§ 3.3 (No Wiener-Filtering)
§ 3.4 Pyramid

\[ f_d(n) \text{ notation } f_d [1] \]

\[ f_d(n) = \sum_{k=0}^{D-1} f_0(\overline{n}) e^{-j2\pi sk} \]

\( D \) is the length of the signal, assume \( T = 1 \) (sampling interval)

To recover the original \( \int_{-\infty}^{\infty} f_d(s) e^{j2\pi st} \, dt \)

Assume that we virtually "xerox" (replicate) the original signal:

\[ f_d(n) \ast \sum_{k=-\infty}^{\infty} s(n-kD) \]

\[ F_{dd}(k) = F_d(s) \cdot \sum s(s - \frac{k}{D}) \]

(sampling in Fourier Domain)
Sampling the Fourier Transform D time over each period!

Isolating the interval \([-\frac{1}{2}, \frac{1}{2}]\) defines the DFT:

\[
\text{DFT} \{ f_d(n) \} = F_d(k) = \sum_{n=0}^{D-1} f_d(n) e^{-j2\pi \frac{k n}{D}}
\]

Remember \( \mathcal{F} \left( s \right) = s \left( s - \frac{k}{D} \right) \)

Discrete argument

FFT is its NlogN implementation.

* Convolution of discrete signals

\[
\text{CONV} ( \quad , \text{same} ) \quad \text{MATLAB}
\]

\( f_d(n) \) of length \( D=8 \)

\( h_d(n) \) of length \( M=4 \)

\( g_{d} = f_{d} \ast h_{d} = \sum_{m=0}^{M-1} f_{d}(m) h_{d}(n-m) \)

'if \( f_{d}=0 \) elsewhere \( g_{d} \) has length \( D+M-1 \)'
Assuming \( f_d = 0 \) elsewhere is in contradiction with our assumption that \( f_d \) was "xeroxed".

\[ \begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
\end{array} \]

\( h_d(n) \) has infinite length conv ('cyclic') (still periodic)

only then \( G_{dd}(k) = F_{dd}(k) \cdot H_{dd}(k) \)

convolution theorem both in discrete?

Short note:

\[
(f_0 x^3 + f_1 x^2 + f_2 x + f_3)(h_0 x + h_1)
= f_0 h_0 x^4 + (f_0 h_1 + f_1 h_0) x^3 + (f_1 h_1 + f_2 h_0) x^2 + (f_2 h_1 + f_3 h_0) x + f_3 h_1
\]

This can only mean a cycle and match (convolution theorem).
*Pyramids*

Typical mistake: \(640 \times 480 = 300k\). Algorithm designed for \(300k\) pixels is too slow for \(12M\) pixels.

\[
\begin{align*}
g & (1:2: \text{size}(g, 1), 1:2: \text{size}(g, 2))
\end{align*}
\]

Subsampling of discrete signals \(\mathbb{F}^2\)

(Wrong)

If we apply a filter which eliminates all frequencies from \(\frac{1}{4}\) to \(\frac{1}{2}\), then subsampling is lossless.

This could happen only with \(\text{rect}(2\pi)\)

\[
\text{rect}(s) \xrightarrow{\mathcal{F}} \begin{cases} 1 & \text{if } |s| < \frac{1}{2} \\ 0 & \text{else} \end{cases}
\]

but \(\mathcal{F}^{-1}\) is the discrete sinc with infinite length.

The art of subsampling is building filters (smoothing, low-pass) which are as close as possible in the frequency domain.

The bad news is that the closer to rect the larger the mask.