

雷达影像的图像处理技术

谢磊 博士 副研究员

国家水运安全工程技术研究中心

2021-03-08

第二章 图像处理技术简介 (二)

一、数字图像空域处理程序示范

1、作业示范

一、原始图像显示

代码如下：

```
I=imread('D:/作业图片/1. jpg');  
imshow(I) ;
```

1、作业示范

二、图像反转

(1) 全反转代码如下：

```
I=imread('D:/作业图片/5. jpg');  
imshow(I);  
J=255-I;  
figure, imshow(J);
```

1、作业示范

二、图像反转

(1) 半反转代码如下：

```
I=imread('D:/作业图片/5. jpg');  
imshow(I);  
J1=abs(128-I);  
figure, imshow(J1);  
J2=uint8(abs(128-double(I)));  
figure, imshow(J2);
```

1、作业示范

三、噪声添加和滤波处理

(1) 添加椒盐噪声及中值滤波代码如下:

```
I=imread('D:/作业图片/6. jpg');  
imshow(I);  
J1=rgb2gray(I);  
J2=imnoise(J1, 'salt & pepper', 0.5);  
figure, imshow(J2);  
I_Filter1=medfilt2(J2, [3 3]); %窗口大小为3×3  
figure, imshow(I_Filter1);  
I_Filter2=medfilt2(J2, [5 5]); %窗口大小为5×5  
figure, imshow(I_Filter2);  
I_Filter3=medfilt2(J2, [7 7]); %窗口大小为7×7  
figure, imshow(I_Filter3);
```

1、作业示范

三、噪声添加和滤波处理

(2) 添加高斯噪声及中值滤波代码如下:

```
I=imread('D:/作业图片/5. jpg');  
imshow(I);  
J1=rgb2gray(I);  
J2=imnoise(J1, 'gaussian', 0, 0.1);  
figure, imshow(J2);  
I_Filter1=medfilt2(J2, [3 3]); %窗口大小为3×3  
figure, imshow(I_Filter1);  
I_Filter2=medfilt2(J2, [5 5]); %窗口大小为5×5  
figure, imshow(I_Filter2);  
I_Filter3=medfilt2(J2, [7 7]); %窗口大小为7×7  
figure, imshow(I_Filter3);
```


1、作业示范

四、通过Sobel算子和拉普拉斯进行边缘提取

```
I=imread('D:/作业图片/5. jpg');  
subplot(2,2,1), imshow(I);  
title('原始图像');  
axis([50,250,50,200]);  
grid on;                                %显示网格线  
axis on;                                %显示坐标系  
I1=im2bw(I);  
subplot(2,2,2), imshow(I1);  
title('二值图像');  
axis([50,250,50,200]);  
grid on;                                %显示网格线  
axis on;                                %显示坐标系  
H=fspecial('sobel');                    %选择sobel算子
```

1、作业示范

```
J=filter2(H,I1);           %卷积运算
subplot(2,2,3), imshow(J);
title('sobel算子锐化图像');
axis([50,250,50,200]);
grid on;                    %显示网格线
axis on;                    %显示坐标系
h=[0 1 0,1 -4 1,0 1 0];    %拉普拉斯算子
J1=conv2(single(I1),single(h),'same'); %卷积运算
subplot(2,2,4), imshow(J1);
title('拉普拉斯算子锐化图像');
axis([50,250,50,200]);
grid on;                    %显示网格线
axis on;                    %显示坐标系
```

二、傅里叶变换

1、什么是频域

一个正常的世界，我们看到的事件都以时间贯穿，人的身高、汽车的轨迹、星系的运动都是随着时间发生改变。这种以时间作为参照来观察动态世界的方法我们称其为时域分析。

我们也想当然的认为，世间万物都在随着时间不停的改变，并且永远不会静止下来。但是换用另一种方法来观察世界的话，我们会发现世界是永恒不变的，这个静止的世界就叫做频域。

1、什么是频域

下面是一段音乐，也是一个随着时间变化的震动：



对于音乐家来说，音乐有更直观的理解：

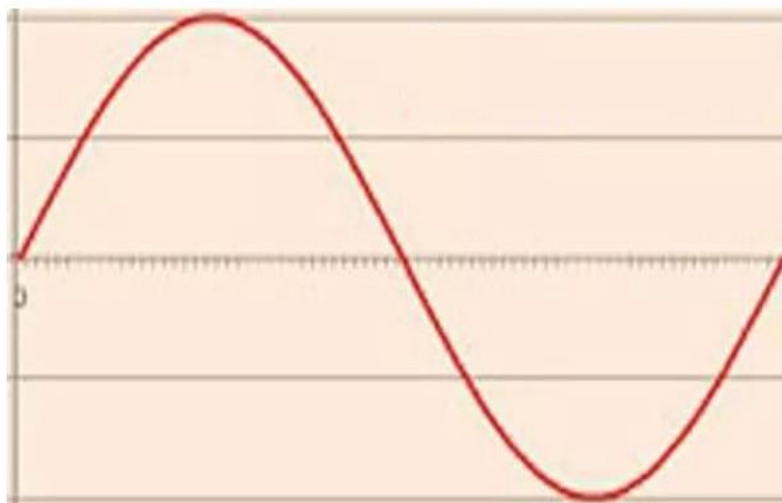


上图是音乐在时域的样子，而下图则是音乐在频域的样子。所以频域这一概念对大家都不陌生，只是从来没意识到而已。

1、什么是频域

在时域，我们可以观察到钢琴的琴弦一会上一会下的摆动，就如同一支涌动的波；而在频域，那只是一个永恒的音符，代表了一个特定的频率。

时域:



频域:



而贯穿了时域与频域的一个重要方法，就是傅里叶分析。

2、傅里叶级数 (Fourier Series) 的频谱

傅里叶生平：1768年生于法国，1807年提出“任何周期信号都可用正弦函数级数表示”，1822年首次发表在“热的分析理论”一书中，1829年狄里赫利第一个给出收敛条件。



2、傅里叶级数 (Fourier Series) 的频谱

傅里叶1795年任巴黎综合工科大学助教，1798年随拿破仑军队远征埃及，受到拿破仑器重，回国后被任命为格伦诺布尔省省长，1817年当选为巴黎科学院院士，1822年成为科学院终身秘书。

傅里叶1807年发表的论文，主要是运用正弦曲线来描述温度分布，但是论文中提了一个在当时具有争议性的决断：任何连续周期信号可以由一组适当的正弦曲线组合而成。当时审查这个论文的人，其中有一位是历史上著名的数学家拉格朗日 (Joseph Louis Lagrange, 1736-1813) 和拉普拉斯 (Pierre Simon de Laplace, 1749-1827)，当拉普拉斯和其它审查者投票通过并要发表这个论文时，拉格朗日坚决反对。在拉格朗日此后生命的六年中，他坚持认为傅立叶的方法无法表示带有棱角的信号，如在方波中出现非连续变化斜率。因此，尽管傅里叶1811年又提交了经修改的论文，且该文获科学院大奖，却仍未正式发表。直到1822年，傅里叶终于出版了专著《热的解析理论》 (Analytical theory of heat, Paris, 1822)。

2、傅里叶级数 (Fourier Series) 的频谱

傅里叶1830年去世，此时距离他最初发表这篇论文已经过了23年了。当然，幸运抑或不幸的是，傅立叶还有其它事情可忙。由于他参加了政治运动，特别是曾经随拿破仑远征埃及，所以法国大革命后，因为会被推上断头台而一直在东躲西藏，所以这篇论文并没有造成太大的遗憾。

现在来看，拉格朗日是正确的，正弦和余弦曲线无法组合成一个带有棱角的信号。但是，我们可以用正弦和余弦曲线来非常逼近地表示它，逼近到两种表示方法不存在能量差别，基于此，傅立叶又是对的。

2、傅里叶级数（Fourier Series）的频谱

傅里叶的贡献在于：他指出任何周期函数都可以表示为不同频率的正弦和/或余弦和的形式，每个正弦和/或余弦和乘以不同的系数（现在称这个和为傅里叶级数）。无论函数多么复杂，只要它是周期的，并且满足某些软的数学条件，都可以用这样的和来表示。

在这个概念提出之后，傅里叶的论文仍然受到质疑，并被全世界的数学家“纠正”了100年。当时，数学思想中函数的规律性是占主导的，因此，“复杂函数可以由简单的正弦和余弦之和来表示”的概念很不直观，遭到怀疑不足为奇。

其实，这就是无穷级数的一种：傅里叶级数。无穷级数的思想恰恰是通过简单的函数形式，逼近复杂的函数，包括泰勒级数、幂级数都是这个道理。

2、傅里叶级数 (Fourier Series) 的频谱

- “周期信号都可表示为谐波关系的正弦信号的加权和”——傅里叶的第一个主要论点
- “非周期信号都可用正弦信号的加权积分表示”——傅里叶的第二个主要论点

$$f_1(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_1 t + b_n \sin n\omega_1 t)$$

直流
分量

基波分量
 $n=1$

谐波分量
 $n>1$

$$\omega_1 = \frac{2\pi}{T_1}$$

$$n\omega_1$$

2、傅里叶级数 (Fourier Series) 的频谱

直流系数 $a_0 = \frac{1}{T_1} \int_{t_0}^{t_0 + T_1} f(t) dt$

余弦分量系数 $a_n = \frac{2}{T_1} \int_{t_0}^{t_0 + T_1} f(t) \cdot \cos n\omega_1 t dt$

正弦分量系数 $b_n = \frac{2}{T_1} \int_{t_0}^{t_0 + T_1} f(t) \cdot \sin n\omega_1 t dt$

2、傅里叶级数 (Fourier Series) 的频谱

傅里叶有限级数：

如果要完全逼近，则需要 $n=\infty$ ；

实际中取 $n=N$ ， N 是有限整数。

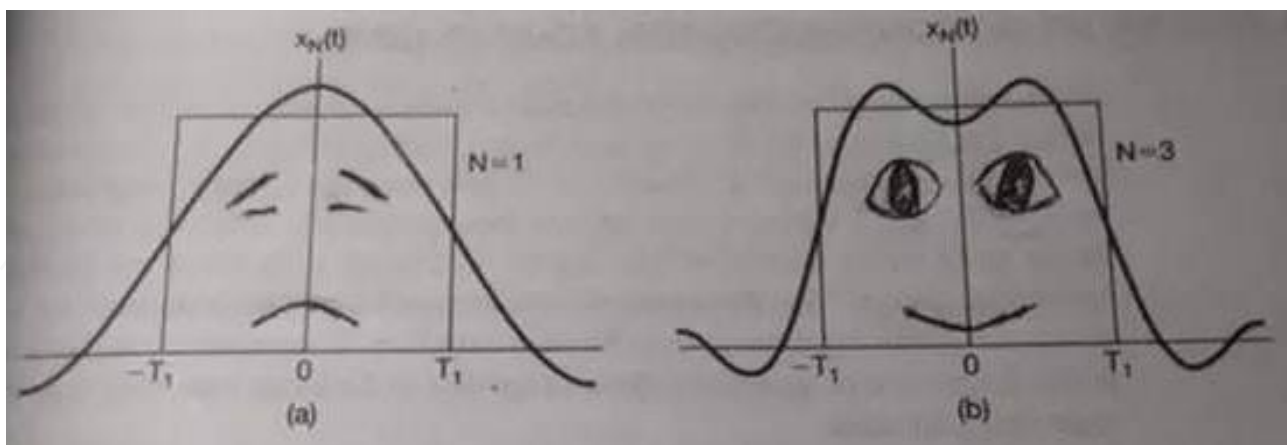
如果 N 愈接近 n ，则其均方误差愈小，若用 $2N+1$ 项逼近，则：

$$S_N(t) = a_0 + \sum_{n=1}^N (a_n \cos \omega_1 t + b_n \sin \omega_1 t)$$

简单的说，就是我们能够用前面说的正弦曲线波叠加出任意一个周期函数来。

即使是非周期函数，也可以通过提供一个非常大的 n 来予以逼近。比如说，可以逼近一个带有90度角的矩形波来。

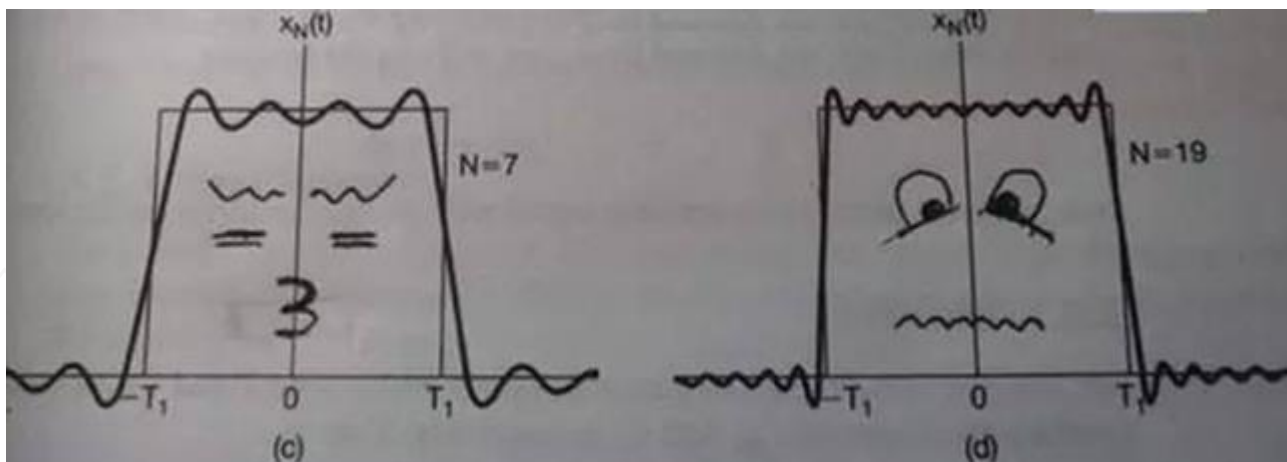
2、傅里叶级数 (Fourier Series) 的频谱



第一幅图是一个郁闷的余弦波： $A_1 \cos(x)$

第二幅图是2个卖萌的余弦波的叠加： $A_1 \cos(x) + A_3 \cos(3x)$

2、傅里叶级数 (Fourier Series) 的频谱



第三幅图是4个哆嗦的余弦波的叠加：

$$A_1 \cos(x) + A_3 \cos(3x) + A_5 \cos(5x) + A_7 \cos(7x)$$

第四幅图是10个余弦波的叠加：……

随着正弦波（余弦波）的数量逐渐增长，他们最终会叠加成一个接近于标准的矩形。

2、傅里叶级数 (Fourier Series) 的频谱

对称方波有限项的傅里叶级数和均方误差如下：

➤ **N=1**

$$S_2 = \frac{2E}{\pi} (\cos \omega_1 t) \quad E_1 \approx 0.05E^2$$

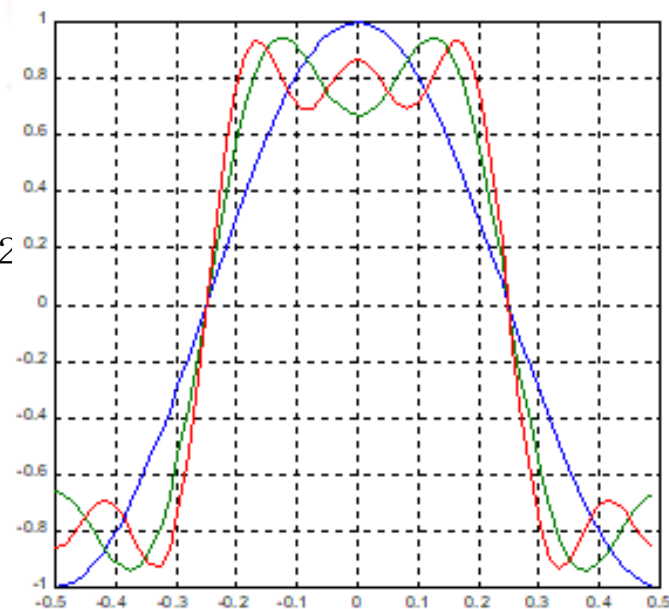
➤ **N=2**

$$S_2 = \frac{2E}{\pi} (\cos \omega_1 t - \frac{1}{3} \cos 3\omega_1 t) \quad E_2 = 0.02E^2$$

➤ **N=3**

$$S_3 = \frac{2E}{\pi} (\cos \omega_1 t - \frac{1}{3} \cos 3\omega_1 t + \frac{1}{5} \cos 5\omega_1 t)$$

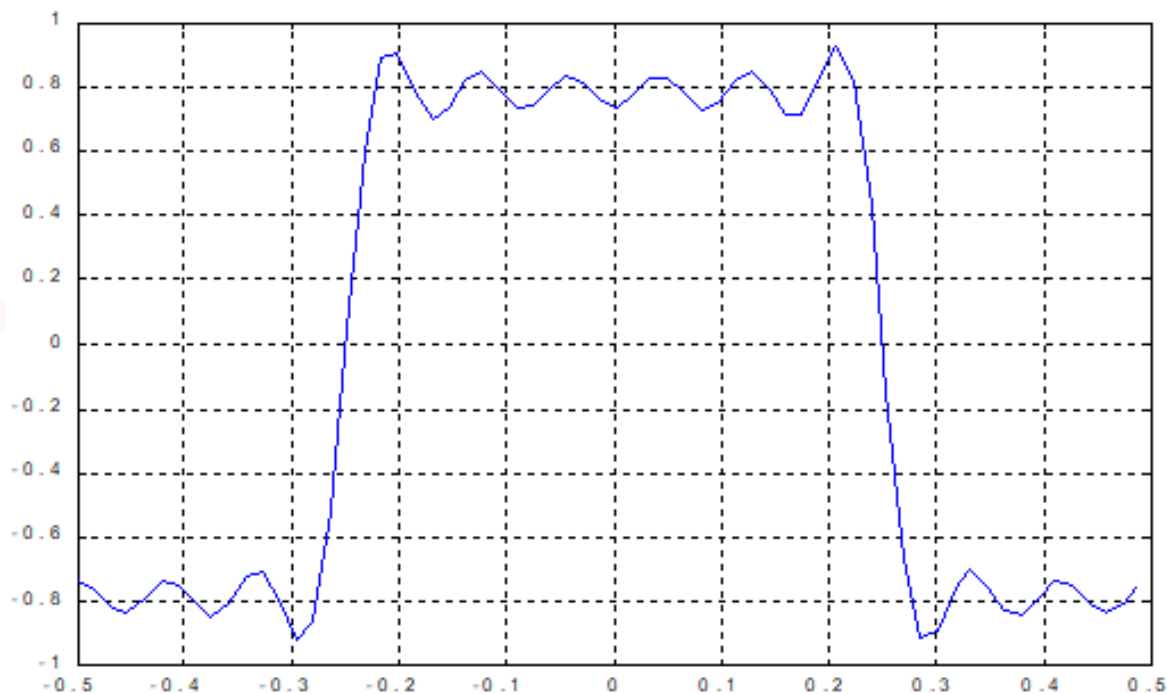
$$E_3 = 0.01E^2$$



2、傅里叶级数 (Fourier Series) 的频谱

有限项的N越大，误差越小，N=11时的傅里叶级数如下：

$$S_9 = \frac{2E}{\pi} \left(\cos\omega_1 t - \frac{1}{3} \cos 3\omega_1 t + \frac{1}{5} \cos 5\omega_1 t + \cdots - \frac{1}{11} \cos 11\omega_1 t \right)$$



2、傅里叶级数 (Fourier Series) 的频谱

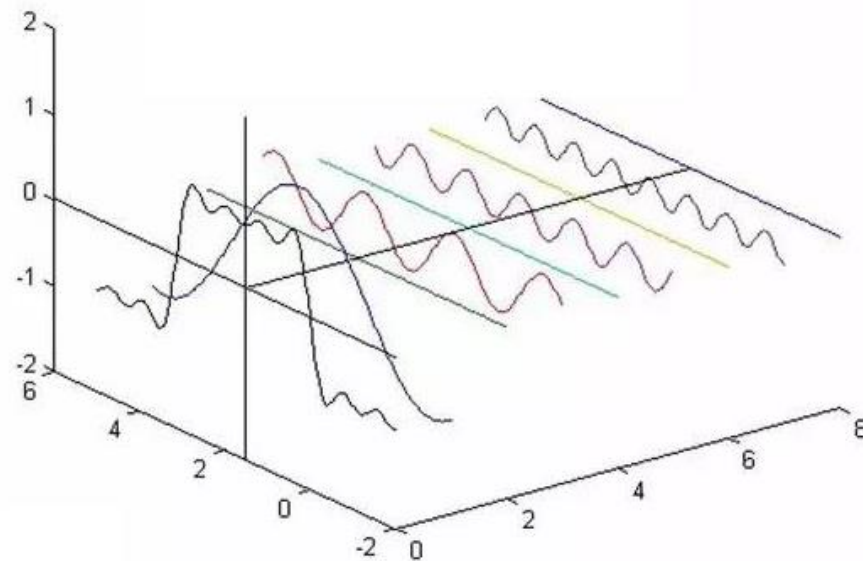
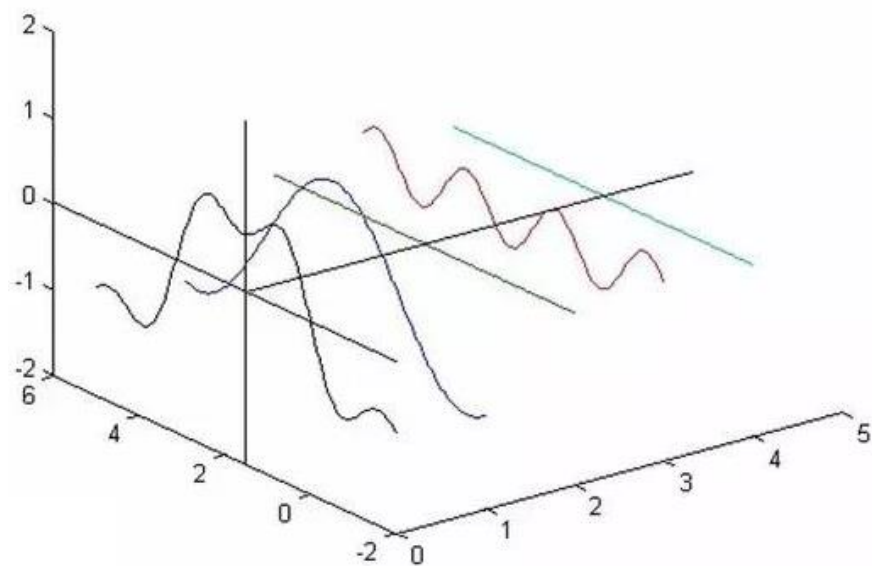
随着叠加的递增，所有正弦波中上升的部分逐渐让原本缓慢增加的曲线不断变陡，而所有正弦波中下降的部分又抵消了上升到最高处时继续上升的部分使其变为水平线。慢慢的，一个矩形方波就这么叠加而成了。

但是，要多少个正弦波叠加起来才能形成一个标准的90度角的矩形波呢？答案是无穷多个，也就是说，只有当傅里叶级数中的 $n=\infty$ 时，才能实现完全逼近。

不仅仅是矩形，我们能想到的任何波形都是可以采用如此方法，用正弦波叠加形成的。

2、傅里叶级数 (Fourier Series) 的频谱

还是上图的正弦波累加成矩形波，我们换一个角度来看：



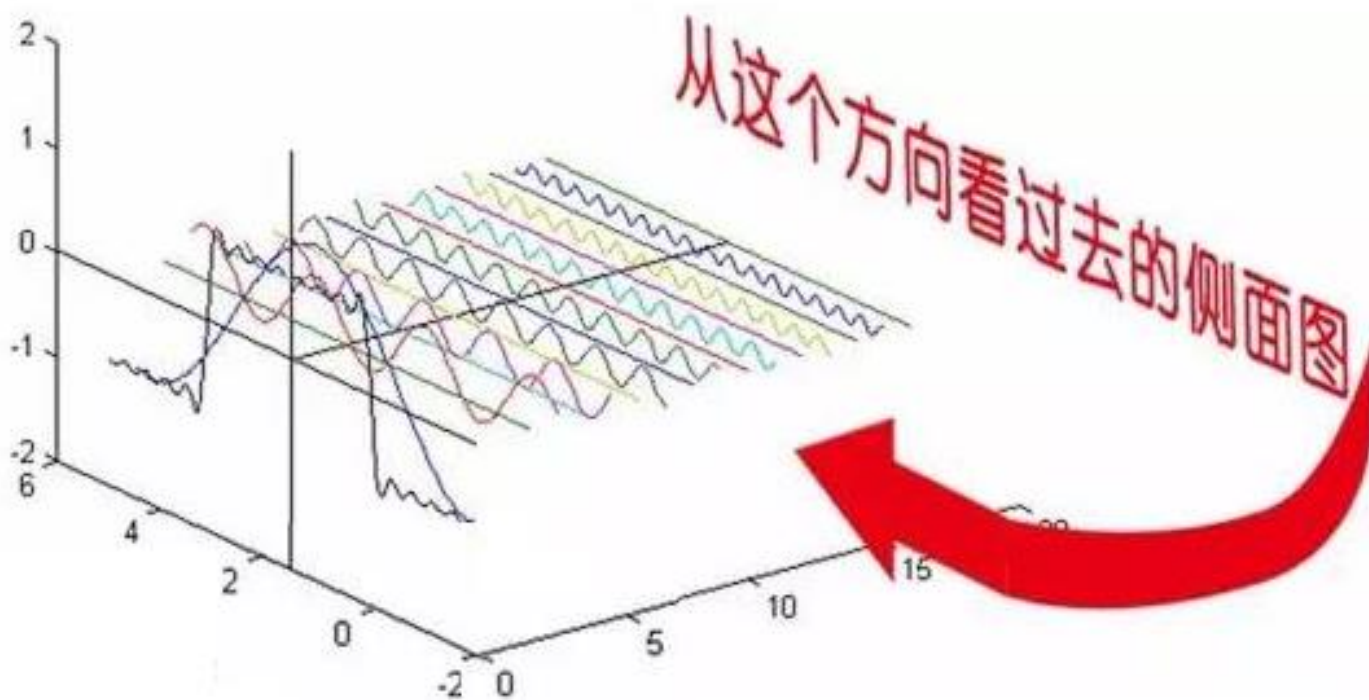
2、傅里叶级数 (Fourier Series) 的频谱

在上面几幅图中，最前面的那条曲线就是所有正弦波叠加而成的总和，也就是越来越接近矩形波的那个图形。而后面依次排列而成的正弦波就是组合为矩形波的各个分量。这些正弦波按照频率从低到高从前向后排列开来，而每一个波的振幅都是不同的。每两个正弦波之间都还有一条直线，那并不是分割线，而是振幅为0的正弦波！也就是说，为了组成特殊的曲线，有些正弦波成分是不需要的。

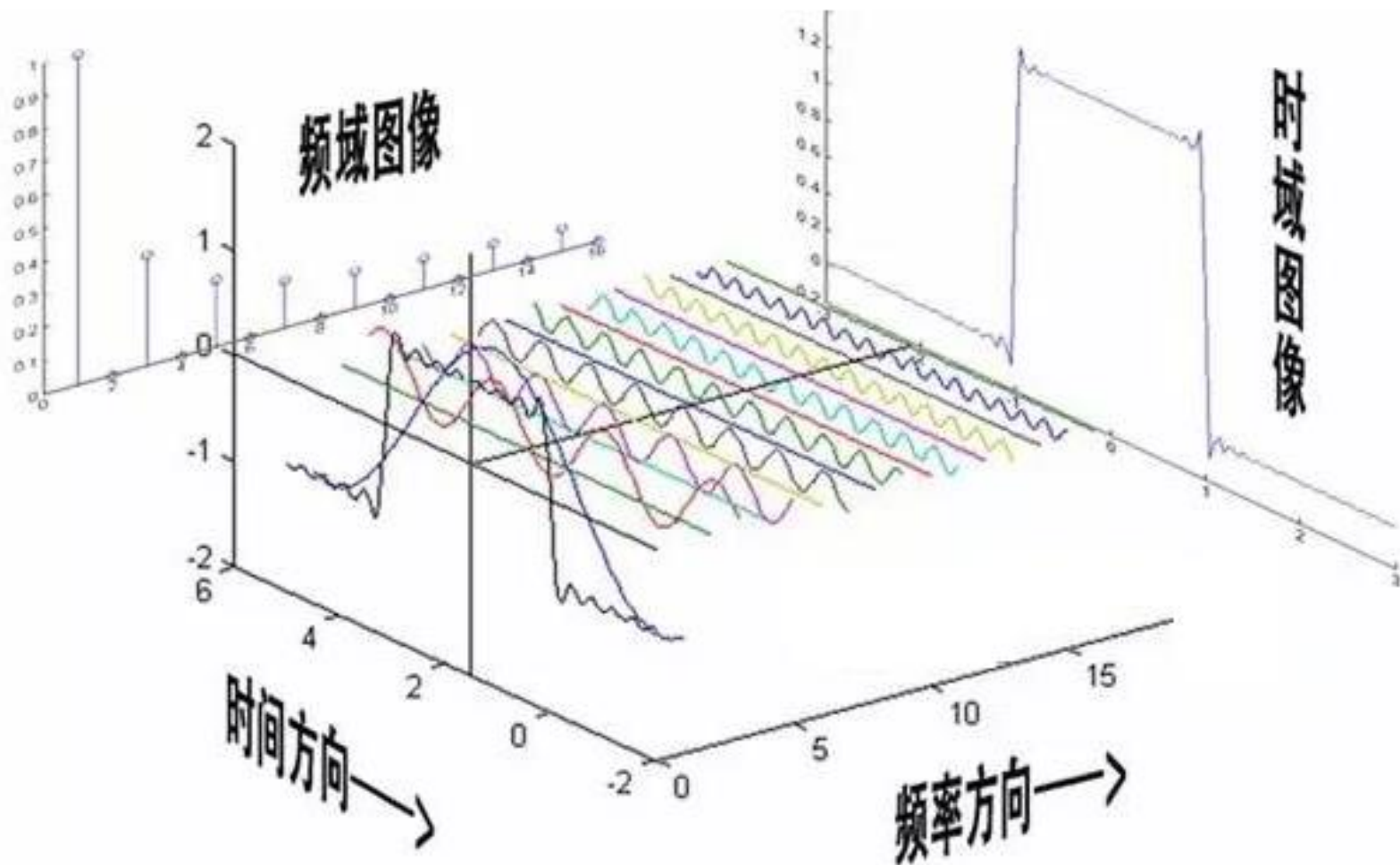
在这里，不同频率的正弦波我们称为频率分量。

2、傅里叶级数 (Fourier Series) 的频谱

现在让我们再换一个角度来看看频域图像，也就是俗称的频谱：



2、傅里叶级数 (Fourier Series) 的频谱



3、傅里叶变化的特点

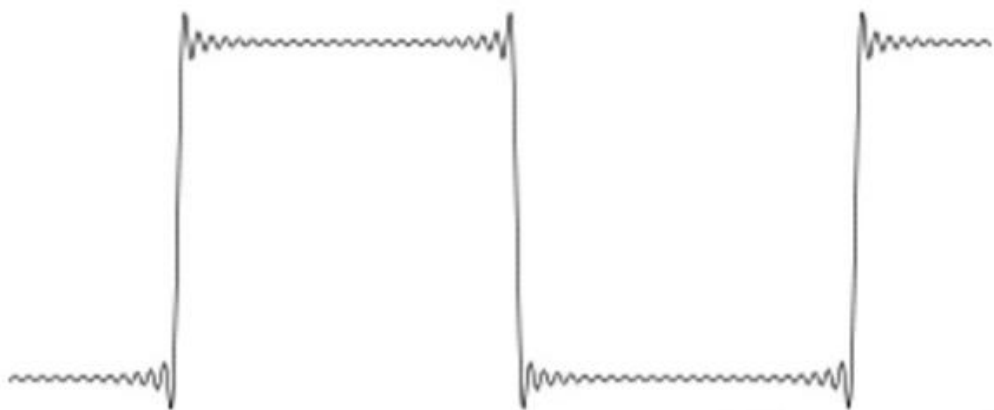
- N 越大，越接近方波；
- 快变信号，高频分量，主要影响跳变沿；
- 慢变信号，低频分量，主要影响顶部；
- 任一分量的幅度或相位发生相对变化时，波形将会失真；
- 存在吉伯斯现象。

4、吉伯斯现象

- 吉伯斯现象Gibbs phenomenon（又叫吉布斯效应）：将具有不连续点的周期函数（如矩形脉冲）进行傅立叶级数展开后，选取有限项进行合成。当选取的项数越多，在所合成的波形中出现的峰起越靠近原信号的不连续点。当选取的项数很大时，该峰起值趋于一个常数，大约等于总跳变值的9%。这种现象称为吉伯斯现象。
- 这个现象的含义是：一个不连续信号 $x(t)$ 的傅里叶级数的截断近似 $x_N(t)$ ，一般来说，在接近不连续点处将呈现高频起伏和超量，而且，若在实际情况下利用这样一个近似式的话，就应该选择足够大的 N ，以保证这些起伏拥有的总能量可以忽略。

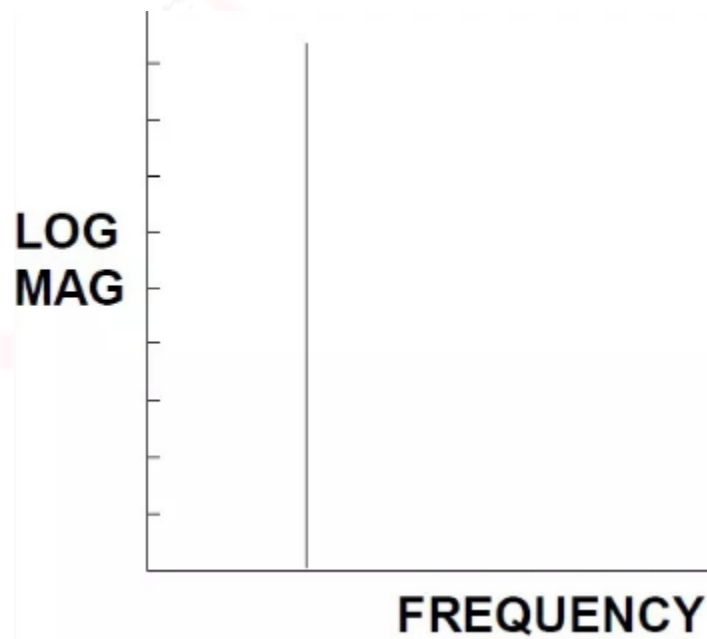
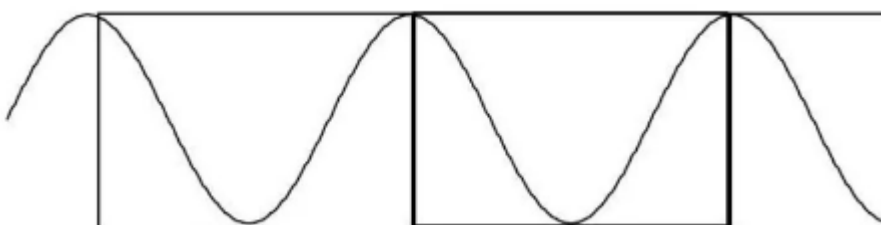
4、吉伯斯现象

- 吉伯斯现象在一维信号上的典型体现，就是方波脉冲的转折处信号出现明显的振荡，以及某些时候力脉冲也会在脉冲的末端位置出现振荡。



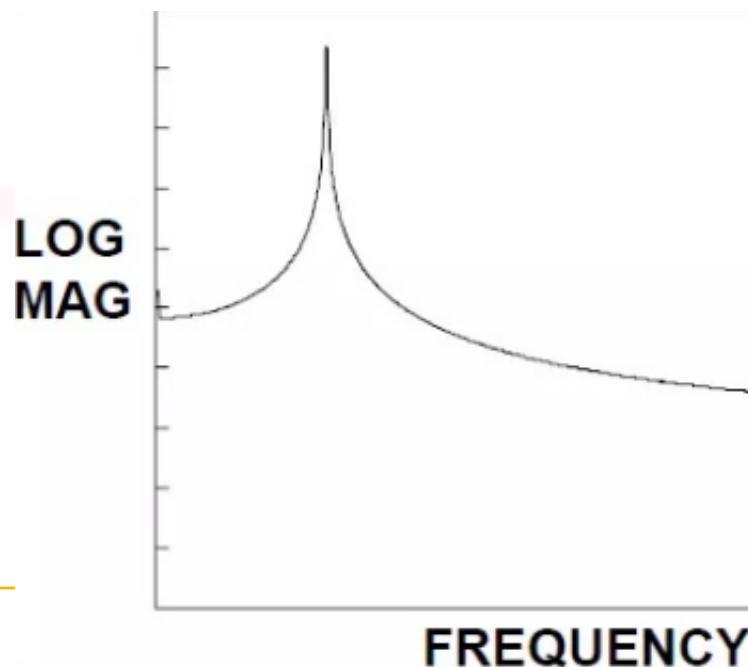
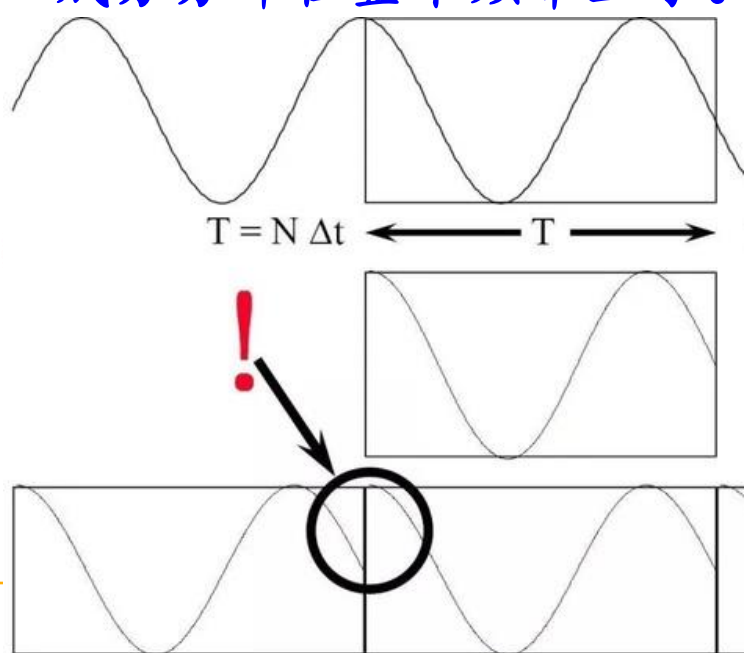
4、吉伯斯现象

- 我们对信号进行FFT处理时，需要做截断，而截断又分为周期截断和非周期截断。假设原始信号是单频正弦波，如果截断的时间长度刚好是信号周期的整数倍，那么，截断后的每帧数据仍是周期信号，可重构得到原始的周期信号。



4、吉伯斯现象

- 倘若信号截断的时间长度不为原始正弦信号周期的整数倍，那么，截断后的信号则不为周期信号。对同一个正弦信号进行非周期截断，截断后的信号起始时刻和结束时刻的幅值明显不等，将这个信号再进行重构，在连接处信号的幅值不连续，出现阶跃，如下图中黑色圆圈区域所示。对这个信号进行FFT分析，得到的频谱就不再是正确的线状谱了，而是出现了明显的频率拖尾效应，频率成分分布在整个频带上了。



4、吉伯斯现象

- ▶ 倘若信号截断的时间长度不为原始正弦信号周期的整数倍，那么，截断后的信号则不为周期信号。对同一个正弦信号进行非周期截断，截断后的信号起始时刻和结束时刻的幅值明显不等，将这个信号再进行重构，在连接处信号的幅值不连续，出现阶跃，如下图中黑色圆圈区域所示。对这个信号进行FFT分析，得到的频谱就不再是正确的线状谱了，而是出现了明显的频率拖尾效应，频率成分分布在整个频带上了。

4、吉布斯现象

- 在时域描述一个不连续的信号时，要求信号有无穷的频率成分。但实际情况是不可能采集到无穷的频率成分。信号采集系统只能采集一定频率范围内的信号，这将导致出现频率截断，考虑到吉布斯现象可知，频率截断会引起时域信号产生“振铃效应”。
- 任何突然不连续或阶跃信号总是会存在出现吉布斯现象，现实世界中可能展示出吉布斯现象的信号包括汽车驶过坑时产生的冲击、力脉冲、爆炸声或者高尔夫球杆击球时产生的振动等等。
- 在数字信号采集系统中，在信号的每一个阶跃处，振铃使得信号出现不一致。信号的幅值出现变化或者完全不变化，这依赖于信号的瞬变时刻与数据采样点数的相对关系。当使用少于合适数目的频率成分来描述信号时，就会产生振铃效应。之前我们用不同数量的正弦波来描述方波信号，可以看出，随着正弦波数量的增加，叠加后的信号越来越接近方波信号，振铃现象越来越弱，振荡的幅值越来越小，持续时间越来越短，信号的斜率越来越陡峭。

4、吉伯斯现象

- 图像的傅立叶变换，由于其变换本身有多种成熟的快速算法（FFT算法），而且性能接近于最佳，从而获得较早的也比较广泛的研究。它的不足之处在于：相邻子图像数据在各个边界不连续造成的所谓Gibbs现象。这是由于图像数据的二维傅立叶变换实质上是一个二维图像的傅立叶展开式。当然这个二维图像应被认为是周期性的。由于子图像的变换系数在边界不连续，而将造成复原的子图像在其边界也不连续。于是由复原子图像构成的整幅复原图像将呈现隐约可见的以子图像尺寸为单位的方块状结构，影响整个图像质量。当子图像尺寸较小时更为严重。
- 解决这个Gibbs现象的方法是后来研究出来的二维余弦变换(DCT)代替二维傅立叶变换。基本思路为：用一个对称的 $2N \times 2N$ 像素的子图像代替原来 $N \times N$ 子图像。由于对称性，子图像作二维傅立叶变换，其变换系数将只剩下实数的余弦项。这样，即可消除Gibbs现象。

4、吉伯斯现象

- 本质上，离散余弦变换 (Discrete Cosine Transform) 也是离散傅里叶变换 (Discrete Fourier Transform)，但是只有实数部分。有这样一个性质：如果一个信号函数在给定区间内满足狄利赫里条件，且该信号为实对称函数，则可以展开成仅含有余弦项的傅里叶级数，即离散余弦变换。所以，我们在构造离散信号的周期函数的时候，要对其进行偶延拓。
- DCT变换较DFT变换具有更好的频域能量聚集度，能够把图像更重要的信息聚集在一块，对于那些不重要的频域区域和系数就能够直接裁剪掉，因此，DCT变换非常适合于图像压缩算法的处理，例如现在的jpeg就是使用了DCT作为图像压缩算法。

4、吉伯斯现象

- DCT没有虚部，本质是傅里叶变换（无损）→图像从空间域快速（没有虚部）地变换到了频率域。

original Lena



DCT of Lena



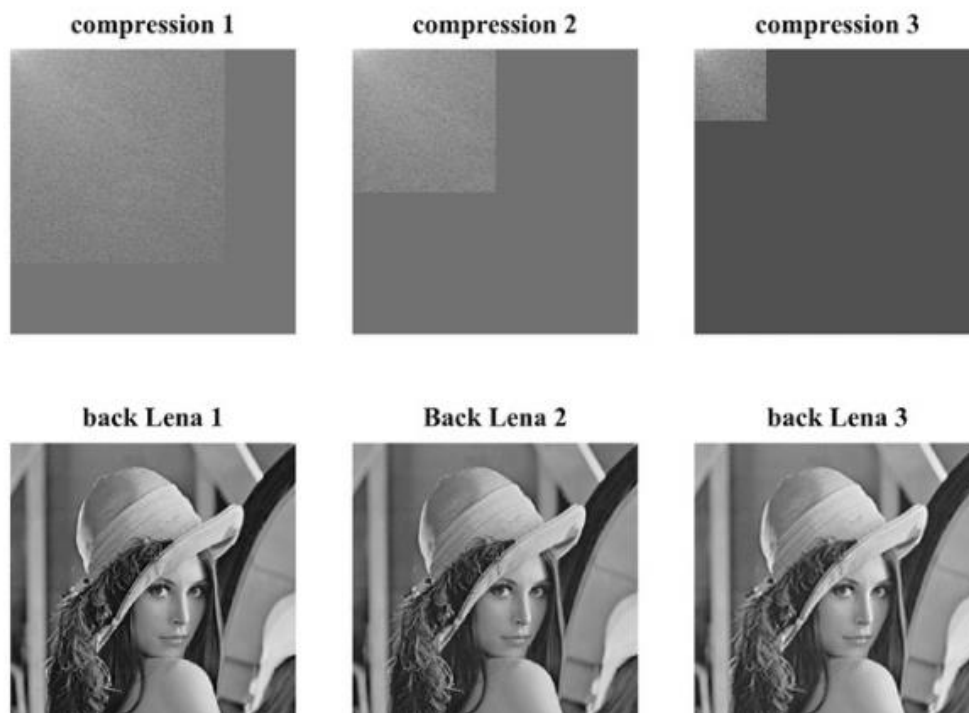
iDCT of Lena



- 从上图中可以看到，左上角亮度高，即Lena的主要能量集中在左上角；而且，从其逆变换可看出，DCT为无损变换。

4、吉伯斯现象

- 为了确定哪种能量、有多少能量集中在左上角，可以对图中的DCT变化图以128像素为步长进行了裁剪，然后再对其进行逆DCT变换。



- 随着压缩比例的增大，图片中的细节部分逐渐消失，明显的高频信息丢失了。因此，DCT使图片的低频成分集中在左上角。

4、吉伯斯现象

- DCT视频编码中地应用最为广泛。1984年，H. 261编码标准开始研究。1990年，H. 261编码标准由国际电信联盟（ITU-T）发布。1993年，MPEG-1编码标准由国际标准化组织/国际电子学委员会（ISO/IEC）发布。自此，H. 26X系列与MPEG-X系列开始了各自的更新换代，当然期间也有合作交易，比如大家比较熟悉的MPEG-2与大家可能不熟悉的H. 262，其实是一个东西；又比如MPEG-4，包括了（早期的）MPEG-4、MPEG-4/AVC（=H. 264）、MPEG-4/HEVC（=H. 265）三个标准。这中间就差了H. 263与MPEG-3没有提。前者随着技术的进步退出了历史舞台，后者则被毙掉了。
- 从H. 261开始，离散余弦变换就被应用于视频编码中。当技术发展到H. 264与H. 265，就具体的数学计算而言，H. 261创立的分块编码已经被后续标准沿用下去了，如H. 264在亮度平面上的基本分块为8x8像素块。

例：函数的傅立叶变换

一、正弦函数的傅里叶变化：

```
t=-4*pi:pi/10:4*pi;  
y=sin(t);  
plot(t,y);  
Y=fft(y);  
figure,plot(abs(Y));
```

例：函数的傅立叶变换

二、多个正弦函数的傅里叶变化：

```
t=-4*pi:pi/10:4*pi;
```

```
y=sin(t)+sin(2*t)+sin(3*t)+sin(4*t)+sin(5*t);
```

```
plot(t, y);
```

```
Y=fft(y);
```

```
figure, plot(abs(Y));
```

三、数字图像频域处理程序示范

数字图像频域处理程序

一、完整的正弦函数傅里叶变化：

```
Ns = 512;           %采样点
n = 0 : Ns-1;
xn = 4.6*sin(2*pi*n/Ns); %离散的正弦信号
Xn = fft( xn );      %fft的结果是复数
Xn = abs(Xn) *2 /Ns;  %求绝对值得到幅值，乘以2，除以Ns，
                      这是由傅里叶变换公式导致的

%用二维图显示原始正弦信号
subplot(3,1,1),plot(n, xn); %画出原始正弦信号
xlabel('n = 0:511'); ylabel('振幅');
title('原始的正弦信号');
```

数字图像频域处理程序

一、完整的正弦函数傅里叶变化：

%用二维图显示信号0~255次的谐波

Nyquist = Ns/2-1; %根据奈奎斯特定理，只需显示前255次的谐波

subplot(3,1,2),plot(n(1:Nyquist), Xn(1:Nyquist));

xlabel('n = 0:255'); ylabel('谐波幅值');

title('0~255次的谐波');%用二维图显示信号频率与幅值的关系

f0 = 50; %假设信号的固有频率为50Hz

f = n*f0; %频率与横轴序列n的映射关系

subplot(3,1,3),plot(f(1:Nyquist), Xn(1:Nyquist)); %根据奈奎斯特定理，只需显示前fs/2部分的频谱

xlabel('f = 0, 50Hz... 255*f0 Hz '); ylabel('幅值');

title('频率与幅值的关系');

数字图像频域处理程序

二、数字图像的傅立叶变换

```
f=zeros(50,50);  
f(5:24,13:17)=1;  
imshow(f);  
F=fft2(f,256,256);  
F2=fftshift(F);  
figure,imshow(log(abs(F2)),[-1 5]);
```


数字图像频域处理程序

注意:

(1) `zeros` 功能是返回一个 $m \times n \times p \times \dots$ 的 `double` 类零矩阵。其中 m, n, p, \dots 必须是非负整数，负整数将被当做0看待。当没有参数时，产生的是标量零矩阵，有参数时产生指定类型的零矩阵，例如 `zeros(3, 2, 'uint8')`。

(2) `imshow(I, n); imshow(I, [low high])`

用指定的灰度范围 `[low high]` 显示灰度图像 `I`。显示结果，图像中灰度值等于或低于 `low` 的都将用黑色显示，而灰度值大于等于 `high` 的都显示为白色，介于 `low` 和 `high` 之间的用其灰度级的默认值的中间色调显示。如果你用了一个空矩阵 `[]` 来代替 `[low high]`，`imshow` 函数将使用 `[min(I(:)) max(I(:))]` 作为第二个参数。

数字图像频域处理程序

注意：

(3) figure，就是控制窗口数量的，例如：

直接显示两个图片：

```
imshow(A1);
```

```
imshow(A2);
```

则会出现两个窗口，一个窗口显示A1，一个窗口显示A2

改为：

```
figure,
```

```
imshow(A1);
```

```
imshow(A2);
```

则只会出现一个窗口，窗口先显示A1，后显示A2

数字图像频域处理程序

注意：

(4) 'true size' 是按图片实际大小显示，窗口会自动调节大小去适应图片尺寸，'true size'是默认情况，跟省略这个参数一样。'not true size'就是图片去适应窗口，窗口大小不动，自动调节图片大小。所以显示出来的图片跟实际尺寸可能不一致。

(5) fft是离散傅立叶变换；fft2是2维离散傅立叶变换。快速傅立叶变换算法只能处理矩阵维数为2的幂次，f矩阵不是，通过对f矩阵进行零填充来调整

(6) 一般在计算图形函数的傅立叶变换时，坐标原点在函数图形的中心位置处，而计算机在对图像执行傅立叶变换时是以图像的左上角为坐标原点。所以使用函数fftshift进行修正，使变换后的直流分量位于图形的中心；

数字图像频域处理程序

三、数字图像的傅立叶变换及逆变换

```
f=zeros(256, 256) ;  
f(102:155, 102:155)=1;  
imshow(f) ;  
F=fft2(f, 256, 256) ;  
F2=fftshift(F) ;  
F3=log(abs(F2)) ;  
figure;  
imshow(F3) ;
```

数字图像频域处理程序

```
F4=F3(101:156, 101:156);  
figure;  
imshow(F4);  
figure;  
f2 = real(ifft2(F));  
imshow(f2);  
figure;  
f3 = real(ifft2(F4));  
imshow(f3);
```

数字图像频域处理程序

四、数字图像的傅立叶变换及逆变换

```
I=imread('D:\作业图片\1.jpg');  
I1=rgb2gray(I);  
imshow(I1);  
J = imnoise(I1,'salt & pepper',0.1); %椒盐  
figure, imshow(J);  
F=fft2(im2double(J));  
F2=fftshift(F);  
F3=log(abs(F2)+1)/5;  
figure;  
imshow(F3);  
f = real(ifft2(F));  
figure;  
imshow(f);
```

数字图像频域处理程序

五、数字图像的傅立叶变换及高斯低通滤波

```
I=imread('D:\作业图片\1. jpg');  
imshow(I);  
I1=rgb2gray(I);  
imshow(I1);  
J = imnoise(I1,'salt & pepper',0.1); %椒盐  
figure, imshow(J);  
s=fftshift(fft2(J));  
figure;  
imshow(s);  
[M,N]=size(s);  
d0=30; %GLPF滤波, d0=5, 15, 30 (程序中以d0=30为例)
```

数字图像频域处理程序

```
n1=floor(M/2);  
n2=floor(N/2);  
for i=1:M  
    for j=1:N  
        d=sqrt((i-n1)^2+(j-n2)^2);  
        h=1*exp(-1/2*(d^2/d0^2));  
        s(i,j)=h*s(i,j);  
    end  
end  
s=ifftshift(s);  
glmg=uint8(real(ifft2(s)));  
figure;  
imshow(glmg);
```


数字图像频域处理程序

六、数字图像的傅立叶变换及巴特沃思低通滤波

```
I=imread('D:\作业图片\1.jpg');  
imshow(I);  
I1=rgb2gray(I);  
imshow(I1);  
J = imnoise(I1,'salt & pepper',0.1); %椒盐  
figure, imshow(J);  
s=fftshift(fft2(J));  
[M,N]=size(s);  
%5级巴特沃思  
n=5;  
d0=30; %BLPF滤波, d0=5, 15, 30 (程序中以d0=30为例)
```

数字图像频域处理程序

```
n1=floor(M/2);  
n2=floor(N/2);  
for i=1:M  
    for j=1:N  
        d=sqrt((i-n1)^2+(j-n2)^2);  
        h=1/(1+(d/d0)^(2*n));  
        s(i,j)=h*s(i,j);  
    end  
end  
s=ifftshift(s);  
b5lmg=uint8(real(ifft2(s)));  
figure;  
imshow(b5lmg);
```

数字图像频域处理程序

- 巴特沃斯滤波器、切比雪夫滤波器、贝塞尔滤波器都是滤波器的一种形式。
- 巴特沃斯滤波器的特点是通频带内的频率响应曲线最大限度平坦，没有起伏，而在阻频带则逐渐下降为零。
- 贝塞尔（Bessel）滤波器是具有最大平坦的群延迟（线性相位响应）的线性过滤器。
- 切比雪夫滤波器是在通带或阻带上频率响应幅度等波纹波动的滤波器，振幅特性在通带内是等波纹。在阻带内是单调的称为切比雪夫I型滤波器；振幅特性在通带内是单调的，在阻带内是等波纹的称为切比雪夫II型滤波器。

数字图像频域处理程序

七、数字图像的傅立叶变换及高斯高通滤波

```
I=imread('D:\作业图片\1.jpg');
```

```
I=imread('autumn.tif');
```

```
imshow(I);
```

```
I1=rgb2gray(I);
```

```
imshow(I1);
```

```
s=fftshift(fft2(I1));
```

```
[M,N]=size(s);
```

```
d0=30; %GLPF滤波, d0=5, 15, 30 (程序中以d0=30为例)
```

数字图像频域处理程序

```
n1=floor(M/2);  
n2=floor(N/2);  
for i=1:M  
    for j=1:N  
        d=sqrt((i-n1)^2+(j-n2)^2);  
        h=1-1*exp(-1/2*(d^2/d0^2));  
        s(i,j)=h*s(i,j);  
    end  
end  
s=ifftshift(s);  
glmg=uint8(real(ifft2(s)));  
figure;  
imshow(glmg);
```

数字图像频域处理程序

八、数字图像的傅立叶变换及巴特沃思高通滤波

```
I=imread('D:\作业图片\1.jpg');  
imshow(I);  
I1=rgb2gray(I);  
imshow(I1);  
s=fftshift(fft2(I1));  
[M,N]=size(s);  
n=2;  
d0=30; %BLPF滤波, d0=5, 15, 30 (程序中以d0=30为例)
```

数字图像频域处理程序

```
n1=floor(M/2);  
n2=floor(N/2);  
for i=1:M  
    for j=1:N  
        d=sqrt((i-n1)^2+(j-n2)^2);  
        h=1/(1+(d0/d)^(2*n));  
        s(i,j)=h*s(i,j);  
    end  
end  
s=ifftshift(s);  
blmg=uint8(real(ifft2(s)));  
figure;  
imshow(blmg);
```

数字图像频域处理程序

九、数字图像的离散余弦变换

```
I=imread('autumn.tif');  
figure, imshow(I);  
J=rgb2gray(I);  
K=dct2(J);  
figure, imshow(log(abs(K)), []), colormap(jet(64)), colorbar;  
K(abs(K)<30)=0;  
L=idct2(K);  
figure, imshow(L, [0, 255]);
```


谢谢！