Multimedia Video Coding & Architectures (5LSE0), Module 03

Scalar Quantization

Peter H.N. de With
(p.h.n.de.with@tue.nl)

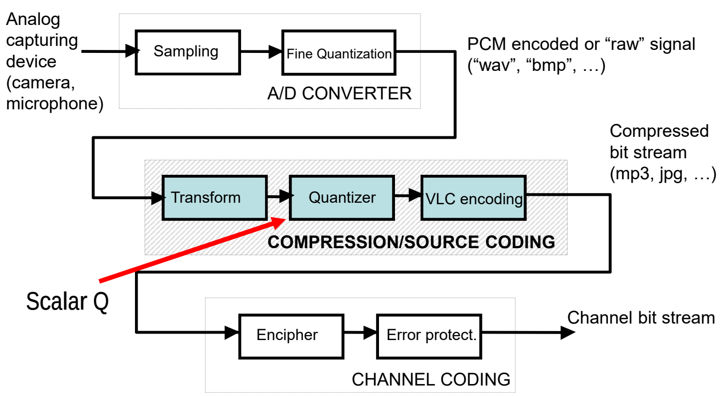
slides version 1.0

Multimedia Video Coding and Architectures (5LSE0), Module 03

Mod 03, Preliminaries

Brief introduction: where are we…?

System Overview



Why can Signals be Compressed? – (1)

Because ~~signal amplitudes~~ are statistically redundant

(When Quantizing:) quantizer representation levels

Question 1:
What is the *shortest* average codeword length that one can achieve for a given signal (or "source")?

$$H(S) = -\sum_{i=1}^{N} P_S(s_i)\log_2[P_S(s_i)] \quad \text{(bit)}$$

$$H(X) = -\int_{-\infty}^{\infty} p_X(x)\log_2[p_X(x)]dx$$

Question 2:
How did you obtain those codewords?

Huffman Coding
Run-length coding

Why can Signals be Compressed? – (2)

Because signal amplitudes are mutually dependent

Question 1:

What is the *best possible* exploitation of the correlation (dependencies) in natural signals?

(Rate-Distortion Theory)

Question 2:

How do we *implement* a system that exploits the correlation in natural signals?

(Compression algorithms: - DPCM
- Subband/wavelet
- Transform/DCT
- Motion compensation)

To be discussed later in course

Why can Signals be Compressed? – (3)

Because infinite accuracy of signal amplitudes is (perceptually) irrelevant

Question 1:

What is the *best possible* trade-off between required bit rate and resulting distortion?

(Rate-Distortion Theory)

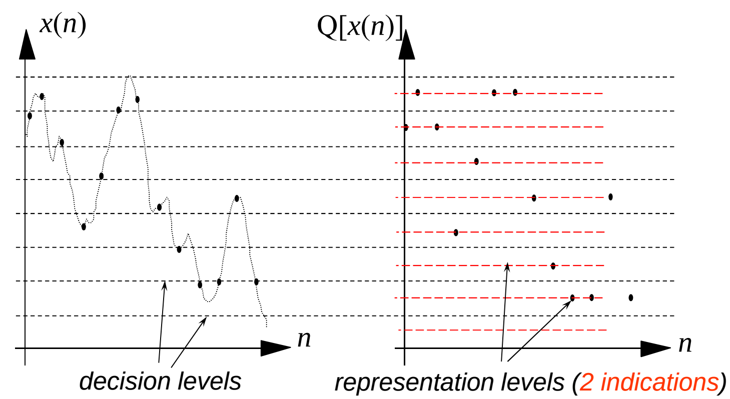
Question 2:

How do we *implement* a system that gives us that best possible trade-off?

(Scalar and Vector Quantization Theory)

5LSE0 - Mod 03
Part 1
Characterization of Quantizers

Quantization Process – (1)



Quantization Process – (2)

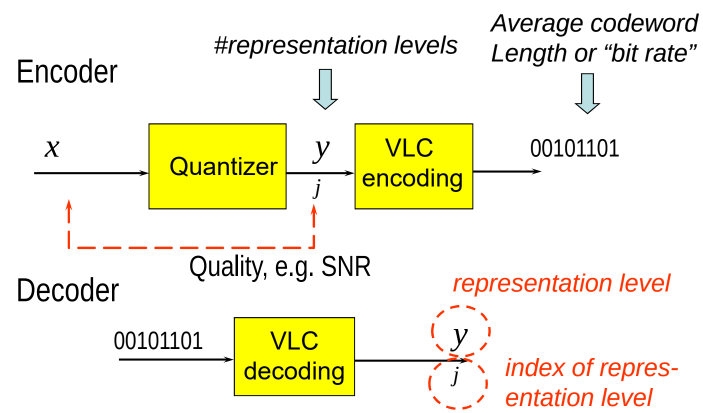
- **Quantization**: Map a continuous-valued signal value $x(n)$ onto a (limited set of) discrete-valued signals values $y(n)$:

$$y(n) = Q[x(n)]$$

such that $y(n)$ is good approximation of $x(n)$.

- **Important:** # bits to represent $y(n)$.
  (= average codeword length per representation level)
- **Design/Optimization** Problem
  - What positions of the "*decision levels*"
  - What positions of the "*representation levels*"

Quantization Process – (3)

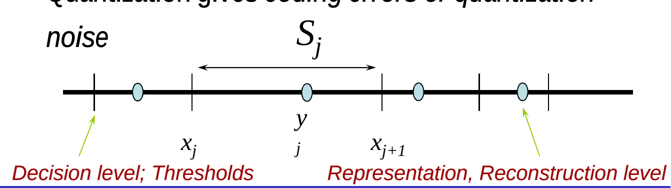


Formal Quantization Definition

- Given amplitude x, divide domain of x into K non-overlapping intervals S_k:

$$S_k = \{x \mid x_k < x \le x_{k+1}\} \qquad k = 1,2,\ldots,K$$

- If x falls in S_j then it is represented by y_j.
- Quantization gives *coding errors or quantization noise*



Quantizer Characteristics – (1)

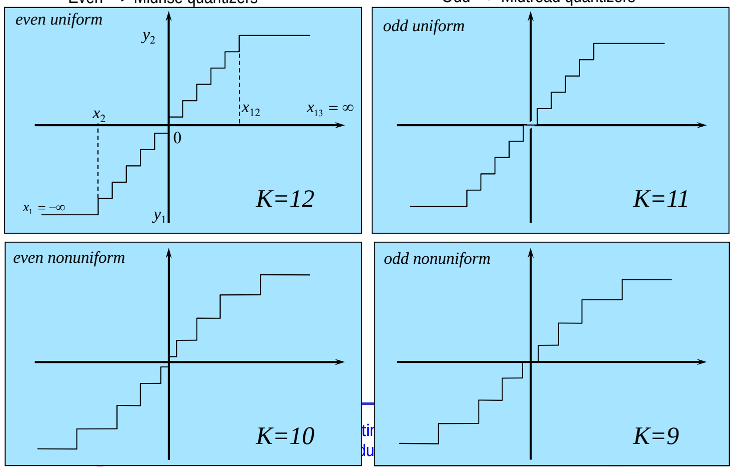
- Signals values are assumed to lie symmetrically around zero

⇨ Quantizers are usually *symmetric about origin*

- Quantizer choices:
  - Odd or even number of levels? } *"structure"*
  - Uniform or non-uniform quantizers?
  - Way of optimizing the quantizer? } *"design"*

Quantizer Characteristics – (2)

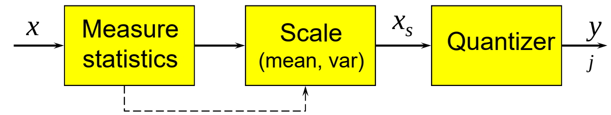
Even = > Midrise quantizers   Odd = > Midtread quantizers



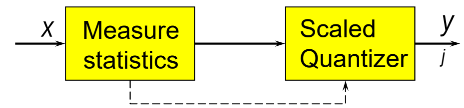
Preprocessing for Quantization – (1)

* Only *prototype* quantizers are designed
  – Signal is assumed to be zero-mean, unity variance

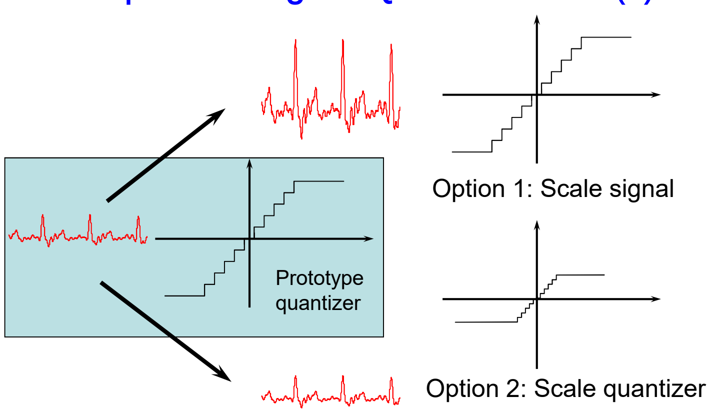
Scale signal values to be quantized



Scale quantizer itself



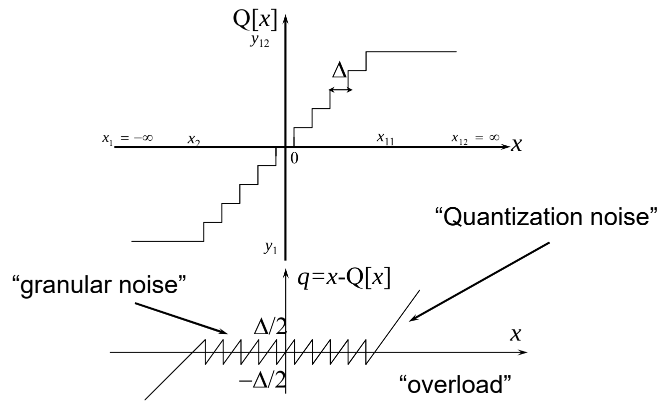
Preprocessing for Quantization – (2)



Quantizer Characteristics – (3)

* Design of quantizer is optimization problem
  – Find optimal values for decision and representation levels
* Two approaches:
  – Exploit knowledge of probability density function (PDF) of *x* to find optimum given #representation levels
    • Uniform quantization: easily solved and implemented
    • Non-uniform quantization: more difficult with limited gain
  – Select uniform quantizer with variable coarseness; determine effective #representation levels during applications
    • More practical; this is what JPEG/MPEG do
    • Choices inspired by theory

Quantization Noise / types – (1)



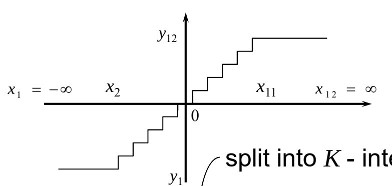
Quantization Noise / Variance – (2)

* Quantification of (average) quantization error:
  Variance of the quantization noise: $q = x - Q[x]$
* Need to model probability density of x: $p_X(x)$
* Quantization noise variance:

$$\sigma_q^2 = \int_{-\infty}^{\infty} \underbrace{\left(x - Q[x]\right)^2}_{\text{Amount of error}} \underbrace{p_X(x)\,\mathrm{d}x}_{\text{Probability of amount}}$$

Amount of error   Probability of amount

Quantization Noise / Variance – (3)



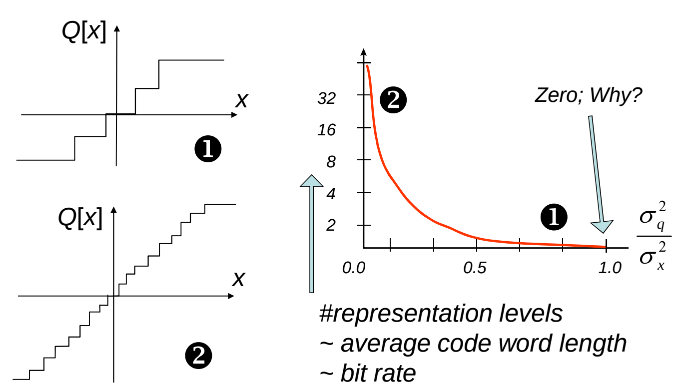
split into K - intervals

$$\sigma_q^2 = \int_{-\infty}^{\infty} \left(x - Q[x]\right)^2 p_X(x)\,\mathrm{d}x = \sum_{k=1}^{K} \int_{x_k}^{x_{k+1}} \left(x - y_k\right)^2 p_X(x)\,\mathrm{d}x$$

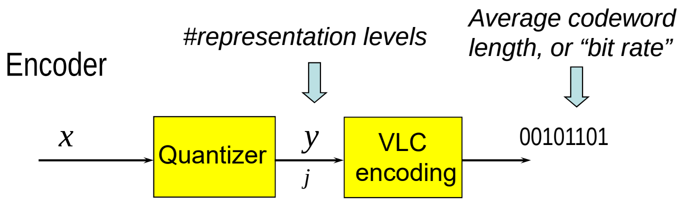
Normalized measure: Signal-to-Noise-Ratio (SNR):

$$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_q^2}\ (\mathrm{dB})$$

Bit Rate versus Distortion – (1)



Terminology for using quantizers



- Representation levels coded in $\lceil \log_2(K) \rceil$ bits: *Fixed-rate quantization*
- Representation levels coded in $H(Y)$ bits: *Quantization with entropy encoding*
- (Later: *Entropy-constrained quantization*)

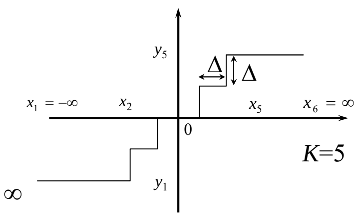
Bit Rate versus Distortion – (2)

- Performance of quantizer is determined by
  - the number of representation levels (bit rate or average codeword length R)
  - the quality σ_q^2 or SNR
- Fixed-rate quantizer design:

  For given K, find the quantizer characteristic with smallest σ_q^2

5 LSE0 - Mod 03, Part 2 Uniform Quantization

R-D Optimal Design: Uniform Quantizer – (1)



- Given:
  $x_1 = -\infty$ $x_{L+1} = \infty$

$$x_{k+1} - x_k = \Delta \qquad y_k = \frac{(x_{k+1} + x_k)}{2}$$

- Find Δ such that σ_q^2 is minimized

R-D Optimal Design: Uniform Quantizer – (2)

$$\min_{\Delta} \sum_{k=1}^{K} \int_{x_k}^{x_{k+1}} (x - y_k)^2 \, p_X(x) \, \mathrm{d}x$$

For values of K>3, this requires numerical optimization

$\sigma_x = 1$

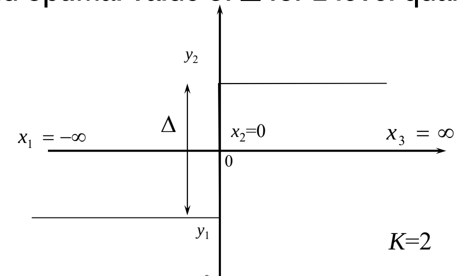
R (bit/sample)	Δ/σx Uniform	Δ/σx Gaussian	Δ/σx Laplace	SNR (dB) Uniform	SNR (dB) Gaussian	SNR (dB) Laplace
1	1.732	1.596	1.414	6.02	4.40	3.01
2	0.866	0.996	1.087	12.04	9.25	7.07
3	0.433	0.586	0.731	18.06	14.27	11.44
4	0.217	0.335	0.461	24.08	19.38	15.96
5	0.108	0.188	0.280	30.10	24.57	20.60
6	0.054	0.104	0.166	36.12	29.83	25.36
7	0.027	0.057	0.096	42.14	35.13	30.23
8	0.013	0.031	0.055	48.17	40.34	35.14

↓ smaller D: "finer" quantizer

→ more difficult to quantize

Example: Two-Level Quantizer – (1)

* Find optimal value of Δ for 2 level quantizer



$$\sigma_q^2 = \sum_{k=1}^{K} \int_{x_k}^{x_{k+1}} (x - y_k)^2 \, p_X(x) \, \mathrm{d}x = \int_{-\infty}^{0} (x + \frac{\Delta}{2})^2 \, p_X(x) \, \mathrm{d}x + \int_{0}^{\infty} (x - \frac{\Delta}{2})^2 \, p_X(x) \, \mathrm{d}x$$

Two-Level Quantizer – (2)

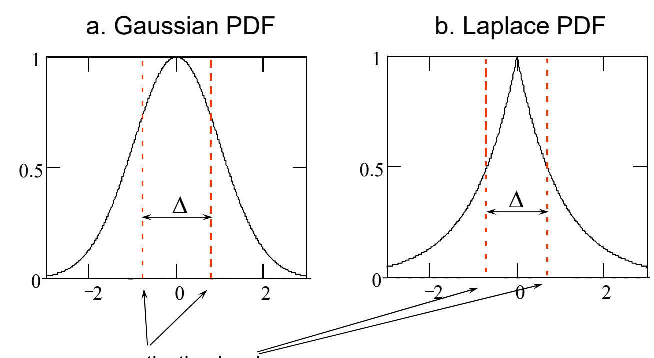
$$\sigma_q^2 = \int_{-\infty}^{0} (x + \frac{\Delta}{2})^2 \, p_X(x) \, \mathrm{d}x + \int_{0}^{\infty} (x - \frac{\Delta}{2})^2 \, p_X(x) \, \mathrm{d}x$$

$$= \sigma_x^2 + \frac{\Delta^2}{4} - 2\Delta \int_{0}^{\infty} x p_X(x) \, \mathrm{d}x$$

$$\Rightarrow \Delta_{\text{optimal}} = \min_{\Delta} \sigma_q^2 = 4 \int_{0}^{\infty} x p_X(x) \, \mathrm{d}x = 2E[|X|]$$

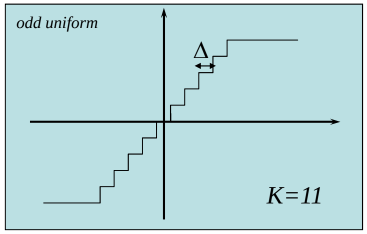
PDF	Δ/σx	σ_q^2	SNR (dB)
Uniform	1.732	0.250	6.02
Gaussian	1.596	0.363	4.40
Laplace	1.414	0.500	3.01
Gamma	1.154	0.667	1.76

Two-Level Quantizer – (3)



Implementation of Uniform Quantizer

* The quantizer
  - Odd
  - Uniform
  - K = 11 outp.levels



can be implemented by:

$$Q[x] = y_j = \Delta\,\text{nint}\left(\frac{x}{\Delta}\right) = \Delta\left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor$$

$$j = \text{nint}\left(\frac{x}{\Delta}\right) = \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor$$

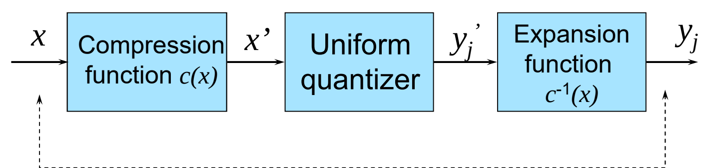
5LSE0 Mod 03, Part 3

Non-Uniform Quantization

Non-Uniform Quantizer / 2 Approaches

1. Preprocessing of x by non-linear function followed by uniform quantizer:
   - Companding (compression-expanding)
   - Simple implementation
   - Popular for audio: logarithmic curves
     - A-law (Europe)
     - and μ-law (USA, Japan)
2. Lloyd-Max quantizers, minimization of σ_q^2
   - Complex design
   - More complex implementation than uniform quantizer
   - Additional gain?

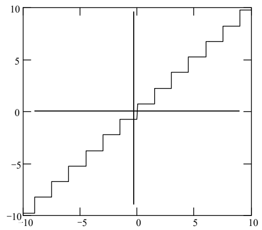
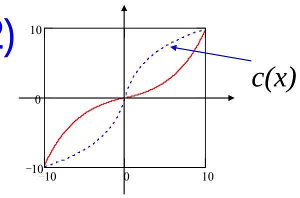
Companding – (1)



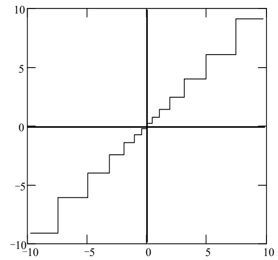
Non-uniform behavior

Uniform quantizer is embedded between two non-uniform functions

Companding – (2)

A/μ-law like companding



A- and μ-Law Definition

A-Law

$$c(x) = \begin{cases} x_{\max} \dfrac{ax / x_{\max}}{1+\ln(a)} & 0 \leq |x|/x_{\max} \leq 1/a \\ x_{\max} \dfrac{\ln(a|x|/x_{\max})}{1+\ln(a)} \operatorname{sgn}(x) & 1/a < |x|/x_{\max} \leq 1 \end{cases}$$

μ -Law

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

Lloyd-Max Quantizer – (1)

* Minimizes quantization noise variance, without enforcing any* structure onto decision thresholds and representation levels

$$\min \sigma_q^2 = \min \sum_{k=1}^{K} \int_{x_k}^{x_{k+1}} (x - y_k)^2 \, p_X(x) \, \mathrm{d}x$$

for:

$x_k \quad k = 2,3,\ldots,K$

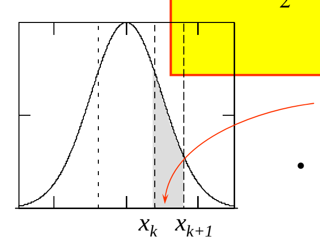
$y_k \quad k = 1,2,\ldots,K$

* *except for symmetry of the quantizer*

Lloyd-Max Quantizer – (2)

* General solution is given by implicit expressions:

$$x_k = \frac{1}{2}\left(y_{k-1} + y_k\right) \qquad y_k = \frac{\int_{x_k}^{x_{k+1}} x p_X(x) \, \mathrm{d}x}{\int_{x_k}^{x_{k+1}} p_X(x) \, \mathrm{d}x}$$



- *Note the structure of the formulas*
  - *Middle of two representation levels*
  - *Weighted average decision levels*

Lloyd-Max Quantizer – (3)

* Property of solution:
  - *if* x_k are known $\Rightarrow y_k$ are known
  - *if* y_k are known $\Rightarrow x_k$ are known
* Iterative design necessary

initial choice, "seed"

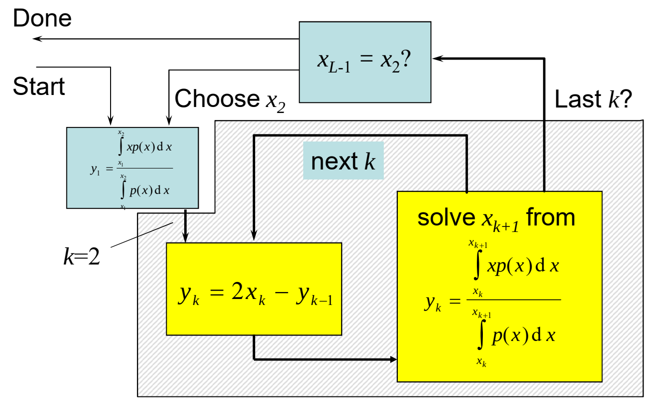
$$x_1 = -\infty$$

$$\text{Assume: } x_2 = g \quad \Rightarrow \quad y_1 = \frac{\int_{x_1}^{x_2} x p(x)\,\mathrm{d}x}{\int_{x_1}^{x_2} p(x)\,\mathrm{d}x} \quad \Rightarrow \quad y_2 = 2x_2 - y_1$$

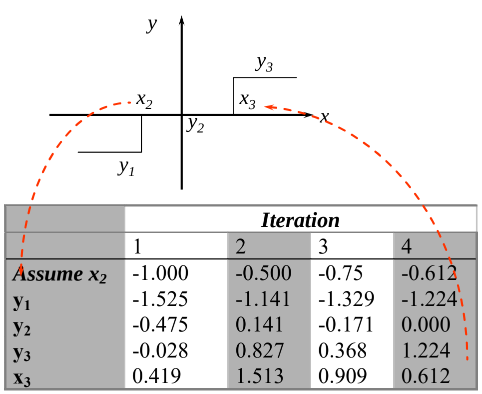
$$\text{Solve } x_3 \text{ from: } y_2 = \frac{\int_{x_2}^{x_3} x p(x)\,\mathrm{d}x}{\int_{x_2}^{x_3} p(x)\,\mathrm{d}x} \quad \Rightarrow \quad y_3 = 2x_3 - y_2$$

etcetera

Lloyd-Max / Iterative Solution Scheme



Lloyd-Max / Example iterative Q design



	Iteration			
	1	2	3	4
Assume x_2	-1.000	-0.500	-0.75	-0.612
y_1	-1.525	-1.141	-1.329	-1.224
y_2	-0.475	0.141	-0.171	0.000
y_3	-0.028	0.827	0.368	1.224
x_3	0.419	1.513	0.909	0.612

Examples of Lloyd-Max Quantizers

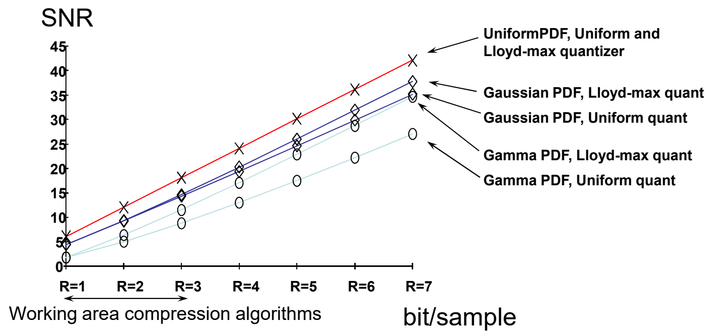
- Example from VcDemo Quantizer

* For *uniform PDF*, Lloyd-Max quantizer is *uniform*

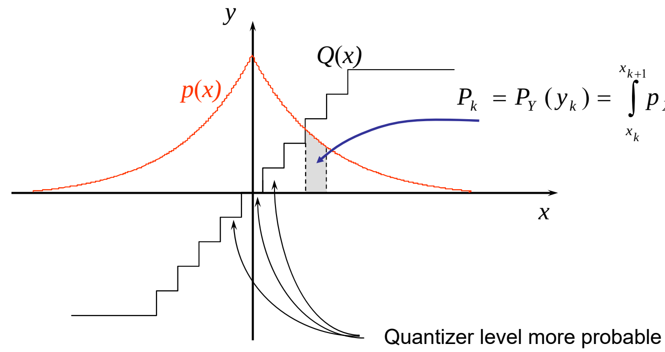
		K=2, 1 bit/sample		K=4, 2 bit/sample		K=8, 3 bit/sample		K=16, 4 bit/sample	
PDF	k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
Gaussian	1	0.000	0.798	0.000	0.453	0.000	0.245	0.000	0.128
	2			0.982	1.510	0.501	0.756	0.258	0.388
	3					1.050	1.344	0.522	0.657
	4					1.748	2.152	0.800	0.942
	5							1.099	1.256
	6							1.437	1.618
	7							1.844	2.069
	8							2.401	2.733
Laplace	1	0.000	0.707	0.000	0.402	0.000	0.233	0.000	0.124
	2			1.127	1.834	0.533	0.833	0.264	0.405
	3					1.253	1.673	0.567	0.729
	4					2.380	3.087	0.920	1.111
	5							1.345	1.578
	6							1.878	2.178
	7							2.597	3.017
	8							3.725	4.432

Uniform quantizer versus Lloyd-Max

For relevant bit rates, Lloyd-Max does often not pay off.



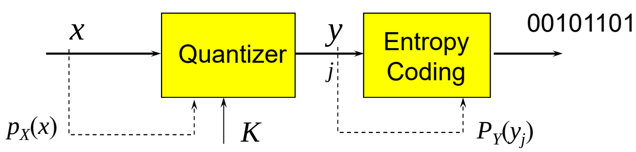
Quantization with Entropy Coding – (1)



$$P_k = P_Y(y_k) = \int_{x_k}^{x_{k+1}} p_X(x)dx$$

- Example from VcDemo Quantizer

Quantization with Entropy Coding – (2)



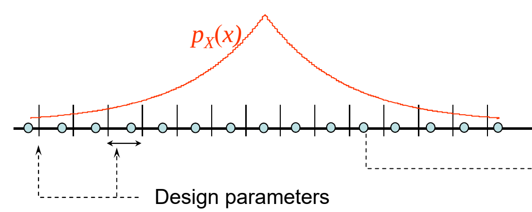
1. Quantizer is optimized for *given* #representation levels K
2. Entropy coder (VLC) is optimized for given probabilities of y_j, yielding average codeword length L

   But, the combination is *not* optimal

   – other quantizer / VLC combinations exist that gives a smaller quantization error variance for same average codeword length

Entropy-Constrained Quantization

- ∗ To find the overall optimal result, quantizer and entropy coder must be *jointly* designed
  - – Complex optimization problem
  - – Reasonable approximations are obtained by *Uniform Threshold Quantizers (UTQ)*

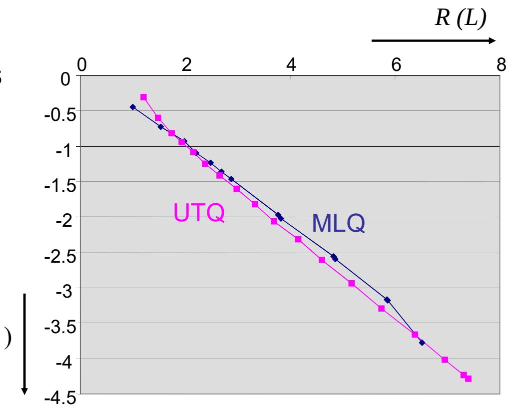


$$y_k = \frac{\int_{x_k}^{x_{k+1}} x p_X(x)dx}{\int_{x_k}^{x_{k+1}} p_X(x)dx}$$

Example MaxLloydQ versus UTQ

45

* Quantizers taken from VcDemo, optimized for Gaussian PDF



$\log_{10}(\sigma^2_{q,proto})$

46

5LSE0 – Mod 03, Part 4
Quantization in practical cases

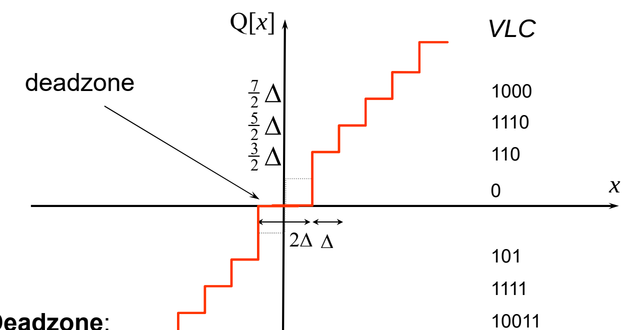
Quantization in Practice – (1)

47

* Uniform quantizers are preferred
  – Implementation and limited loss compared to Lloyd-Max
  – Easily scalable (one parameter: step size Δ)
* *Odd* quantizers are often preferred over *even* because of the presence of a representation level at zero
  – In good compression scheme many (near-)zero values occur
  – Zeroes efficiently coded by an entropy coder (run length coding)
* Audio: Companding is usual
* Image/video coding : No companding
  – Uniform quantizer with *deadzone* is typical

Quantization in Practice – (2)

48

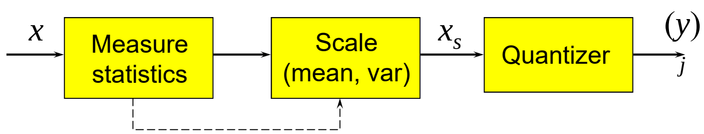


Deadzone:
• Improves noise robustness of coding system
• “Stimulates” truncation to zero: can be coded efficiently

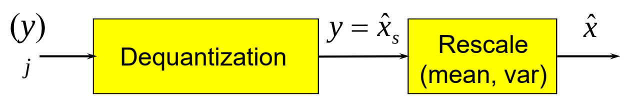
Overall Quantization Error – (1)

* Only *prototype* quantizers are designed with σ_q^2

Sending side



Receiving side



Overall Quantization Error – (2)

* Scaling procedure: $x_s = \dfrac{x-\mu_x}{\sigma_x}$
* Rescaling procedure: $\hat{x} = \sigma_x y + \mu_x = \sigma_x \hat{x}_s + \mu_x$
* Overall effect: Quantization noise variance is scaled with : σ_x^2

$$\sigma_q^2 = E[(\hat{x}-x)^2] = E[(\sigma_x \hat{x}_s - \sigma_x x_s)^2] = \sigma_x^2 E[(\hat{x}_s - x_s)^2]$$
$$= \sigma_x^2 \sigma_{q,\text{prototype}}^2$$

$\Rightarrow$ *Quantization noise variance is linearly proportional to the variance of the signal to be quantized*

The following sheets were used in previous classes and are included for reference only

Quantizer Example - (2)

Back to MLQuantizers
Back to Prob. Calculation

	Repres. Level	Probability	Codewords (positive)	Codewords (negative)
1	±9.247	0.00007	00010101101010	00010101101011
2	±6.875	0.00031	000101011011	000101011010
3	±5.519	0.00077	0001010111	00010101100
4	±4/562	0.00152	0001010000	0001010001
5	±3.823	0.00269	000101001	000101010
6	±3.217	0.00444	00011010	00011011
7	±2.703	0.00699	1011010	1011011
8	±2.253	0.01068	0001011	0001100
9	±1.855	0.01590	000111	101100
10	±1.497	0.02318	10111	000100
11	±1.175	0.03316	01000	01001
12	±0.884	0.04658	1100	1101
13	±0.622	0.06441	0101	1010
14	±0.388	0.08797	111	0000
15	±0.180	0.11987	011	100
16	0.000	0.16292	001	

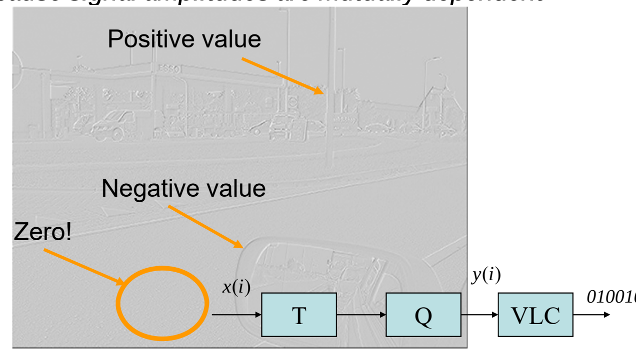
* Simple binary coding would require $\lceil \log_2(31) \rceil$ =5 bits
* $\boldsymbol{H(S)}$ = 3.876 bit/repr.level
* $\boldsymbol{L}$ = 3.912 bit/repr.level

Application to Signal Coding

* We will see that efficient "transforms" used in compression produce
  – A lot of "zero" values
  – Some (significant) non-zero values
* Typical symbol sequences to be coded: "5 1 0 0 0 0 0 0 0 3 0 0 6 0 0 0 0 1 0 0 0 0 ..."
  – Will be done by {zero-run, non-zero symbol} pairs
  – Here: "{0,5}, {0,1}, {7,3}, {2,6}, {4,1}, ..."
  – The pairs will now be assigned a Huffman code

Why can Signals be Compressed?

Because signal amplitudes are mutually dependent



The following sheets are included for further reference

Note on Quantization Noise Variance

* We wish to calculate the variance of *q*:

$$\sigma_q^2 = E[q^2] - \underbrace{E[q]^2}_{=0} = \int_{-\infty}^{\infty} q^2 p_q(q)dq$$

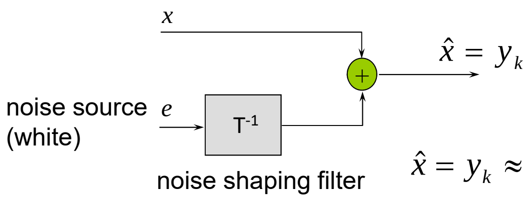
* Try to calculate $p_q(q)$, but with non-linear relation of $q=x-Q[x]$, this is not easy
* Much easier, take following short-cut (E-operator):

$$\sigma_q^2 = E[q^2] = E[(x-Q[x])^2]$$

$$= \int_{-\infty}^{\infty} (x-Q[x])^2 \, p_X(x)\, \mathrm{d}\, x$$

Modeling of Quantizers

- Quantizer is often the only non-linear element in a compression system
- Linear (approximating) models are sometimes used to analyze the performance of a quantizer / compression system



$$\hat{x} = y_k \approx x + T^{-1} e$$