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.
Introduction

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.
System Overview

Transmitter

Frame differencer

$\hat{f}_n$

$\hat{f}_{n-1}$

Reconstruction buffer

$\hat{f}_n$

$\hat{e}$

Frame difference decomposer

$M$ VQ encoders

$e$

$e^1$ $e^M$

$\hat{e}^1$ $\hat{e}^M$

Frame difference recomposer

$\hat{f}_n$ $\hat{f}_{n-1}$

Bit interleaver

$u$

$u^1$ $u^M$

TCM encoder

$x$

VTC Spring 2006
Dallas
University of Southampton
System Overview

\[ f_n \]

\[ \hat{f}_{n-1} \]

\[ e \]

\[ e^1 \]

\[ e^M \]

\[ u^1 \]

\[ u^M \]

\[ \hat{e} \]

\[ \hat{e}^1 \]

\[ \hat{e}^M \]

Frame differencer

Frame difference decomposer (M sub-frames)

Frame difference recomposer

Reconstruction buffer

\[ \hat{f}_n \]

\[ x \]

\[ TCM \text{ encoder} \]
The TCM decoder and the VQ decoders iteratively exchange mutually extrinsic Log Likelihood Ratio (LLR) soft information.

This extrinsic information is exploited as \textit{a priori} information during decoding.

The \textit{a priori} information is subtracted from the resultant \textit{a posteriori} information to obtain more reliable extrinsic information for use in the next iteration.
System Overview

\[ L^1_p(u) = L^2_e(u) + L^1_e(u) \]

\[ L^1_a(u) = L^2_e(u) \]

\[ L^2_a(u) = L^1_e(u) \]

\[ L^2_p(u) = L^2_e(u) + L^1_e(u) \]

\[ L^2_e(u) = L^1_e(u) \]

\[ \tilde{e}_1 \]

\[ \tilde{e}_M \]

Frame difference recomposer

\[ \tilde{f}_n \]

\[ \tilde{f}_{n-1} \]

Reconstruction buffer

LLR partitioner

(M sub-frames)

LLR concatenator

\[ \tilde{e} \]

LLR deinterleaver

TCM decoder

(1)

(2)
Trellis-Based Vector Quantization

Frame Difference Decomposition

- The Frame Difference (FD) $e$ is decomposed into $M$ number of sub-frames $[e^1 \ldots e^M]$ for the sake of reducing trellis-based VQ complexity.
- Each FD sub-frame $e^m$, $m \in [1 \ldots M]$, comprises $J$ number of video blocks $[e^m_1 \ldots e^m_J]$ chosen as one Macro Block (MB) from each MB group.
- This ensures that the sub-frames have similar characteristics.
- For example:

```
Macro-block grouping boundaries. Each macro-block group comprises
$M = 33$ macro-blocks.
Video block, comprising
$(8 \times 8)$ pixels.
Macro-block, comprising
$J^{MB} = 4$ video blocks.
FD sub-frame $e^m$, comprising
$J = 12$ video blocks.
FD $e$, 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 $(8 \times 8)$ pixels.

Macro-block, comprising $J_{\text{MB}} = 4$ video blocks.

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

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

<table>
<thead>
<tr>
<th>Index $k$</th>
<th>Video blocks $(J^k_x \times J^k_y) = J^k$</th>
<th>Bits $I^k$</th>
<th>RVLC code $RVLC^k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(2 \times 2) = 4$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$(1 \times 2) = 2$</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>$(1 \times 1) = 1$</td>
<td>3</td>
<td>101</td>
</tr>
<tr>
<td>4</td>
<td>$(1 \times 1) = 1$</td>
<td>4</td>
<td>1001</td>
</tr>
<tr>
<td>5</td>
<td>$(1 \times 1) = 1$</td>
<td>5</td>
<td>10001</td>
</tr>
</tbody>
</table>

VQ tiles: $VQ^1, VQ^2, VQ^3, VQ^4, VQ^5$
**Trellis-Based Vector Quantization**

<table>
<thead>
<tr>
<th>Index $k$</th>
<th>Video blocks $J^k_x \times J^k_y = J^k$</th>
<th>Bits $I^k$</th>
<th>RVLC code $RVLC^k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(2 \times 2) = 4$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$(1 \times 2) = 2$</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>$(1 \times 1) = 1$</td>
<td>3</td>
<td>101</td>
</tr>
<tr>
<td>4</td>
<td>$(1 \times 1) = 1$</td>
<td>4</td>
<td>1001</td>
</tr>
<tr>
<td>5</td>
<td>$(1 \times 1) = 1$</td>
<td>5</td>
<td>10001</td>
</tr>
</tbody>
</table>

VQ tiles:

- VQ$^1$
- VQ$^2$
- VQ$^3$
- VQ$^4$
- VQ$^5$

Video block, comprising $(8 \times 8)$ pixels.
Vector Quantization Code Constraints

- Each \( J \) video block FD sub-frame \( e^m \) must be represented by an exact tessellation of VQ tiles from the VQ codebook.
- The corresponding transmission sub-frame \( u^m \) 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:

\[
e^m = \begin{bmatrix} e_1^m & e_2^m & e_3^m & e_4^m & e_5^m & e_6^m & e_7^m & e_8^m & e_9^m & e_{10}^m & e_{11}^m & e_{12}^m \end{bmatrix}
\]

\[
u^m = \{1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1\}
\]

Transmission sub-frame, comprising \( I = 17 \) bits.
Trellis-Based Vector Quantization

Video block, comprising (8 × 8) pixels.

Macро-block, comprising $J_{MB} = 4$ video blocks.

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

$\hat{e}^m = e_1^m e_2^m e_3^m e_4^m e_5^m e_6^m e_7^m e_8^m e_9^m e_{10}^m e_{11}^m e_{12}^m$

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

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

<table>
<thead>
<tr>
<th>$T$</th>
<th>$k^T$</th>
<th>$J_{k^T}$</th>
<th>$I_{k^T}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_a$</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>$T_b$</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>$T_c$</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$T_d$</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$T_e$</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>$T_f$</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
Trellis-Based Vector Quantization

Vector Quantization Trellis Structure

- Each transition \( T \) in the trellis represents a possible application of the \( J^kT \)-block VQ tile \( \text{VQ}^kT \) and the corresponding \( I^kT \)-bit RVLC \( \text{RVLC}^kT \).

- The trellis represents every possible \( J \)-block FD sub-frame \( \hat{e}^m \) and \( I \)-bit transmission sub-frame \( u^m \) combination.

- For example:
Trellis-Based Vector Quantization

FD sub-frame $\hat{e}_m$, comprising $J = 12$ video blocks

Transmission sub-frame $u^m$, comprising $I = 17$ bits
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 \( a \text{ priori} \) soft information \( L_{a}^{2}(u^{m}) \).
- \( A \ text{ posteriori} \) soft information \( L_{p}^{2}(u^{m}) \) is obtained by considering the \( a \text{ posteriori} \) probabilities of transitions on vertical cross sections through the trellis.
- Similarly, a MMSE soft FD sub-frame reconstruction \( \tilde{e}^{m} \) is obtained by considering horizontal cross sections through the trellis.
Simulation parameters

- 100 frames of ‘Lab’ QCIF (176 × 144)-pixel 10 fps video sequence
- \( M = 33, J = 12, I = 45, K = 512, 14.85 \text{ kbps} \)
- 3/4-rate TCM using 16QAM modulation
- Bandwidth efficiency of \( \eta = 2.00 \text{ bit/s/Hz} \)
- \( E_b/N_0 = 3.96 \text{ dB at Rayleigh fading channel capacity limit} \)
- Two schemes: 0.1s latency (1485 bit interleaver length) and 5.0s latency (74250 bit interleaver length)
Results

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 $E_b/N_0 > 5.25 \, \text{dB}$, just 1.29 dB from the channel capacity limit.
Results

TCM EXIT characteristic:

- \( E_b/N_0 = 4 \text{ dB} \)
- \( E_b/N_0 = 5.25 \text{ dB} \)
- \( E_b/N_0 = 6 \text{ dB} \)
- \( E_b/N_0 = 7 \text{ dB} \)
- \( E_b/N_0 = 8 \text{ dB} \)
- \( E_b/N_0 = 9 \text{ dB} \)
- \( E_b/N_0 = 10 \text{ dB} \)
- \( E_b/N_0 = 11 \text{ dB} \)

EXIT trajectory:

- Low latency \( E_b/N_0 = 6 \text{ dB} \)
- High latency \( E_b/N_0 = 6 \text{ dB} \)
Results

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.

![PSNR Performance Graph]

- VQ-TCM high latency
- VQ-TCM low latency
- VQ-based bench-marker
- MPEG4-based bench-marker

Average PSNR [dB]

E_b/N_0 [dB]

1 iter
2 iter
8 iter

1.4 dB
0.7 dB
Results

- VQ-TCM high latency
- VQ-TCM low latency
- VQ-based benchmarker
- MPEG4-based benchmarker

Average PSNR [dB]

E_b/N_0 [dB]

1.4 dB
0.7 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.