BRIGHTCOVE

Block Codes with Embedded Quantization Step Size Information

Yuriy Reznik
Brightcove, Inc.
Seattle, WA, USA

DCC

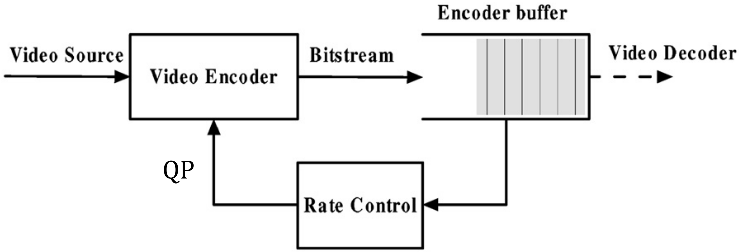
Data Compression Conference

March 21-24, 2023
Snowbird, UT

Motivational example

Rate control module in video coding

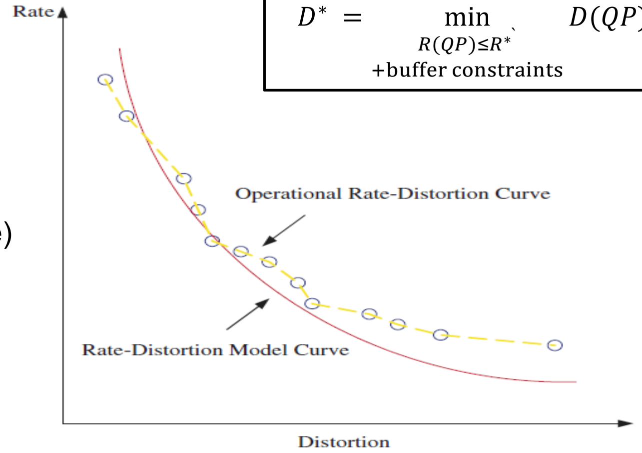
- A unit ensuring that overall bitrate approaching target rate R
  - achieving best quality (or minimum distortion D)
  - within certain constraints (decode buffer size, max rate, etc.)
- It does it by adjusting quantization step sizes in the bitstream:
  - Pictures/slices = have “quant” or “QP” parameter in headers
  - Macroblocks/CTUs = allow transmission of “Delta QPs”
- Two levels of rate adaptations:
  - Frame-level bit allocation and QP derivation, and
  - Macroblock or CTU-level bit allocation and DeltaQP derivation



Models used to implement rate control methods

- Typically influenced by information-theory concepts (*):
  - rate-distortion characteristic of a source (e.g. RD of Gaussian source)
  - operational rate-distortion characteristic, which is expected to be similar to an idealized rate-distortion curve

$$D^* = \min_{R(QP) \leq R^*} D(QP)$$
+buffer constraints



Questions

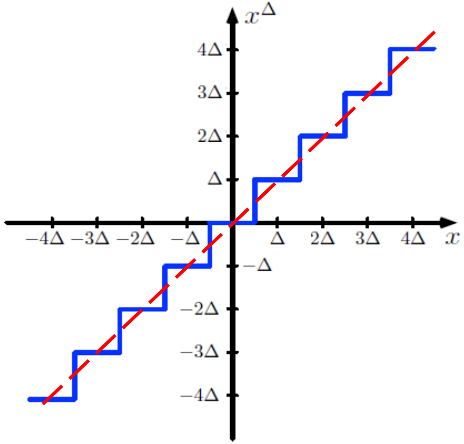
- When we transmit QPs do we still solve the classic “quantization problem”?
- How does transmission of QPs affect the performance of such codes?
- Does the use of classic R(D) models still appropriate in this application?

(*) T. Cover and J. Thomas, “Elements of Information Theory”, Wiley, NY, 1991.
(**) H. Chen, K-N. Ngan, “Recent advances in rate control for video coding,” Signal Processing: Image Communication, vol. 22, no. 1, 2007, pp 19-38

Some known facts

Uniform quantization

- Effectively a map: $x \rightarrow x^{\Delta}$
  - x – real-valued random variable, $x \sim p(x)$, $h(x) = -\int p(x) \log p(x) dx$
  - x^{Δ} – quantized output, $x^{\Delta} \sim P(x^{\Delta})$, $H(x^{\Delta}) = -\sum_i P(x_i^{\Delta}) \log P(x_i^{\Delta})$
  - Δ – step size
- The simplest example: $x^{\Delta} = \Delta \cdot \lfloor x/\Delta + 1/2 \rfloor$ (uniform mid-tread quantizer)



Performance in high-fidelity regime

- If $p(x)$ is Riemann-integrable, then with $\Delta \rightarrow 0$, the following holds (*):

$$H(x^{\Delta}) \rightarrow -\log(\Delta) + h(x)$$

Operational rate-distortion function of uniform quantizer

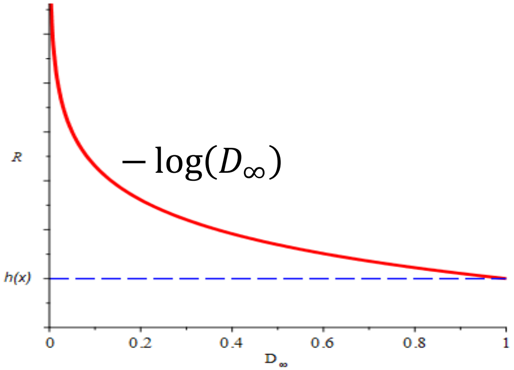
- R – encoding bitrate, $R \geq H(x^{\Delta})$
- D_{∞}– ℓ_{∞}- type distortion:

$$D_{\infty} = \max_x |x - x^{\Delta}(x)| = \Delta/2$$

- Then, with $\Delta, D_{\infty} \rightarrow 0$:

$$R \rightarrow -\log(D_{\infty}) + h(x) + O(1)$$

- The $-\log(D_{\infty})$ term is the most important.



(*) T. Cover and J. Thomas, “Elements of Information Theory”, Wiley, NY, 1991.

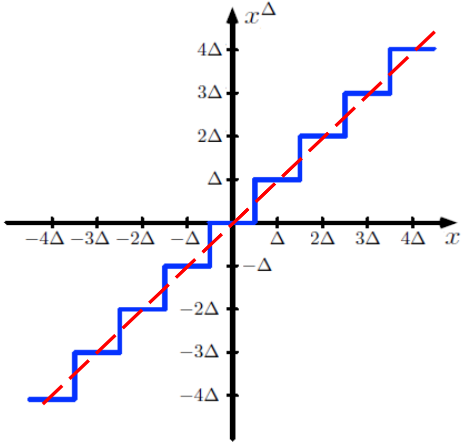
BRIGHTCOVE

Code with embedded step size

Quantizer

- Input: $x_1, \ldots, x_n$ – samples from variable x; quantized output: $x_1^\Delta, \ldots, x_n^\Delta$
- Step size (q – integer, C – constant):

$$\Delta(\mathrm{q}) = C/q$$



Block code

- Send parameter q, encoded any monotonic code for integers
- Send quantized samples $x_1^\Delta, \ldots, x_n^\Delta$, encoded by arithmetic codes for $x^\Delta \sim P(x^\Delta)$
- Bitstream :

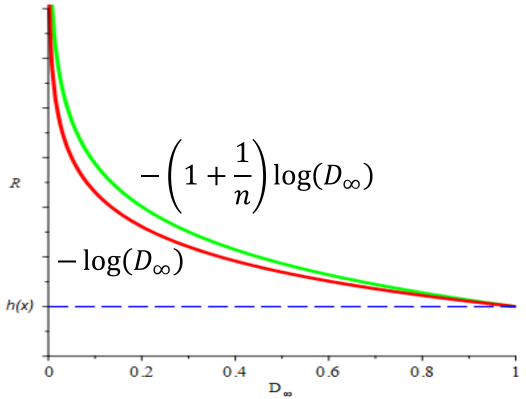
$$< q >< x_1^\Delta >, \ldots, < x_n^\Delta >$$

Operational rate-distortion function

- n – block length
- $D_\infty = \max_i |x_i - x_i^\Delta| \le \Delta/2$ – distortion
- R_n – per-sample bitrate:

$$R_n > \frac{1}{n}\log(q) + H(x^\Delta) \xrightarrow{\Delta \to 0} -\left(1 + \frac{1}{n}\right)\log(D_{n,\infty}) + h(x) + O(1)$$

- In comparison with regular uniform quantizer, we observe that the transmission of $\Delta(q)$ increases the bitrate of a block code by a factor of $\left(1 + \frac{1}{n}\right)$



BRIGHTCOVE

A related mathematical problem

Consider now the following problem:

- Given n irrational numbers: $\xi_1, \ldots, \xi_n$, find integers $p_1, \ldots, p_n$ and q, such that

$$\frac{p_1}{q} \approx \xi_1, \ldots, \frac{p_n}{q} \approx \xi_n$$

- This problem is remarkably old and known in mathematics as *simultaneous Diophantine approximations* (named after Diophantus of Alexandria, 200s BC)

Performance of Diophantine approximations:

- There exists infinitely many integers $p_1, \ldots, p_n$ and q, such that (*):

$$\max_i |\xi_i - p_i/q| < \frac{1}{1+1/n} q^{-(1+1/n)}$$

- This is a significant improvement over a trivial bound:

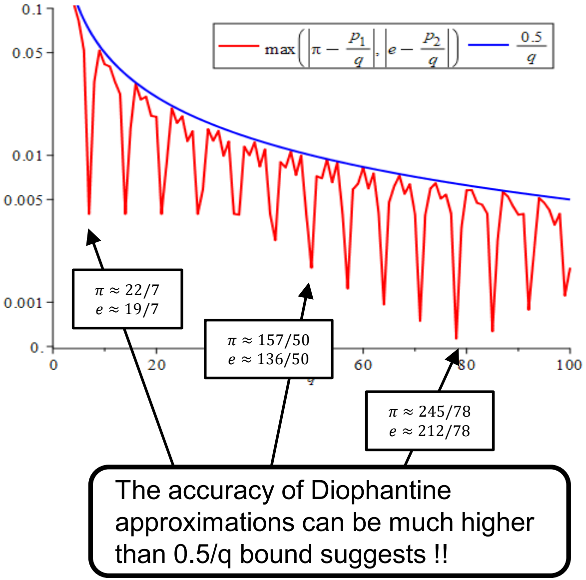
$$\max_i |\xi_i - p_i/q| \leq 0.5\, q^{-1}$$

Connection to quantization:

- Given a block $x_1, \ldots, x_n$, and quantizer $\Delta(q) = C/q$, we see that $\xi_i = \frac{x_i}{C}, i = 1, \ldots, n$ maps quantizer design to the Diophantine approximation problem!
- However, in earlier analysis, we assumed that $D_\infty = \max_i |x_i - x_i^\Delta| \leq \frac{1}{2}\Delta = \frac{1}{2} C/q$, which is a reasonable bound when we don't know much about sample values or q
- But if we know the samples, and selectively choose q, then the existence of much higher accuracy approximations makes a difference!

Example:

- $\xi_1 = \pi \approx 3.14159 \ldots$
- $\xi_2 = e \approx 2.7182 \ldots$
- Best approximations with q<100:



(*) J. Cassels, "An Introduction to Diophantine Approximations", Cambridge University Press, 1957.

BRIGHTCOVE

Achievable performance

Main result

- Theorem 1. Given a block of samples $x_1, \dots, x_n$, *there exist infinitely many values of quantization parameter* q, such that the resulting rate-distortion performance of a block code with embedded quantization step size parameter satisfies:

  $$R_n \leq -\log(D_\infty) + h(x) + O(1)$$

  This inequality holds in high-fidelity ($\Delta(q) \to 0$) regime.

Proof

- The result follows by applying accuracy limit for Diophantine approximations: $D_\infty = \max_i |x_i - x_i^\Delta| \leq C \frac{1}{1+1/n} q^{-(1+1/n)}$

Discussion

- Compared to our earlier estimate: $R_n \geq -\left(1 + \frac{1}{n}\right) \log(D_\infty) + \cdots$, this means that the leading $\left(1 + \frac{1}{n}\right)$ factor can be avoided!
- This means, that block codes with embedded quantization step size information, may, theoretically, be as efficient as codes that do not transmit such information!
- Good news for practical applications! But how to design such codes?

BRIGHTCOVE

Example code construction

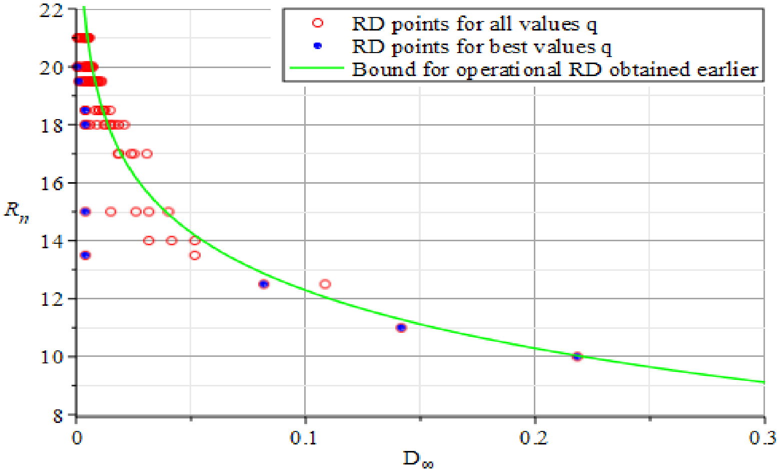
Input

- Source: x – random variable, $x \in [0, x_{\max})$, uniformly distributed, $x_{\max} = 100$
- Input samples: $x_1 = \pi \approx 3.14159 \ldots, x_2 = e \approx 2.7182 \ldots$

Code construction

- Find p_1, p_2, and q such that: $x_1 \approx p_1/q,\ \ x_2 \approx p_2/q$
- Send q by using Levenstein code
- Send p_1 and p_2 by binary codes using $\lceil \log_2(q \cdot x_{\max}) \rceil$ bits

R/D performance (q=1..100):



Example codes:

q	p_1	p_2	$< q >$	$< p_1 >$	$< p_2 >$	R_n [bits]	D_∞	$0.5/q$
2	6	5	1100	00000110	00000101	4/2+8=10	0.21828	0.25000
3	9	8	1101	000001001	000001000	4/2+9=11	0.14159	0.16666
5	16	14	1110001	000010000	000001110	7/2+9=12.5	0.08171	0.10000
7	22	19	1110011	0000010110	0000010011	7/2+10= 13.5	0.00399	0.07142
36	113	98	1111000100100	000001110001	0000110 0010	13/2+12=18.5	0.00394	0.01388
57	179	155	1111000111001	0000010110011	0000010011011	13/2+13=19.5	0.00124	0.00877
78	245	212	11110010001110	0000011110101	0000011010100	14/2+13=20	0.00056	0.00641

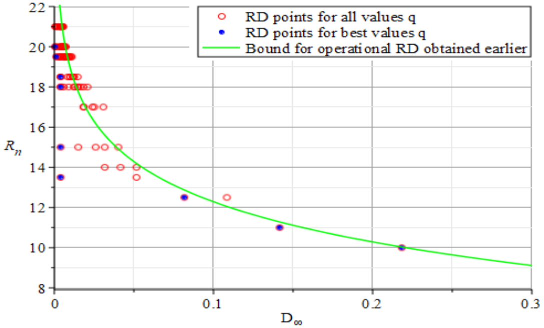
Observations:

- By varying q, the RD points can be all over the place.
- There are few points q for which RD performance is much better. Their existence is predicted by Diophantine theory.
- The bound for RD model obtained earlier misses most of such good operating points!

Conclusions

Results

- Discovered connection between uniform quantization and Diophantine approximation problem
- Showed that block codes that transmit step sizes may (in theory) be asymptotically as efficient as codes that do not carry such information
- Showed that simple RD models don't predict behavior such codes well



Applications & consequences

- The discovered phenomena may help with improving designs of rate control algorithms and performance of encoders in general
- But such improvements may require much more compute power!
  - The problem of finding best Diophantine approximations is known to be NP-complete. Related discussion and results can be found in (*).
  - Finding good near-optimal solutions is a non-trivial problem!
- More work... More fun!

(*) M. Groetschel, L. Lovacz, and A. Schrijver, "Geometric algorithms and combinatorial optimization", Springer, Berlin, 1988.

THANK YOU