Multimedia Software Systems
CS4551

Digital Data Acquisition

Analog Signal/Digital Data

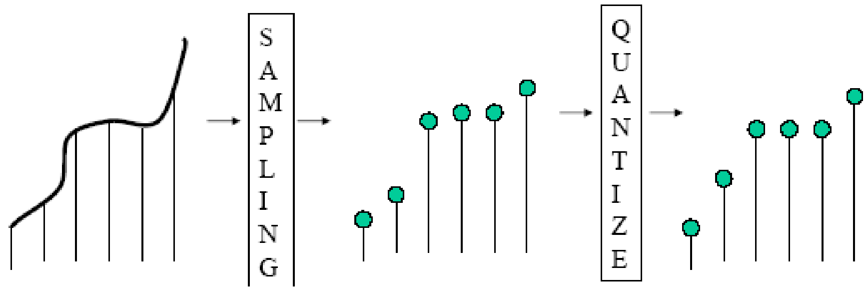
- **Analog signal** : continuous signal
- **Digital data**: discrete numbers (codes)

- **Digitization**: a process that converts analog signal to digital codes
- **Interpolation**: a process that recovers the original signal from digital codes

- **Analog-to-Digital Converter** (ADC): the device that converts analog signal to digital numbers
- **Digital-to-Analog Converter** (DAC): the device that converts digital codes to analog signal

Digitization

- The **digitization** of a signal comprises two steps:
  - **Sampling:** *extracting a subset of the signal*
  - **Quantization :** *approximating continuous range values by a set of discrete values (in b-bits)*



Fraction of a Second

1.0	second [s]
0.001 [thousandth]	millisecond [ms]
0.000 001 [millionth]	microsecond [μs]

Sampling and Sampling Period *T*

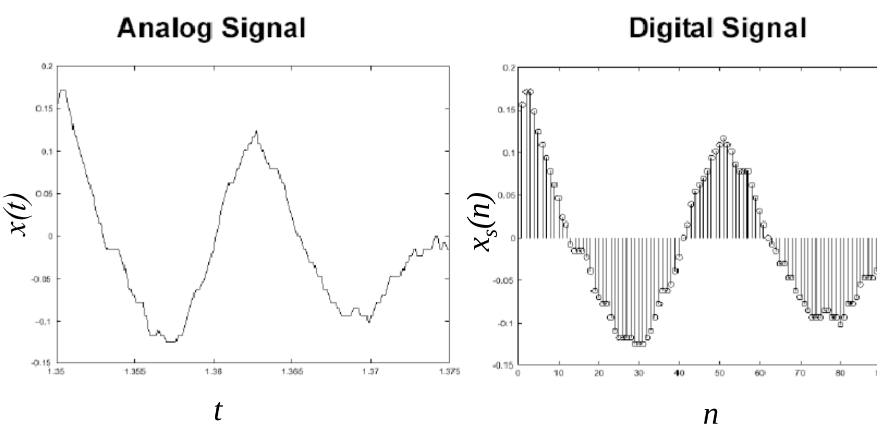
- Sampling of a signal $x(t)$ produces samples $x_s(n)$ that is n^{th} sampled signal value at time $n*T$.

$$x(t) \rightarrow x(nT) = x_s(n)$$

where $n = 1 \ldots$ and T is the sampling period

Examples:
- If T = 1ms/5ms/10ms, how many samples do we get per second?
- If T = 2ms, $x_s(1) = x(1*2ms)$, $x_s(2) = x(2*2ms)$

Sampling Example



Sampling Frequency *F*

- For a signal $x(t)$, this step produces $x(nT) = x_s(n)$, where T is the sampling period.
- $F=1/T$ is the sampling frequency.
  - If $T=1ms$, $F=1/0.001=1000$.
  - If $T=5ms$, $F=1/0.005=200$.
  - If $T=10ms$, $F=1/0.01=100$. F=100 means that sampling produces 100 samples per second.
- Sampling frequency is measure in Hz.
  - E.g F=100Hz, F=2Khz, etc

Sampling Issue

- The inverse transformation is called **Interpolation**
  - $x(t)$ from $x_s(n)$
- Issues
  - If the sampled signal is interpolated, how do you ensure that you get back the original signal? (How fast should we sample?)

Quantization

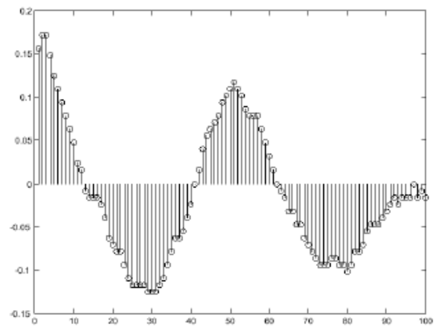
- The value at every sampled location is digitized. The digital domain has a finite bit representation. The sampled value is approximated to the nearest digital value.

$$x_s(n) \text{ -> } x_q(n)$$

- Formally
  - $x_q(n)=Q[x_s(n)]$, *where Q* is a quantization function which maps the values of $x_s(n)$ into N levels with a quantization step Δ.
  - Typically, $2^{b-1}<N \leq 2^b$ so that we need ***b*** bits to represent one quantized sample.
- Issues
  - What is the correct quantization step?
  - Quantization errors may result!

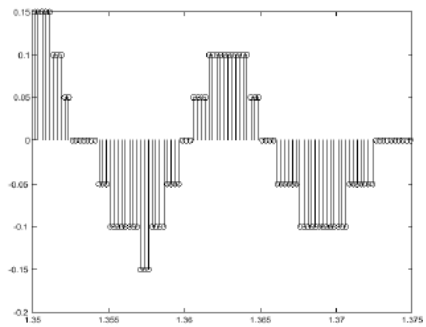
Quantization

$x_s(n)$

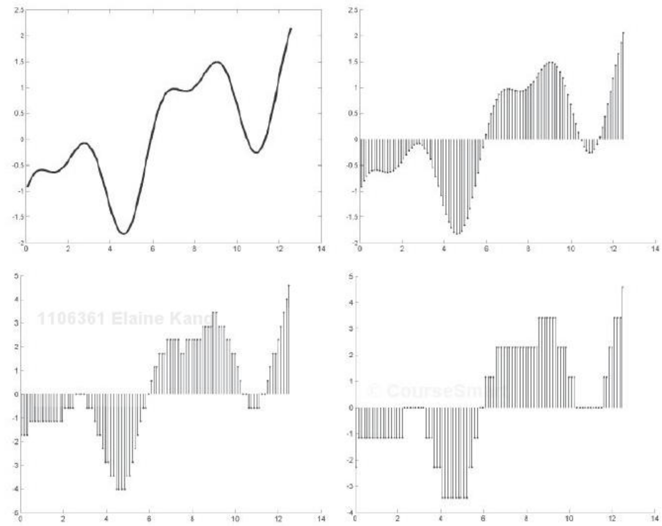


n

$x_q(n)$

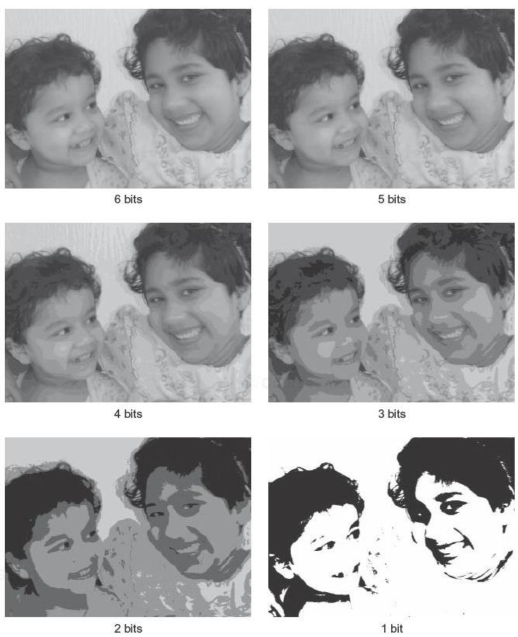


t, Δ=0.05



Original analog signal (upper left) is shown sampled and quantized at different quantization levels. For quantization, 8 bits (256 levels), 4 bits (16 levels), and 3 bits (8 levels) were used to produce the digital signals on the top right, bottom left, and bottom right, respectively.

CSULA CS451 Multimedia Software Systems by Eun-Young Kang



CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Digitization

- The **digitization** of a signal *x(t)* comprises two steps:

Sampling

$x(t) \rightarrow x(nT) = x_s(n)$

Quantization

$x_s(n) \rightarrow x_q(n)$

Digitization

$x(t) \rightarrow x_s(n) \rightarrow x_q(n)$

Bit Rate

- How many bits do you get per second (bits/second)?

Bit rate =(*number of samples per second*) x (*bits per sample*)

- As sampling rate increases, bit rate increases
- As quantization bits used increase, bit rate increases

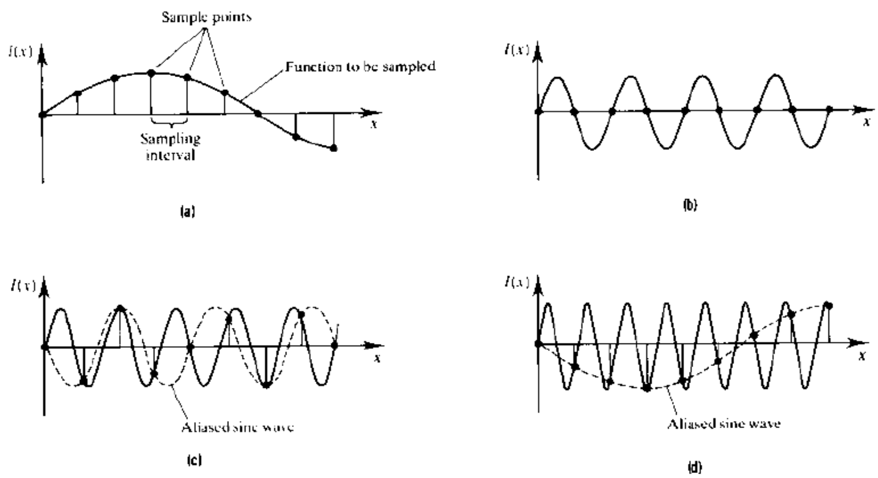
Bit Rate Examples

- Digitized audio?
  - Sampling frequency: F= 44.1 KHz (sampling period: T= 1/F = 0.02 ms)
  - Quantization with b =16 bits (N= 2^{16}=65,536)
  - Bit-rate = b F(= b/T) = 44.1K * 16 = 705.6 Kb/s (1K = 10^3 bit)
  - Example: 1 minute of uncompressed stereo music takes more than 10 MB!
- Digitized speech?
  - Sampling frequency: F = 8 KHz (why?)
  - Quantization with b = 16 bits
  - Bit-rate = bF = 8K * 16 = 128 Kb/s (1K = 10^3 bit)

What is the correct sampling rate?

- If *F* is too large *(T* is too small), we obtain too high a bitrate
- If *F* is too small *(T* is too large), too much information is lost in the sampling process
- We want to capture as much information as necessary to represent the signal correctly

Sampling Example



CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Sampling Rate

- We can see that the sampling rate affects the reconstructed signal.
- How can we have a correct sampling rate depending on a signal?
- Sampling Theorem: The *minimum sampling rate* for "correct" sampling depends on the frequency characteristics of the signal

CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Nyquist-Shannon's Sampling Theorem

- Nyquist-Shannon's Sampling Theorem
  - A continuous signal can be completely reconstructed from a set of equally spaced samples **iff** the sampling frequency is greater than **twice** the **highest frequency** of the signal
- Nyquist rate f = 2B, where B is the highest frequency of the signal
- Issue - How can we know the highest frequency of a signal?
  - Fourier Transform : An equivalent frequency spectrum representation of a signal in the temporal/spatial domain

Frequency of a Signal

x(t) → **Fourier Transform:** → *X(f)*

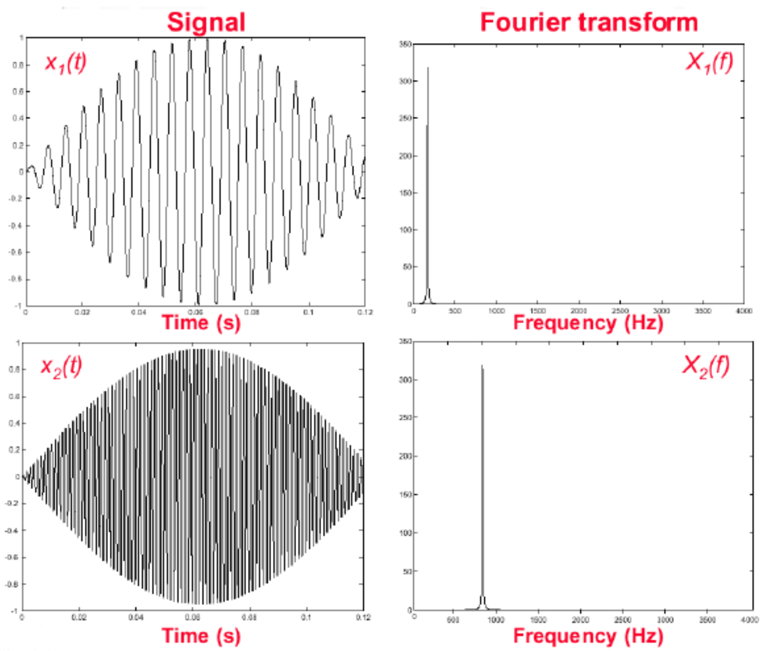
$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-i2\pi f x}\, dt$$

from spatial/temporal domain to frequency domain

x(t) ← **Inverse Fourier Transform:** ← *X(f)*

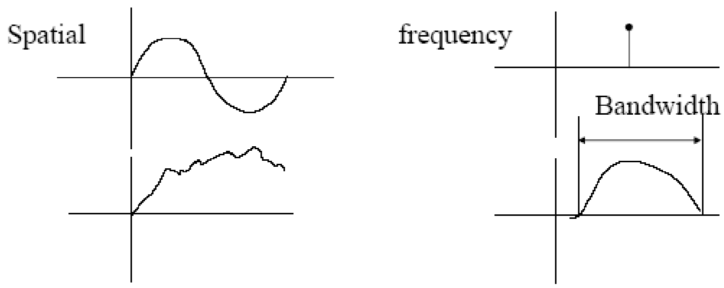
$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{i2\pi u x}\, df$$

from frequency domain back to spatial/temporal domain



Band-limited Signals

- Fourier Transform *X(f)* of a signal *x(t)*
  - An equivalent frequency spectrum representation of a signal in the spatial domain
  - Reminder: frequency is measured in Hz
- **Band (B) of a signal**: highest frequency “contained” in the signal
- If the highest frequency in *X(f)* is B, we say ***x(t)* is Band Limited to B**

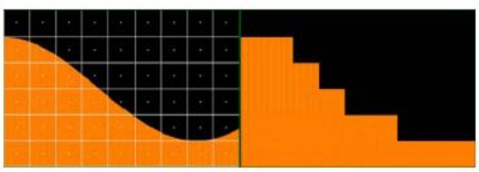


Band-limited Signals

- Sampling Theorem Revisited
  - Let *x(t)* have a maximum frequency B. Then we can "perfectly" interpolate *x(t)* from its sampled version $x_s(n)$ only if sampling rate F > 2B (i.e. the sampling period T < 1/(2B))
- Otherwise – aliasing – "Aliasing" is a form of non–linear, signal–dependent distortion

Aliasing Examples

- Audio aliasing
  - without aliasing
  - with aliasing
- Image aliasing



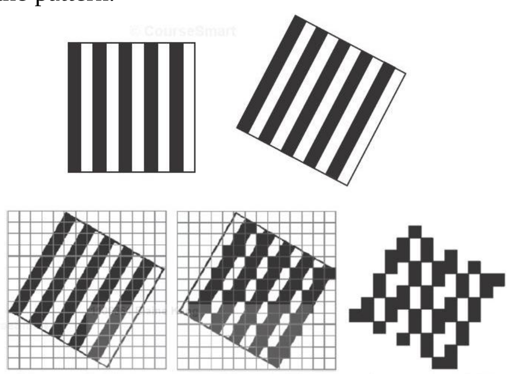
- Examples of Spatial aliasing and Temporal aliasing

Spatial Domain Aliasing

CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Moiré Patterns

- Moiré effect (an example of aliasing)
  - A different pattern will result in in the reconstructed image if the pattern in the original image is not sampled at a rate that is at least twice the rate of repetition of the pattern.



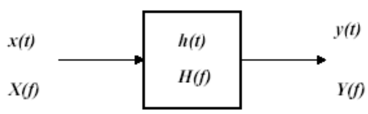
CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Moiré Patterns

Moiré patterns in images—the left image shows aliasing problems due to texture patterns on the tablecloth and the woman's clothes. These artifacts are minimized when the image is passed through a suitable low-pass filter as shown in the right image.

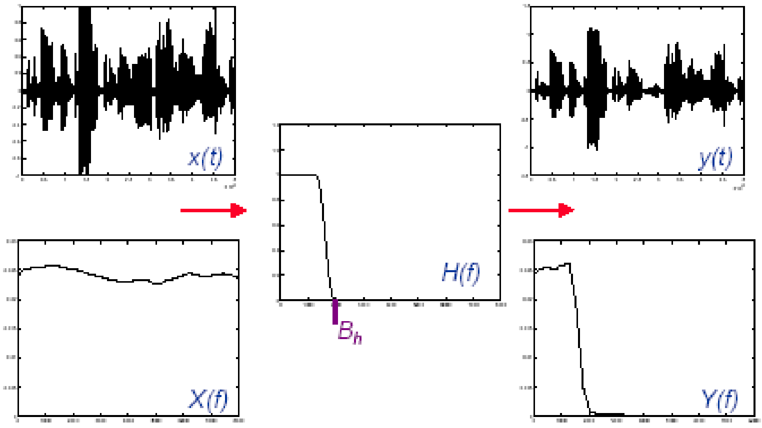
Signal Filtering

- A "*filter*" H(f) is an operator characterized by its frequency response :



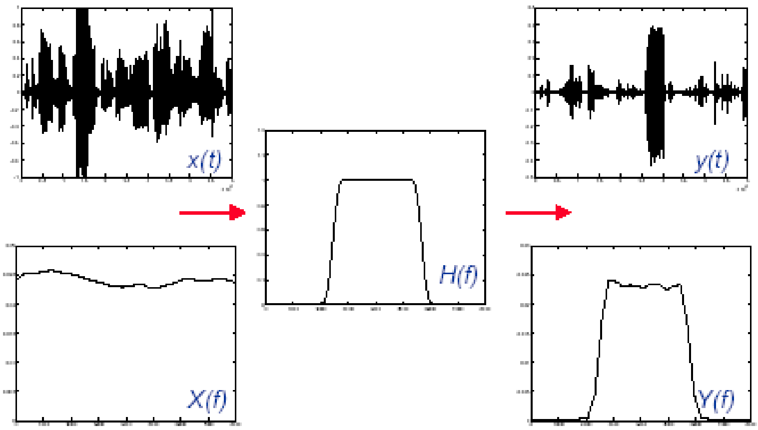
  - The Fourier transform of *y(t)* is *Y(f)=H(f)X(f)*
  - The band B_y of *y(t)* is smaller the band B_h of the filter
  - Filters can be *low-pass, band-pass* or *high-pass*

Example: Low-Pass Filter



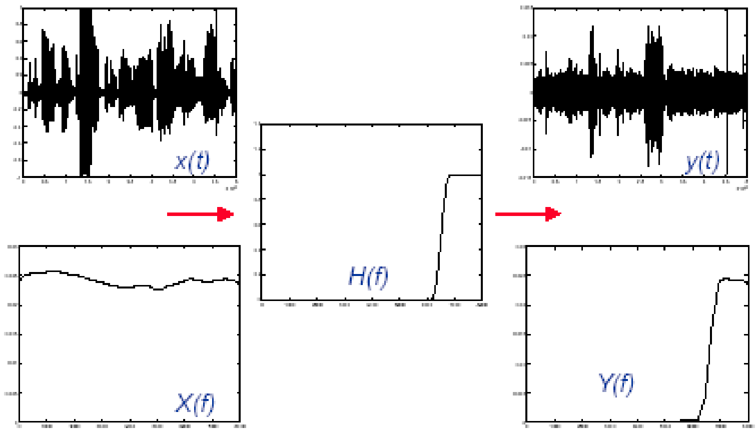
CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Example: Band-pass Filter

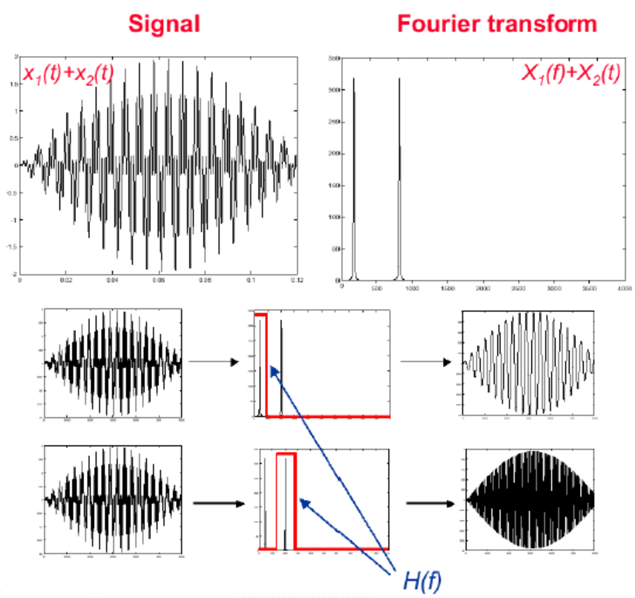


CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Example: High-pass Filter



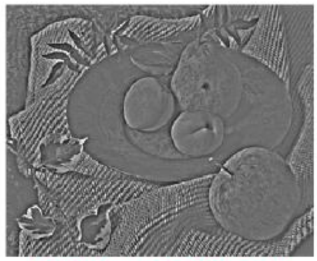
CSULA CS451 Multimedia Software Systems by Eun-Young Kang



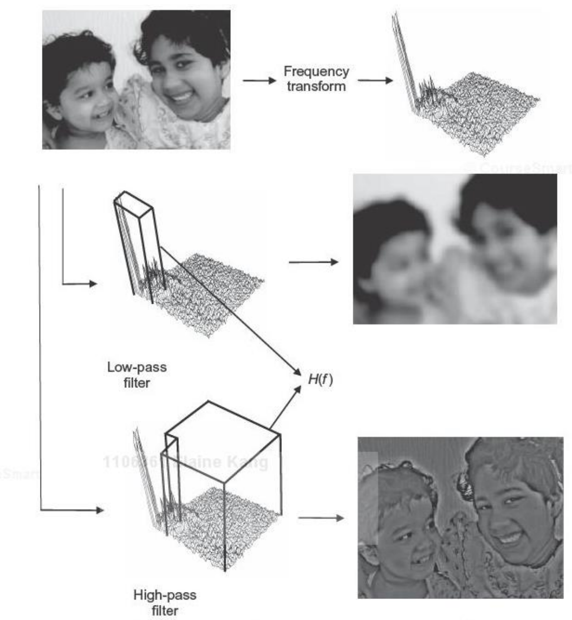
CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Original

Low-pass filtered



High-pass filtered



Prefiltering (Filtering Before Sampling)

- If we need to sample a signal that has band B using a sampling rate F<2B, we have two choices:
  - Sample the signal as it is, obtaining aliasing
  - **Or**
  - Prefilter the signal with a filter that has band B_h and use F = 2 B_h accordingly,
    - In this case, we lose all the frequencies between B and B_h (equal to F/2), but do not get aliasing.

    => In other words, if we use F lower than the highest frequency of the original signal, we cannot recover all frequencies above F/2. But no aliasing for the data lower than F/2.

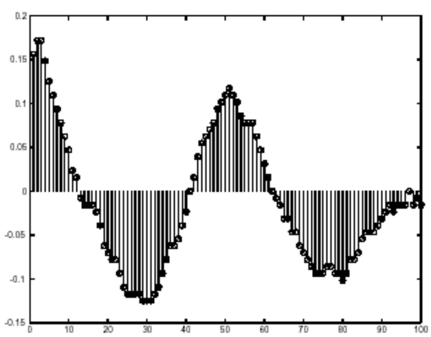
Examples Using Filters in Compression

- Audio Filtering Example
  - An microphone captures up to 100KHz (i.e. cutoff frequency is 100 KHz).
    - We should sample at 200KHz.
    - If Quantized at 16 bits per sample -> 3 Mbs!
  - But our hearing system can only detect frequencies up to ~20KHz.
  - Thus, prefilter the signal
    - Use a low-pass filter with filter frequency 20KHz. Then, we sample the signal at 40KHz producing only 640 Kb/s
    - In fact, for CD Audio we use a sampling frequency of 44.1KHz

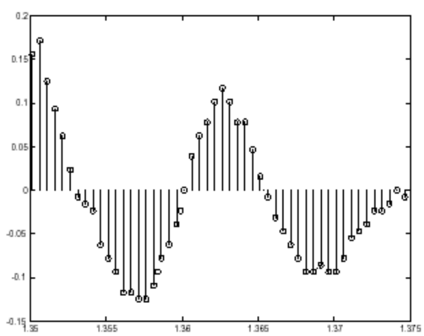
Subsampling (Decimation)

- Given $x_s(n)$, subsampling by M means generating a signal $y_s(n) = x_s(Mn)$ where $M > 1$
- Example: A continuous signal $x(t)$, band-limited to B=4KHz is sampled without aliasing with F=10 KHz.
- Suppose now we subsample the resulting signal by 2. This means sampling the original signal with rate F=5KHz (which gives *aliasing*)
- Solution: digital low-pass filter before subsampling.

Subsampling Example



$x(n)$



$y(n)=x(2n)$

Original Image

CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Subsampled by 2

Without prefiltering

With prefiltering

CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Subsampled by 4

Without prefiltering With prefiltering

What is the correct quantization step?

- If too many quantization levels $N(=2^b)$: too high a bit-rate
- If too few quantization levels $N(=2^b)$: distortion due to quantization error (a.k.a. quantization noise)

Statistical Definitions

- **Mean** of the signal *x(n)*, for a large sample space M:

$$\mu_x = (\sum_{n=1}^{n=M} x(n))/M$$

- The **variance** of the signal *x(n):*

$$\sigma_x^2 = (\sum_{n=1}^{n=M} (x(n)-\mu_x)^2)/M$$

- The power of the quantization error, σ_e^2 , is the variance of the signal $e(n)=x_q(n)-x(n)$

- The **signal-to-quantization noise ratio** (measured in *decibels dB*)

$$SQNR = 10\log_{10}(\sigma_x^2/\sigma_e^2) = 20\log_{10}(\sigma_x/\sigma_e)$$

Quantization - Facts

- Now, assume that $N= 2^b$, where b is the number of bits for representing the quantized value.
- Then SQNR = *20* x b x *log2* = 6.03 b (dB).
  - Thus, the SQNR increases by 6 dB for every bit we add when quantizing an input sample. So 16bits provide a maximum SQNR of 96dB.

Some Useful Numbers

Speech/Audio Type	Frequency Range	Sampling Rate	Bits/Sample	Uncompressed Bit Rate
Narrowband Speech	200-3200 Hz	8 kHz	16	128 kb/s
Wideband Speech	50-7000 Hz	16 kHz	16	256 kb/s
CD Audio	20-20000 Hz	44.1 kHz	16 x 2 channels	1.41 Mb/s

Quantization Example (1)

6 bits

5 bits

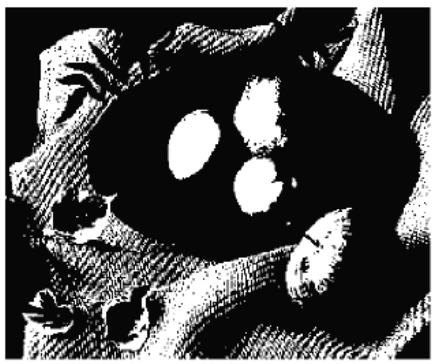
Quantization Example (2)

4 bits

3 bits

CSULA CS451 Multimedia Software Systems by Eun-Young Kang

Quantization Example (3)



2 bits

1 bits

CSULA CS451 Multimedia Software Systems by Eun-Young Kang