Pyramid Vector Quantization for Video Coding

Jean-Marc Valin
Daala Coding Party
Sep 2013
Motivations

- Pyramid vector quantization is a key technique used in Opus (both SILK and CELT parts)
- Investigate PVQ for a video codec (Daala)
- Potential advantages
  - Preserves energy (details) even when details are imperfect (instead of blurring)
  - Implicit activity masking
  - Better representation of coefficients
Gain-Shape Quantization

- Represent a vector as magnitude multiplied by unit-norm vector (radius + point on sphere)
  - Amount of texture vs exact details
- Code magnitude separately
  - Adjust resolution of the sphere based on the magnitude
Pyramid Vector Quantizer (PVQ)

- Place $K$ unit pulses in $N$ dimensions
  - Up to $N = 1024$ dimensions
- Normalize to unit norm ($L_2$)

$$S(N, K) = \left\{ \frac{y}{\|y\|}, \ y \in \mathbb{Z}^N : \sum_{i=0}^{N-1} |y_i| = K \right\}$$
Codebook for $N=3$ and different $K$

<table>
<thead>
<tr>
<th>$K$</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5.25</td>
</tr>
<tr>
<td>4</td>
<td>6.04</td>
</tr>
<tr>
<td>6</td>
<td>7.19</td>
</tr>
<tr>
<td>8</td>
<td>8.01</td>
</tr>
<tr>
<td>11</td>
<td>8.92</td>
</tr>
<tr>
<td>16</td>
<td>10.00</td>
</tr>
<tr>
<td>23</td>
<td>11.05</td>
</tr>
<tr>
<td>32</td>
<td>12.00</td>
</tr>
</tbody>
</table>
Distortion, $N$ and $K$

$$D = \frac{N^2}{(24K^2)}$$

Fewer pulses needed
PVQ vs Scalar Quantization

-6 dB/bit
Prediction

• Unlike CELT, we want to predict the vectors
• PVQ on the residual loses energy preservation
• Apply prediction in the normalized vector
  – Use Householder reflection to align prediction with one axis
  – Encode magnitude of the residual as an angle
2-D Projection

- Input
2-D Projection

- Input+prediction
2-D Projection

- Input+prediction
- Compute reflection plane
2-D Projection

- Input+prediction
- Compute reflection plane
- Apply reflection
2-D Projection

- Input+prediction
- Compute reflection plane
- Apply reflection
- Compute/code angle

\[ \theta \]
2-D Projection

- Input+prediction
- Compute reflection plane
- Apply reflection
- Compute/code angle
- Code other dimensions
Activity Masking

- Artefacts are easier to detect on flat areas than on textured areas.
  - Code unit-norm vector with a resolution that depends on the gain (texture).

- Code companded gain $g_c = g^y$
  - Implicit activity masking built into the bitstream.
Open Questions

- How to split into bands
- Avoid wasting bits on still video
- Quantization matrix
- Take advantage of correlation/prediction in gain and angle
- Rate-Distortion Optimization
  - Fast RDO PVQ search?