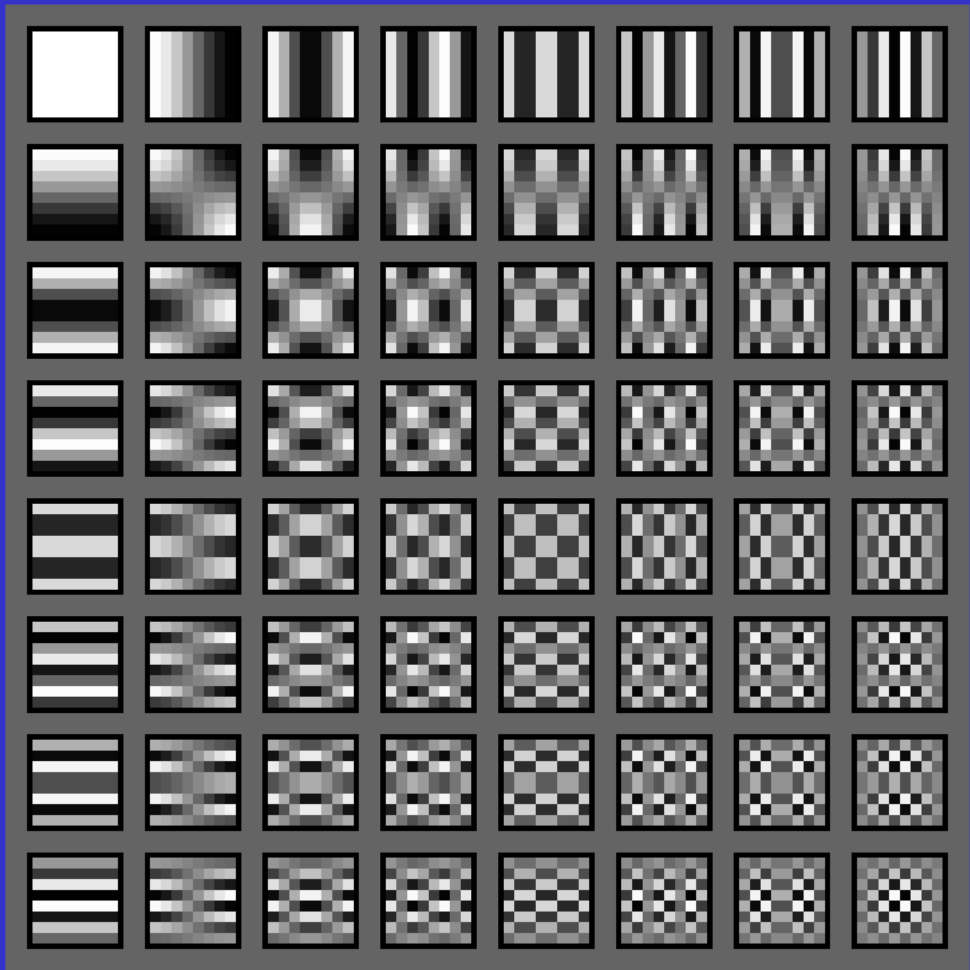
当 N=8时，DCT 变换矩阵如下：

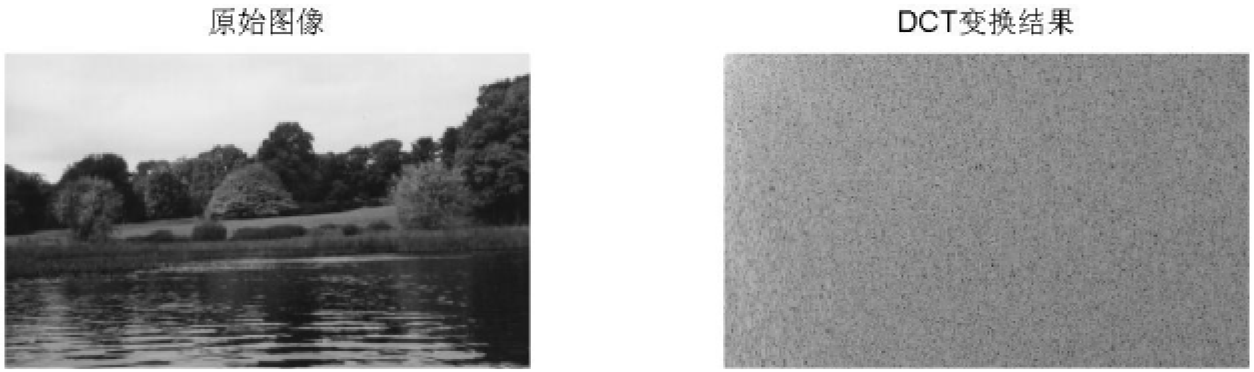
$$
\mathbf{T}^t = \begin{bmatrix}
0.354 & 0.49 & 0.462 & 0.416 & 0.354 & 0.278 & 0.191 & 0.098 \\
0.354 & 0.416 & 0.191 & -0.098 & -0.354 & -0.49 & -0.462 & -0.278 \\
0.354 & 0.278 & -0.191 & -0.49 & -0.354 & 0.098 & 0.462 & 0.416 \\
0.354 & 0.098 & -0.462 & -0.278 & 0.354 & 0.416 & -0.191 & -0.49 \\
0.354 & -0.098 & -0.462 & 0.278 & 0.354 & -0.416 & -0.191 & 0.49 \\
0.354 & -0.278 & -0.191 & 0.49 & -0.354 & -0.098 & 0.462 & -0.416 \\
0.354 & -0.416 & 0.191 & 0.098 & -0.354 & 0.49 & -0.462 & 0.278 \\
0.354 & -0.49 & 0.462 & -0.416 & 0.354 & -0.278 & 0.191 & -0.098
\end{bmatrix}
$$

一维DCT 基向量（ N=8）



二维 DCT 基图像（N=8）




原始图像
DCT变换结果

小于10的系数取为0

重构后的图像

DCT变换编码

- DCT变换编码方法：

  1）编码过程：

原图像 → DCT变换 → 除以量化矩阵 → 取整 → 压缩图像

  2）解码过程：

压缩图像 → 乘以量化矩阵 → DCT逆变换 → 取整 → 解压图像

DCT变换编码

例:

原图像为:
$$F=\begin{bmatrix}59&60&58&57\\61&59&59&57\\62&59&60&58\\59&61&60&56\end{bmatrix}$$

DCT变换

$$D_1=\begin{bmatrix}120.5&119.5&118.5&114.0\\-0.27&-0.65&-1.58&0.38\\-2.50&1.50&-0.50&-1.00\\0.65&-0.27&0.11&0.92\end{bmatrix}$$

Huffman:42bits

除以量化矩阵，取整

$$C=\begin{bmatrix}16&11&11&16\\12&12&14&19\\14&13&16&24\\14&17&22&29\end{bmatrix}$$

$$D=\begin{bmatrix}8&11&11&7\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

Huffman:28bits

DCT变换编码

原图

解压图

图像信号的正交变换－离散K-L变换

- 一般变换的变换核矩阵是固定不变的，而K-L变换则随各批图像的统计性质不同而有不同的变换核矩阵，即变换核矩阵是由某批图像的统计性质来确定的。
- 比如一幅图像通过卫星传送了N次，这时由于电波传播的影响，N幅图像互有差异，这样也可以进行统计。

- 一维K-L变换
- 对随机向量的集合， $f=\{f_1,f_2,\cdots,f_n\}$
- 其均值向量为 $\mu=E\{f\}\approx\frac{1}{N}\sum_{i=1}^{N}f_i$
- 其斜方差阵 $\sum_f=E\{(f-\mu)(f-\mu)^T\}\approx\frac{1}{N}[\sum_{i=1}^{N}(f_if_i^T)]-\mu\mu^T$
- 特征值和特征向量 $\sum_f\phi_i=\lambda_i\phi_i, 0\le i\le N-1$

$$F=\phi^{*T}f$$

$$\phi=[\phi_1,\phi_2,\cdots,\phi_n]$$

$$f=\phi F=\sum_{i=1}^{N-1}F(i)\phi_i$$

二维$K-L$变换：

图象集合 $f(x,y)=\{f_1(x,y),\cdots,f_l(x,y)\}$

$$f_i=\begin{bmatrix} f_{i,0} \\ f_{i,1} \\ \\ f_{i,M-1} \end{bmatrix} \quad f_{i,j}=\begin{bmatrix} f_i(j,0) \\ f_i(j,1) \\ \\ f_i(j,N-1) \end{bmatrix}$$

f向量的斜方差阵 $\sum_f=E\{(f-\mu_f)(f-\mu_f)^T\}$

$\mu_f=E\{f\}$为均值向量， e_i和λ_i是$\sum_f$的特征向量和对应的特征值，

设它们按照降序排列，即$\lambda_1\succ\lambda_2\succ\cdots\succ\lambda_{MN}$,相应的特征向量为$e_1,e_2,\cdots,e_{mn}$

$$A=\begin{bmatrix} e_{11} & e_{12} & e_{1,MN} \\ e_{21} & e_{22} & e_{2,MN} \\ \\ e_{MN,1} & e_{MN,2} & e_{MN,MN} \end{bmatrix}$$

则$K-L$变换写成矩阵 $F=A(f-\mu_f)$

图像信号的正交变换-沃尔什和哈达玛变换

沃尔什变换包括+1和-1构成的完备正交基，它与数字逻辑的两个状态相对应，因此更加利于计算机处理，它的存储空间和运算速度提高了很多，这一点对图像的实时处理非常有意义。

一维离散沃尔什变换：

$$g(x,u)=\frac{1}{N}\prod_{i=0}^{n}(-1)^{b_i(x)b_{n-1-i}(u)}$$

$$x=01,2,\ldots,\ N-1$$

$$u=0,1,2,\ldots,\ N-1$$

$$N=2^n$$

图像信号的正交变换-沃尔什和哈达玛变换

$b_k(z)$是z的二进制表示的第k位值，或者是0，或者是1。
如z=6，其二进制表示110，则存在：

$$b_0(z)=0,\ b_1(z)=1,\ b_2(z)=1$$

一维离散沃尔什变换定义：

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

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

图像信号的正交变换-沃尔什和哈达玛变换

当n=2,N=4时，沃尔什变幻核是:

$$G_4 = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$W = G \times f$$

图像信号的正交变换-沃尔什和哈达玛变换

二维离散沃尔什变换定义：

$$W(u,v)=\frac{1}{N^2}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1}f(x,y)\prod_{i=0}^{n}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

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

$$W=G\times f\times G$$

图像信号的正交变换-沃尔什和哈达玛变换

例：

$$f=\begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\qquad G_4=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&1&-1\\1&-1&-1&1\end{bmatrix}$$

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

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

图像信号的正交变换-沃尔什和哈达玛变换

哈达玛变换是沃尔什变换的特例，其优点在于变换核函数具有简单的递推关系。

一维离散哈达玛变换：

$$g(x,u)=\frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

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

图像信号的正交变换-沃尔什和哈达玛变换

哈达玛变换是沃尔什变换的特例，其优点在于变换核函数具有简单的递推关系。

一维离散哈达玛变换的递推关系：

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

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

图像信号的正交变换-沃尔什和哈达玛变换

二维离散哈达玛变换定义：

$$H(u,v)=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)(-1)^{\sum_{i=0}^{n-1}[b_i(x)b_i(u)+b_i(y)b_i(v)]}$$

图像信号的正交变换-沃尔什和哈达玛变换

例：

$$f=\begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\qquad G_4=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}$$

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

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

图像信号的正交变换-图像的正交基表示

图像信号的变换域表示：

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)g(x,y,u,v)$$

$$f(x,y)=\sum_{u=0}^{M}\sum_{v=0}^{N-1}F(u,v)h(x,y,u,v)$$

$g(x,y,u,v)$ 正变换核函数

$h(x,y,u,v)$ 逆变换核函数

图像信号的正交变换-图像的正交基表示

图像信号的变换域表示：

$g(x,y,u,v)$ 正变换核函数

$h(x,y,u,v)$ 逆变换核函数

$$g(x,y,u,v) = P(u,x)Q(v,y)$$

$$h(x,y,u,v) = P_1(u,x)Q_1(v,y)$$

$$F = PfQ^T$$

$$f = P_1 f Q_1^T$$

图像信号的正交变换-图像的正交基表示

基本图像和基本频谱：

$$P=[P_1,P_2,\dots,P_{N-1}]$$

$$Q=[Q_1,Q_2,\dots,Q_{N-1}]$$

$$F=\sum_i\sum_j f_{ij}P_iQ_j^T$$

$$f=\sum_i\sum_j F_{ij}P^*{}_iQ^{*T}_j$$

- FFT变换公式及几种应用
- DCT 变换公式及编码过程