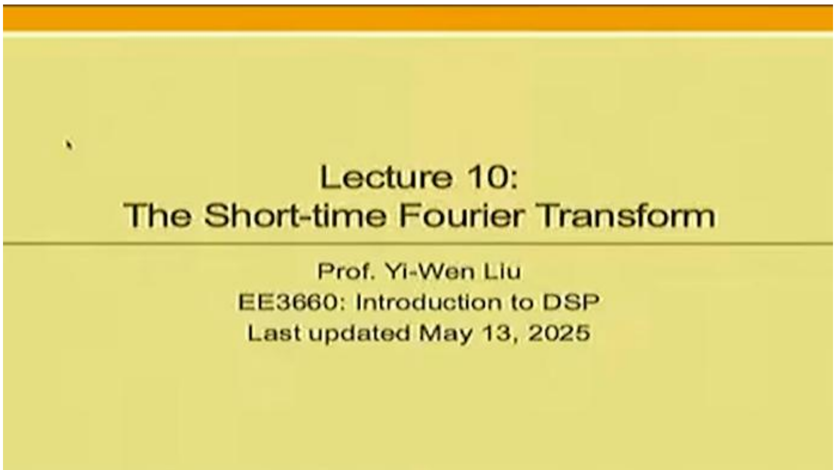
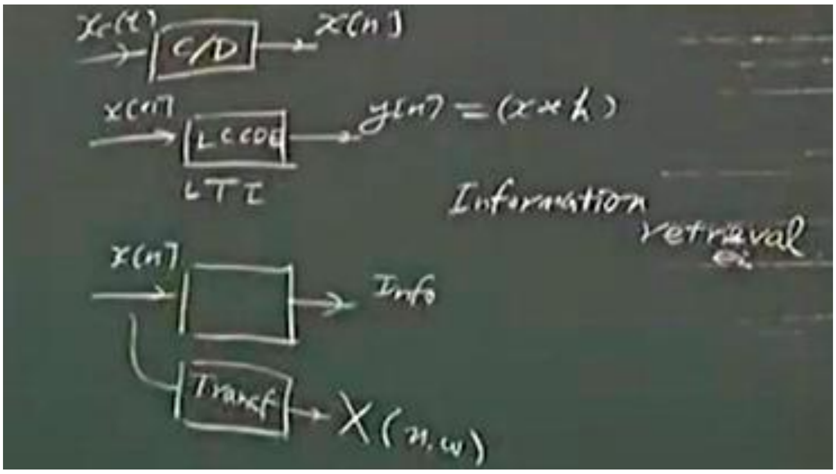
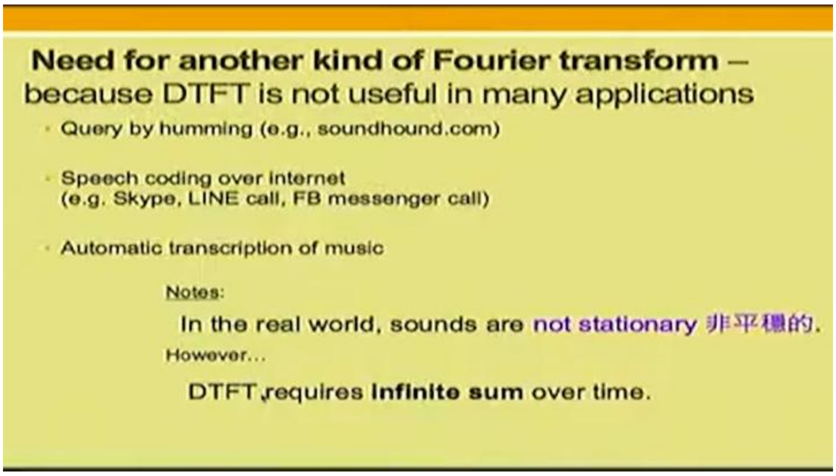
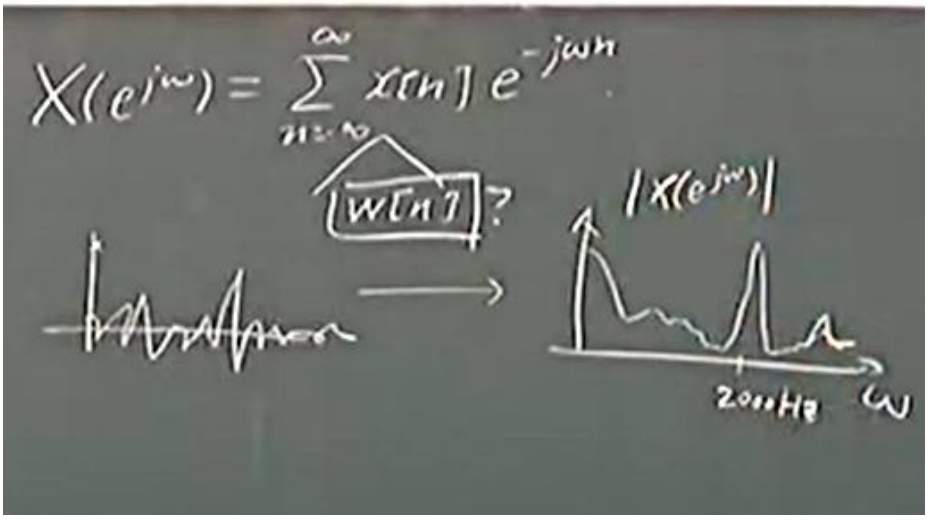
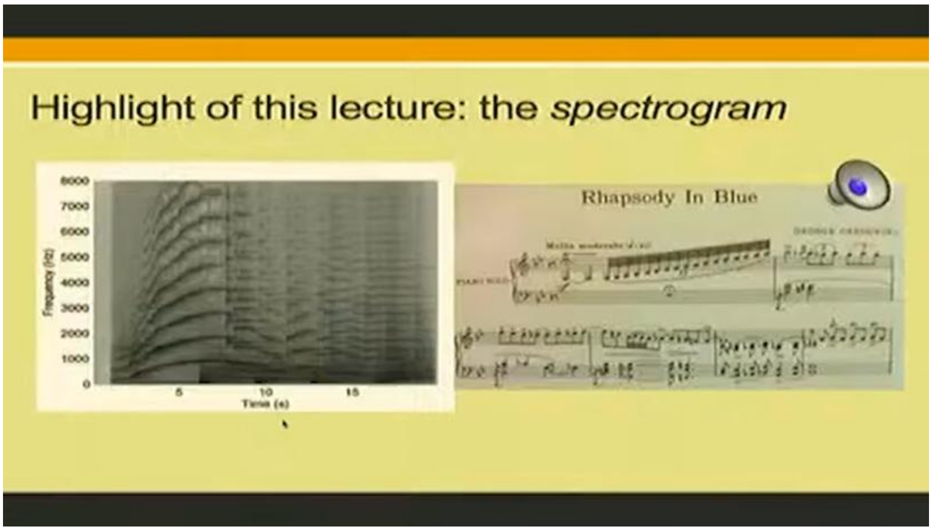
Lecture10 The Short-time Fourier TransformIntro. to DSP – Lec 11








$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$
w[n] ?
$|X(e^{j\omega})|$
2000Hz
ω


Highlight of this lecture: the *spectrogram*
Frequency (Hz)
Time (s)
Rhapsody In Blue

A: The discrete Fourier transform (DFT)

Summary: $\mathbf{F}^{-1} = \frac{1}{L}\mathbf{F}^{*}.$

$$\begin{bmatrix} S(0) \\ S(2\pi/L) \\ S(4\pi/L) \\ \vdots \\ \cdot \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{L-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(L-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & w^{L-1} & w^{2(L-1)} & \cdots & w^{(L-1)(L-1)} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{L-1} \end{bmatrix} \quad w = e^{-j2\pi/L}$$

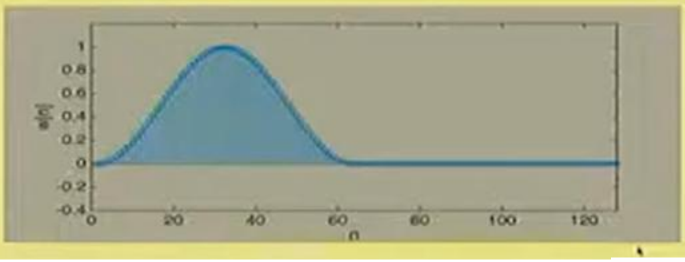
$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{L-1} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & v & v^2 & \cdots & v^{L-1} \\ 1 & v^2 & v^4 & \cdots & v^{2(L-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & v^{L-1} & v^{2(L-1)} & \cdots & v^{(L-1)(L-1)} \end{bmatrix} \begin{bmatrix} S(0) \\ S(2\pi/L) \\ S(4\pi/L) \\ \vdots \\ \cdot \end{bmatrix}$$

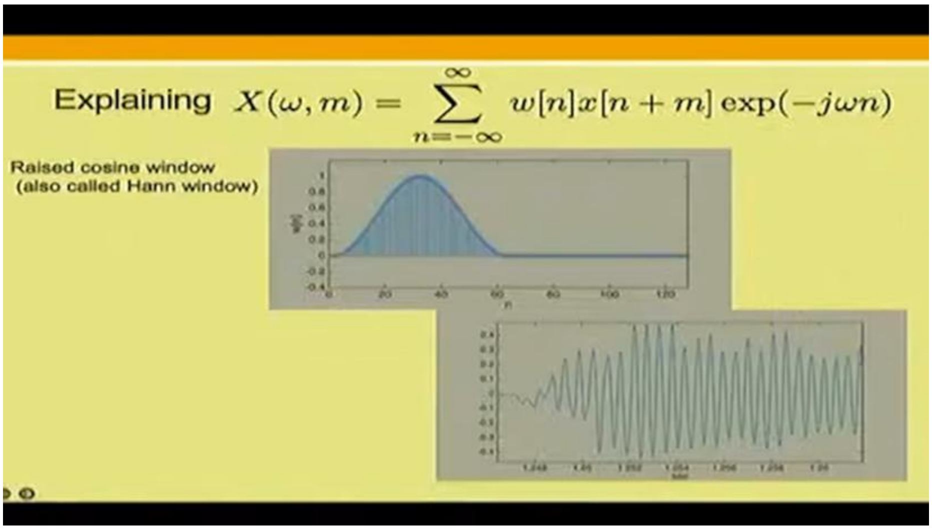
$$v = w^{-1} = e^{j2\pi/L}$$

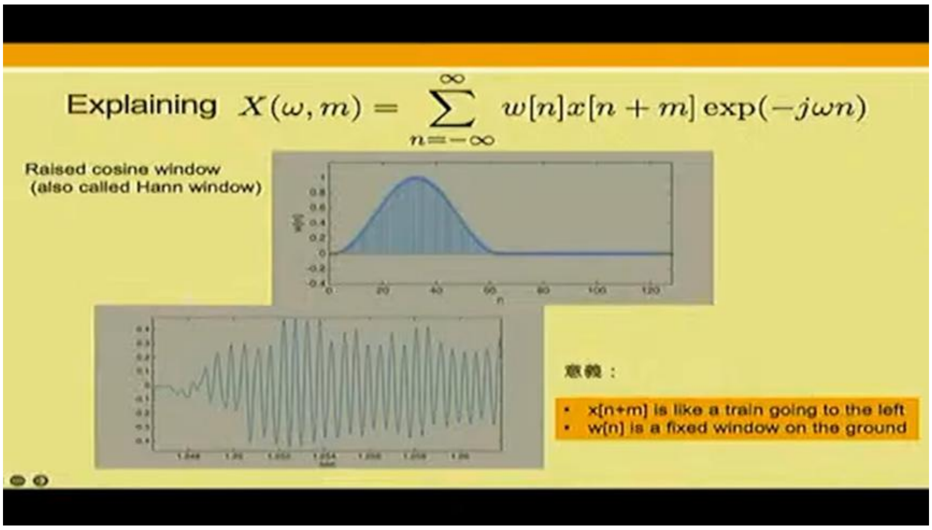
Short-time Fourier transform: converting a sound to an image

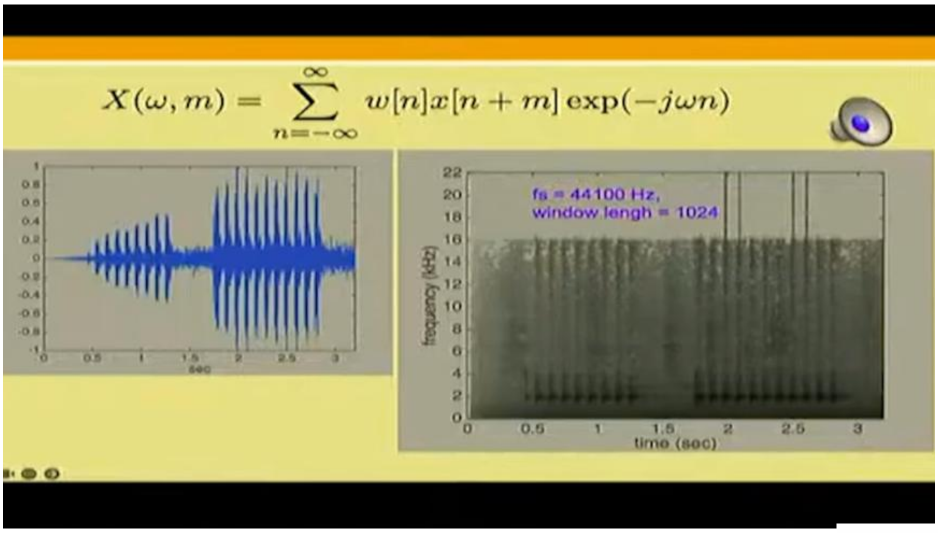
$$X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m]\exp(-j\omega n)$$

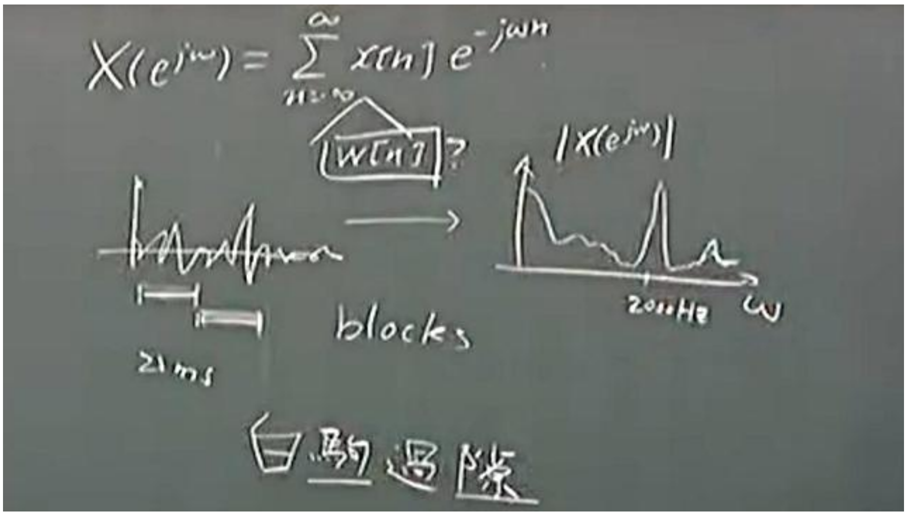
w[n]: a **window** with finite number of non-zero points.

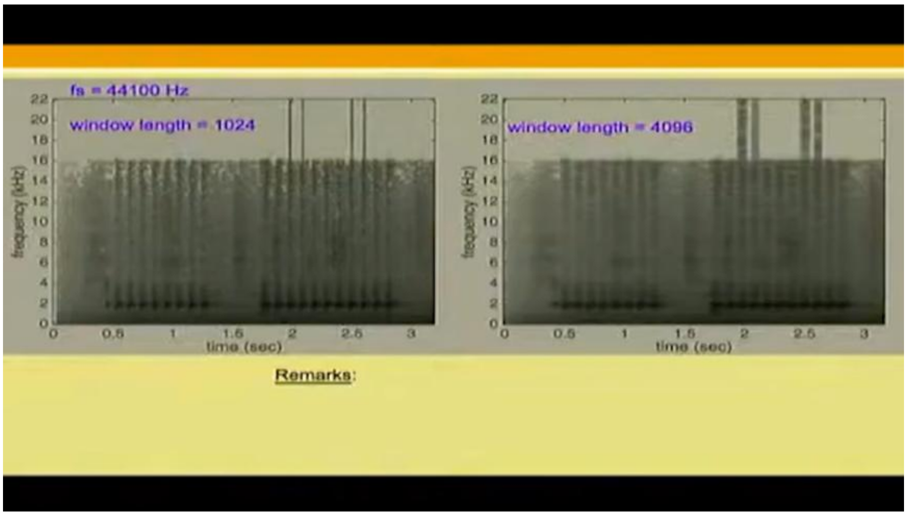


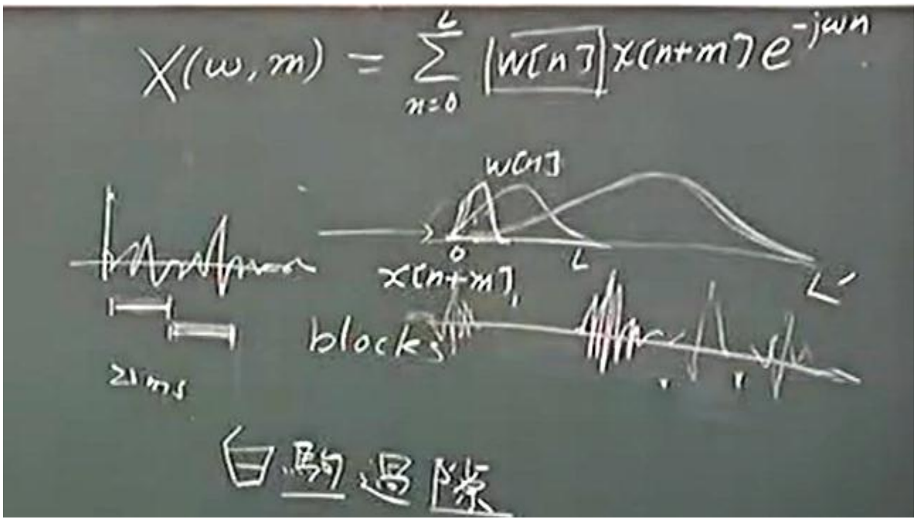

Explaining $X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m]\exp(-j\omega n)$
Raised cosine window
(also called Hann window)

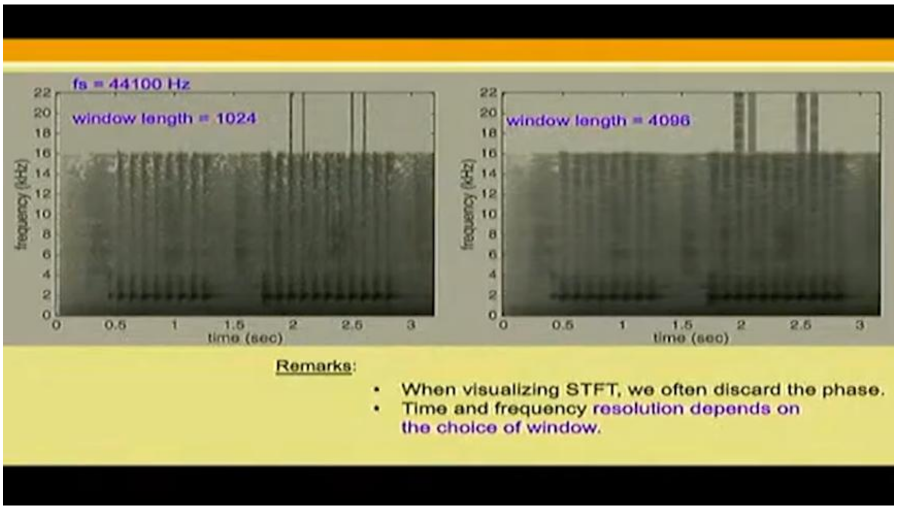

Explaining $X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m]\exp(-j\omega n)$
Raised cosine window
(also called Hann window)
意義：
x[n+m] is like a train going to the left
w[n] is a fixed window on the ground

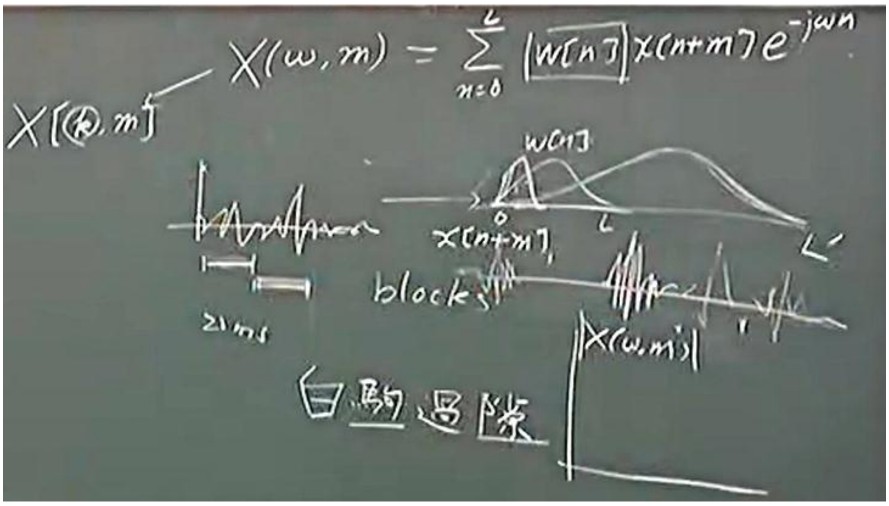

$X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m]\exp(-j\omega n)$
fs = 44100 Hz,
window lengh = 1024
frequency (kHz)
time (sec)
sec

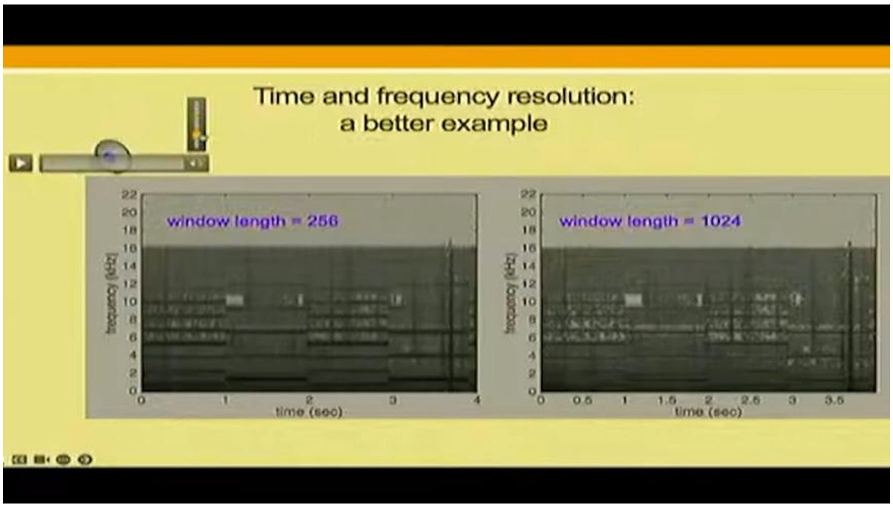

X(e^{jω}) = Σ x[n] e^{-jωn}
w[n]?
|X(e^{jω})|
blocks
21ms
2000Hz
ω
白駒過隙


fs = 44100 Hz
window length = 1024
window length = 4096
frequency (kHz)
time (sec)
Remarks:

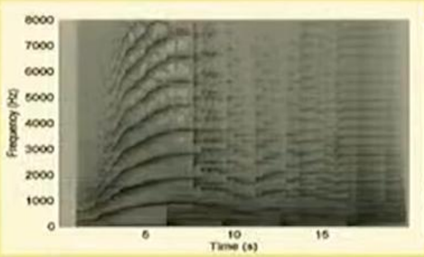

X(ω, m) = Σ_{n=0}^{L} w[n] x[n+m] e^{-jωn}
w[n]
x[n+m]
L
L'
blocks
21ms
白駒過隙


fs = 44100 Hz
window length = 1024
window length = 4096
frequency (kHz)
time (sec)
Remarks:
• When visualizing STFT, we often discard the phase.
• Time and frequency resolution depends on the choice of window.


$X(\omega, m) = \sum_{n=0}^{L} \boxed{w[n]} x[n+m] e^{-j\omega n}$
X[k, m]
w[n]
x[n+m]
block
21ms
白駒過隙
|X(ω,m)|


Time and frequency resolution:
a better example
window length = 256
window length = 1024
frequency (kHz)
time (sec)

Summary:
sounds are time-varying spectra as we hear them.



$$X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m]\exp(-j\omega n)$$

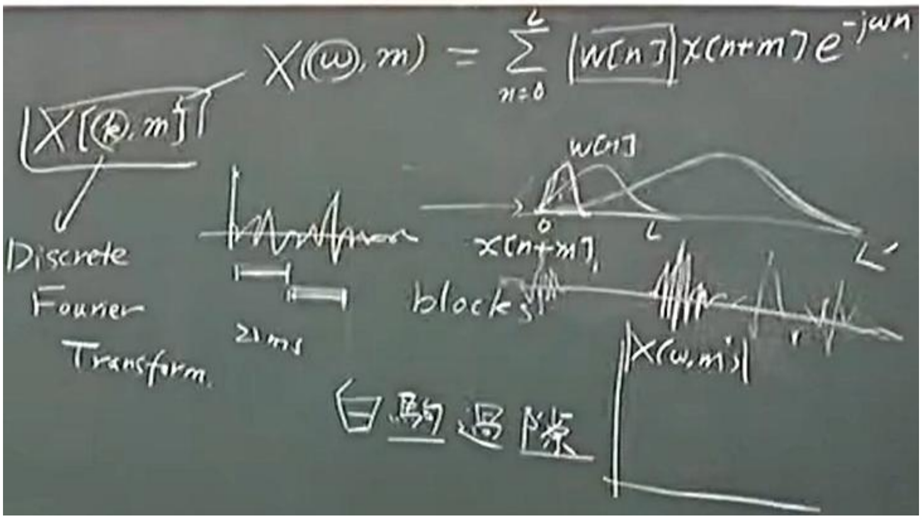
Discrete Fourier Transform (DFT)

Definition: Assume that a block of signal $\mathbf{s}$ contains L points,

$$\mathbf{s} = [s_0, s_1, \ldots, s_{L-1}]^T$$

The DFT of $\mathbf{s}$ is defined as a discrete spectrum $\boldsymbol{S}$ that also contains L points:

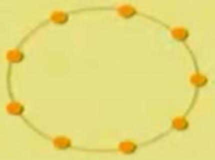
$$S_k = \sum_{n=0}^{L-1} s_n \exp\left(-jk\frac{2\pi}{L}n\right), \quad k = 0, 1, 2, \ldots, L-1.$$

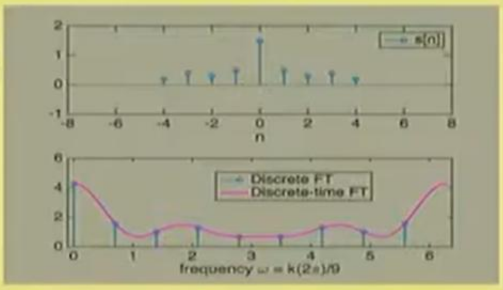


Why is this called *discrete* Fourier transform?

Answer: Because the *discrete* FT is the *discrete-time* FT sampled in frequency.

$$S_k = \sum_{n=0}^{L-1} s_n \exp\left(-jk\frac{2\pi}{L}n\right)$$
$$= S(\omega)\Big|_{\omega=k\frac{2\pi}{L}}$$



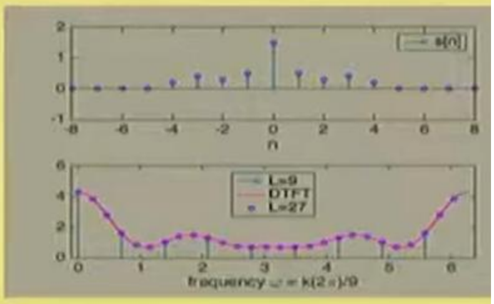


Relation between DFT and DTFT

- DFT is DTFT sampled in frequency.
- Remarks:
  For a finite-length signal:

  DTFT is the limit of DFT when L approaches infinity.

  * zero-padding

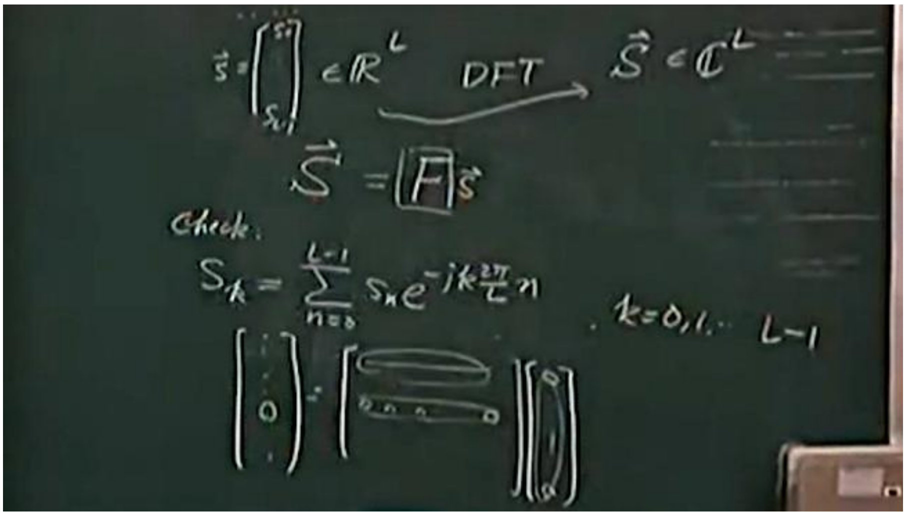


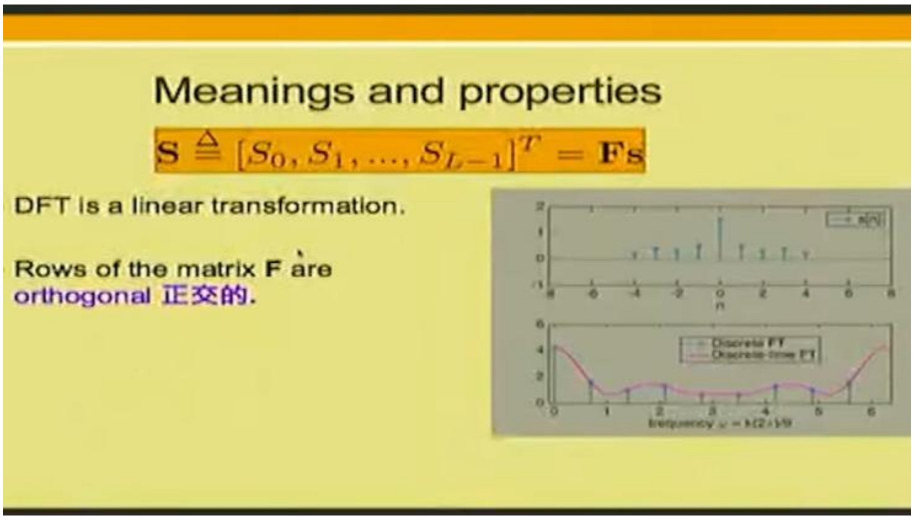
Matrix representation of DFT

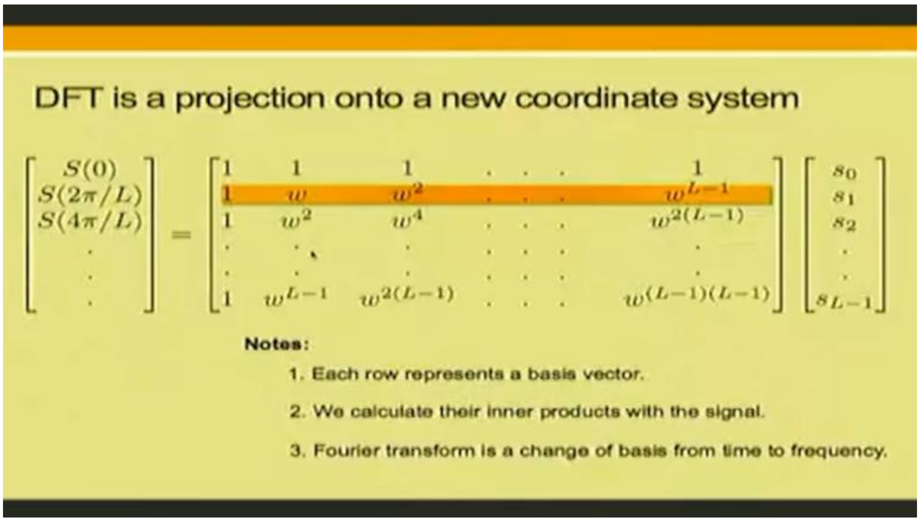
$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 \\ 1 & w & w^2 & \cdot & \cdot & \cdot & w^{L-1} \\ 1 & w^2 & w^4 & \cdot & \cdot & \cdot & w^{2(L-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & w^{L-1} & w^{2(L-1)} & \cdot & \cdot & \cdot & w^{(L-1)(L-1)} \end{bmatrix}$$

where $\quad w = e^{-j2\pi/L}$

Note that: $\boxed{\mathbf{S} \triangleq [S_0, S_1, \ldots, S_{L-1}]^T = \mathbf{F}\mathbf{s}}$


$\vec{s} = \begin{bmatrix} s_0 \\ \vdots \\ s_{L-1} \end{bmatrix} \in \mathbb{R}^L \xrightarrow{DFT} \vec{S} \in \mathbb{C}^L$
$\vec{S} = \boxed{F}\vec{s}$
Check:
$S_k = \sum_{n=0}^{L-1} s_n e^{-jk\frac{2\pi}{L}n}$, $k = 0, 1, \ldots L-1$


Meanings and properties
$\mathbf{S} \triangleq [S_0, S_1, \ldots, S_{L-1}]^T = \mathbf{F}\mathbf{s}$
DFT is a linear transformation.
Rows of the matrix **F** are orthogonal 正交的.
Discrete FT
Discrete-time FT


DFT is a projection onto a new coordinate system
$$\begin{bmatrix} S(0) \\ S(2\pi/L) \\ S(4\pi/L) \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{L-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(L-1)} \\ \vdots & \ddots & \vdots & & \vdots \\ 1 & w^{L-1} & w^{2(L-1)} & \cdots & w^{(L-1)(L-1)} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{L-1} \end{bmatrix}$$
Notes:
1. Each row represents a basis vector.
2. We calculate their inner products with the signal.
3. Fourier transform is a change of basis from time to frequency.

2B: The Fast Fourier transform (FFT) and its inverse

Regarding the computation load

Recall that

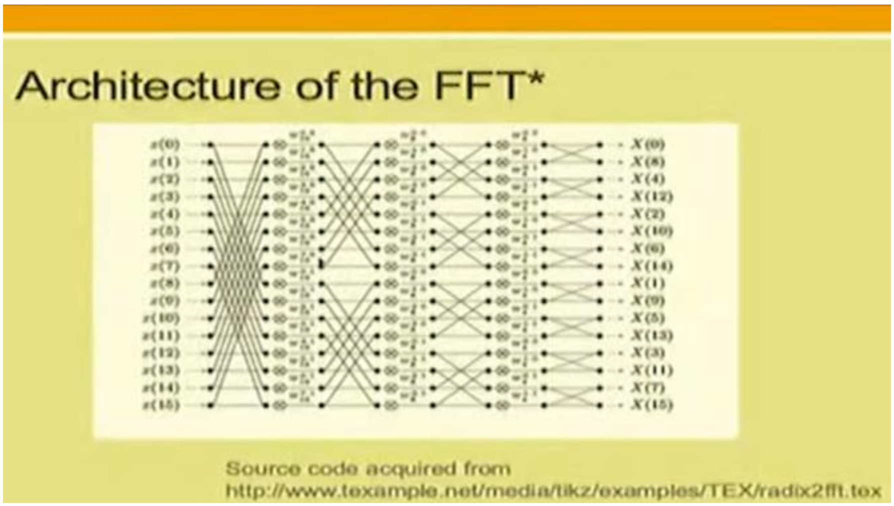
$$\begin{bmatrix} S(0) \\ S(2\pi/L) \\ S(4\pi/L) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdot\ \cdot\ \cdot & 1 \\ 1 & w & w^2 & \cdot\ \cdot\ \cdot & w^{L-1} \\ 1 & w^2 & w^4 & \cdot\ \cdot\ \cdot & w^{2(L-1)} \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot & \cdot \\ 1 & w^{L-1} & w^{2(L-1)} & \cdot\ \cdot\ \cdot & w^{(L-1)(L-1)} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \cdot \\ \cdot \\ s_{L-1} \end{bmatrix}$$

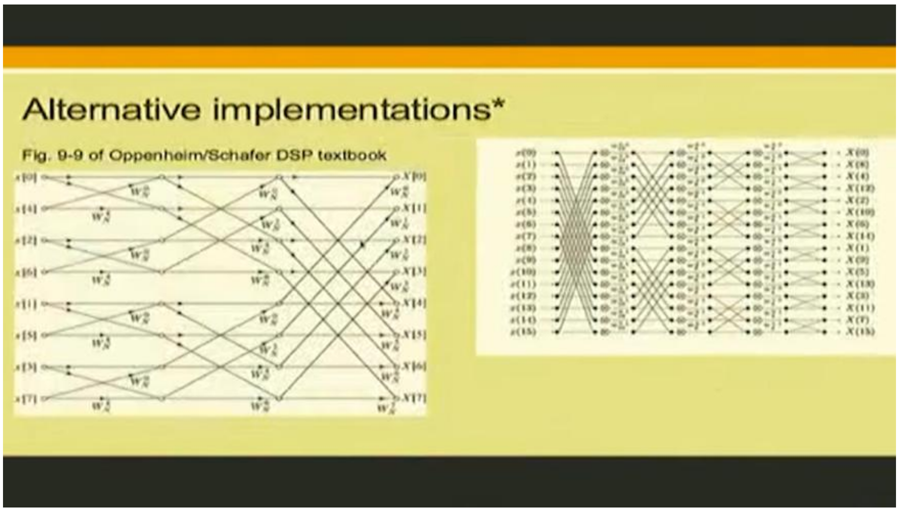
Therefore, direct computation involves L^2 multiplications.

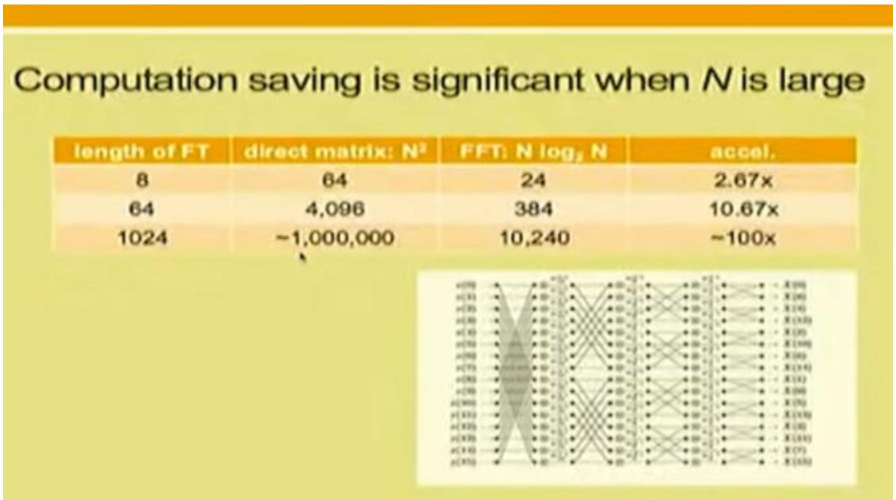
Is there any way we can reduce the computation load?

The fast Fourier transform (FFT)

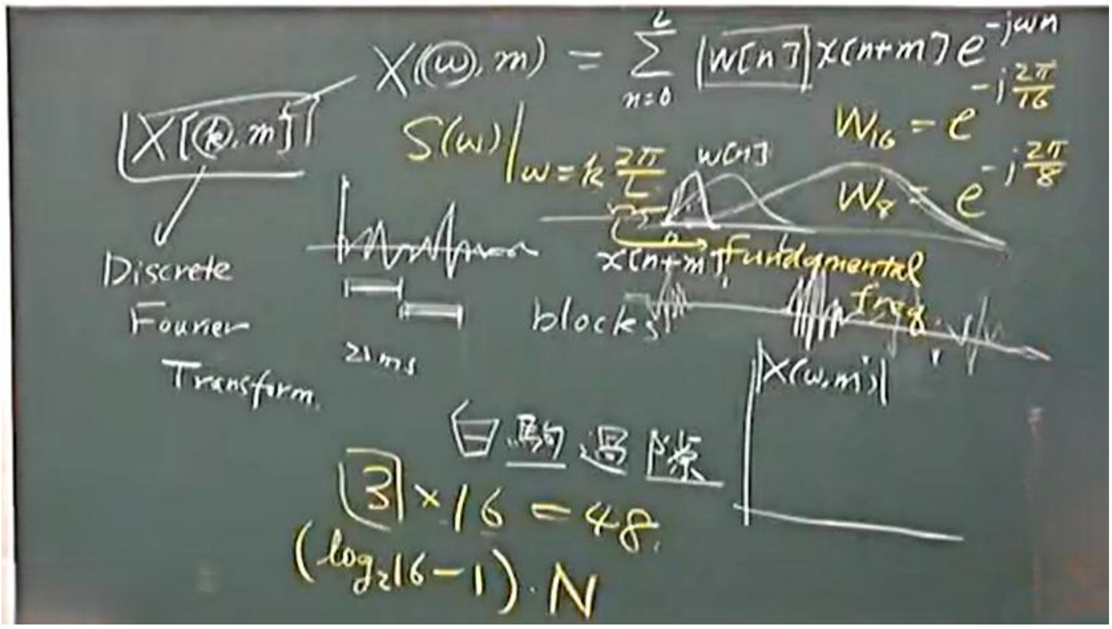
- It turns out that it is possible to reduce the computation from $O(N^2)$ to $O(N \log_2 N)$, and this is generally referred to as FFT.
  - Cooley, J. W. and Tukey, O. W. "An Algorithm for the Machine Calculation of Complex Fourier Series." *Math. Comput.* **19**, 297-301, 1965.
  - Often named one of the 10 most important algorithms in history.
  - Original algorithm is most efficient if $N = 2^p$, e.g. N = 1024.







length of FT	direct matrix: N^2	FFT: $N \log_2 N$	accel.
8	64	24	2.67x
64	4,096	384	10.67x
1024	~1,000,000	10,240	~100x


$X(\omega, m) = \sum_{n=0}^{L} w[n]\, x[n+m]\, e^{-j\omega n}$
$|X[k,m]|$
$S(\omega)\big|_{\omega = k\frac{2\pi}{L}}$
$W_{16} = e^{-j\frac{2\pi}{16}}$
$W_8 = e^{-j\frac{2\pi}{8}}$
w[n]
x[n+m]
Discrete
Fourier
Transform
21 ms
blocks
fundamental
freq.
$|X(\omega, m)|$
白駒過隙
$3 \times 16 = 48$
$(\log_2 16 - 1) \cdot N$