• New tool for video coding
• Original contribution \(\rightarrow\) lattices embedding
• Design of a complete vector quantizer
  ▶ multistages quantizing method
  ▶ determination of the optimal lattice
  ▶ labeling of the codebook points
  ▶ processing of the outlying source vectors
  ▶ bit allocation

**Perspectives**

• Progressive image coding
• Codebook updating (adaptive coding)
Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion
### Image sequences coding

<table>
<thead>
<tr>
<th>image sequence</th>
<th>image number</th>
<th>PSNR [dB]</th>
<th>entropy [bpp]</th>
<th>maximal time of encoding [s/image]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salesman</td>
<td>200</td>
<td>33.86</td>
<td>0.238</td>
<td>1.5</td>
</tr>
<tr>
<td>MissAmerica</td>
<td>107</td>
<td>39.38</td>
<td>0.064</td>
<td>1.3</td>
</tr>
</tbody>
</table>
Region-based coder

- Very low bit rate
- Motion estimation
  → polygonal shapes (Nzomigni95, Pateux96)
- Dyadic wavelet transform (Mallat89, Daubechies88)
- Multiresolution codebook (Antonini91)
- Bit allocation → threshold 0.2 bpp [final rate 0.175 bpp]
Coding of the image sequence “MissAmerica”
Coding of the image sequence “Salesman”
Coding of the image sequence “MissAmerica”

![Graph showing rate (bpp) vs. image number for different subbands and gains (dB) vs. image number for PSNR, Gp, and Gq.](image-url)
Coding of the image sequence “Salesman”
### Image sequences coding

<table>
<thead>
<tr>
<th>image sequence</th>
<th>number of images</th>
<th>PSNR [dB]</th>
<th>entropy [bpp]</th>
<th>maximal time of encoding [s/image]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salesman</td>
<td>200</td>
<td>39.07</td>
<td>0.201</td>
<td>2</td>
</tr>
<tr>
<td>MissAmerica</td>
<td>107</td>
<td>38.98</td>
<td>0.162</td>
<td>2</td>
</tr>
<tr>
<td>Claire</td>
<td>200</td>
<td>38.03</td>
<td>0.157</td>
<td>2</td>
</tr>
</tbody>
</table>
Codebooks design (before bit allocation)

Image sequence “Salesman”

<table>
<thead>
<tr>
<th>subband label</th>
<th>training sequence size</th>
<th>cpu time [s]</th>
<th>number of code vectors</th>
<th>entropy [bpp]</th>
<th>training ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5 images</td>
<td>6.75</td>
<td>43</td>
<td>0.992</td>
<td>884</td>
</tr>
<tr>
<td>B</td>
<td>10 images</td>
<td>13.95</td>
<td>358</td>
<td>0.427</td>
<td>186</td>
</tr>
<tr>
<td>C</td>
<td>10 images</td>
<td>14.13</td>
<td>434</td>
<td>0.496</td>
<td>153</td>
</tr>
<tr>
<td>D</td>
<td>148 images</td>
<td>48.50</td>
<td>1248</td>
<td>0.087</td>
<td>189</td>
</tr>
</tbody>
</table>

Bit allocation

Threshold 0.2 bpp → final rate 0.188 bpp

<table>
<thead>
<tr>
<th>subband label</th>
<th>number of code vectors</th>
<th>entropy [bpp]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>19</td>
<td>0.416</td>
</tr>
<tr>
<td>B</td>
<td>64</td>
<td>0.111</td>
</tr>
<tr>
<td>C</td>
<td>108</td>
<td>0.134</td>
</tr>
<tr>
<td>D</td>
<td>1234</td>
<td>0.086</td>
</tr>
</tbody>
</table>
MPEG-based coder

- Motion estimation → “block matching”
- DCT 2x2, intra-band configuration
- Codebooks

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th></th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>bpp</td>
<td>0.5</td>
<td></td>
<td>bpp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th></th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bpp</td>
<td>bpp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• 4 steps
  ▶ training sequences → open loop coder
  ▶ codebooks design
  ▶ bit allocation
  ▶ image sequence coding → closed loop coder

• Formulae
  ▶ $PSNR = 10 \cdot \log_{10} \frac{255^2}{\left(\frac{1}{N_x N_y} \cdot d(e, e_q)\right)}$
  ▶ prediction gain $G_p = 10 \cdot \log_{10} \frac{255^2}{\left(\frac{1}{N_x N_y} \cdot d(e)\right)}$
  ▶ quantization gain $G_q = 10 \cdot \log_{10} \frac{d(e)}{d(e, e_q)}$
  ▶ codebook entropy

• QCIF image sequence

• Sparc-Station 5 [110 Mhz] computer
Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion
Conclusion

• Partition of the space according to :
  ▶ the source distribution
  ▶ the rate vs. distortion tradeoff
• Simple labeling method
• $\mathbb{Z}^k \rightarrow$ simple processing for the outlying source vectors
• Fast quantizing $\rightarrow$ complexity $O(h.k)$
• Bit allocation
Optimal bit allocation

Singular value of $\lambda$ (Shoham88)

- From a first point of the hull, by successive calculations of singular values $\rightarrow$ global convex hull
- $\lambda \rightarrow$ search the BFOS criterion with the maximal value among the subbands
Lagrange multiplier

- \( N^M \) combination of quantizers → complex
- \( \min(D + \lambda R) \iff \sum_{j=0}^{M-1} \min(d_{j,i} + \lambda r_{j,i}) \)

- Algorithm (general form)
  1. convex hull of each subband → directly obtained when growing the tree
  2. global convex hull → search \( C \)
Bit allocation

- $M$ subbands
- $\min D$ subject to $R \leq R_d$
- Lagrangian methods (Shoham88, Ramchandran93)
  - $N$ quantizers $q_{j,i}$ for each subband $j$
    - different configurations of the tree
  - for one combination of quantizers
    \[
    D = \frac{1}{M} \sum_{j=0}^{M-1} d_{j,i} \quad \text{and} \quad R = \frac{1}{M} \sum_{j=0}^{M-1} r_{j,i}
    \]
  - cluster of points $\to$ search on the convex hull
Processing of a source vector whose energy is too large

- **Detection**
  - $F = \frac{b_{\text{min}} \cdot \rho}{\sqrt{\mathcal{E}_{\text{max}}}}$
  - $L_\infty$ norm of a vector $\mathbf{u}$ within the cube:
    \[ L_\infty(\mathbf{u}) = \max_{i=1,\ldots,k} |u_i| \leq (b_{\text{min}} \times \rho) \]
  - $L_\infty$ norm of a vector $\mathbf{x}$ which can be quantized:
    \[ L_\infty(\mathbf{x}) = \max_{i=1,\ldots,k} |x_i| \leq \sqrt{\mathcal{E}_{\text{max}}} \]

- **Processing**
  - If $|x_i|_{i=1,\ldots,k} > \sqrt{\mathcal{E}_{\text{max}}}$ \implies $x_i = \text{sign}(x_i) \cdot \sqrt{\mathcal{E}_{\text{max}}}$
Labeling of the codebook vectors

1. Look-up table → index of the truncated lattice points
2. Scan the tree in order to number the nodes
3. Re-scan the tree and store for each node:
   - the children numbers
   - the father number
   - the index of the corresponding lattice point
   - the entropy code word (for the leaves)
Optimal lattice

\[ \Rightarrow Z^k \text{ is optimal} \]
Unbalanced tree design (greedy approach)

- Partition adapted to the source distribution
- “Dead zone”
Quantization scheme

\[ F = b \cdot \rho / \sqrt{\mathcal{E}_{max}} \]

- A tree-structured codebook
- Progressive splitting \( \rightarrow b = b_{min} = 3 \)
Hierarchy of embedded lattices
• Packing radius of the support lattice: $\rho$

• Packing radius of the dilated lattice:

$$b \cdot \rho \text{ avec } b \in \mathbb{R} \quad / \quad b > 1$$

$\Rightarrow$ $b = 2n + 1 \quad / \quad n \in \mathbb{N}^*$
Embedded Lattices

- Support Lattices $\mathbb{Z}^k$, $D_k$, $E_8$, $\Lambda_{16}$ → fastest quantizing algorithms

- Embedding:
  by contracting it, embed a truncated lattice in its Voronoï cell

- Optimal embedding:
  the rescaled truncated lattice covers exactly the Voronoï cell

- Sub-optimal embedding:
  the rescaled truncated lattice covers maximally the Voronoï cell
Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion
Conclusion

- Pruning algorithm
  - global approach
  - storage of the complete tree
- Greedy algorithm
  - local approach
  - limited storage
Greedy approach

\[ (d(S^0), r(S^0)) \]

\[ \Delta d \]

\[ \Delta r \]

\[ \implies \text{Splitting of the leaf for which } \lambda(n_i) \text{ is maximal} \]
Pruning principle

\( \rightarrow \) Pruning of the branch for which \( \lambda(n_i) \) is minimal
Tree pruning

- BFOS algorithm (Breiman84)
  1. complete tree $\mathcal{T}$
  2. successive pruning
- Characterisation of each branch $S_{n_i}$
  - increase in distortion if $S_{n_i}$ is removed $\Delta d(S_{n_i})$
  - decrease in rate if $S_{n_i}$ is removed $\Delta r(S_{n_i})$
  - BFOS criterion $\lambda(n_i) = \Delta d(S_{n_i})/\Delta r(S_{n_i})$
Training

Characterisation of each node \( n_i \)

- Probability of reaching \( n_i \)
  \[
  P(n_i) = \frac{\text{card}(C_{n_i})}{\text{card}(SA)}
  \]

- Average distortion
  \[
  d(n_i) = \frac{1}{\text{card}(C_{n_i})} \cdot \sum_{x \in C_{n_i}, x \in SA} \| x - y_{n_i} \|^2
  \]

- Entropy code length
  \[
  r(n_i) = -\log_2 P(n_i)
  \]
• Encoding
  ▶ complexity $O(\log_B L)$
  ▶ tree storage

• Decoding
  ▶ leaves
  ▶ progressive reconstruction

• Unbalanced tree $\rightarrow$ variable rate
  ▶ pruning approach (Breiman84, Chou89)
  ▶ greedy approach (Makhoul85, Riskin91)
Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion
Conclusion

- Advantage
  - fast quantization
  - predefined codebook
- Drawback → method for simple sources

⇒ TSLVQ : hierarchical packing of embedded truncated lattices
Labeling of the lattice points

- Index calculation $\rightarrow$ product code (Lamblin88, Moureaux94, Onno95)
  - sub-index for the sphere energy
  - sub-index for the point position
Projection within a sphere

- Sphere radius $\sqrt{E_t}$
- Training sequence $SA = \{x_j = (x_1, \ldots, x_k)^T \mid j = 0, 1, 2, \ldots\}$
- Vector energy $E(x) = L_2(x)$
- $E_{\text{max}} = \max_{x} \{E(x) \mid x \in SA\}$
- Scaling factor $F = \sqrt{E_t/E_{\text{max}}}$

- Real source $\rightarrow$ vectors with energy greater than $E_{\text{max}}$ are processed separately
Coding scheme

- **Lattice Truncation**
  - **Shape**
    - i.i.d Gaussian source $\rightarrow$ sphere $(L_2)$
    - i.i.d Laplacian source $\rightarrow$ pyramid $(L_1)$
    - correlated GG source $\rightarrow$ ellipse (ponderated $L_2$)
  - Truncation energy (Fisher86) $E_t$ $\rightarrow$ points number

- **Source normalisation**
  - before quantization
  - scaling factor
Characteristics

- Packing radius $\rho$
  \[
  \begin{array}{c}
  \bullet \\
  \bullet \\
  \bullet \\
  \bullet \\
  \bullet \\
  \circ \\
  \bullet \\
  \bullet \\
  \bullet \\
  \bullet \\
  \end{array}
  \]

- Series Theta, Nu (Gaidon93), modified Theta (Moureaux94)
  \[
  \begin{array}{c}
  \bullet \\
  \bullet \\
  \bullet \\
  \bullet \\
  \circ \\
  \bullet \\
  \bullet \\
  \bullet \\
  \bullet \\
  \end{array}
  \]

- Best quantizing lattices: $A_2$, $D_4$, $E_8$, $\Lambda_{16}$

- Fast quantizing lattices (Conway and Sloane83): $Z^k$, $D_k$, $E_8$, $\Lambda_{16}$
  $\iff$ complexity $O(k)$
Lattices $\Lambda$

- Regular arrangement of identical spheres $\rightarrow$ spheres centers
  $0 \rightarrow$ origin
- $Z^k = \{y = (y_1, y_2, \ldots, y_k)^T \mid y_i \in \mathbb{Z}\}$
- $D_k = \{y = (y_1, y_2, \ldots, y_k)^T \mid y \in Z^k, \sum_{i=1}^{k} y_i = 0 \mod(2)\}$
Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion
(nearly) Optimal VQ

- Training
- Encoding $\rightarrow$ exhaustive research $O(L)$
Performance evaluation

• Rate [bpp]
  ▶ Rate constrained : $R = \frac{1}{k} \cdot \log_2 L$
  ▶ Entropy constrained : $R \simeq H(D)$

• Distorsion

$$D = \frac{1}{k} \sum_{i=1}^{L} \int_{C_i} L_2(x, y_i) \cdot p_X(x) \, dx$$
- Coding

\[ x \rightarrow \text{Encoding} \rightarrow i \rightarrow \text{Channel } i \rightarrow \text{Decoding} \rightarrow y_i \]

- Encoding, "Nearest Neighbour" rule

\[ C_i = \{ x \in \mathbb{R}^k / Q(x) = y_i, \text{ s.t. } d(x, y_i) \leq d(x, y_j), \forall j \neq i \} \]
Principe

- Vector Quantizer, dimension $k$, size $L$

  $Q : \mathbb{R}^k \rightarrow \mathcal{D}$

  $x \mapsto Q(x) = y_i$

  $\mathcal{D} = \{y_i \in \mathbb{R}^k / i = 1, 2, ..., L\}$

- $\mathbb{R}^k$ partition into $L$ Voronoï cells

  $C_i = \{x \in \mathbb{R}^k / Q(x) = y_i\}$
Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion
- Hybrid approach

\[ I(t-1) \rightarrow I(t) \rightarrow \text{Motion Estimation} \rightarrow \text{Motion Compensation} \rightarrow \text{Transform} \rightarrow \text{TSLVQ} \rightarrow \text{Entropic Coder} \rightarrow \text{index} \rightarrow \text{predicted I} \rightarrow \text{Motion Information} \rightarrow \text{quantized errors} \rightarrow \text{reconstructed I} \rightarrow \text{inv. Transform} \rightarrow \text{inv. TSLVQ} \rightarrow \text{reconstructed I} \]
- Standards design (H261, MPEG1&2, MPEG4)
- Image coding scheme

![Diagram of image coding process]
Plan

1. **Context of the study**
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion
Tree-Structured Lattice Vector Quantization for the Compression of Digital Image Sequences

Vincent Ricordel

Tampere University of Technology
Signal Processing Laboratory