Scalar and Vector Quantization

National Chiao Tung University
Chun-Jen Tsai
11/06/2014

Basic Concept of Quantization

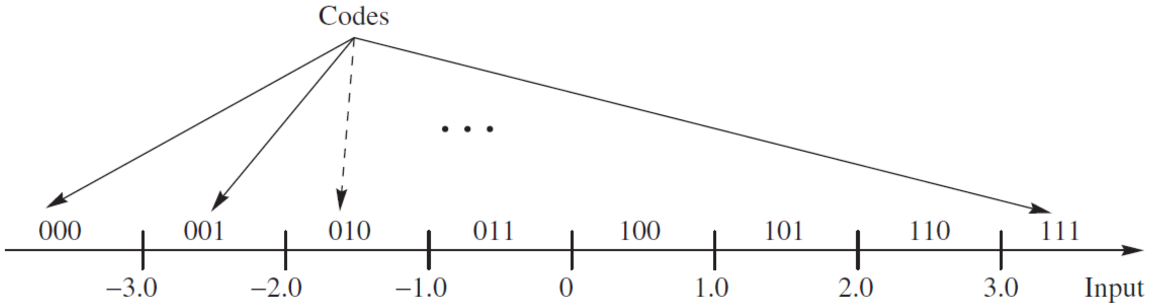
- Quantization is the process of representing a large, possibly infinite, set of values with a smaller set
- Example: real-to-integer conversion
  - Source: real numbers in the range $[-10.0, 10.0]$
  - Quantizer: $Q(x) = \lfloor x + 0.5 \rfloor$
    - $[-10.0, -10.0] \rightarrow \{ -10, -9, \ldots, -1, 0, 1, 2, \ldots, 9, 10\}$
- The set of inputs and outputs of a quantizer can be scalars (scalar quantizer) or vectors (vector quantizer)

The Quantization Problem

- Encoder mapping
  - Map a range of values to a codeword
  - Irreversible mapping
  - If source is analog $\rightarrow$ A/D converter
- Decoder mapping
  - Map the codeword to a (fixed) value representing the range
  - Knowledge of the source distribution can help us pick a better value representing each range
  - If output is analog $\rightarrow$ D/A converter
- Informally, the encoder mapping is called the quantization process, and the decoder mapping is called the inverse quantization process

Quantization Examples

- 3-bit Quantizer
  - Encoder (A/D)



- Decoder (D/A)

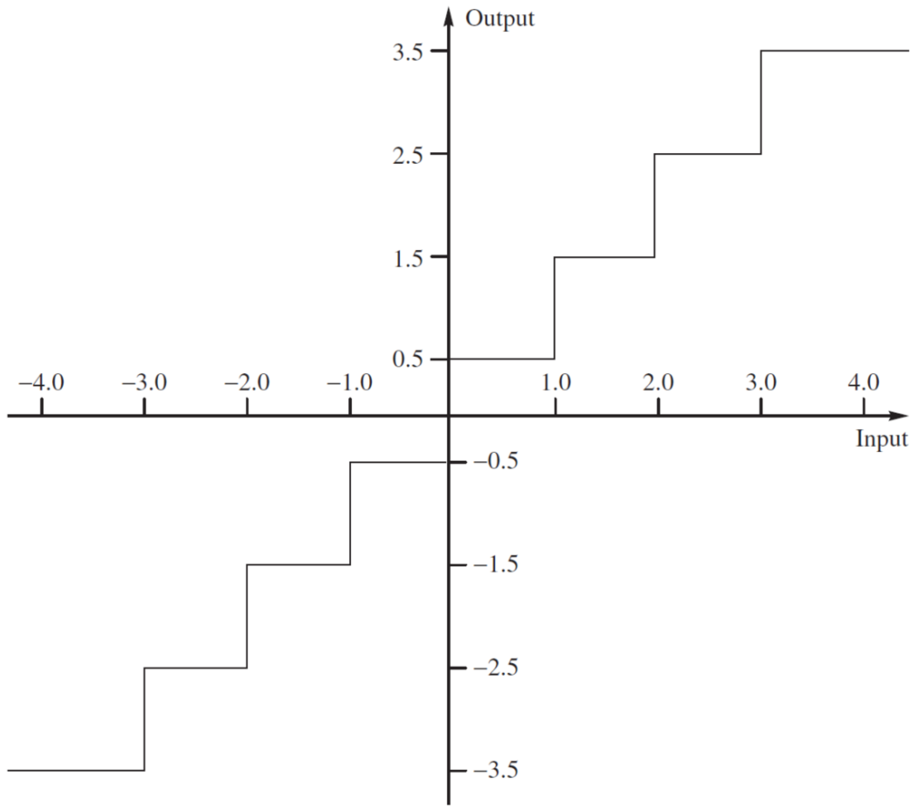
Input Codes	Output
000	−3.5
001	−2.5
010	−1.5
011	−0.5
100	0.5
101	1.5
110	2.5
111	3.5

- Digitizing a sine wave

t	$4\cos(2\pi t)$	A/D Output	D/A Output	Error
0.05	3.804	111	3.5	0.304
0.10	3.236	111	3.5	−0.264
0.15	2.351	110	2.5	−0.149
0.20	1.236	101	1.5	−0.264

Quantization Function

- ❑ A quantizer describes the relation between the encoder input values and the decoder output values
  - Example of a quantization function:



Quantization Problem Formulation

- Input:
  - X – random variable
  - $f_X(x)$ – probability density function (pdf)
- Output:
  - $\{b_i\}_{i=0..M}$ decision boundaries
  - $\{y_i\}_{i=1..M}$ reconstruction levels
- Discrete processes are often approximated by continuous distributions
  - Example: Laplacian model of pixel differences
  - If source is unbounded, then the first and the last decision boundaries = $\pm\infty$ (they are often called “saturation” values)

Quantization Error

- If the quantization operation is denoted by $Q(\cdot)$, then

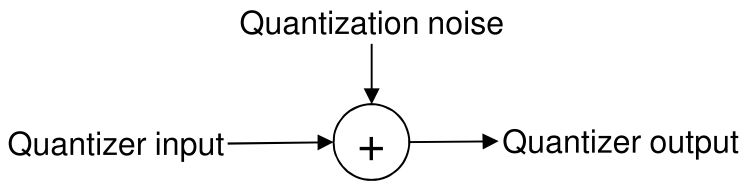
$$Q(x) = y_i \text{ iff } b_{i-1} < x \le b_i.$$

The mean squared quantization error (MSQE) is then

$$\sigma_q^2 = \int_{-\infty}^{\infty} \left(x - Q(x)\right)^2 f_X \, dx$$

$$= \sum_{i=1}^{M} \int_{b_{i-1}}^{b_i} \left(x - y_i\right)^2 f_X \, dx$$

- Quantization error is also called quantization noise or quantizer distortion, e.g., additive noise model:



Quantized Bitrate with FLC

- If the number of quantizer output is M, then the rate (per symbol) of the quantizer output is $R = \lceil \log_2 M \rceil$
  - Example: $M = 8 \rightarrow R = 3$

- Quantizer design problem:
  - Given an input pdf $f_X(x)$ and the number of levels M in the quantizer, find the decision boundaries $\{b_i\}$ and the reconstruction levels $\{y_i\}$ so as to minimize the mean squared quantization error

Quantized Bitrate with VLC

- For VLC representation of quantization intervals, the bitrate depends on decision boundary selection

- Example: eight-level quantizer:

y_1	1110
y_2	1100
y_3	100
y_4	00
y_5	01
y_6	101
y_7	1101
y_8	1111

$$R = \sum_{i=1}^{M} l_i P(y_i)$$

$$= \sum_{i=1}^{M} l_i \left[\int_{b_{i-1}}^{b_i} f_X(x) dx \right]$$

Optimization of Quantization

- Rate-optimized quantization
  - Given: Distortion constraint $\sigma_q^2 \leq D^*$
  - Find: $\{ b_i \}$, $\{ y_i \}$ binary codes
  - Such that: R is minimized

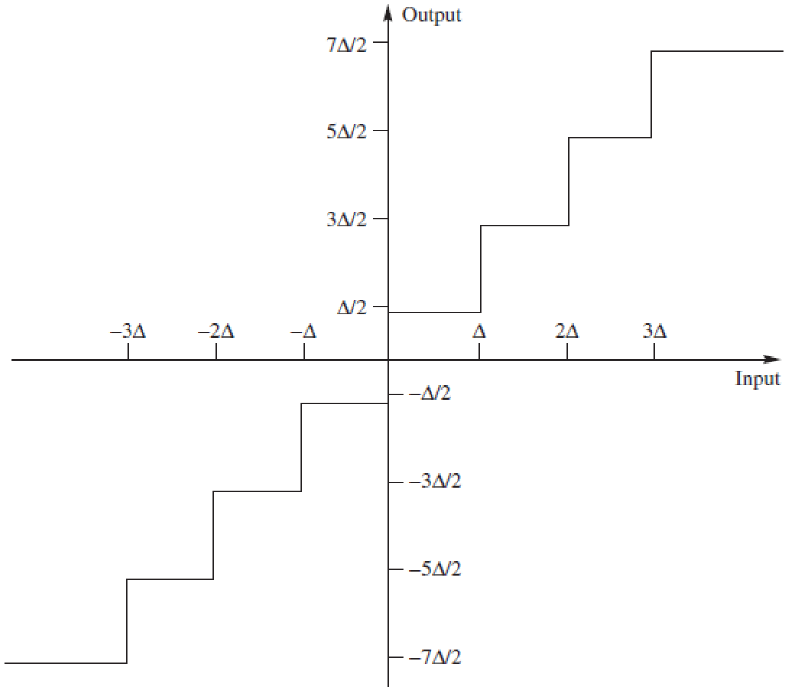
- Distortion-optimized quantization
  - Given: Rate constraint $R \leq R^*$
  - Find: $\{ b_i \}$, $\{ y_i \}$ binary codes
  - Such that: σ_q^2 is minimized

Uniform Quantizer

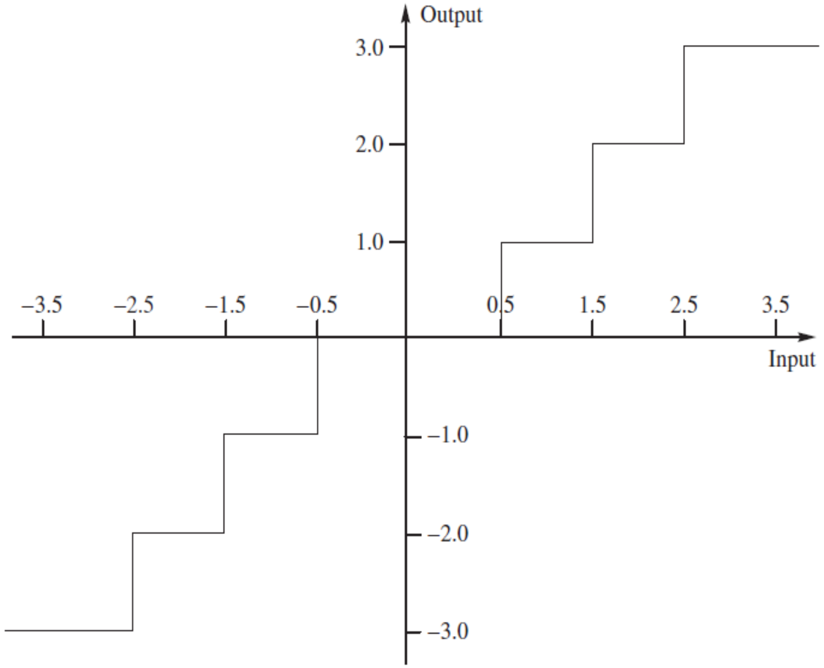
- All intervals are of the same size
  - Boundaries are evenly spaced (step size:Δ), except for out-most intervals
- Reconstruction
  - Usually the midpoint is selected as the representing value
- Quantizer types:
  - Midrise quantizer: zero is not an output level
  - Midtread quantizer: zero is an output level

Midrise vs. Midtread Quantizer

- Midrise



Midtread



Uniform Quantization of Uniform Source

- If the source is uniformly distributed in $[-X_{max}, X_{max}]$, the output is quantized by an M-level uniform quantizer, then the quantization step size is

$$\Delta = \frac{2X_{\max}}{M},$$

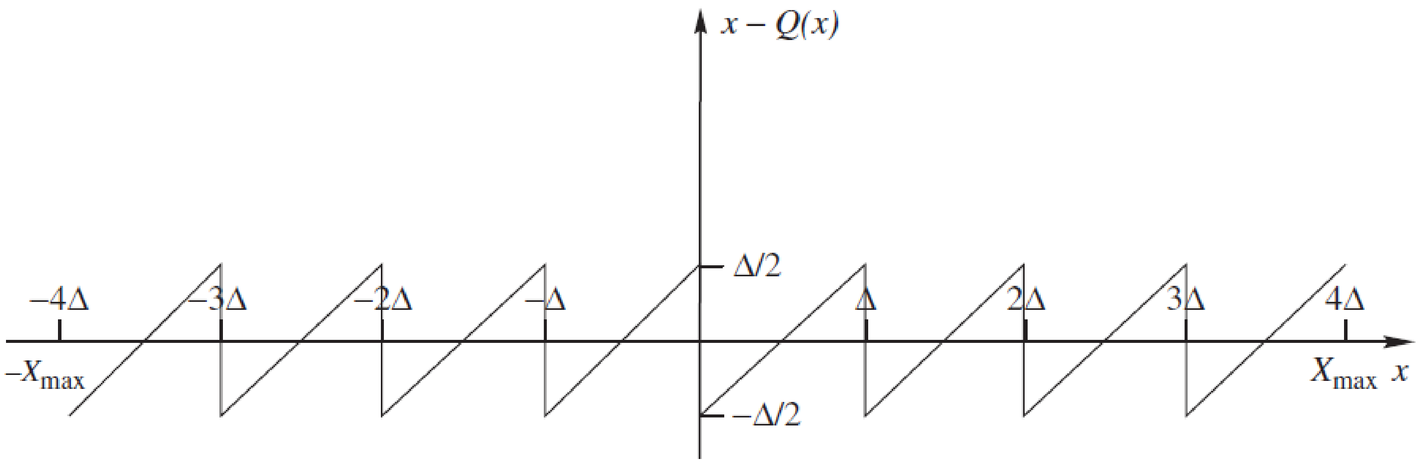
and the distortion is

$$\sigma_q^2 = 2\sum_{i=1}^{M/2}\int_{(i-1)\Delta}^{i\Delta}\left(x - \frac{2i-1}{2}\Delta\right)^2 \frac{1}{2X_{\max}}\,dx = \frac{\Delta^2}{12}.$$

Alternative MSQE Derivation

- We can also compute the "power" of quantization error $q = x - Q(x)$, $q \in [-\Delta/2, \Delta/2]$ by:

$$\sigma_q^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} q^2 dq = \frac{\Delta^2}{12}.$$



The SNR of Quantization

- For n-bit uniform quantization of an uniform source of $[-X_{max}, X_{max}]$, the SNR is $6.02n$ dB, where $n = \log_2 M$:

$$10\log_{10}\left(\frac{\sigma_s^2}{\sigma_q^2}\right) = 10\log_{10}\left(\frac{(2X_{\max})^2}{12}\cdot\frac{12}{\Delta^2}\right)$$

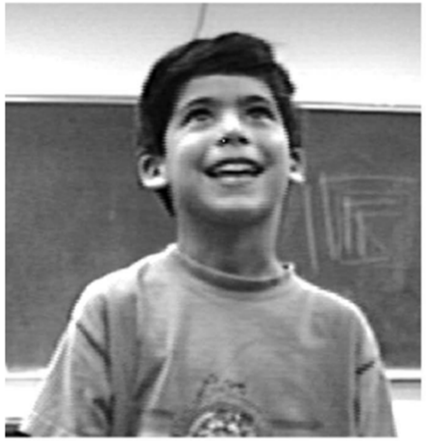
$$= 10\log_{10}\left(\frac{(2X_{\max})^2}{12}\frac{12}{\left(\frac{2X_{\max}}{M}\right)^2}\right) = 10\log_{10} M^2$$

$$= 20\log_{10} 2^n = 6.02n \text{ dB}.$$

Example: Quantization of Sena

- Darkening and contouring effects of quantization

8 bits / pixel



1 bits / pixel

2 bits / pixel

3 bits / pixel

Quantization of Non-uniform Sources

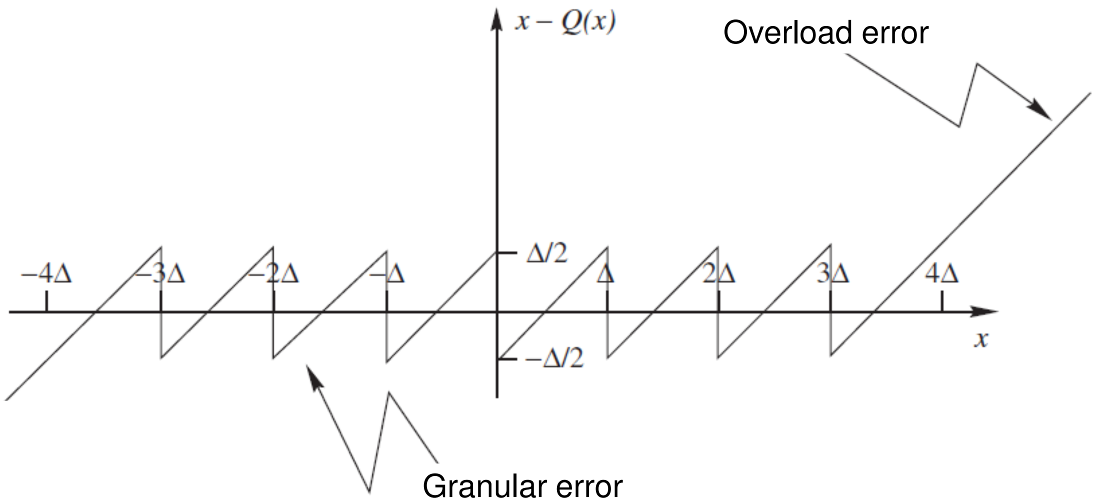
- Given a non-uniform source, $x \in [-100, 100]$, $P(x \in [-1, 1]) = 0.95$, and we want to design an 8-level (3-bit) quantizer.
- A naïve approach uses uniform quantizer ($\Delta = 25$):
  - 95% of sample values represented by only two numbers: –12.5 and 12.5, with a maximal quantization error of 12.5 and minimal error of 11.5
- If we use $\Delta = 0.3$ (two end-intervals would be huge)
  - Max error is now 98.95 (i.e. 100 – 1.05), however, 95% of the time the error is less than 0.15

Optimal Δ that minimizes MSQE

- Given pdf $f_X(x)$ of the source, let's design an M-level mid-rise uniform quantizer that minimizes MSQE:

$$\sigma_q^2 = 2\sum_{i=1}^{\frac{M}{2}-1}\int_{(i-1)\Delta}^{i\Delta}\left(x-\frac{2i-1}{2}\Delta\right)^2 f_X(x)dx \quad \rightarrow \text{Granular error}$$

$$+2\int_{\left(\frac{M}{2}-1\right)\Delta}^{\infty}\left(x-\frac{M-1}{2}\Delta\right)^2 f_X(x)dx. \quad \rightarrow \text{Overload error}$$



Solving for Optimum Step Sizes

- Given an $f_X(x)$ and M, we can solve for Δ numerically:

$$\frac{d\sigma_q^2}{d\Delta} = -\sum_{i=1}^{\frac{M}{2}-1}(2i-1)\int_{(i-1)\Delta}^{i\Delta}\left(x-\frac{2i-1}{2}\Delta\right)f_X(x)dx$$

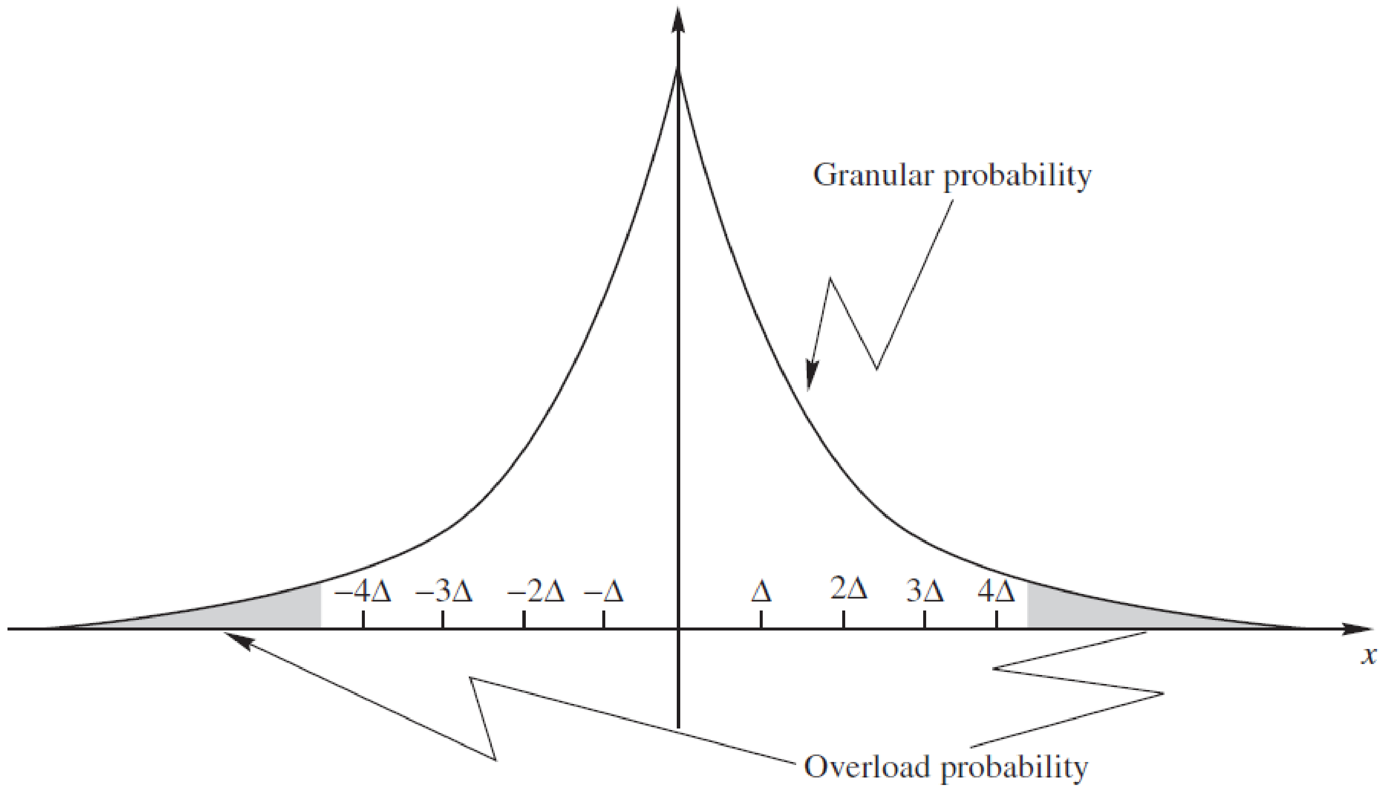
$$-(M-1)\int_{\left(\frac{M}{2}-1\right)\Delta}^{\infty}\left(x-\frac{M-1}{2}\Delta\right)f_X(x)dx = 0.$$

- Optimal uniform quantizer Δ for different sources:

Alphabet Size	Uniform		Gaussian		Laplacian	
	Step Size	SNR	Step Size	SNR	Step Size	SNR
2	1.732	6.02	1.596	4.40	1.414	3.00
4	0.866	12.04	0.9957	9.24	1.0873	7.05
6	0.577	15.58	0.7334	12.18	0.8707	9.56
8	0.433	18.06	0.5860	14.27	0.7309	11.39
10	0.346	20.02	0.4908	15.90	0.6334	12.81
12	0.289	21.60	0.4238	17.25	0.5613	13.98
14	0.247	22.94	0.3739	18.37	0.5055	14.98
16	0.217	24.08	0.3352	19.36	0.4609	15.84
32	0.108	30.10	0.1881	24.56	0.2799	20.46

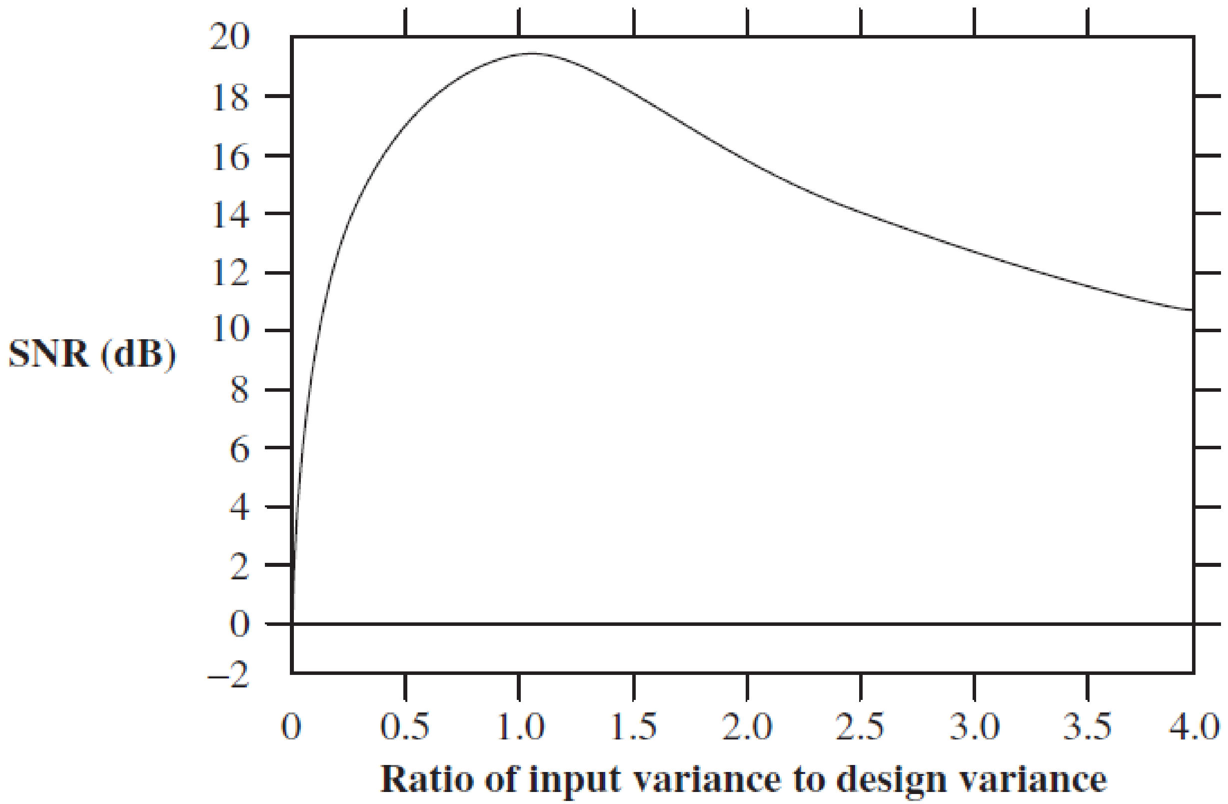
Overload/Granular Regions

- Selection of the step size must trade off between overload noise and granular noise



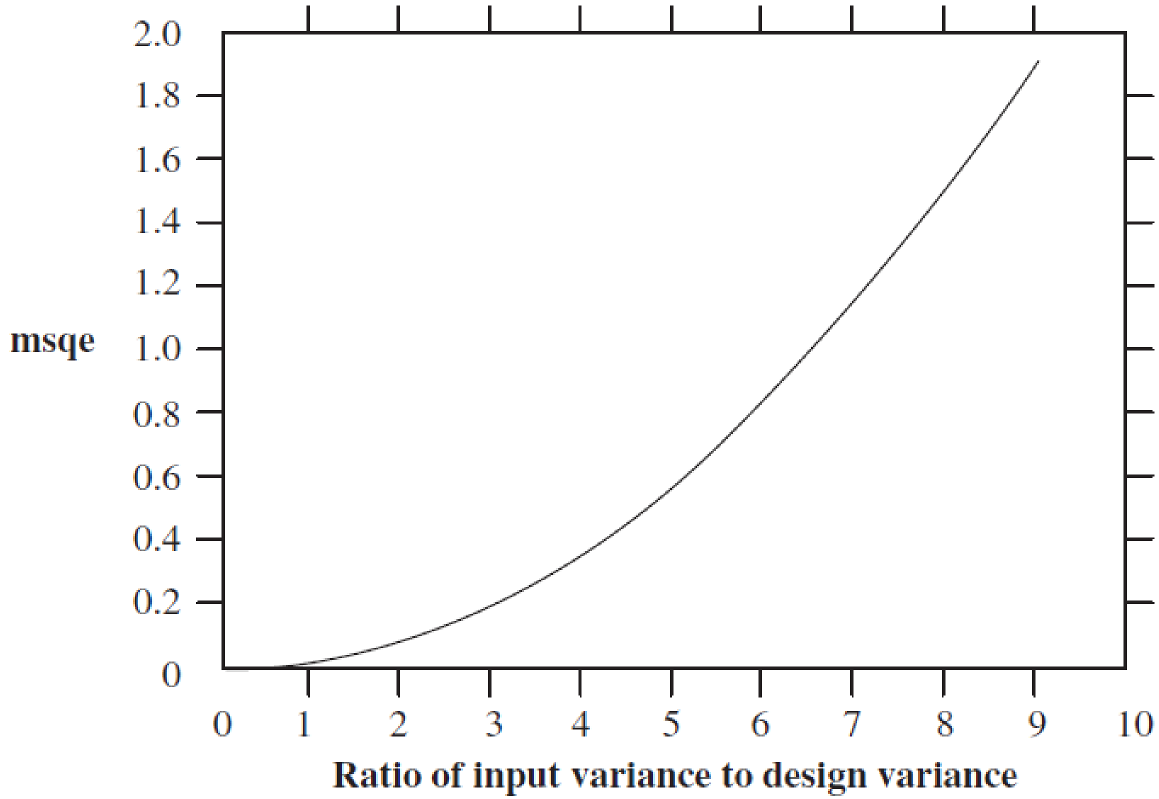
Variance Mismatch Effects (1/2)

- Effect of variance mismatch on the performance of a 4–bit uniform quantizer



Variance Mismatch Effects (2/2)

- The MSQE as a function of variance mismatch with a 4–bit uniform quantizer



Distribution Mismatch Effects

- Given 3-bit quantizer, the effect of distribution mismatch for different sources (SNR errors in dB):
  - Form left-to-right, we assume that the sources are uniform, Gaussian, Laplacian, and Gamma, and compute the optimum MSQE step size for uniform quantizer
  - The resulting Δ gets larger from left-to-right

Input Distribution	Uniform Quantizer	Gaussian Quantizer	Laplacian Quantizer	Gamma Quantizer
Uniform	18.06	15.56	13.29	12.41
Gaussian	12.40	14.27	13.37	12.73
Laplacian	8.80	10.79	11.39	11.28
Gamma	6.98	8.06	8.64	8.76

$\rightarrow$ if there is a mismatch, larger than "optimum" Δ gives better performance
$\rightarrow$ 3-bit quantizer is too coarse

Adaptive Quantization

- We can adapt the quantizer to the statistics of the input (mean, variance, pdf)
- Forward adaptive (encoder-side analysis)
  - Divide input source in blocks
  - Analyze block statistics
  - Set quantization scheme
  - Send the scheme to the decoder via side channel
- Backward adaptive (decoder-side analysis)
  - Adaptation based on quantizer output only
  - Adjust Δ accordingly (encoder-decoder in sync)
  - No side channel necessary

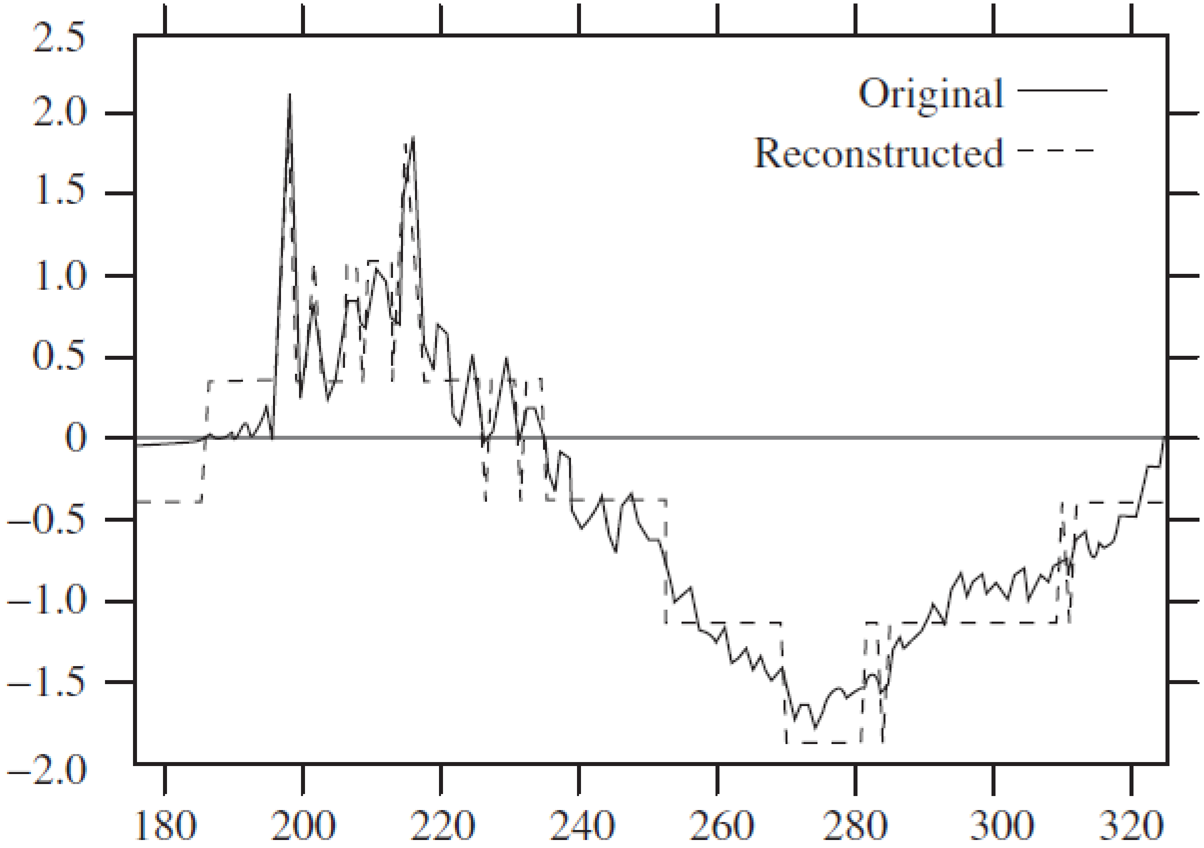
Forward Adaptive Quantization (FAQ)

- Choosing analysis block size is a major issue
- Block size too large
  - Not enough resolution
  - Increased latency
- Block size too small
  - More side channel information
- Assuming a mean of zero, signal variance is estimated by

$$\hat{\sigma}_q^2 = \frac{1}{N} \sum_{i=0}^{N-1} x_{n+i}^2 .$$

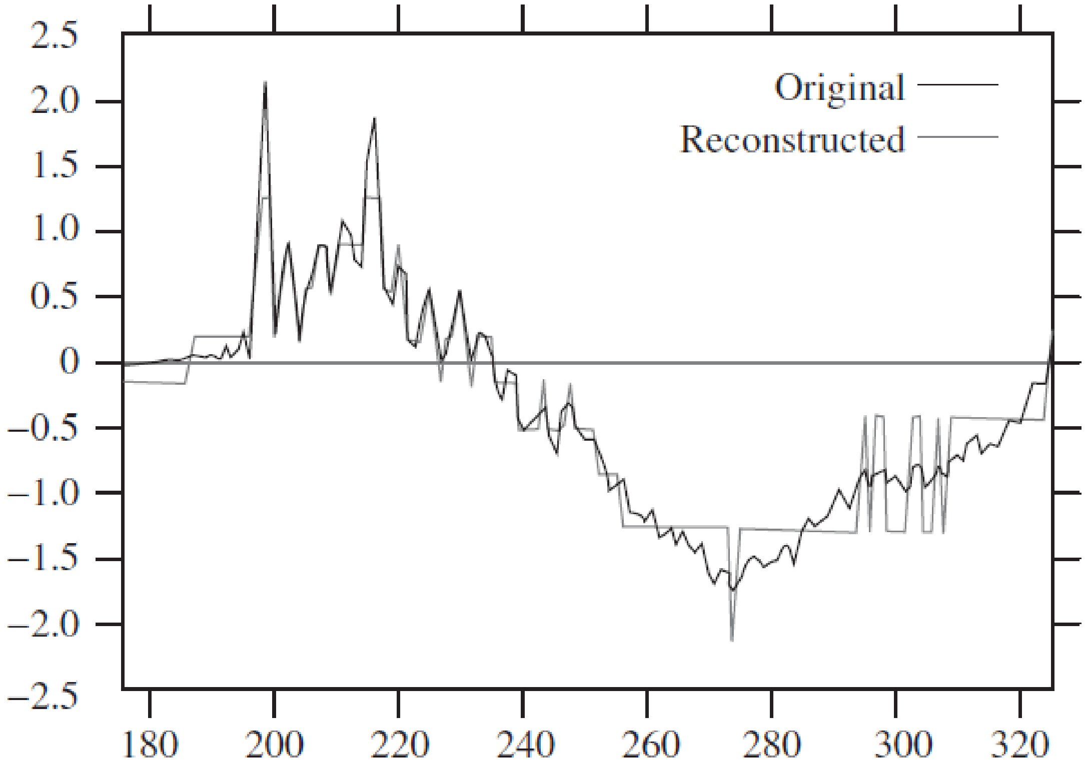
Speech Quantization Example (1/2)

- 16-bit speech samples → 3-bit fixed quantization



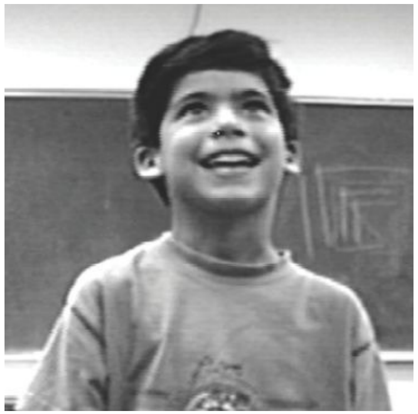
Speech Quantization Example (2/2)

- 16-bit speech samples → 3-bit FAQ
  - Block = 128 samples
  - 8-bit variance quantization

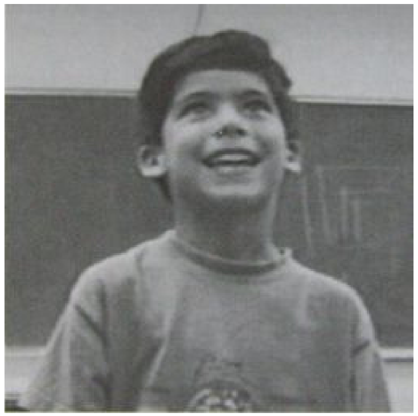


FAQ Refinement

- So far, we assumed uniform pdf over maximal ranges, we can refine it by computing the range of distribution adaptively for each block
- Example: Sena image, 8×8 blocks, 2×8-bit for range per block, 3-bit quantizer



Original 8 bits/pixel



Quantized 3.25 bits/pixel

Backward Adaptive Quantization (BAQ)

- Key idea: only encoder sees input source, if we do not want to use side channel to tell the decoder how to adapt the quantizer, we can only use quantized output to adapt the quantizer
- Possible solution:
  - Observe the number of output values that falls in outer levels and inner levels
  - If they match the assumed pdf, Δ is good
  - If too many values fall in outer levels, Δ should be enlarged, otherwise, Δ should be reduced
- Issue: estimation of pdf requires large observations?

Jayant Quantizer

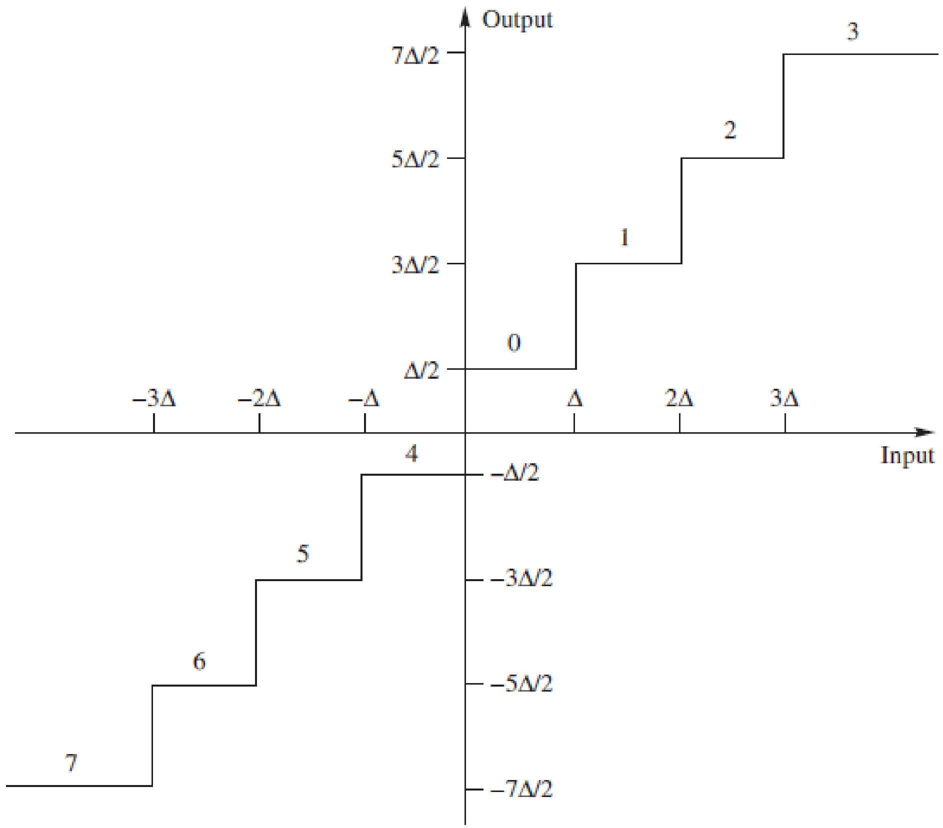
- N. S. Jayant showed in 1973 that Δ adjustment based on few observations still works fine:
  - If current input falls in the outer levels, expand step size
  - If current input falls in the inner levels, contract step size
  - The total product of expansions and contraction should be 1
- Each decision interval k has a multiplier M_k
  - If input s_{n-1} falls in the k^{th} interval, step size is multiplied by M_k
  - Inner-level $M_k < 1$, outer-level $M_k > 1$
  - Step size adaptation rule:

$$\Delta_n = M_{l(n-1)}\Delta_{n-1},$$

where $l(n-1)$ is the quantization interval at time $n-1$.

Output Levels of 3-bit Jayant Quantizer

- The multipliers are symmetric:
  - $M_0 = M_4, M_1 = M_5, M_2 = M_6, M_3 = M_7$



Example: Jayant Quantizer

- $M_0 = M_4 = 0.8$, $M_1 = M_5 = 0.9$
- $M_2 = M_6 = 1.0$, $M_3 = M_7 = 1.2$, $\Delta_0 = 0.5$
- Input: 0.1, –0.2, 0.2, 0.1, –0.3, 0.1, 0.2, 0.5, 0.9, 1.5

n	Δ_n	Input	Output Level	Output	Error	Update Equation
0	0.5	0.1	0	0.25	0.15	$\Delta_1 = M_0 \times \Delta_0$
1	0.4	−0.2	4	−0.2	0.0	$\Delta_2 = M_4 \times \Delta_1$
2	0.32	0.2	0	0.16	0.04	$\Delta_3 = M_0 \times \Delta_2$
3	0.256	0.1	0	0.128	0.028	$\Delta_4 = M_0 \times \Delta_3$
4	0.2048	−0.3	5	−0.3072	−0.0072	$\Delta_5 = M_5 \times \Delta_4$
5	0.1843	0.1	0	0.0922	−0.0078	$\Delta_6 = M_0 \times \Delta_5$
6	0.1475	0.2	1	0.2212	0.0212	$\Delta_7 = M_1 \times \Delta_6$
7	0.1328	0.5	3	0.4646	−0.0354	$\Delta_8 = M_3 \times \Delta_7$
8	0.1594	0.9	3	0.5578	−0.3422	$\Delta_9 = M_3 \times \Delta_8$
9	0.1913	1.5	3	0.6696	−0.8304	$\Delta_{10} = M_3 \times \Delta_9$
10	0.2296	1.0	3	0.8036	0.1964	$\Delta_{11} = M_3 \times \Delta_{10}$
11	0.2755	0.9	3	0.9643	0.0643	$\Delta_{12} = M_3 \times \Delta_{11}$

Picking Jayant Multipliers

- We must select $\Delta_{\min}$ and $\Delta_{\max}$ to prevent underflow and overflow of step sizes
- Selection of multipliers
  - Total production of expansion/contractions should be 1

$$\prod_{k=0}^{M} M_k^{n_k} = 1.$$

  - Scaled to probability of events in each interval, we have

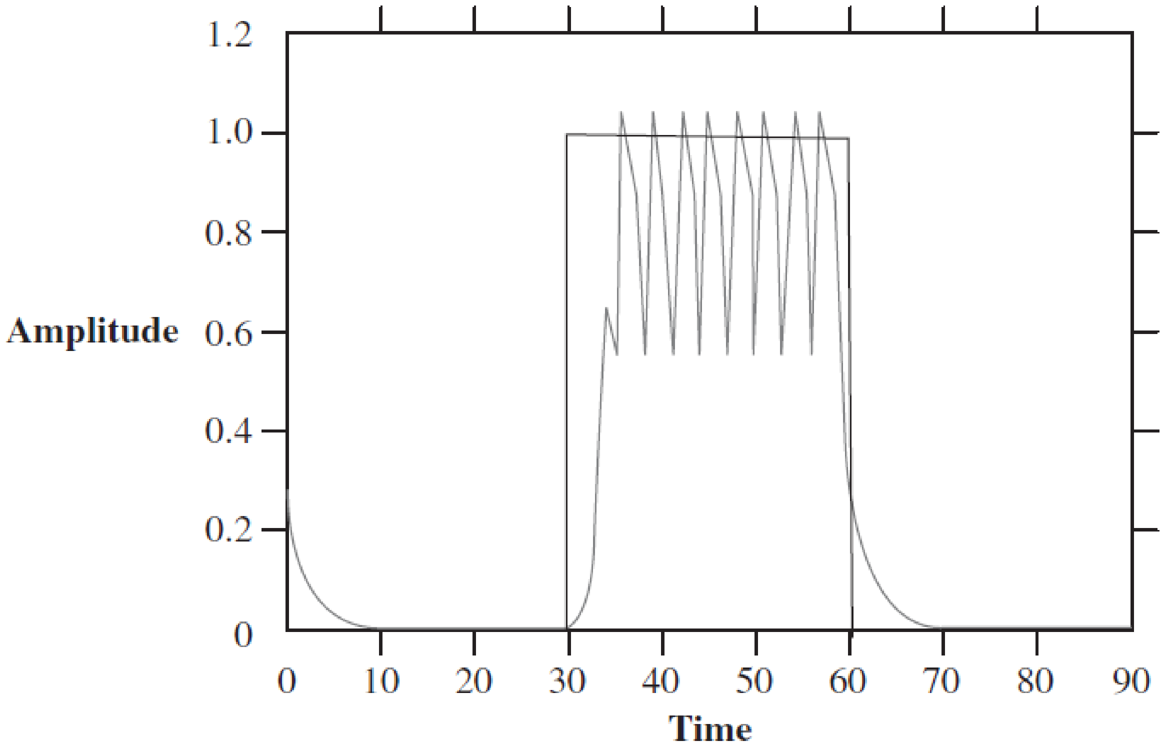
$$\prod_{k=0}^{M} M_k^{P_k} = 1, \text{ where } P_k = \frac{n_k}{N}, \ N = \sum_{k=0}^{M} n_k.$$

  - Pick $\gamma > 1$, and let $M_k = \gamma^{l_k}$, we have

$$\sum_{k=0}^{M} l_k P_k = 0, \rightarrow \gamma \text{ and } l_k \text{ are chosen, } P_k \text{ is known.}$$

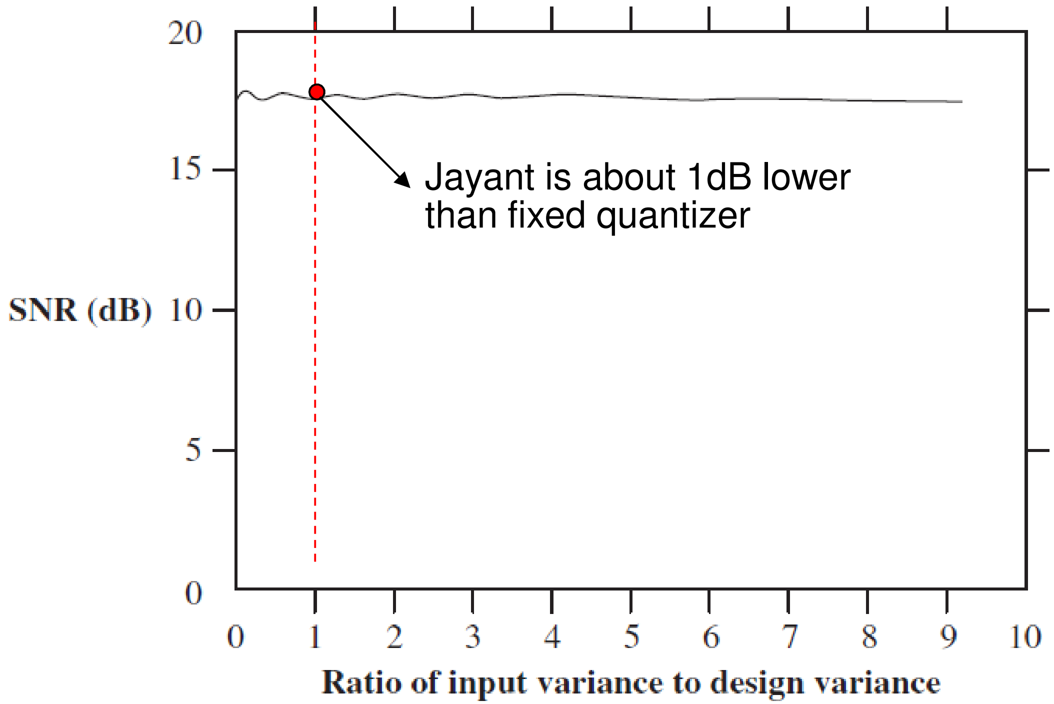
Example: Ringing Problem

- Use a 2-bit Jayant quantizer to quantize a square wave
  - $P_0 = 0.8, P_1 = 0.2 \rightarrow$ pick $l_0 = -1, l_1 = 4, \gamma \sim 1$.



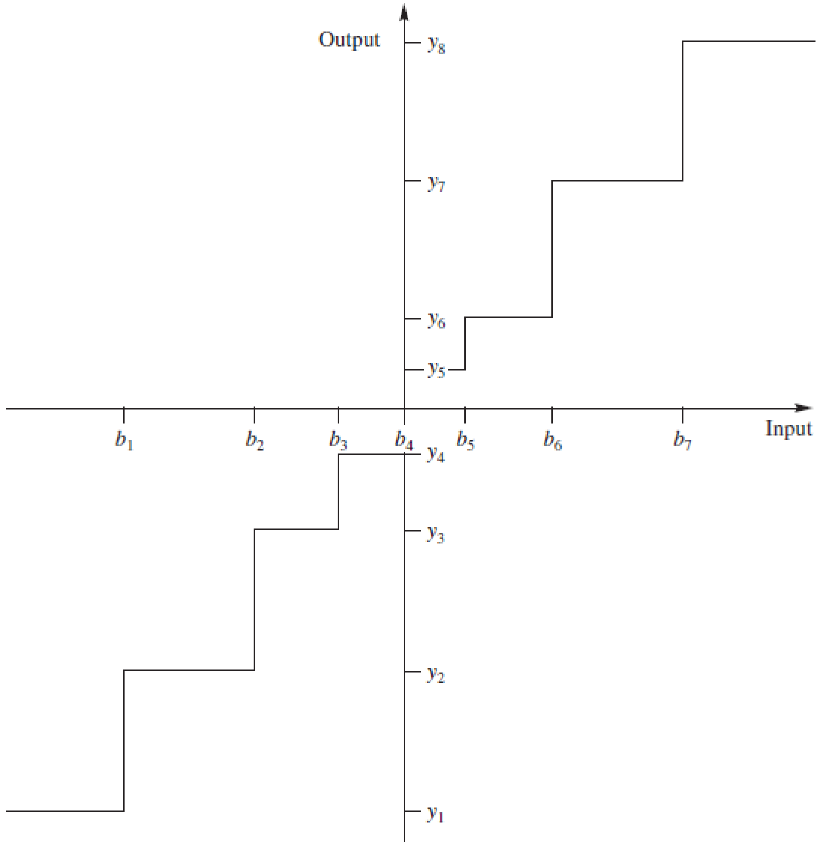
Jayant Quantizer Performance

- To avoid overload errors, we should expand Δ rapidly and contracts Δ moderately
- Robustness over changing input statistics



Non-uniform Quantization

- ❑ For uniform quantizer, decision boundaries are determined by a single parameter Δ.
- ❑ We can certainly reduce quantization errors further if each decision boundaries can be selected freely



pdf-optimized Quantization

- Given $f_X(x)$, we can try to minimize MSQE:

$$\sigma_q^2 = \sum_{i=1}^{M} \int_{b_{i-1}}^{b_i} (x - y_i)^2 f_X(x)dx.$$

- Set derivative of σ_q^2 w.r.t. y_j to zero and solve for y_j, we have:

$$y_j = \frac{\int_{b_{j-1}}^{b_j} x f_X(x)dx}{\int_{b_{j-1}}^{b_j} f_X(x)dx}.$$

y_j is the center of mass of f_X in $[b_{j-1}, b_j)$

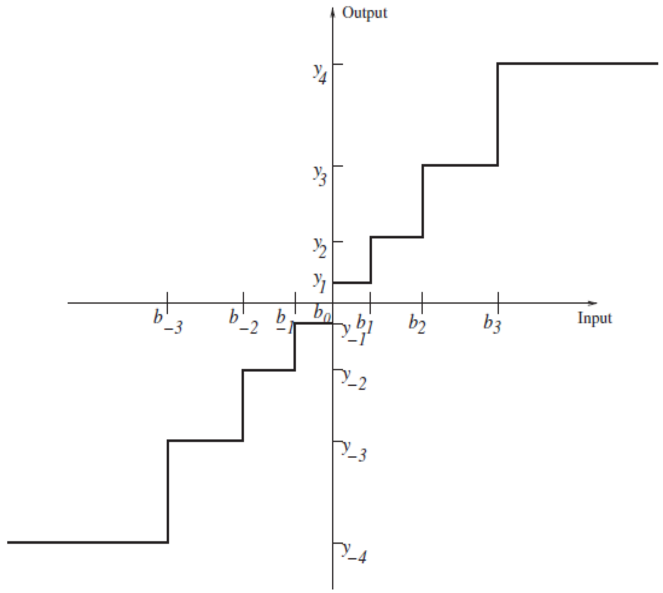
If y_j are determined, the b_j's can be selected as:

$$b_j = (y_{j+1} + y_j)/2.$$

Lloyd-Max Algorithm (1/3)

- Lloyd-Max algorithm solves y_j and b_j iteratively until an acceptable solution is found

- Example: For midrise quantizer, $b_0 = 0$, $b_{M/2}$ is the largest input, we only have to find $\{ b_1, b_2, \ldots, b_{M/2-1}\}$ and $\{ y_1, y_2, \ldots, y_{M/2-1}\}$.



Lloyd-Max Algorithm (2/3)

- Begin with $j = 1$, we want to find b_1 and y_1 by

$$y_1 = \int_{b_0}^{b_1} x f_X(x)dx \Big/ \int_{b_0}^{b_1} f_X(x)dx.$$

- Pick a value for y_1 (e.g. $y_1 = 1$), solve for b_1 and compute y_2 by

$$y_2 = 2b_1 + y_1,$$

and b_2 by

$$y_2 = \int_{b_1}^{b_2} x f_X(x)dx \Big/ \int_{b_1}^{b_2} f_X(x)dx.$$

- Continue the process until all $\{b_j\}$ and $\{y_j\}$ are found

Lloyd-Max Algorithm (3/3)

- If the initial guess of y_1 does not fulfills the termination condition:

$$\left| y_{M/2} - \hat{y}_{M/2} \right| \le \varepsilon,$$

where

$$\hat{y}_{M/2} = 2b_{M/2-1} + y_{M/2-1},$$

$$y_{M/2} = \int_{b_{M/2-1}}^{b_{M/2}} x f_X(x)dx \Big/ \int_{b_{M/2-1}}^{b_{M/2}} f_X(x)dx.$$

we must pick a different y_1 and repeat the process.

Example: *pdf*-Optimized Quantizers

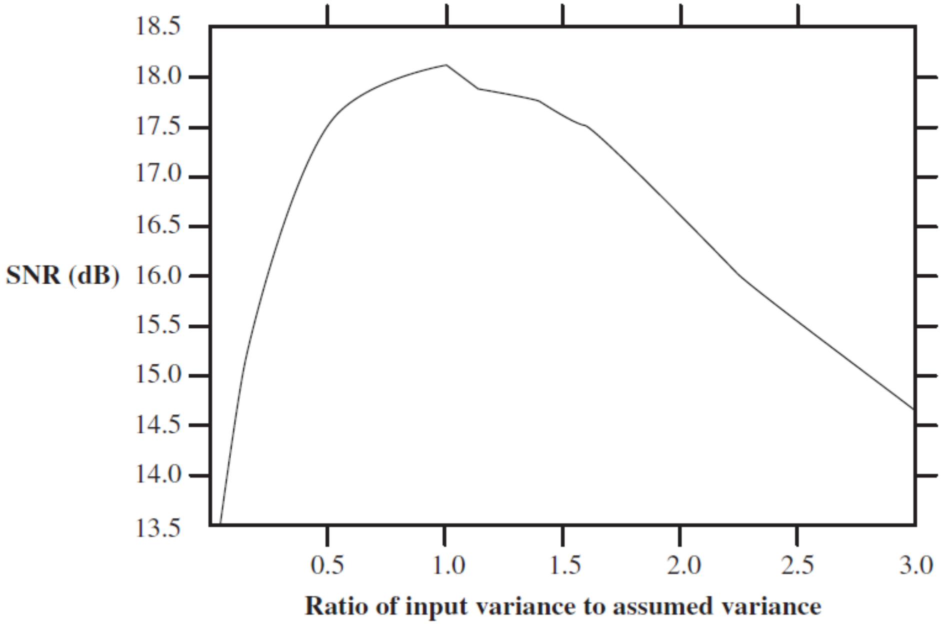
- We can achieve gain over the uniform quantizer

Levels	Gaussian b_i	Gaussian y_i	Gaussian SNR	Laplacian b_i	Laplacian y_i	Laplacian SNR
4	0.0	0.4528		0.0	0.4196	
	0.9816	1.510	9.3 dB (9.24)	1.1269	1.8340	7.54 dB (7.05)
6	0.0	0.3177		0.0	0.2998	
	0.6589	1.0		0.7195	1.1393	
	1.447	1.894	12.41 dB (12.18)	1.8464	2.5535	10.51 dB (9.56)
8	0.0	0.2451		0.0	0.2334	
	0.7560	0.6812		0.5332	0.8330	
	1.050	1.3440		1.2527	1.6725	
	1.748	2.1520	14.62 dB (14.27)	2.3796	3.0867	12.64 dB (11.39)

Mismatch Effects

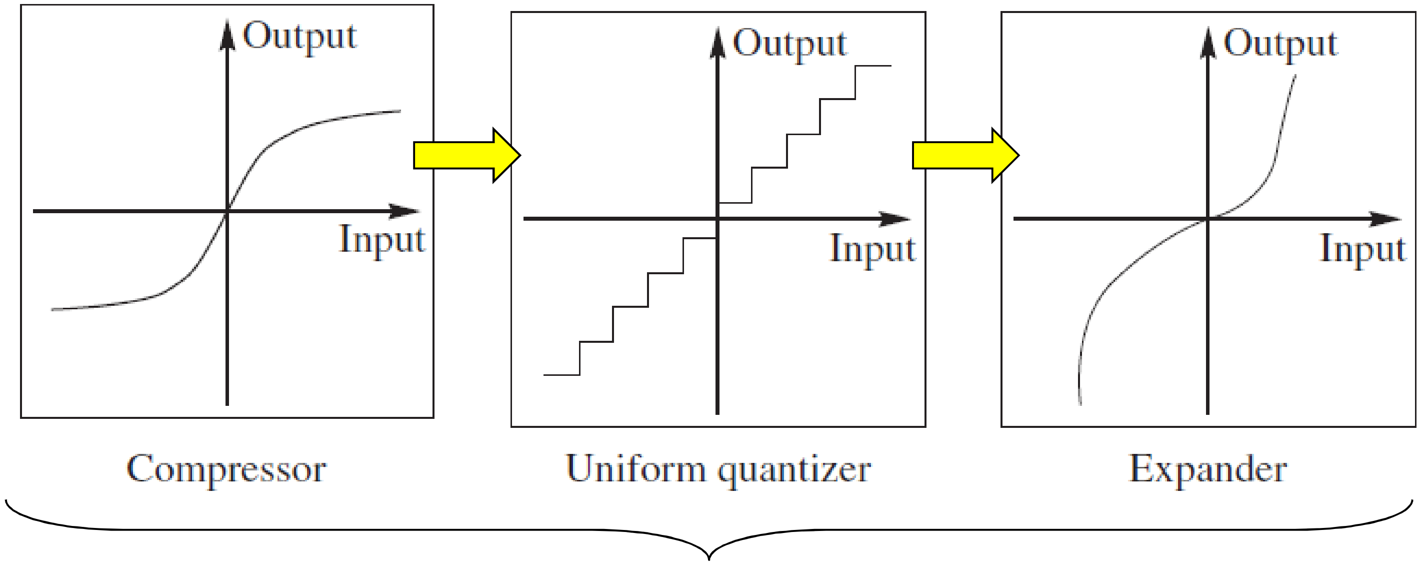
- Non-uniform quantizers also suffer mismatch effects.
- To reduce the effect, one can use an adaptive non-uniform quantizer, or an adaptive uniform quantizer plus companded quantization techniques

Variance mismatch on a 4-bit Laplacian non-uniform quantizer.



Companded Quantization (CQ)

- In companded quantization, we adjust (i.e. re-scale) the intervals so that the size of each interval is in proportion to the probability of inputs in each interval

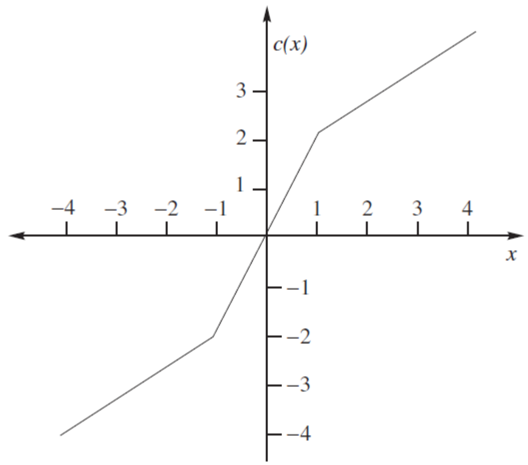


equivalent to a non-uniform quantizer

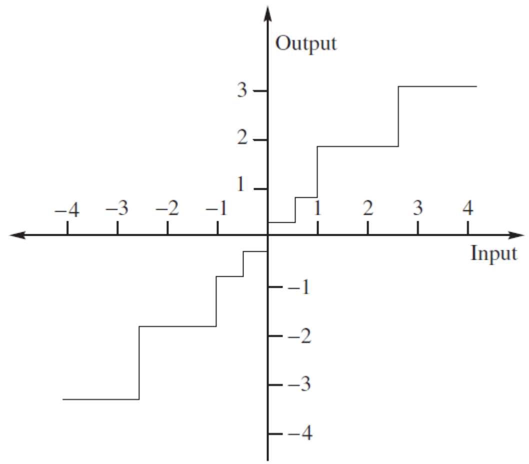
Example: CQ (1/2)

- The compressor function:

$$c(x) = \begin{cases} 2x & \text{if } -1 \le x \le 1 \\ \frac{2x}{3} + \frac{4}{3} & x > 1 \\ \frac{2x}{3} - \frac{4}{3} & x < -1 \end{cases} .$$



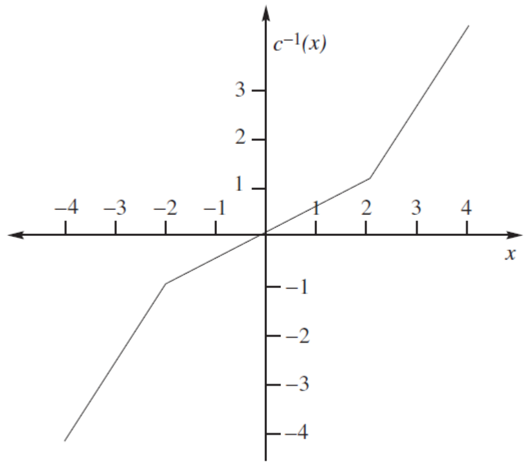
- The uniform quantizer: step size $\Delta = 1.0$



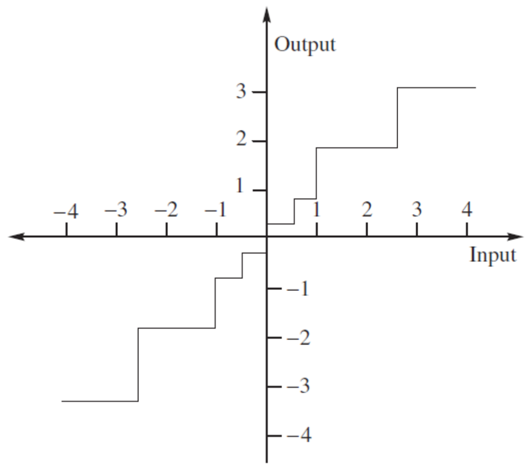
Example: CQ (2/2)

- The expander function:

$$c^{-1}(x) = \begin{cases} \frac{x}{2} & \text{if } -2 \le x \le 2 \\ \frac{3x}{2} - 2 & x > 2 \\ \frac{3x}{2} + 2 & x < -2 \end{cases}.$$



- The equivalent non-uniform quantizer



Remarks on CQ

- If the level of quantizer is large and the input is bounded by x_{max}, it is possible to choose a $c(x)$ such that the SNR of CQ is independent to the input pdf:

$$\text{SNR} = 10\log_{10}(3M^2) - 20\log_{10}\alpha,$$

  where $c'(x) = x_{max} / (\alpha|x|)$ and a is a constant.

- Two popular CQ for telephones: μ-law and A-law
  - μ-law compressor

$$c(x) = x_{\max}\frac{\ln(1+\mu\frac{|x|}{x_{\max}})}{\ln(1+\mu)}\operatorname{sgn}(x).$$

  - A-law compressor

$$c(x) = \begin{cases} \frac{A|x|}{1+\ln A}\operatorname{sgn}(x), & 0 \le \frac{|x|}{x_{\max}} \le \frac{1}{A} \\ x_{\max}\cdot\frac{1+\ln\frac{A|x|}{x_{\max}}}{1+\ln A}\operatorname{sgn}(x), & \frac{1}{A} \le \frac{|x|}{x_{\max}} \le 1 \end{cases}.$$

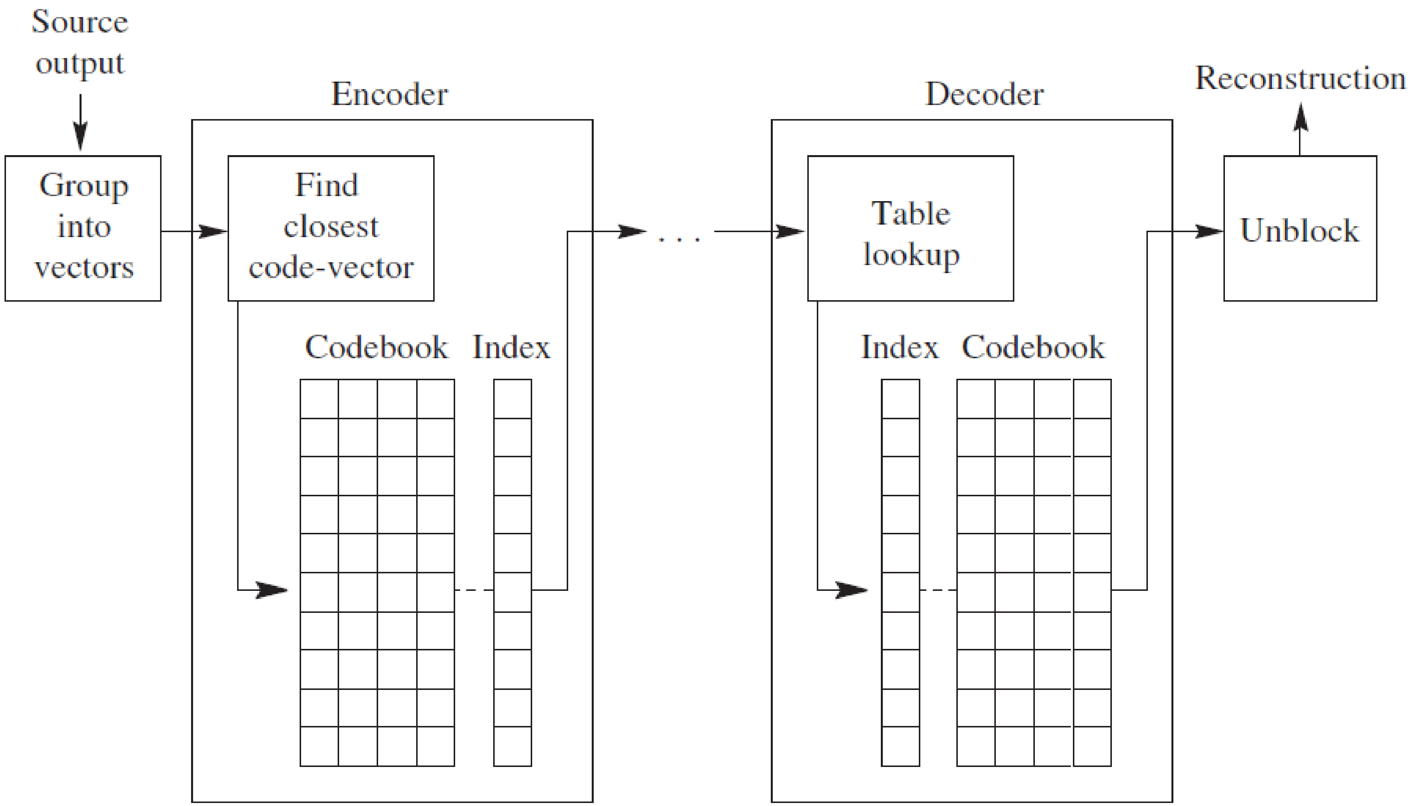
Entropy Coding of Quantizer Outputs

- The levels of the quantizer is the alphabet of entropy coders, for M-level quantizer, FLC needs $\log_2 M$ bits per output
- Example of VLC coded output of minimum MSQE quantizers:
  - Note: non-uniform quantizer has higher entropies since high probability regions uses smaller step sizes

Number of Levels	Gaussian		Laplacian	
	Uniform	Nonuniform	Uniform	Nonuniform
4	1.904	1.911	1.751	1.728
6	2.409	2.442	2.127	2.207
8	2.759	2.824	2.394	2.479
16	3.602	3.765	3.063	3.473
32	4.449	4.730	3.779	4.427

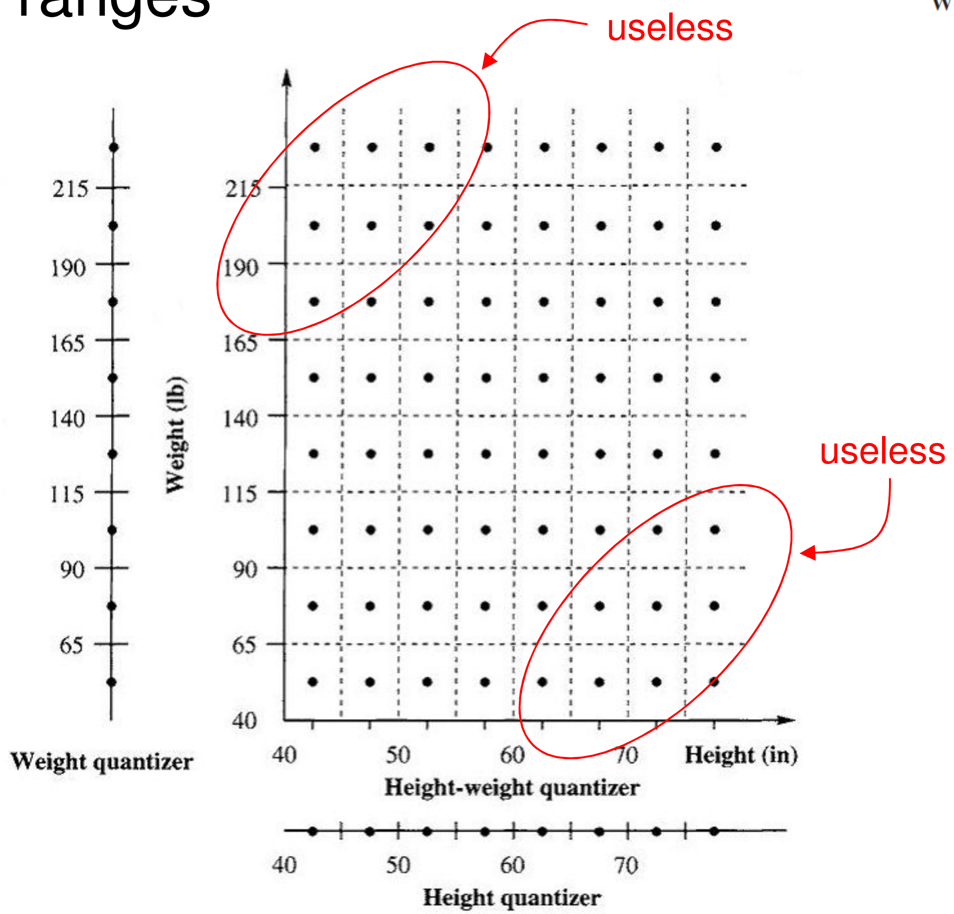
Vector Quantization

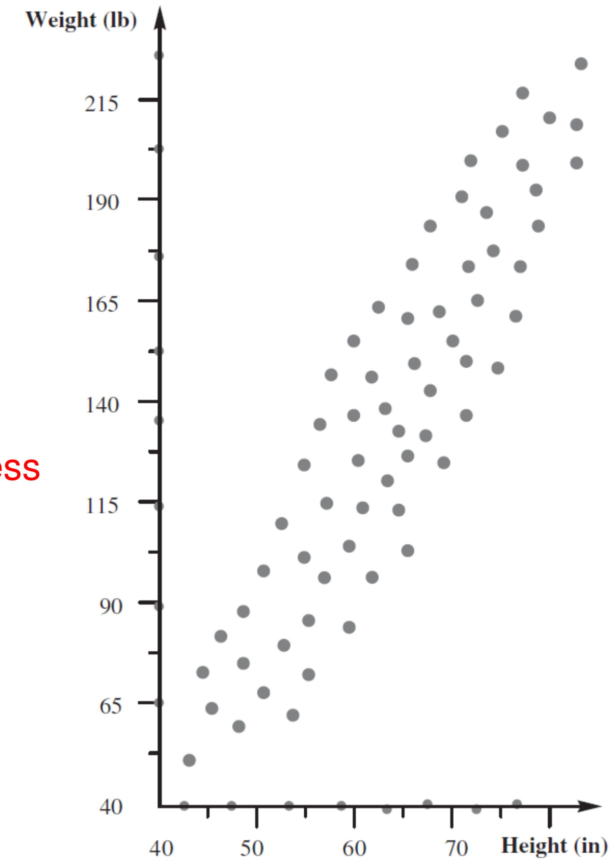
- Vector quantization groups source data into vectors
  - A vector quantizer maintains a set of vectors called the codebook. Each vector in the codebook is assigned an index.



Why Vector Quantization (1/2)?

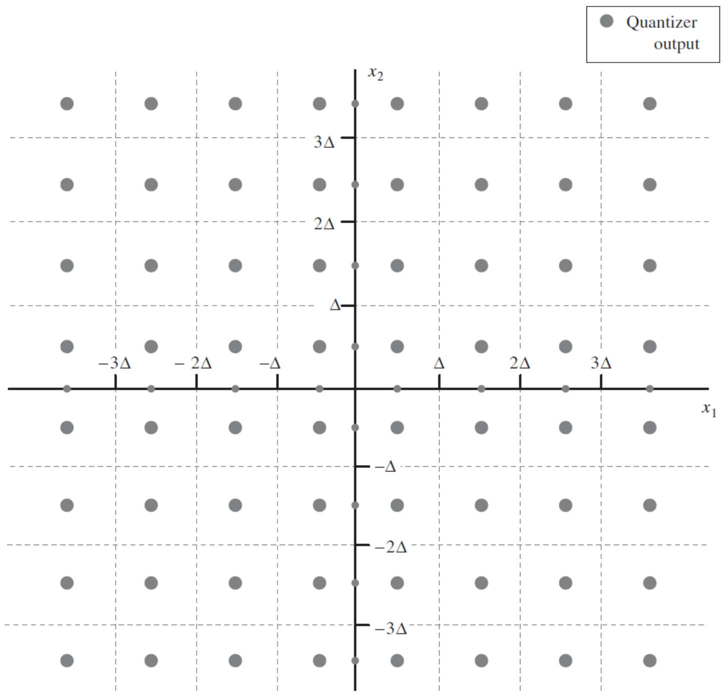
- ❑ Correlated multi-dimensional data have limited valid ranges



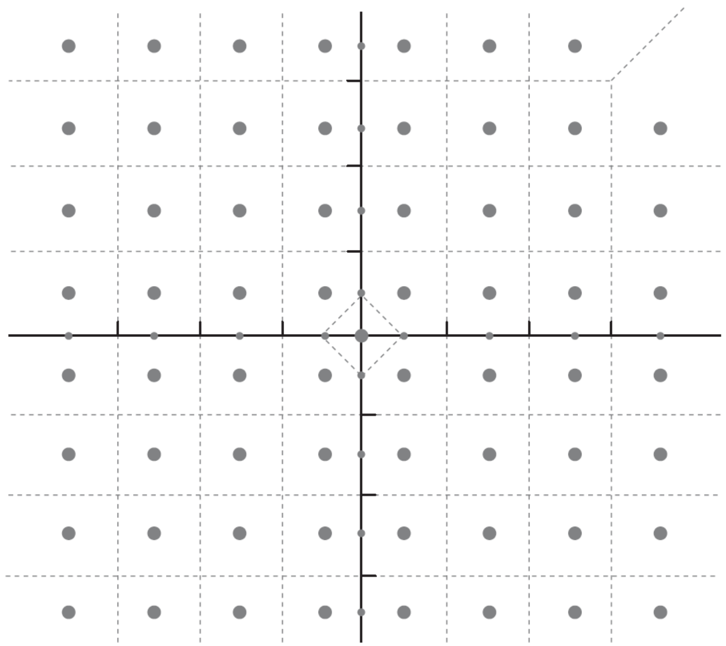


Why Vector Quantization (2/2)?

- Looking at the data from a higher dimension allow us to better fit the quantizer structure to the joint pdf
  - Example: quantize the Laplacian source data two at a time:



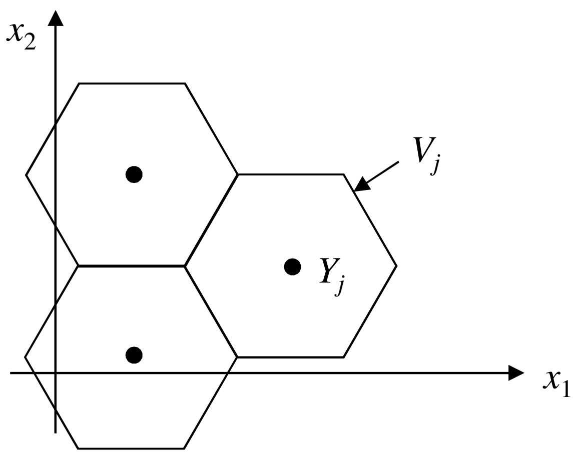
11.44 dB



11.73 dB

Vector Quantization Rule

- Vector quantization (VQ) of X may be viewed as the classification of X into a discrete number of sets
  - Each set is represented by a vector output Y_j
- Given a distance measure $d(x, y)$, we have
  - VQ output: $Q(X) = Y_j$ iff $d(X, Y_j) < d(X, Y_i), \forall\, i \neq j$.
  - Quantization region: $V_j = \{ X: d(X, Y_j) < d(X, Y_i), \forall\, i \neq j\}$.



Codebook Design

- The set of quantizer output points in VQ is called the codebook of the quantizer, and the process of placing these output points is often referred to as the codebook design

- The k-means algorithm† is often used to classify the outputs
  - Given a large set of output vectors from the source, known as the training set, and an initial set of k representative patterns
  - Assign each element of the training set to the closest representative pattern
  - After an element is assigned, the representative pattern is updated by computing the centroid of the training set vectors assigned to it
  - When the assignment process is complete, we will have k groups of vectors clustered around each of the output points

† The idea is the same as the scalar quantization problem in Stuart P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. on Information Theory*, Vol. 28, No. 2, March 1982.

The Linde-Buzo-Gray Algorithm

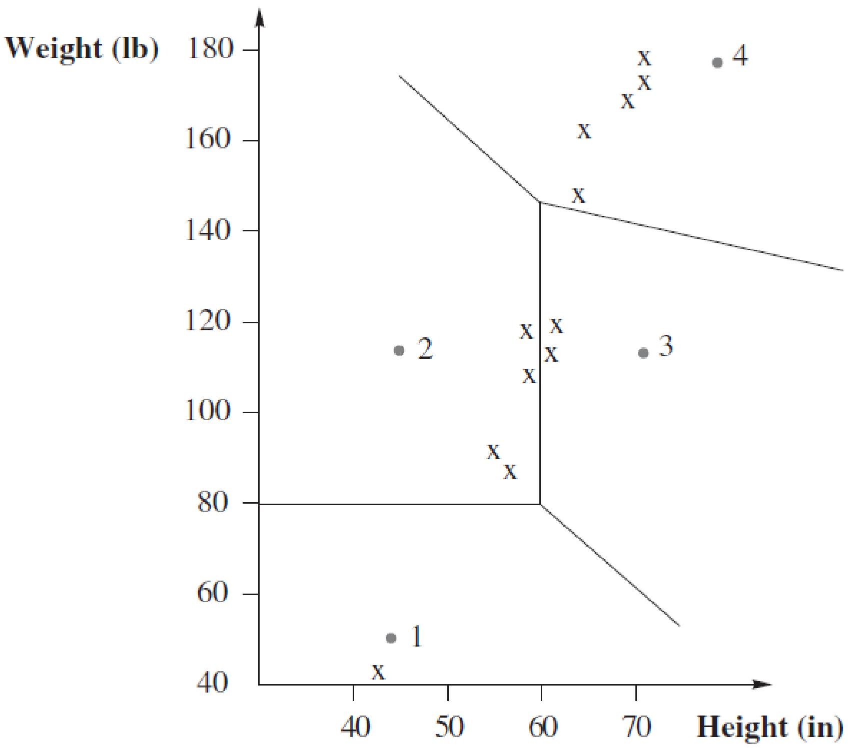
1. Start with an initial set of reconstruction values $\{Y_i^{(0)}\}_{i=1..M}$ and a set of training vectors $\{X_n\}_{n=1..N}$. Set $k = 0$, $D^{(0)} = 0$. Select threshold ε.
2. The quantization regions $\{V_i^{(k)}\}_{i=1..M}$ are given by

$$V_j^{(k)} = \{X_n: d(X_n, Y_i) < d(X_n, Y_j), \forall\, j \neq i\},\ i = 1, 2, \ldots, M.$$

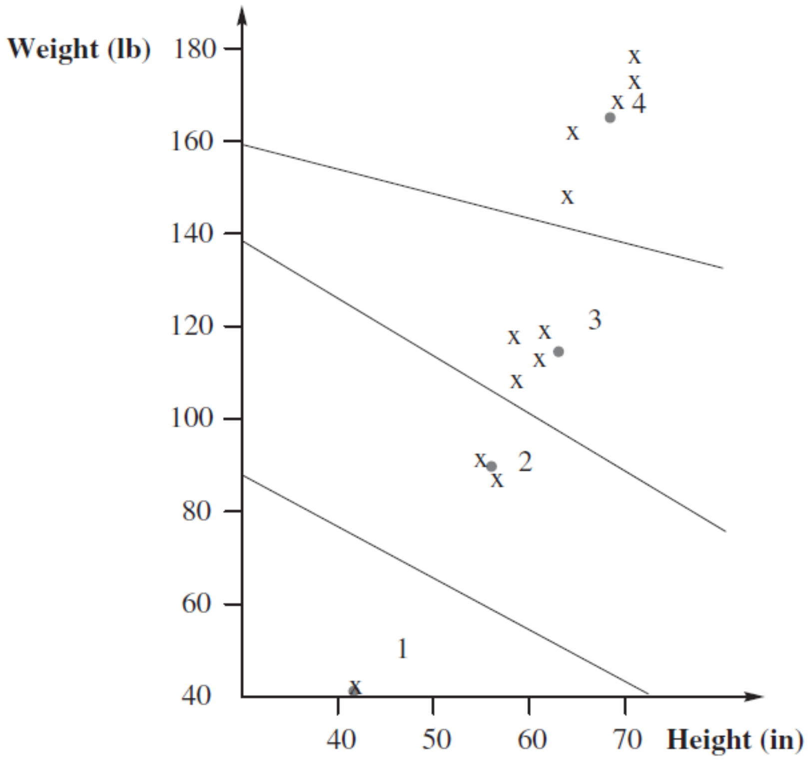
3. Compute the average distortion $D^{(k)}$ between the training vectors and the representative value
4. If $(D^{(k)} - D^{(k-1)})/D^{(k)} < \varepsilon$, stop; otherwise, continue
5. Let $k = k + 1$. Update $\{Y_i^{(k)}\}_{i=1..M}$ with the average value of each quantization region $V_i^{(k-1)}$. Go to step 2.

Example: Codebook Design

- Initial state

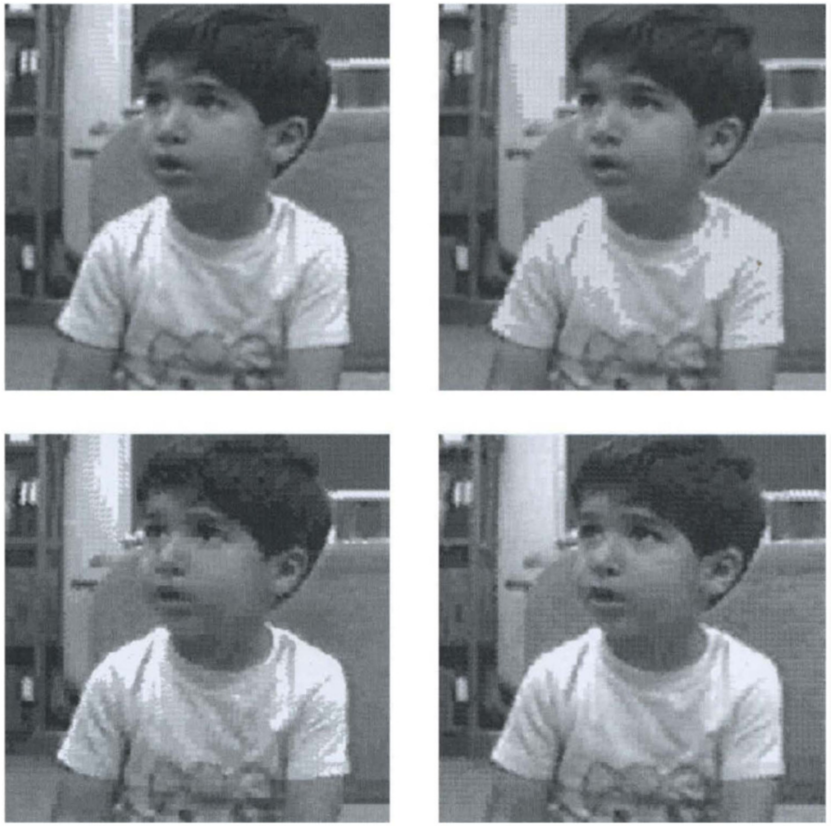


- Final state



Impact of Training Set

- The training sets used to construct the codebook have significant impact on the performance of VQ



Images quantized at 0.5 bits/pixel, codebook size 256