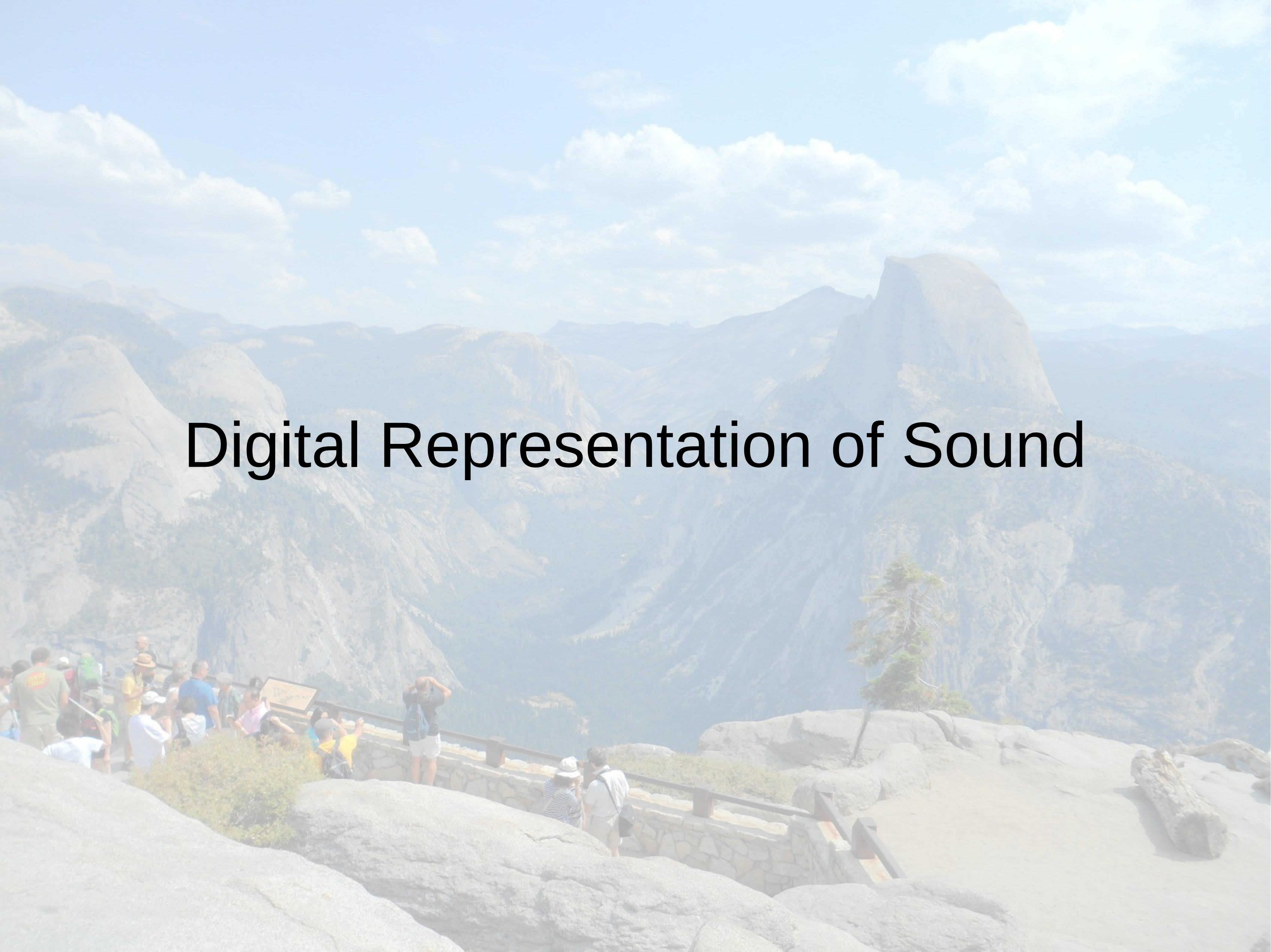


Digital Representation of Sound

Sound as Voltage

- Ears and microphones translate sound waves into electrical signals (voltages)
- Analog tape recorders
  - Store an essentially continuous signal
  - Recording is degraded during playback
- Digital samplers
  - Discretize the continuous signal
  - Information is lost in the process
  - Samples are faithfully represented

Digital Sampling

- Sampling rate
  - Voltage is sampled at regular time intervals:

$$t_n = \frac{n}{R}$$

    where *R* is the sampling rate

- Sample resolution
  - Voltages are assigned integer values (quantization)
  - A fixed number of bits is allowed for each sample

Common Sampling Rates

- 8000 Hz – telephony
- 11025 Hz, 22050 Hz – low quality audio for computer applications
- 44100 Hz – CD audio quality
- 48000 Hz, 96000 Hz – DVD audio

8 vs 16 Bit Sampling

- 8 bit samples
  - Unsigned integer values in range [0..255]
  - Midpoint value is 128
- 16 bit samples
  - Signed integer values in range [-32768..32767]
  - Midpoint value is 0

16 Bit Sampling Example

- Suppose that the voltage at time *t* seconds is

  $$V(t) = 23456(7t^2 - 1)$$

  and the sampling rate is $R = 5\ Hz$.

- Using 16 bit sampling, the first 4 samples values are

  $$V(0) = -23456$$
  $$V(1/5) = -16888.32 \Rightarrow -16888$$
  $$V(2/5) = 2814.72 \Rightarrow 2815$$
  $$V(3/5) = 35653.12\,(>32767) \Rightarrow 32767$$

8 Bit Sampling Example

- Assume the input voltage is

$$V(t) = 135(7t^2 - 1)$$

  *a*nd the sampling rate is *R = 5 Hz*

- With 8 bit sampling, the first 4 sample values are

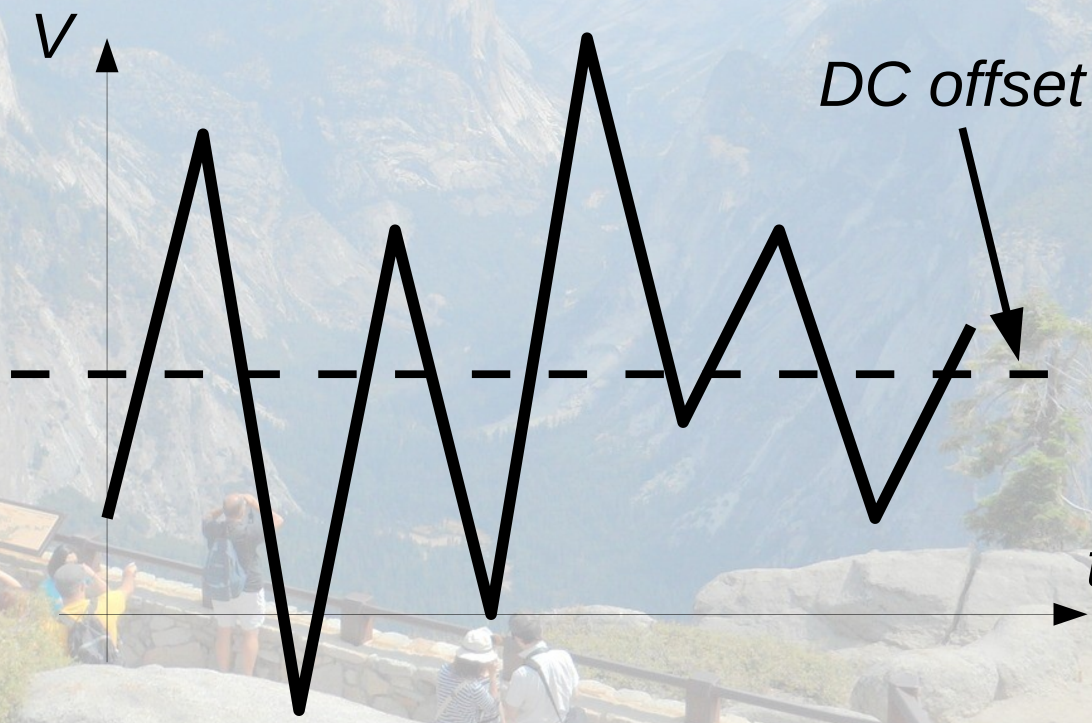
$$V(0) = -135 \Rightarrow -135 + 128 = -7 (<0) \Rightarrow 0$$
$$V(1/5) = -97.2 \Rightarrow -97.2 + 128 = 30.8 \Rightarrow 31$$
$$V(2/5) = 16.2 \Rightarrow 16.2 + 128 = 144.2 \Rightarrow 144$$
$$V(3/5) = 205.2 \Rightarrow 205.2 + 128 = 332.2 (>255) \Rightarrow 255$$

Other Sample Resolutions

- 24 and 32 bit (signed) integer
- 32 bit floating point
- Used for audio mixing and mastering to avoid overflow when combining several audio sources
- Final version is down-sampled to 16 bits

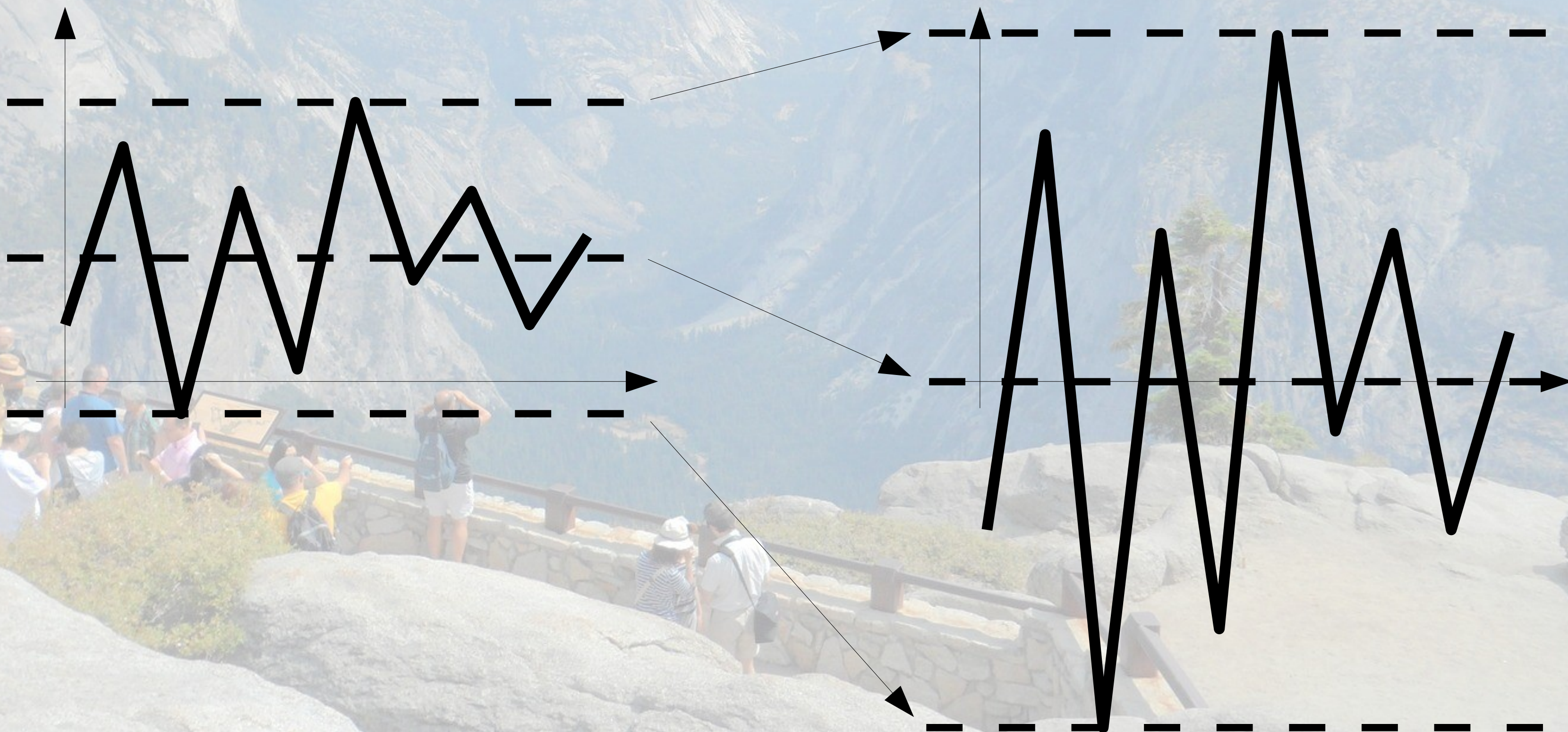
DC Offset

- The **DC Offset** of a signal is the average of the sample values



Normalization

- Normalized audio data
  - DC offset of 0
  - Volume set to a maximum (specified) value



Normalization Example

- *Example:* we normalize the values

$$47, -102, 63, 95$$

to a maximum of 200.

  - The DC offset is:

$$(47 - 102 + 63 + 95) / 4 = 25.75$$

  - Remove the DC offset from the samples:

$$47 - 25.75 = 21.25, \quad -102 - 25.75 = -127.75,$$

$$63 - 25.75 = 37.25, \quad 95 - 21.25 = 69.25$$

Normalization Example (continued)

- Scale samples by $m = 200/127.75$ to maximize:

$$21.25 * m = 33.27, \quad -127.75 * m = -200,$$

$$37.25 * m = 58.32, \quad 69.25 * m = 108.41$$

- If we are using (say) 16 bit resolution, we should round to the nearest integer. The normalized samples are thus:

$$33, -200, 58, 108$$

- *Remark:* to avoid numerical overflow, we should perform intermediate computations using floating point arithmetic

Decibel Scale

- Decibels measure the logarithmic change between two voltage/amplitude values
- A voltage change from V_0 to V_1 corresponds to

$$20\log\left(\frac{V_1}{V_0}\right)\ \ \textit{decibels}$$

- *Example:* increasing the voltage from *300* to *500* corresponds to an increase of

$$20\log(500/300) \approx 4.44\ dB$$

Gain and Volume

- Change of volume: multiply signal values by a **gain factor** *g*
  - Input: *x(t)*, output: *y(t) = g x(t)*
  - The gain is often specified in decibels:

$$g = 10^{dB/20}$$

- Maximum volume/level is specified relative to the maximum possible volume
  - *Example: 16* bit audio, volume of *-3* dB

$$(max\, value) = (2^{15} - 1)\, 10^{-3/20} \approx 23197.3$$

Nyquist Limit

- For a sampling rate *R*, only signals with a frequency *f* with

$$f \leq R/2$$

  can be represented
- A periodic wave requires at least 2 sample points per cycle: one for the maximum, one for the minimum

Implications of the Nyquist Limit

- Since humans can here frequencies in the (approximate) range

  20 – 20,000 Hz

  a sampling rate of at least 40 kHz is needed to reproduce all sounds within the human range of hearing
- Canine hearing is approximately 40-60,000 Hz, so a compact disc cannot represent all sounds that are audible to a dog

Bit Resolution and Noise

- Using a lower bit resolution introduces noise into the signal
- The signal to noise ratio is given by

$$SNR = \frac{(range\ of\ values)}{(smallest\ difference)}$$

(usually measured in decibels)

- 8 bits: *SNR = 48 dB*, 16 bits: *SNR = 96 dB*

WAVE files

- Most common format for storing uncompressed audio data
- Simple file format
- 8 and 16 bit sample resolutions
- 1 and 2 channel (mono and stereo) data
- Allows for some types of compression, although this is not commonly used

Interpolation (1)

- Digital audio data is obtained by sampling a continuous signal at a given rate *R*

$$samples: \quad x_0, x_1, \ldots, x_{N-1}$$

- There are situations when we need the same signal sampled at a different rate *R'*
  - Change the sampling rate of a WAVE file
  - Speed up and slow down audio effects
  - Time delay audio effects

Interpolation (2)

- The approximate original continuous signal must be reconstructed using an interpolation technique
- Ideally, the reconstruction should not introduce frequency artifacts
  - For down-sampling ($R' < R$), the frequency spectrum should be the same as the original up to the Nyquist frequency ($R'/2$)
  - For up-sampling ($R' > R$), the frequency spectrum should be the same as the original for all frequencies

Band-limited Interpolation (1)

- Nearly ideal interpolation
- Computationally expensive [$O(N^2)$]
- Reconstruction of continuous signal

$$y(t) = \sum_{m=0}^{N-1} x_m sinc(Rt - m)$$

$$sinc(x) = \left| \begin{matrix} 1 \quad if\ x = 0 \\ \sin(\pi x)/\pi x \quad if\ x \neq 0 \end{matrix} \right|$$

Band-limited interpolation (2)

- Resample at rate *R'*

$$samples: \quad y_0, y_1, \ldots, y_{N'-1}$$

$$y_n = y(n/R') = \sum_{m=0}^{N-1} x_m \, sinc(\alpha n - m)$$

where $\alpha = R/R'$

- Number of output samples

$$N' = floor((N-1)/\alpha) + 1$$

Linear Interpolation

- Computationally efficient [*O(N)*]
- Introduces frequency artifacts
- Reconstruction of continuous signal

$$k = floor(Rt)$$
$$y(t) = x_k + (Rt - k)(x_{k+1} - x_k)$$

provided that *k* = *0, 1, …, N-2*

Applications

- Data may be sped up (or slowed down) by a factor of α on playback
  - Output data: $y_0, y_1, \ldots, y_{N'-1}$
  - Datum y_j is interpolated value at time $t = \alpha j/R$
    - $y_j = y(\alpha j/R)$
    - Fractional index αj: $y_j = x_{\alpha j}$
- Data sampled at the rate R can be stored as data with a different sampling rate R'
  - Effective speed up factor is $\alpha = R/R'$

Interpolation Example

- Input: $x_0 = 16$, $x_1 = 55$, $x_2 = -20$, $x_3 = 34$
- Want interpolated values for speed up by a factor of $\alpha = 1.2$

$$y_0 = x_0 = 16$$

$$y_1 = x_{1.2} = 55 + (-20 - 55)(1.2 - 1) = 40$$

$$y_2 = x_{2.4} = -20 + (34 - (-20))(2.4 - 2) = 1.6$$

Output values: *16, 40, 2* (rounded)

Resampling Example

- Input: $x_0 = 40$, $x_1 = 14$, $x_2 = -26$, $x_3 = 8$ sampled at *8 Hz*
- Want interpolated values for *10 Hz* sampling

  ($\alpha = 8/10 = 0.8$)

  $y_0 = x_0 = 40$

  $y_1 = x_{0.8} = 40 + (14 – 40)(0.8 – 0) = 19.2 \rightarrow 19$

  $y_2 = x_{1.6} = 14 + (-26 – 14)(1.6 – 1) = -10$

  $y_3 = x_{2.4} = -26 + (8 - (-26))(2.4 – 2) = -12.4 \rightarrow -12$