Fourier变换

Introduction to Wavelet Theory and it's Applications

Hello Viewers. In this video, the wavelet transform theory and its applications is explained.This video includes following components,* Fourier Transform and...

https://www.youtube.com/watch?v=F_QvT_8kOfc

- Fourier变换基本知识
  - 连续Fourier变换表达式：
  - 离散Fourier变换表达式：
  - Fourier变换的局限性
  - Fourier变换的改进
  - 二维Fourier变换
    - 2D Continuous Fourier变换表达式：
    - 2D Discrete Fourier Transform（2D DFT）的表达式：

Fourier变换基本知识

连续Fourier变换表达式：

$$F(\omega)=\int_{-\infty}^{\infty} f(t) e^{-j \omega t} d t$$

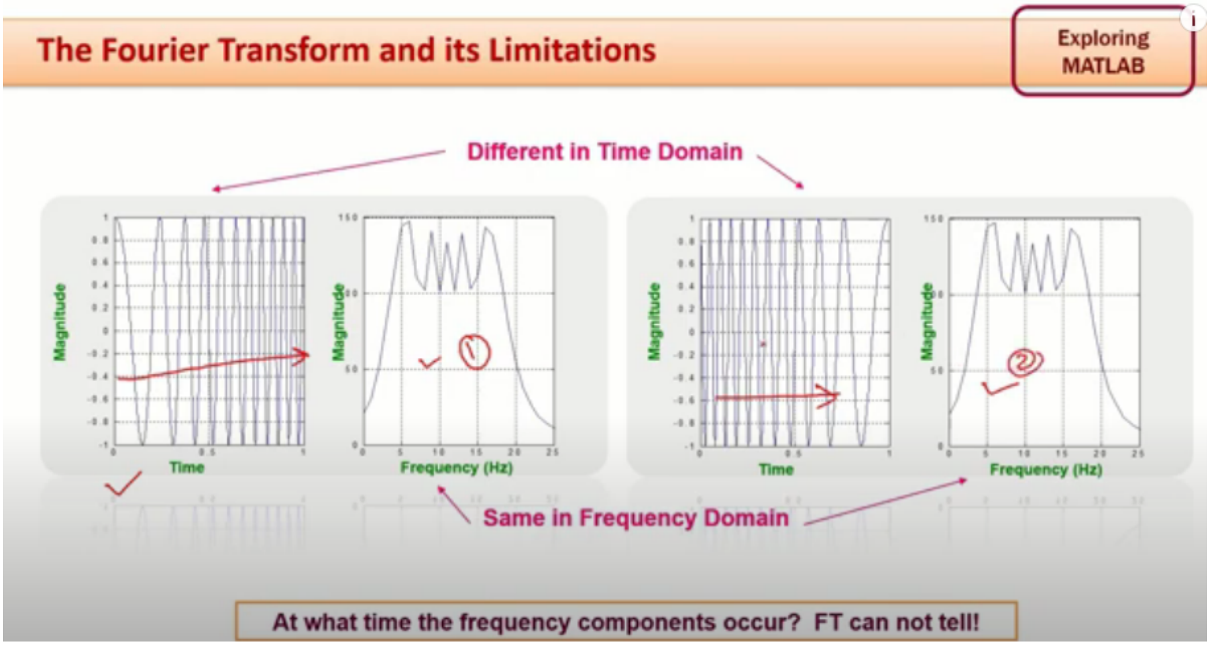
离散Fourier变换表达式：

$$F(\omega)=\sum_{-\infty}^{+\infty} f(t) e^{-j \omega t} d t$$

本质上是将信号$f(t)$分解为一系列sine/cosine谐波的线性组合

Fourier变换的局限性

- 无法反映**非稳态信号（随着时间频率改变）**时域的变化



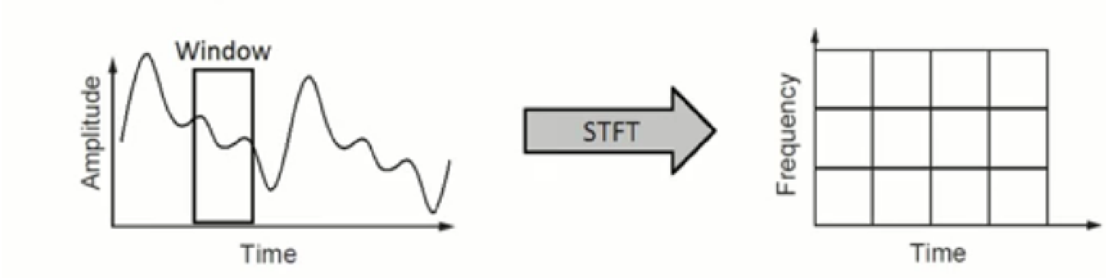
Fourier变换的改进

- 采用Short-Time Fourier Transform（STFT）1946

  每次只使用一个窗口内的信号进行变换，该窗口内的信号认为是稳态信号

$$STFT_x^{\omega}(t',\omega)=\int_t [x(t)\cdot W(t-t')]\cdot e^{-j\omega t}dt$$

  其中，$W(t-t')$表示窗口函数，由于采用了窗口，STFT得到的变换系数是二维的：



- 存在问题：

时域窗口窄，会导致poor frequency resolution；时域窗口宽，则会得到high frequency resolution

二维Fourier变换

- 对于一个二维图像，二维Fourier变换本质上可以理解为在一组由Sine、Cosine函数构成的二维基底下，求取每个基底对应的相关值，这些相关值构成了这个二维图像在Fourier变换基下的表达

2D Continuous Fourier变换表达式：

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)e^{-j2\pi(ux+vy)}dxdy$$

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(u,v)e^{j2\pi(ux+vy)}dudv$$

其中，u,v表示频率取值，$e^{-j2\pi(ux+vy)}=\cos 2\pi(ux+vy)-j\sin 2\pi(ux+vy)$，即基底是由Sine、Cosine函数构成的

2D Discrete Fourier Transform（2D DFT）的表达式：

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

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

- 相当于，对于每一个频域中的点(u_0,v_0)，都对应时域中的一个基底：$b(x,y)=e^{-j2\pi(u_0x/M+v_0y/N)}=\cos 2\pi(u_0x/M+v_0y/N)-j\sin 2\pi(u_0x/M+v_0y/N)$

  其中，x,y为变量。比如，当取$(u_0,v_0)=(0,0)$时，$b(x,y)=1-j0$，即此时的基底就是一个取值恒定的图像。我们可以取不同的(u_0,v_0)，分别绘制其对应的基底图像：

  https://s3-us-west-2.amazonaws.com/secure.notion-static.com/de780e13-bb3d-49fb-803f-8932ebc00dac/d3b4f1602784b09e64a0234252fc8576.mp4

  左图表示取不同的u,v值，右图表示该取值下对应的基底图像（注意只显示了实部）

绘制上述过程的Matlab代码如下：

```
Img_M = 20;     % 设置图像大小
Img_N = 20;
[nx,ny] = ndgrid([0:Img_M-1]-(Img_M-1)/2,[0:Img_N-1]-(Img_N-1)/2 ); % 计算x·y可能取值
figure,
set(gcf,'position',[100,100,1500*0.6,520*0.6]);
skip = 1;
for u = [0:skip:Img_M-1]
  for v = [0:skip:Img_N-1]
    subplot(1,2,1)
    b = zeros(Img_M,Img_N);
    b(u+1,v+1)=1;
    imagesc(b);colormap gray;
    xlabel('u');ylabel('v');
    title(['u=',num2str(u),',','v=',num2str(v)])
    subplot(1,2,2)
    imagesc(real(exp(-i*2*3.1416*(u*nx/Img_M+v*ny/Img_N))));colormap gray;
    xlabel('x');ylabel('y');
    pause(0.1)
  end
end
```

- 根据2D DFT的表达式，**Fourier变换的本质是去求空域图像$f(x,y)$与每个基底(u_0,v_0)之间的相关值**，将该相关值放在频域图像坐标(u_0,v_0)处，遍历每个可能的(u_0,v_0)即可得到最终的Fourier变换后的图像

  可以编写Matlab代码实现上述过程：

```
clear
close all
Img=double(imread('lena.jpg'));
Img=imresize(Img,[150,150]);
[Img_M, Img_N] = size(Img);
Output=zeros(Img_M,Img_N);
SumOutner = 0;
%% 2D Discrete Fourier Transform
[nx,ny]=ndgrid([0:Img_M-1],[0:Img_N-1]);
du=1;
for u = [0:Img_M-1]
 dv=1;
for v = [0:Img_N-1]
    SumOutner=sum(sum(Img.*exp(-1i*2*3.1416*(u*nx/Img_M+v*ny/Img_N))));
    Output(du,dv) = SumOutner;
    dv=dv+1;
end
du=du+1;
end
%% Calculate Spectrum and show
figure,set(gcf,'position',[100,100,1800*0.6,520*0.6]);
subplot(1,3,1)
imagesc(Img);colormap gray;title('原始图像')
```

```
subplot(1,3,2)
imagesc(log(abs(fftshift(Output))));colormap gray;title('Fourier变换后图像')
subplot(1,3,3)
imagesc(log(abs(fftshift(fft2(Img)))));colormap gray;title('Matlab自带函数变换后图像')
```

代码运行结果为：