VTC 2006 Fall
Dallas

Iterative Joint Video and Channel Decoding in a Trellis-Based Vector Quantized Video Codec and Trellis Coded Modulation Aided Wireless Videophone

R. G. Maunder, J. Kliewer, S. X. Ng, J. Wang, L-L. Yang, L. Hanzo

School of ECS
University of Southampton

Outline

- ❑ Introduction
- ❑ System Overview
- ❑ Trellis-Based Vector Quantization
- ❑ Results
- ❑ Conclusions and Future Work

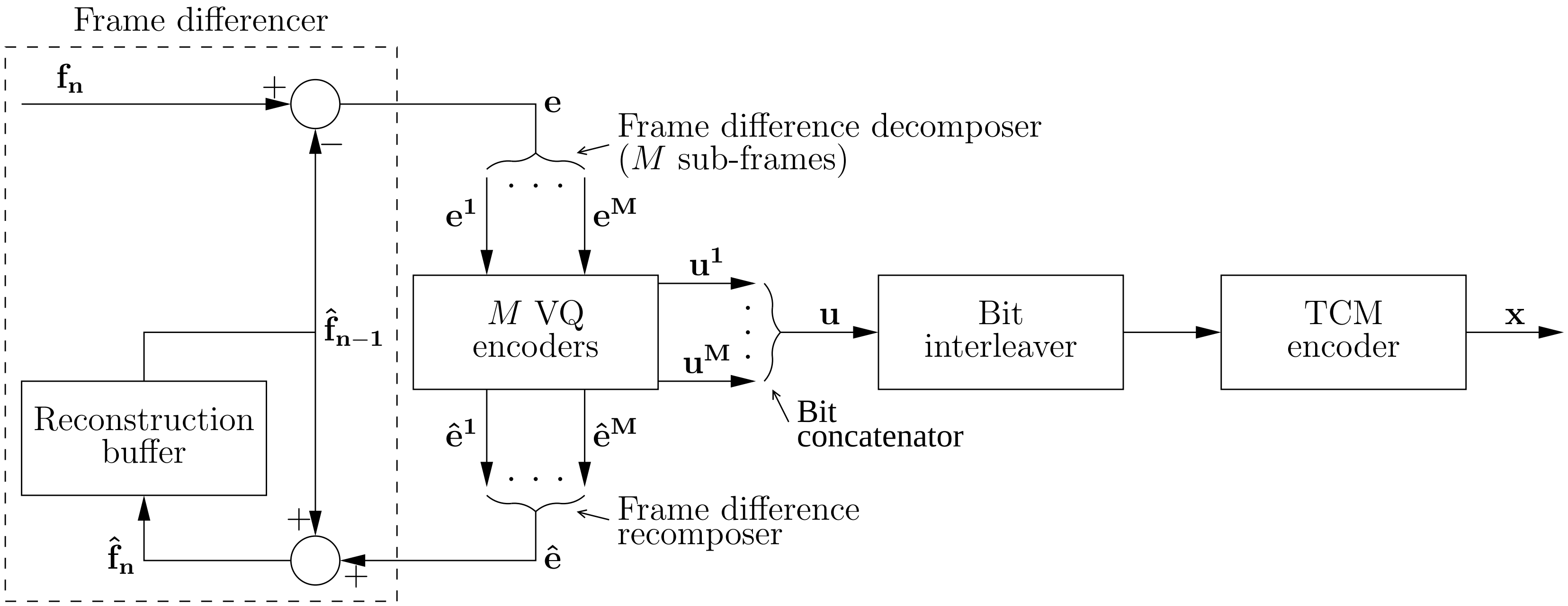
Shannonian Video and Channel Coding

- Video and channel coding may be performed independently without penalty.
- However, this requires a number of conditions to be met, including:
  - infinite complexity and
  - infinite latency.
- Hence, Shannonian video and channel coding is not practical.
- This motivates joint video and channel coding.

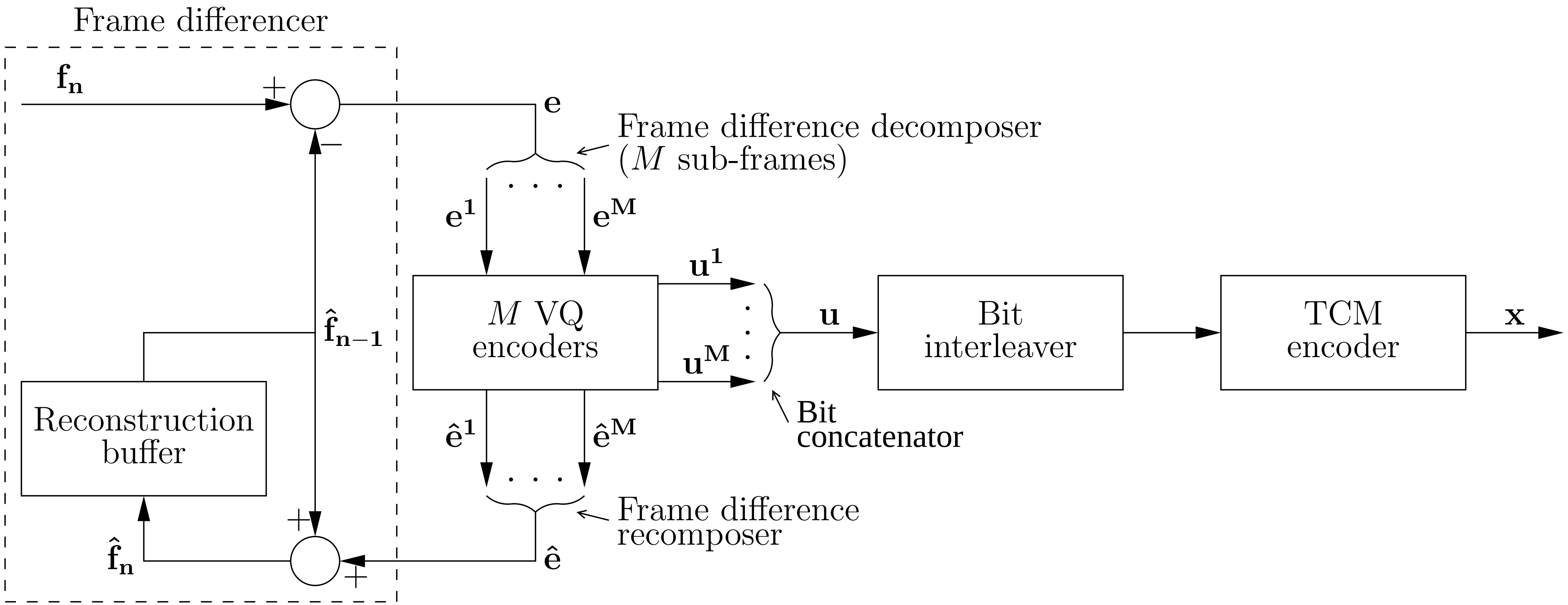
Joint Video and Channel Coding

- As in the Shannonian approach, the video codec achieves compression.
- However, like the channel codec, the video also has a Forward Error Correction (FEC) capability.
- In the receiver, video and channel decoding are performed jointly.
- We employ:
  - a frame differencing and trellis-based Vector Quantization (VQ) video codec and
  - a Trellis Coded Modulation (TCM) channel codec.

Transmitter

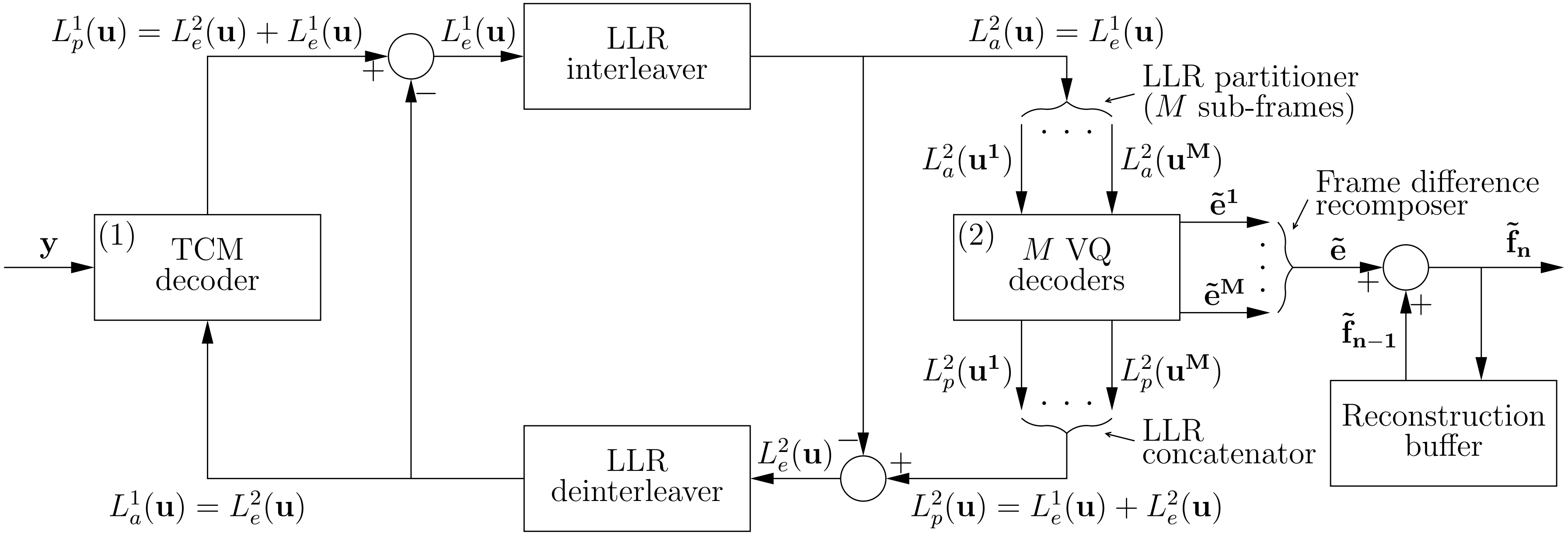


System Overview

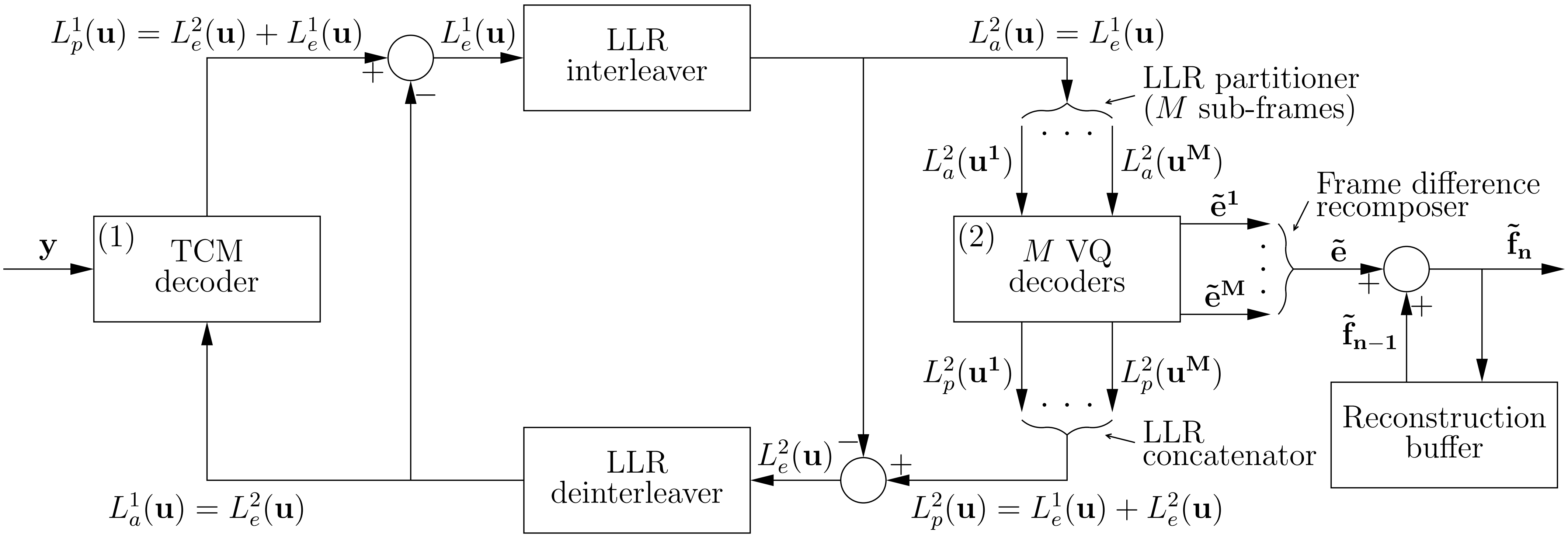


Receiver

- The TCM decoder and the VQ decoders iteratively exchange mutually extrinsic Log Likelihood Ratio (LLR) soft information.
- This extrinsic information is exploited as *a priori* information during decoding.
- The *a priori* information is subtracted from the resultant *a posteriori* information to obtain more reliable extrinsic information for use in the next iteration.



System Overview



$L_p^1(\mathbf{u}) = L_e^2(\mathbf{u}) + L_e^1(\mathbf{u})$

$+$

$-$

$L_e^1(\mathbf{u})$

LLR interleaver

$L_a^2(\mathbf{u}) = L_e^1(\mathbf{u})$

LLR partitioner (M sub-frames)

$L_a^2(\mathbf{u^1})$

$L_a^2(\mathbf{u^M})$

$\mathbf{y}$

(1) TCM decoder

(2) M VQ decoders

$\tilde{\mathbf{e}}^\mathbf{1}$

$\tilde{\mathbf{e}}^\mathbf{M}$

Frame difference recomposer

$\tilde{\mathbf{e}}$

$+$

$\tilde{\mathbf{f}}_\mathbf{n}$

$+$

$\tilde{\mathbf{f}}_\mathbf{n-1}$

Reconstruction buffer

$L_p^2(\mathbf{u^1})$

$L_p^2(\mathbf{u^M})$

LLR concatenator

$L_e^2(\mathbf{u})$

$-$

$+$

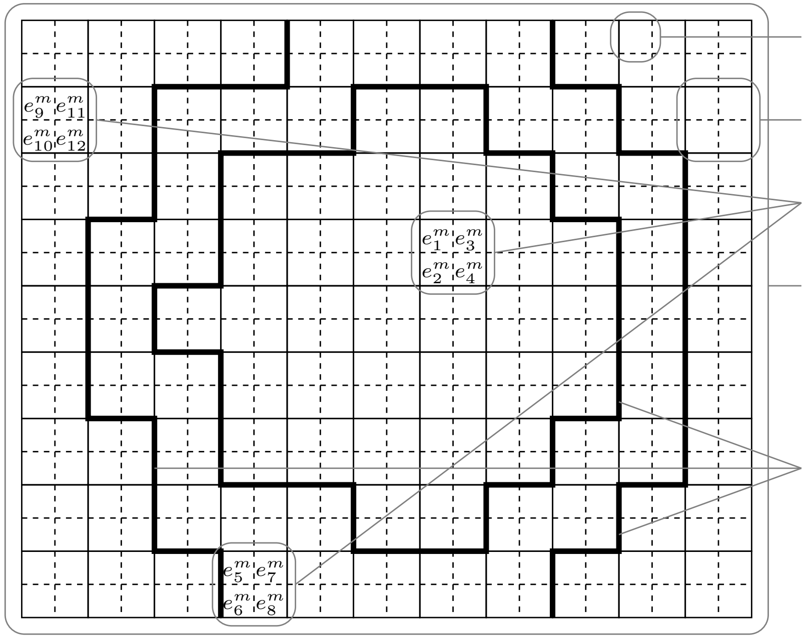
LLR deinterleaver

$L_a^1(\mathbf{u}) = L_e^2(\mathbf{u})$

$L_p^2(\mathbf{u}) = L_e^1(\mathbf{u}) + L_e^2(\mathbf{u})$

Frame Difference Decomposition

- The Frame Difference (FD) $\mathbf{e}$ is decomposed into M number of sub-frames $[\mathbf{e^1} \ldots \mathbf{e^M}]$ for the sake of reducing trellis-based VQ complexity.
- Each FD sub-frame $\mathbf{e^m}$, $m \in [1 \ldots M]$, comprises J number of video blocks $[e_1^m \ldots e_J^m]$ chosen as one Macro Block (MB) from each MB group.
- This ensures that the sub-frames have similar characteristics.
- For example:



Video block, comprising (8×8) pixels.

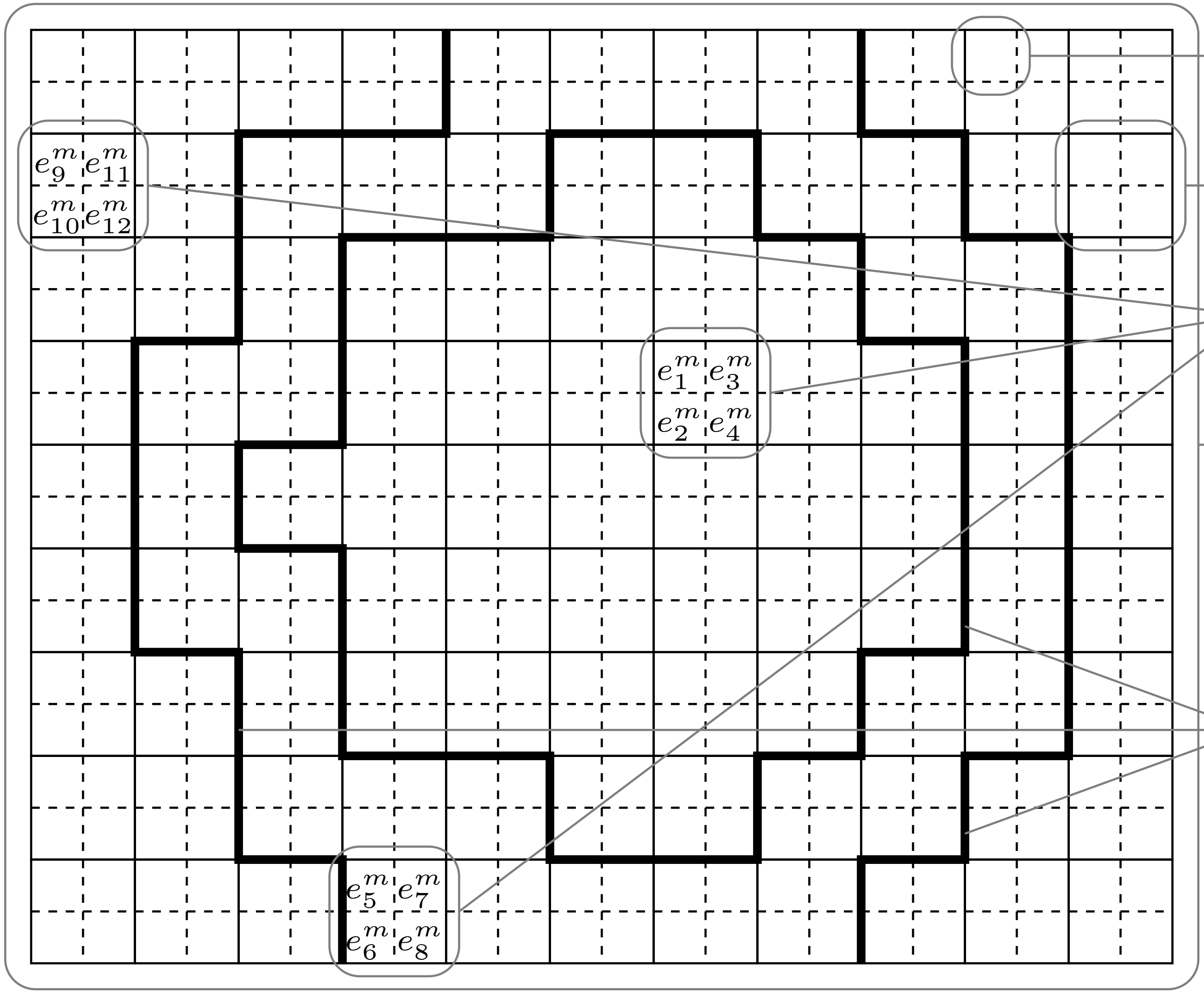
Macro-block, comprising $J^{\mathrm{MB}} = 4$ video blocks.

FD sub-frame $\mathbf{e^m}$, comprising $J = 12$ video blocks.

FD $\mathbf{e}$, comprising $M \cdot J = 396$ video blocks.

Macro-block grouping boundaries. Each macro-block group comprises $M = 33$ macro-blocks.

Trellis-Based Vector Quantization



Video block, comprising (8×8) pixels.

Macro-block, comprising $J^{\mathrm{MB}} = 4$ video blocks.

FD sub-frame $\mathbf{e^m}$, comprising $J = 12$ video blocks.

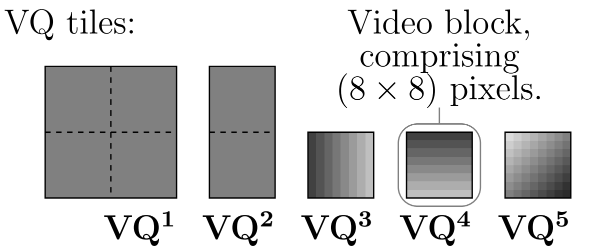
FD $\mathbf{e}$, comprising $M \cdot J = 396$ video blocks.

Macro-block grouping boundaries. Each macro-block group comprises $M = 33$ macro-blocks.

Vector Quantization Codebook

- The LBG-algorithm designed VQ codebook comprises K number of VQ tiles $[\mathbf{VQ^1} \ldots \mathbf{VQ^K}]$ of various dimensions.
- This allows the efficient encoding of large areas of low FD activity.
- Each VQ tile $\mathbf{VQ^k}$, $k \in [1 \ldots K]$, is represented by a minimum free-distance 2 Reversible Variable Length Code (RVLC) $\mathbf{RVLC^k}$ with an entropy-considered length.
- For example:

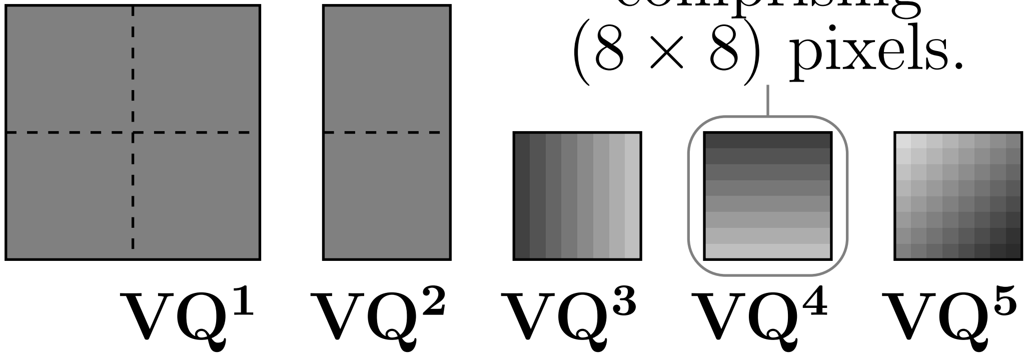
Index k	Video blocks $(J_x^k \times J_y^k) = J^k$	Bits I^k	RVLC code $\mathbf{RVLC^k}$
1	$(2 \times 2) = 4$	1	0
2	$(1 \times 2) = 2$	2	11
3	$(1 \times 1) = 1$	3	101
4	$(1 \times 1) = 1$	4	1001
5	$(1 \times 1) = 1$	5	10001



Trellis-Based Vector Quantization

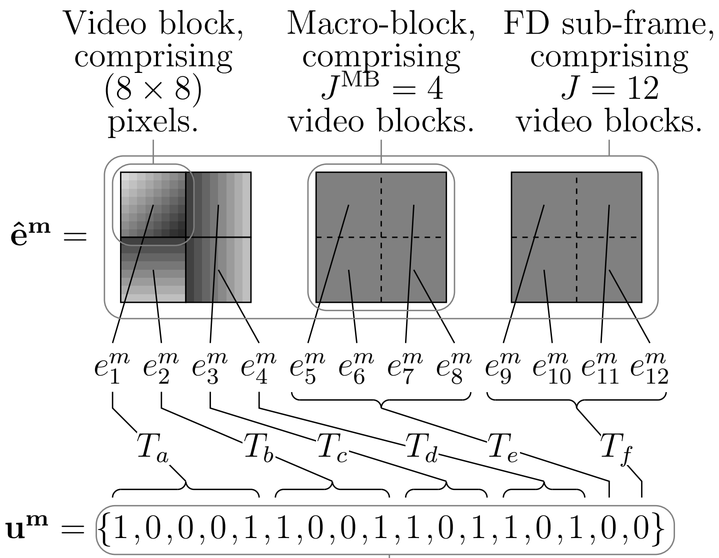
Index k	Video blocks $(J_x^k \times J_y^k) = J^k$	Bits I^k	RVLC code $\mathbf{RVLC^k}$
1	$(2 \times 2) = 4$	1	0
2	$(1 \times 2) = 2$	2	11
3	$(1 \times 1) = 1$	3	101
4	$(1 \times 1) = 1$	4	1001
5	$(1 \times 1) = 1$	5	10001

VQ tiles:



Vector Quantization Code Constraints

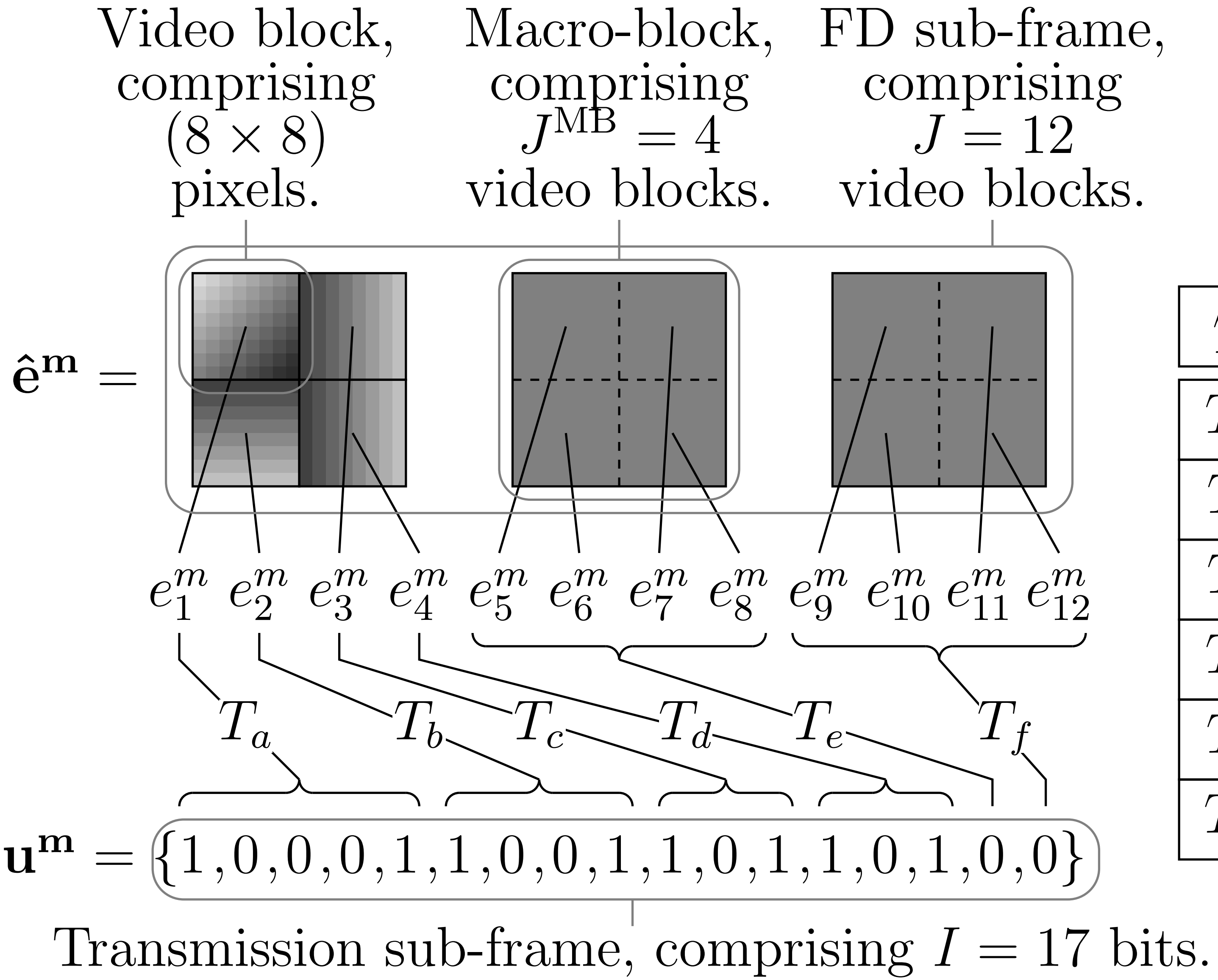
- Each J video block FD sub-frame $\mathbf{e^m}$ *must* be represented by an exact tessellation of VQ tiles from the VQ codebook.
- The corresponding transmission sub-frame $\mathbf{u^m}$ is formed by concatenating the corresponding RVLC codes and *must* comprise I number of bits $[u_1^m \ldots u_I^m]$.
- These code constraints must be adhered to during VQ encoding and may be exploited to achieve FEC during VQ decoding.
- For example:



Transmission sub-frame, comprising $I = 17$ bits.

T	k^T	J^{k^T}	I^{k^T}
T_a	5	1	5
T_b	4	1	4
T_c	3	1	3
T_d	3	1	3
T_e	1	4	1
T_f	1	4	1

Trellis-Based Vector Quantization



Video block, comprising (8×8) pixels.

Macro-block, comprising $J^{\mathrm{MB}} = 4$ video blocks.

FD sub-frame, comprising $J = 12$ video blocks.

$\hat{\mathbf{e}}^{\mathbf{m}} =$

e_1^m e_2^m e_3^m e_4^m e_5^m e_6^m e_7^m e_8^m e_9^m e_{10}^m e_{11}^m e_{12}^m

T_a T_b T_c T_d T_e T_f

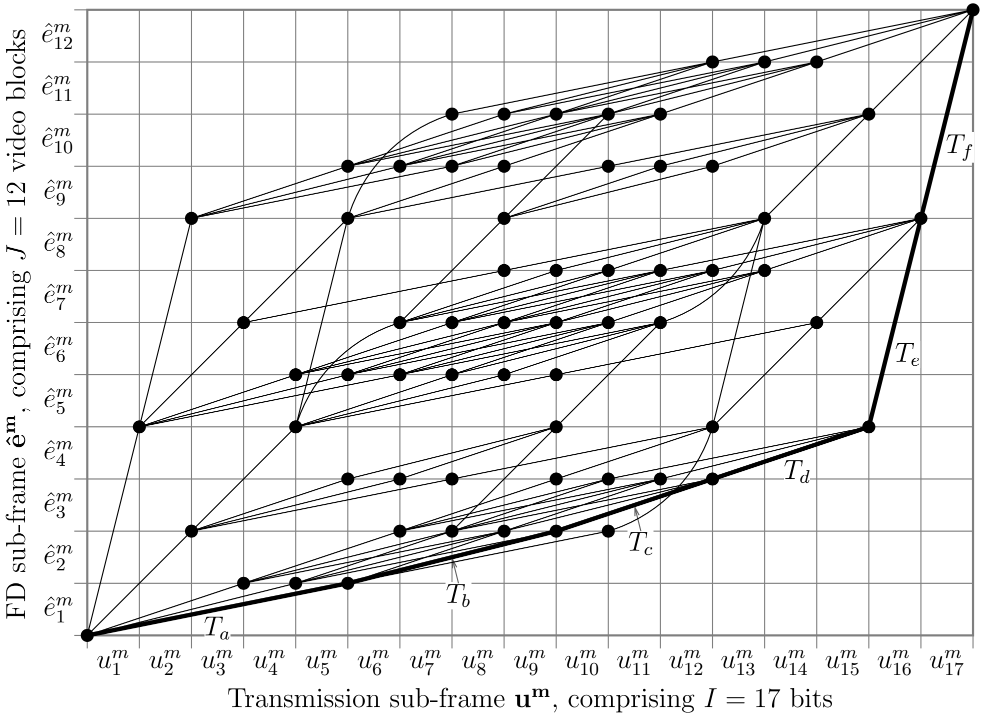
$\mathbf{u}^{\mathbf{m}} = \{1,0,0,0,1,1,0,0,1,1,0,1,1,0,1,0,0\}$

Transmission sub-frame, comprising $I = 17$ bits.

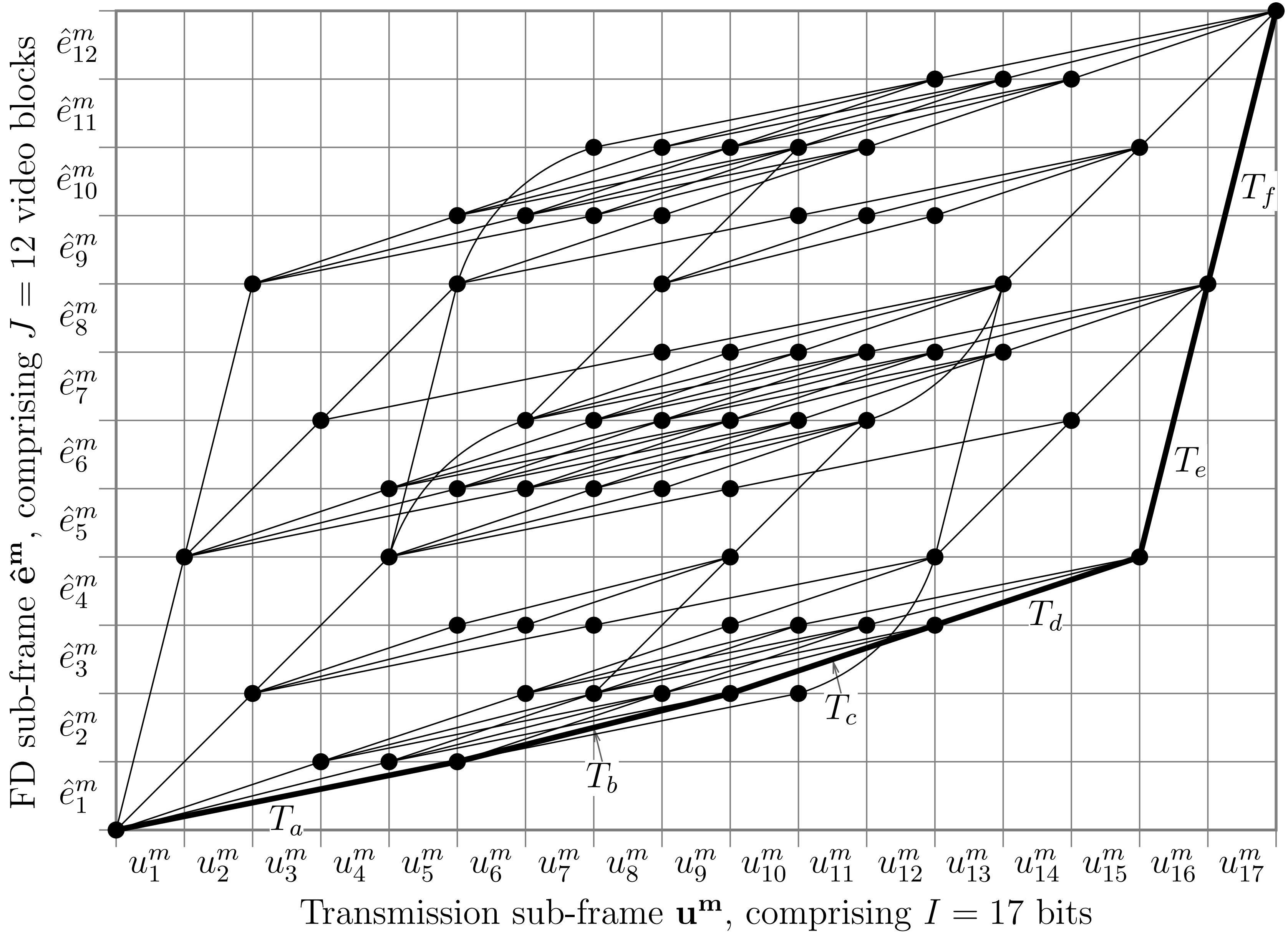
T	k^T	J^{k^T}	I^{k^T}
T_a	5	1	5
T_b	4	1	4
T_c	3	1	3
T_d	3	1	3
T_e	1	4	1
T_f	1	4	1

Vector Quantization Trellis Structure

- Each transition T in the trellis represents a possible application of the J^{k^T}-block VQ tile $\mathbf{VQ}^{\mathbf{k^T}}$ and the corresponding I^{k^T}-bit RVLC $\mathbf{RVLC}^{\mathbf{k^T}}$.
- The trellis represents every possible J-block FD sub-frame $\hat{\mathbf{e}}^{\mathbf{m}}$ and I-bit transmission sub-frame $\mathbf{u}^{\mathbf{m}}$ combination.
- For example:



Trellis-Based Vector Quantization



Vector Quantization Encoding and Decoding

- VQ encoding performs the Viterbi algorithm on the trellis using a mean squared error distortion metric.
- This ensures that the VQ code constraints are adhered to and obtains the optimal Minimum Mean Squared Error (MMSE) encoding.
- The trellis is also employed during VQ decoding to guarantee the recovery of valid information and to provide a FEC capability by exploiting the VQ code constraints and the minimum free distance of the RVLCs.
- The BCJR algorithm is performed on the basis of the transmission sub-frame *a priori* soft information $L_a^2(\mathbf{u}^{\mathbf{m}})$.
- *A posteriori* soft information $L_p^2(\mathbf{u}^{\mathbf{m}})$ is obtained by considering the *a posteriori* probabilities of transitions on vertical cross sections through the trellis.
- Similarly, a MMSE soft FD sub-frame reconstruction $\tilde{\mathbf{e}}^{\mathbf{m}}$ is obtained by considering horizontal cross sections through the trellis.

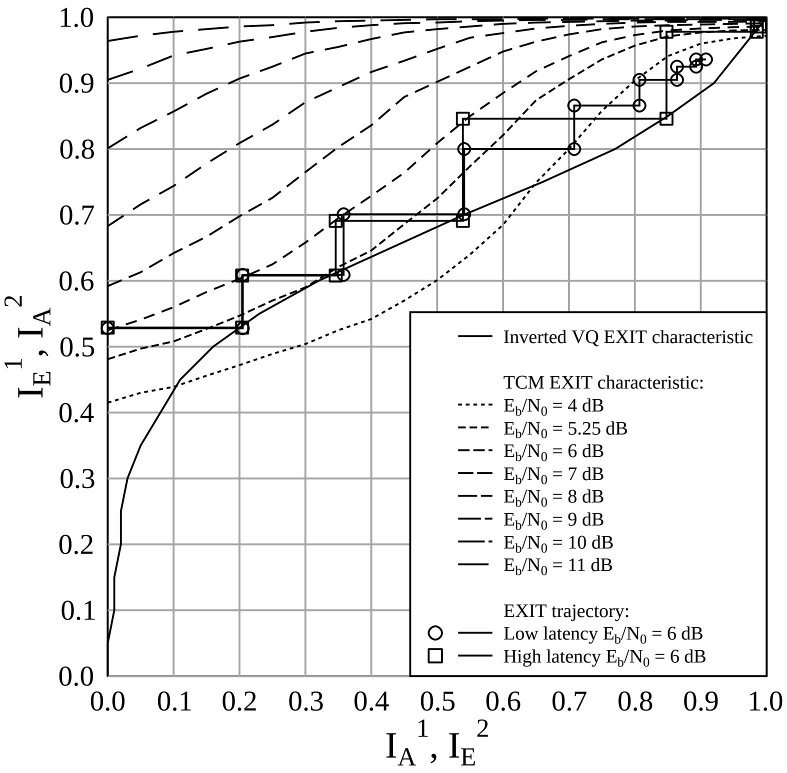
Simulation parameters

- ❏ 100 frames of ‘Lab’ QCIF (176×144)-pixel 10 fps video sequence

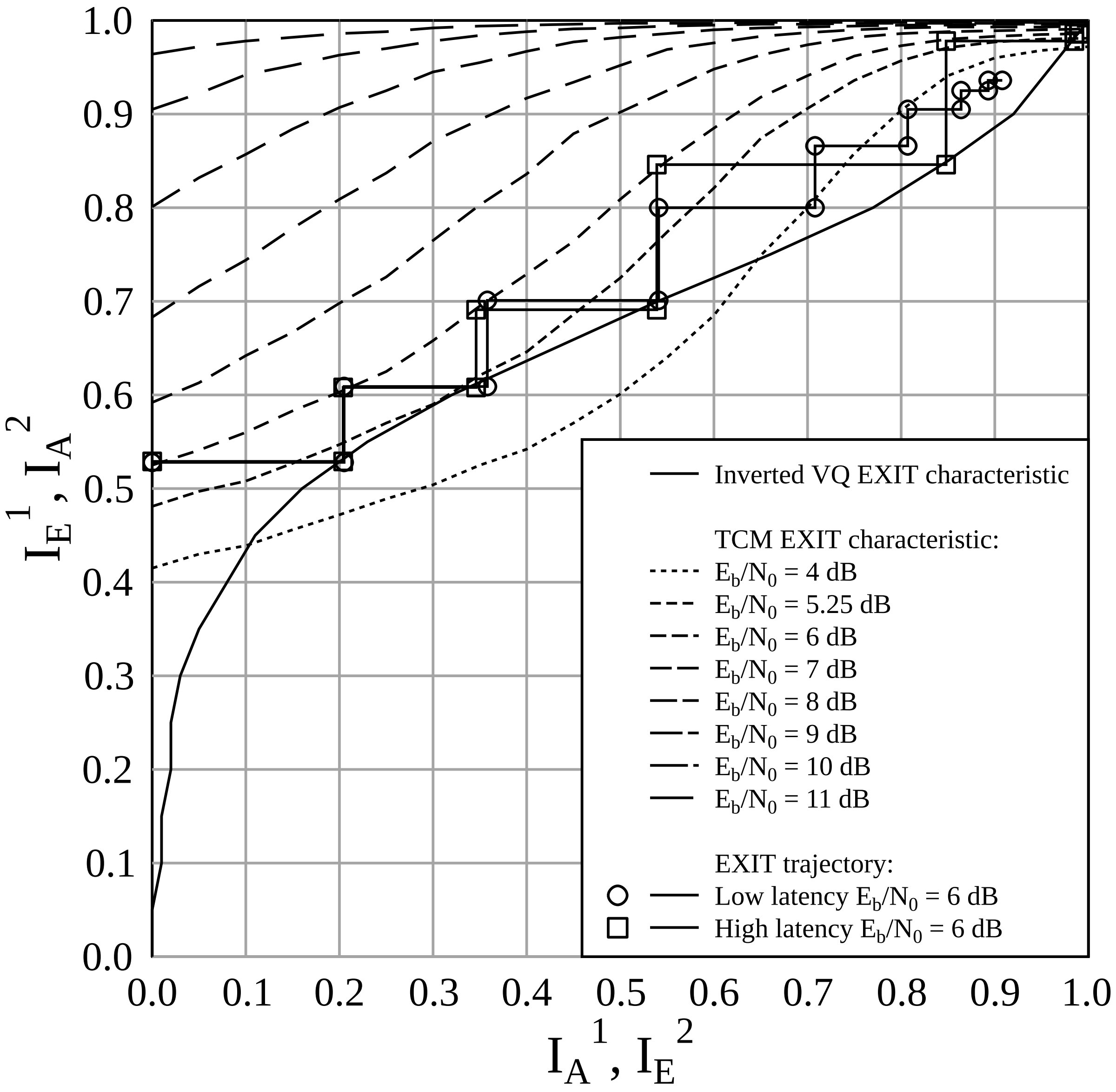
- ❏ $M = 33$, $J = 12$, $I = 45$, $K = 512$, 14.85 kbps
- ❏ $3/4$-rate TCM using 16QAM modulation
- ❏ Bandwidth efficiency of $\eta = 2.00$ bit/s/Hz
- ❏ $E_b/N_0 = 3.96$ dB at Rayleigh fading channel capacity limit
- ❏ Two schemes: 0.1s latency (1485 bit interleaver length) and 5.0s latency (74250 bit interleaver length)

EXIT chart

- Using minimum free-distance 2 RVLCs ensures that VQ decoding can achieve unity extrinsic mutual information and an infinitesimally low decoding error.
- Tunnel for $E_b/N_0 > 5.25$ dB, just 1.29 dB from the channel capacity limit.

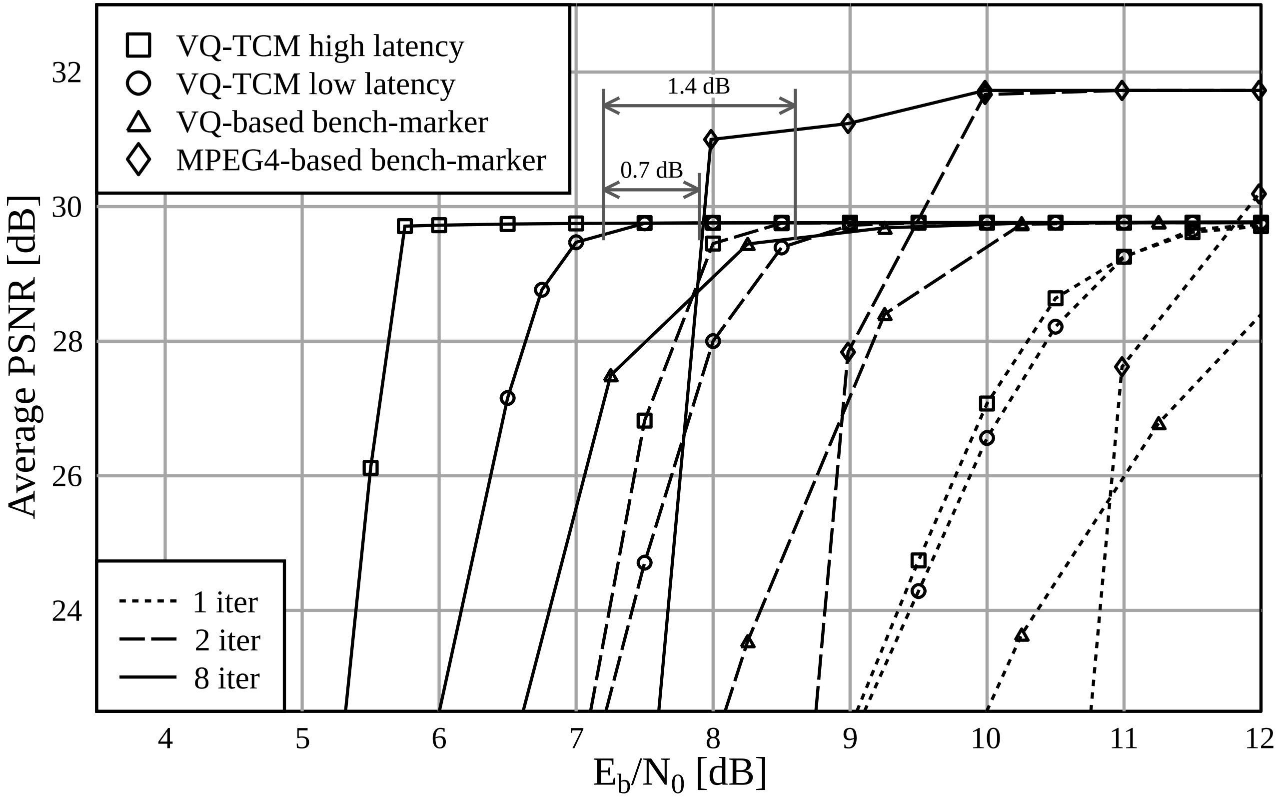


Results

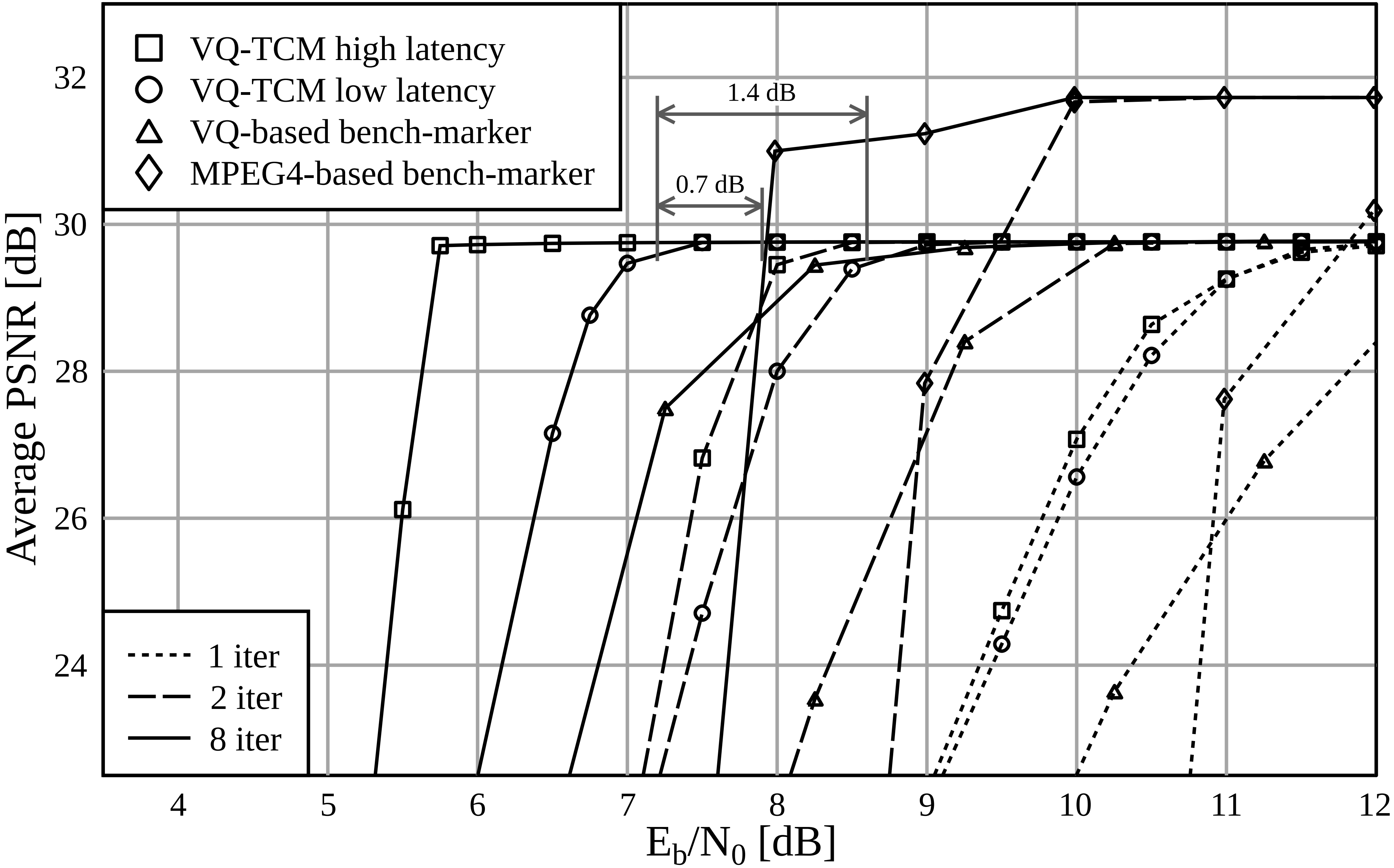

I_E^1, I_A^2
I_A^1, I_E^2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Inverted VQ EXIT characteristic
TCM EXIT characteristic:
E_b/N_0 = 4 dB
E_b/N_0 = 5.25 dB
E_b/N_0 = 6 dB
E_b/N_0 = 7 dB
E_b/N_0 = 8 dB
E_b/N_0 = 9 dB
E_b/N_0 = 10 dB
E_b/N_0 = 11 dB
EXIT trajectory:
○ Low latency E_b/N_0 = 6 dB
□ High latency E_b/N_0 = 6 dB

PSNR performance

- VQ- and MPEG4-based bench-markers employ iterative channel decoding, have a 0.1s latency and have the same computational complexity as our approach.



Results


□ VQ-TCM high latency
○ VQ-TCM low latency
△ VQ-based bench-marker
◇ MPEG4-based bench-marker
1 iter
2 iter
8 iter
1.4 dB
0.7 dB
Average PSNR [dB]
32
30
28
26
24
4
5
6
7
8
9
10
11
12
E_b/N_0 [dB]

Conclusions and Future Work

- A trellis structure describes the complete set of VQ code constraints.
- Optimal MMSE VQ encoding is achieved by employing the Viterbi algorithm.
- BCJR VQ decoding exploits VQ code constraints to achieve FEC, to guarantee the recovery of valid video information and to allow MMSE soft reconstruction.
- Iterative decoding convergence to an infinitesimally low decoding error is possible within 1.29 dB of the channel capacity limit.
- Future work will consider the application of the proposed method to standard video codecs, such as H.264/AVC.