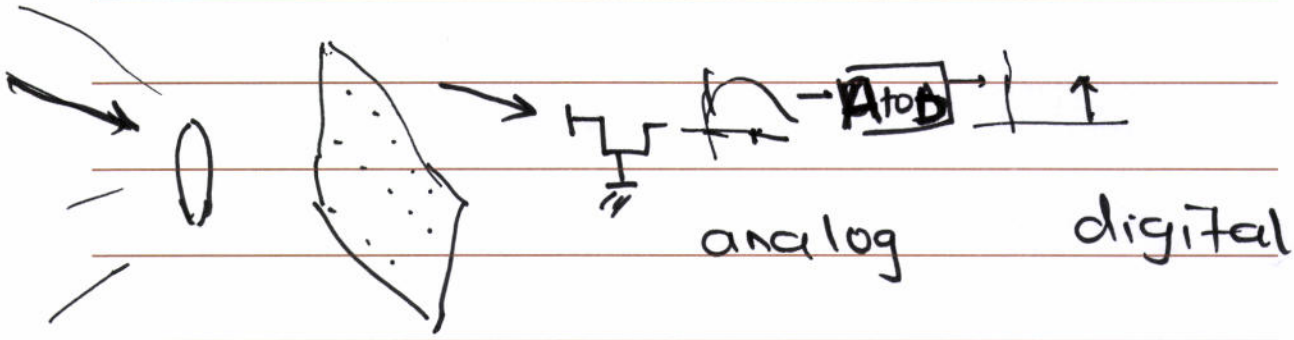
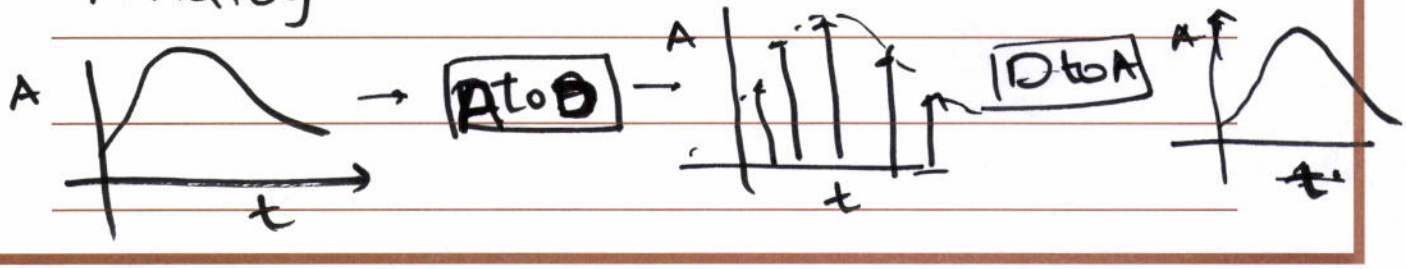
Camera



analog digital

Analog



continuous discrete analog

A to D

sampling

Quantization

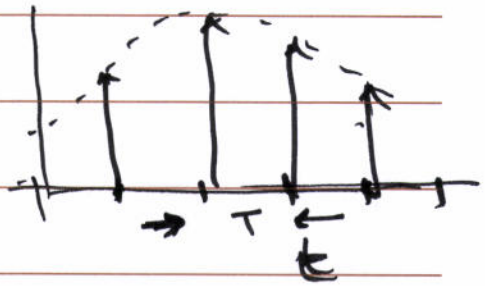
Sampling

$$y = x(t)$$

$$x_s[n] = x(nT)$$

T is sampling period

$$x_s[1] = x(T)$$
$$x_s[2] = x(2T)$$
$$x_s[3] = x(3T)$$
$$\vdots$$



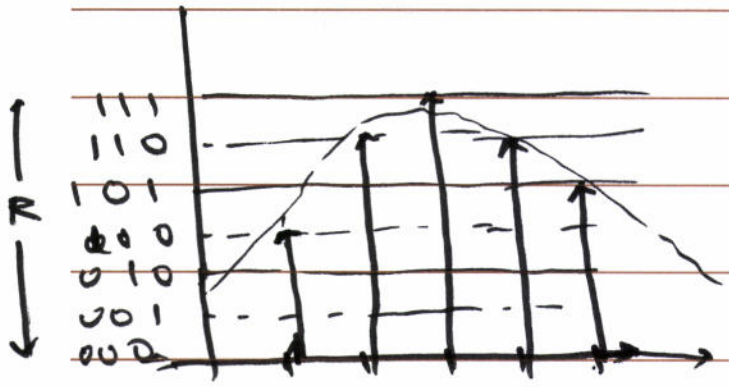
$\therefore$ $f = \frac{1}{T}$ sampling frequency

$T \uparrow\uparrow \rightarrow f \downarrow\downarrow$ data decreases

$T \downarrow\downarrow \rightarrow f \uparrow\uparrow$ data increases

Quantization

$$x_q[n] = Q\left[x_s[n]\right]$$



range R

each sample → b bits

$\frac{1}{2}\left(\frac{R}{2^b - 1}\right) \Rightarrow$ max error

Bitrate

= bits/second

$= \frac{\text{samples}}{\text{second}} \times \frac{\text{bits}}{\text{samples}}$

$= F \times b$

F = sampling freq

b = bits/sample

~~What should T be?~~

LTI - Linear Time Invariant

$y = x(t)$

$x_1 \rightarrow y_1 \quad x_2 \rightarrow y_2$

$c_1 x_1 + c_2 x_2 \rightarrow c_1 y_1 + c_2 y_2$

$y_1 = x(t_1) \quad y_2 = x(t_2)$

$x(c_1 t_1 + c_2 t_2) = c_1 y_1 + c_2 y_2$

$y = kx$ linear?

x_1 $y_1 = kx_1$

x_2 $y_2 = kx_2$

$x_1 + x_2$ $y = k(x_1 + x_2) = kx_1 + kx_2$

$= y_1 + y_2$

$y = x^2$ linear? NO

$y = \frac{dx}{dt}$

$y = x + 9$

x_1 : $y_1 = x_1 + 9$

x_2 : $y_2 = x_2 + 9$

$x_1 + x_2$: $y = x_1 + x_2 + 9 \neq y_1 + y_2$

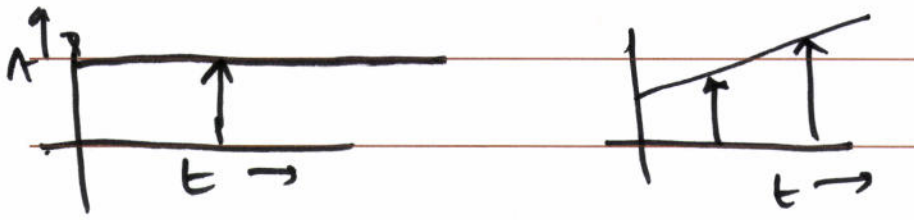
Time domain view:

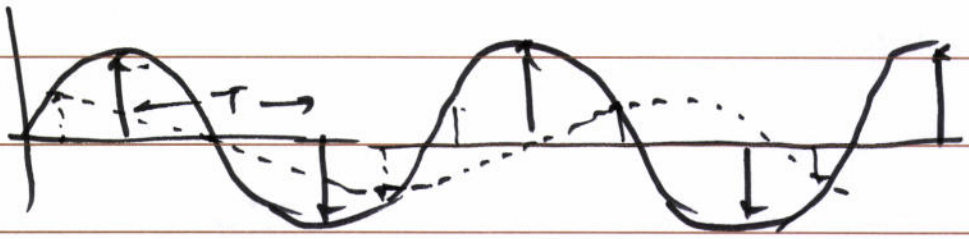
If your system is LTI, & you known how it responds to a delta function →

$x(t) \longrightarrow \boxed{h(t)} \longrightarrow y(t)$

$y(t) = h(t) \circledast x(t)$

What should T be?



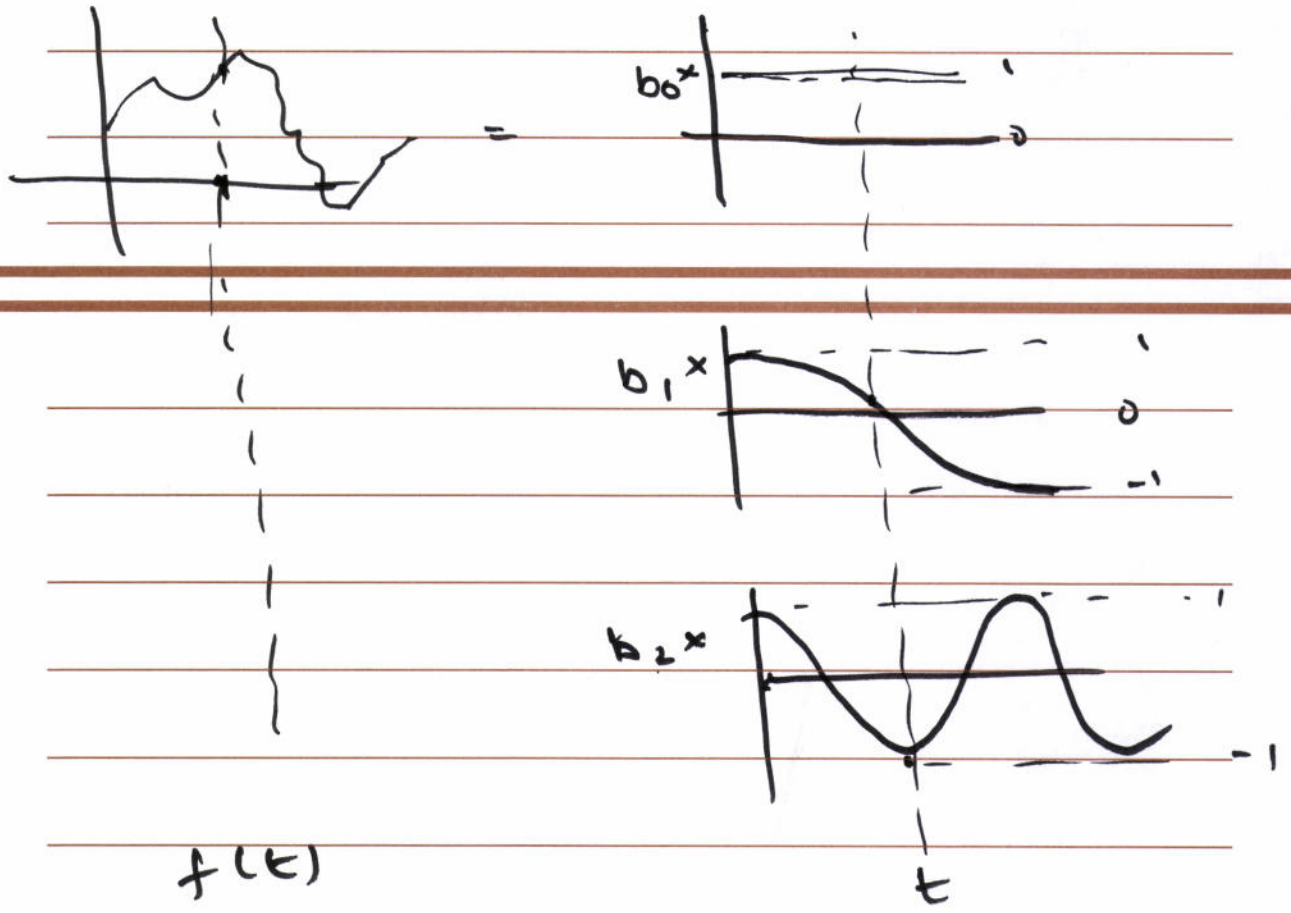


F > 2x max freq. in signal

$$f(t) = \sum a_i \sin i\omega t + \sum b_i \cos i\omega t$$

$$f(t) = \sum b_i \cos i\omega t$$

$$f(t) = b_0 + b_1 \cos\omega t + b_2 \cos 2\omega t + \cdots$$

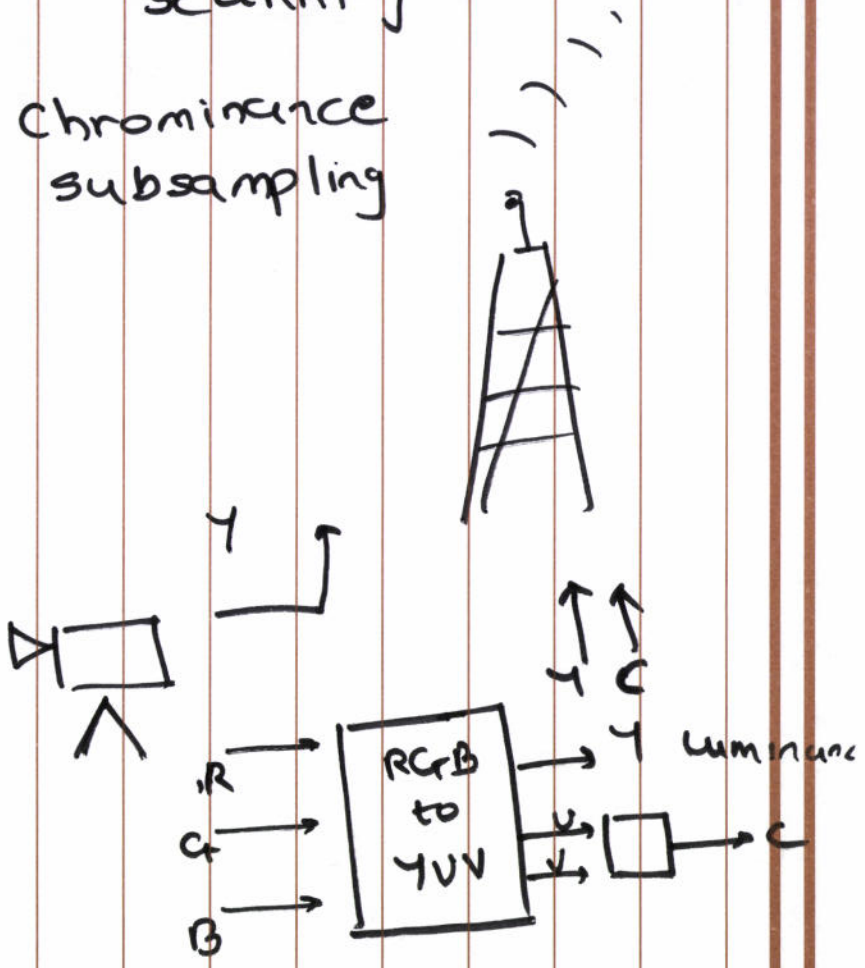


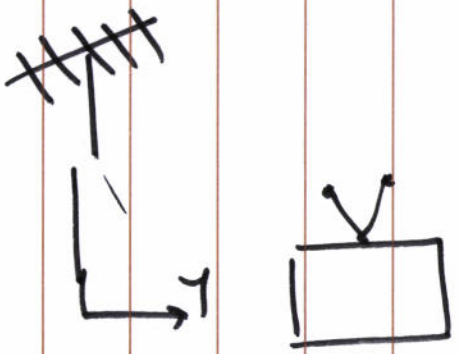
$$b_i = \int_{-\infty}^{\infty} f(t) \cos i\omega t \, dt$$

synchronization

interlaced scanning

Chrominance subsampling





RGB → YUV

YUV 4:4:4

for every four pixels

keep all Y 4×8 = 32

keep all U 4×8 = 32

keep all V 4×8 = 32

96

avg bits/pixel = 96/4

= 24

YUV 4:2:2

for every four pixels

keep all Y 4×8 = 32

keep only 2 U 2×8 = 16

keep only 2 V 2×8 = 16

64

avg bits/pixel = 64/4

= 16

YUV 4:2:0 4:1:1

for every four pixels

keep all Y → 4×8 = 32

keep 1 U → 1×8 = 8

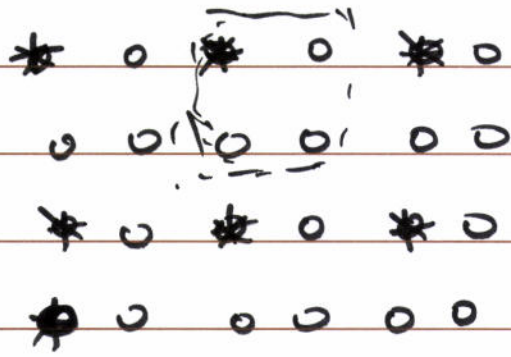
keep 1 V → 1×8 = 8

48

avg bits/pixel = $\frac{48}{4}$

= 12

4:2:0



4:1:1