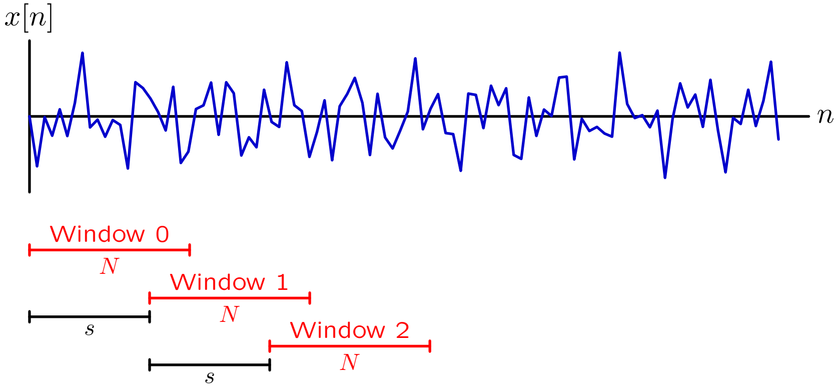
STFT

Short-time Fourier transforms are based on the analysis of a sequence of finite-length portions of an input signal.



spectrogram

Motivating Example: Song Recognition

Song detection systems (like Shazam)

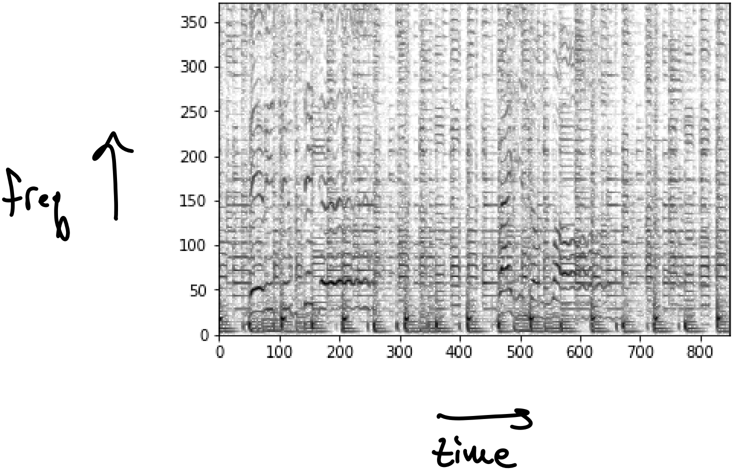
Goal: given a (potentially corrupted) recording of a portion of a song, determine what song that recording came from

Example: `noisy.wav`

How is this possible?

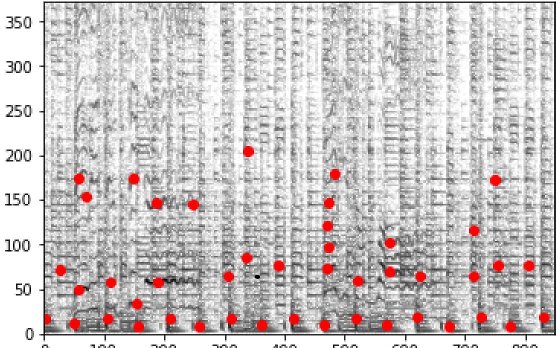
Song Recognition: Fingerprinting

Typical song recognition systems work by producing a *fingerprint* of a song. For example, we can start with a spectrogram of a song:



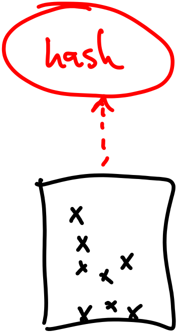
Song Recognition: Fingerprinting

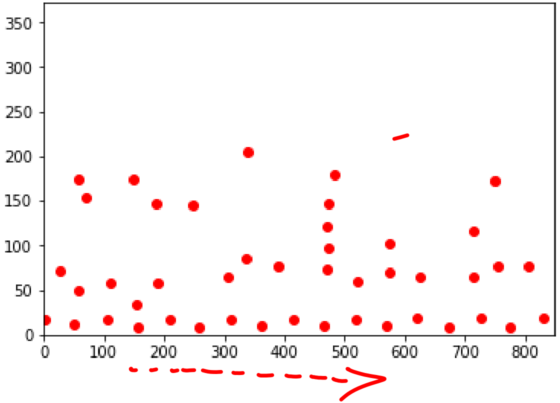
Then, we can find peaks within that spectrogram to produce a sparse representation of the song:



Song Recognition: Fingerprinting

Then, we can find peaks within that spectrogram to produce a sparse representation of the song:





This representation is (surprisingly) robust against a lot of kinds of corruption we might expect to experience in the world.

Song Recognition

We can then compute a similar fingerprint for the noisy recording, and look for a match across the song (and across the other songs in our database).

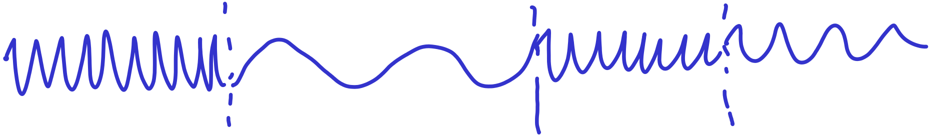
This process works really well, but it is helped dramatically by having a relatively clean spectrogram of the original song to start with (so that we can find reliable information about the peak frequencies).

.

Spectrograms

Consider a small sound in `cos0.wav`, consisting of a single cosine alternating between two frequencies.

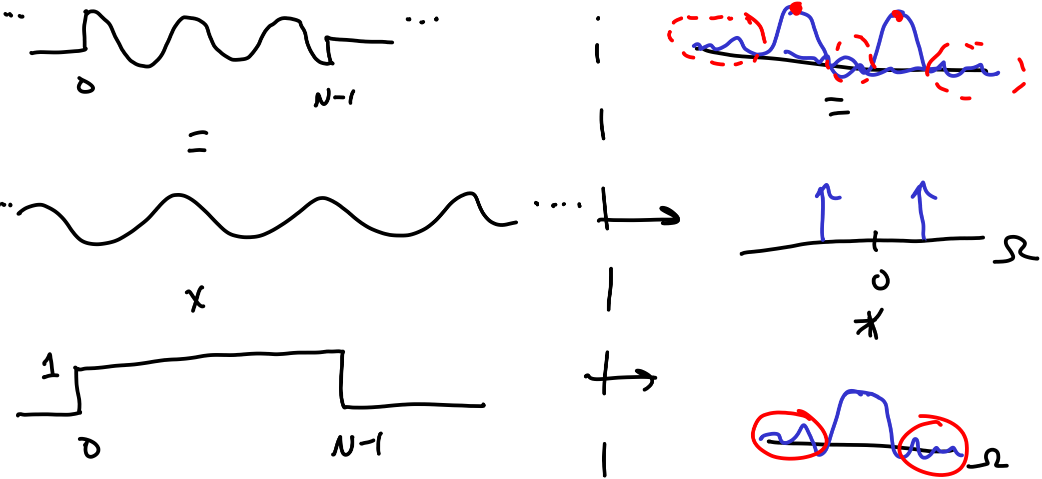
What do we expect its spectrogram to look like?



Windowing

DTFT

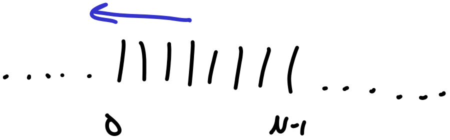
To understand what we just saw, let's think just about the first "window" of the spectrogram, and how it was computed.



Characterizing Windows

Each sequence of length N can be thought of as the product of $x[n]$ times a shifted version of a rectangular window $w[n]$ of length N:

$$w[n] = \begin{cases} 1 & \text{if } 0 \le n < N \\ 0 & \text{otherwise} \end{cases}$$



Find the DTFT of $w[n]$.

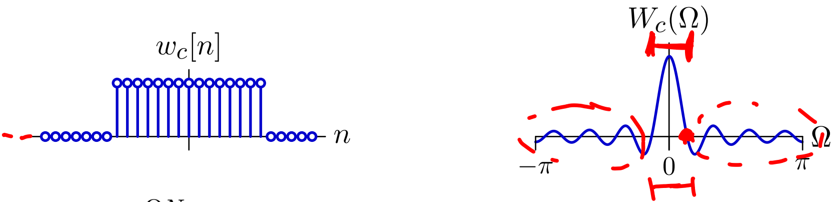
$$W(\Omega) = \sum_{n=-\infty}^{\infty} w[n] e^{-j\Omega n} = \sum_{n=0}^{N-1} e^{-j\Omega n} = \boxed{\frac{e^{j\Omega 0} - e^{-j\Omega N}}{e^{j\Omega 0} - e^{-j\Omega}}}$$

$$\frac{e^{-j\Omega\frac{N}{2}}\left(e^{j\Omega\frac{N}{2}} - e^{-j\Omega N/2}\right)}{e^{-j\Omega/2}\left(e^{j\Omega/2} - e^{-j\Omega/2}\right)} = \boxed{\frac{\sin(\Omega\frac{N}{2})}{\sin(\Omega/2)}}\, e^{-j\Omega\left(\frac{N-1}{2}\right)}$$

$$= W_c(\Omega)\, e^{-j\Omega\left(\frac{N-1}{2}\right)}$$

Characterizing Rectangular Windows

What is the half-width of the center lobe of $W_c(\Omega)$.



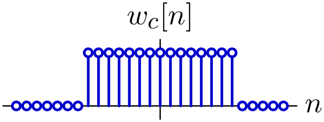
$$W_c(\Omega) = \frac{\sin \frac{\Omega N}{2}}{\sin \frac{\Omega}{2}}$$

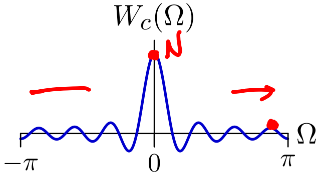
Notice that $W_c(\Omega) > 0$ for $-\frac{2\pi}{N} < \Omega < \frac{2\pi}{N}$. Thus the half-width is $\frac{2\pi}{N}$

The half-width of the central lobe decreases as N increases.

Characterizing Rectangular Windows

Characterize the heights of the sidelobes.





$$W_c(\Omega) = \frac{\sin \frac{\Omega N}{2}}{\sin \frac{\Omega}{2}}$$

Central lobe has height $W_c(0) = N$.

Heights of the sidelobes approach $\frac{1}{\sin(\pi/2)} = 1$ as $\Omega \to \pi$.

Ratio of the tallest to shortest lobe is $\sim N$.

Characterizing Rectangular Windows: Summary

The effect of windowing in time is to convolve in frequency, which blurs the frequency representation.

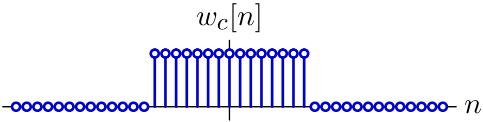
Two important metrics characterize blur:

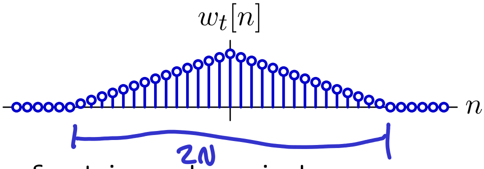
- half-width of central lobe $= \frac{2\pi}{N}$ (which decreases with N).
- amplitudes of side lobes $> \frac{1}{N}$ (which decreases with N).

Increasing N improves both metrics but the improvement is only linear in N while the computation cost (per window) grows faster than N (N^2 for direct convolution and $N \log N$ for the FFT).

Characterizing Other Windows

What if we use a triangular window instead of a rectangular window?





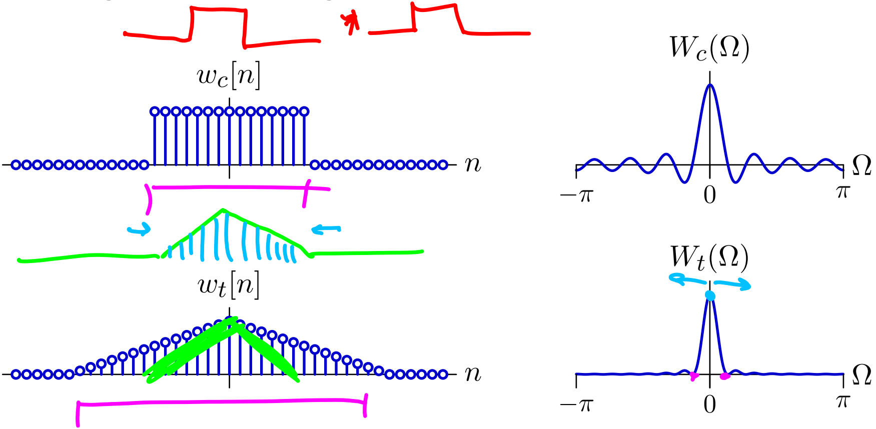
Characterize the blurring properties of a triangular window.

$$W_t(\Omega) = W_c^2(\Omega)$$

$$w_t[n] = (w_c * w_c)[n]$$

Rectangular and Triangular Windows

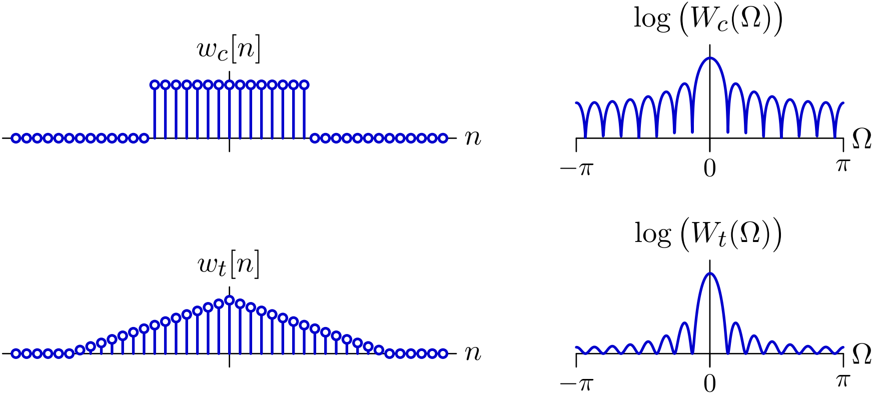
Plot rectangular and triangular windows and their transforms.



The triangular window has a narrower central lobe and smaller side-lobes.

Rectangular and Triangular Windows

Use semilog axes for the transforms.

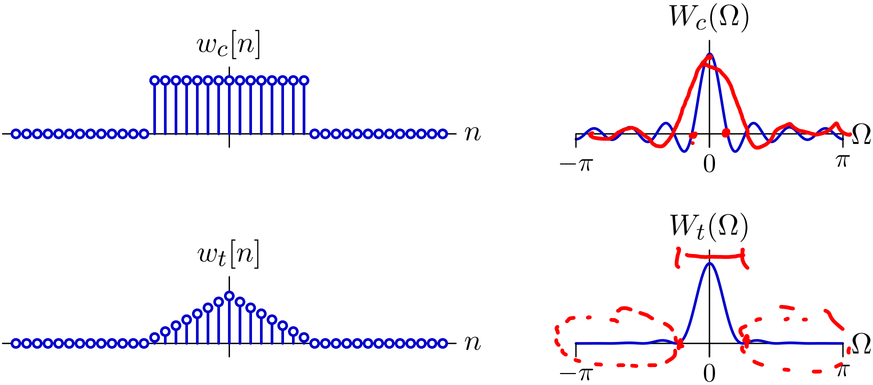


Notice that the triangle window has the same number of sidelobes as the rectangular window, but their amplitudes are much smaller.

However, the triangular window has nearly double the length (in time domain) of the square window!

Rectangular and Triangular Windows

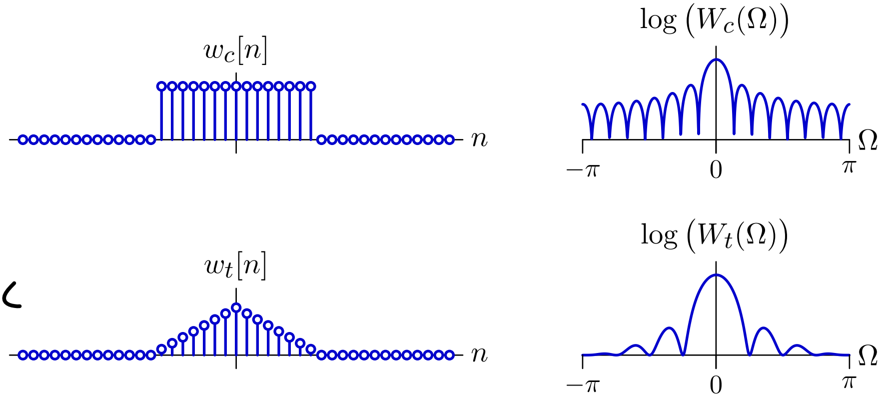
Compare rectangular and triangular windows of equal length.



The sidelobes of the triangular window are still much smaller than those of the rectangular window. Why?

Rectangular and Triangular Windows

Compare rectangular and triangular windows of equal length.



The central lobe of the triangular window is twice as wide as that of the rectangular window.

Rectangular and Triangular Windows: Summary

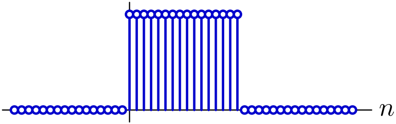
The central lobe of a triangular window is twice as wide as that of the rectangular window with the same length N.

The sidelobes of the triangular window are much smaller than those of a rectangular window.

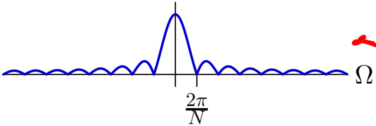
Other Window Functions

N=16

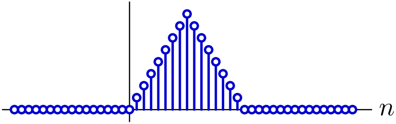
$$w_1[n] = u[n] - u[n-N]$$



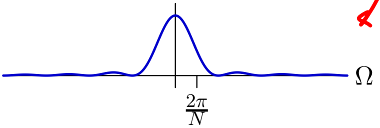
$$|W_1(\Omega)|$$



$$w_2[n] = \big(N/2 - \big|N/2 - n\big|\big)w_1[n]$$

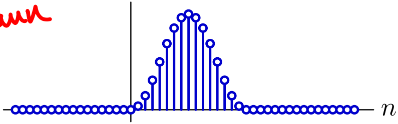


$$|W_2(\Omega)|$$

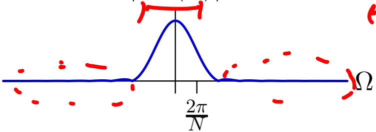


$$w_3[n] = \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)w_1[n]$$

Hann

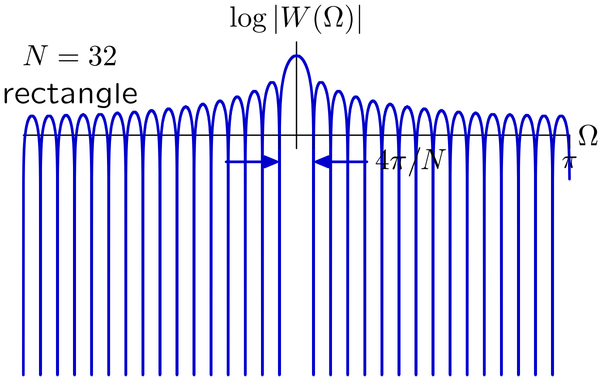


$$|W_3(\Omega)|$$



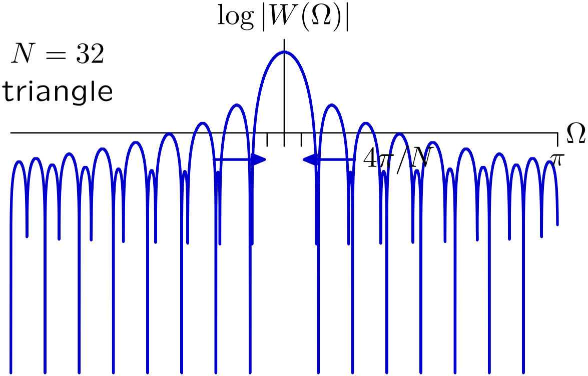
Windowing

Shape in time affects shape in frequency.



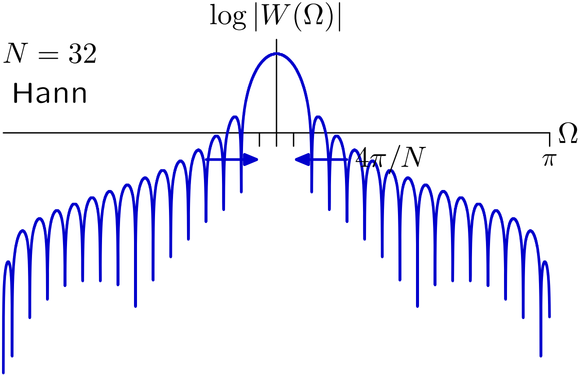
Windowing

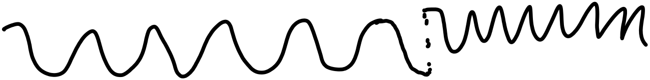
Shape in time affects shape in frequency.



Windowing

Shape in time affects shape in frequency.





Windowing

For many purposes (e.g., spectrograms), the smaller sidelobes are more important than the half-width of the central lobe, so higher order windows (triangles, Hann, etc.) are commonly used.

Regardless of the choice of window shape, increasing N still increases frequency resolution (at the cost of time resolution).