MELPe Speech Compression using Bit-plane based Arithmetic Coding

Chris Parrish
University of Washington
2004
Overview

- MELPe Speech Coding
  - LPC modeling
  - Vector Quantization of LSP
- Arithmetic Coding of LSP
  - DCT Transform
  - Bit Plane Quantization
  - Arithmetic Coding
- Results
- Future work and Improvements
MELPe

- Low bit-rate LPC based coder
  - 2.4 or 1.2 Kbps
  - D.o.D. Standard
  - Mixed pulse and noise excitation
  - Noise Pre-processing

Original

MELPe 2.4 Kbps
Linear Predictive Coding

- Analysis / Synthesis Technique over a window of speech (22.5 ms)
- Models vocal tract with a filter
- Excite filter with pulse train (voiced) or white noise (unvoiced)
Parameter Bit Allocation

Line Spectral Pairs

A form of prediction coefficient. Strong correlation, usually 10 parameters per frame

Excitation Signal

Voiced / Unvoiced decision, Pitch Period Estimation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Voiced</th>
<th>Unvoiced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Spectral Pairs</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Fourier Magnitude</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>Gain</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Pitch</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Bandpass Voicing</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>Aperiodic</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Error Protection</td>
<td>-</td>
<td>13</td>
</tr>
<tr>
<td>Sync</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TOTAL / Frame</td>
<td>54</td>
<td>54</td>
</tr>
</tbody>
</table>
Vector Quantization

- MELPe uses vector quantization to code LSP and Fourier magnitudes
- VQ Groups parameters to be coded into “vectors”

  Create a codebook of representative vectors
  Encode a vector by finding closest codebook entry
  Store / send codebook index

![Diagram showing vector quantization process]

Source Data → Group into Vectors → Find Closest Vector in Codebook → Store Index

"lossy"
LSP Example

Example LSP parameters:

- LSP 10
- LSP 9
- LSP 8
- LSP 7
- LSP 6
- LSP 5
- LSP 4
- LSP 3
- LSP 2
- LSP 1
2-D DCT of LSP

- Invertible (one to one, onto)
- Discrete Cosine Transform provides energy compaction
- Most of the coefficients are approximately zero
New Coder Diagram

- Group frames of LSP parameters into a block
- Transform block with 2-D DCT
- Quantize with bit plane coding
- Perform lossless compression on bit planes with arithmetic coder
Bit Planes

- A method to quantize the coefficients to be compressed
- Allows for a progressive encoder / decoder
- Allows a variable bit rate, more bits for hard to code frames.
- Truncation at a particular bit plane favors larger coefficients

<table>
<thead>
<tr>
<th>Coefficient 1</th>
<th>Coefficient 2</th>
<th>Coefficient 3</th>
<th>Coefficient 4</th>
<th>Coefficient 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 1 1 1 1 0</td>
<td>0 0 1 0 1 1 0 1</td>
<td>0 0 0 0 1 0 1 0</td>
<td>0 1 0 0 0 1 0 1</td>
<td>0 0 1 0 1 1 1 0</td>
</tr>
</tbody>
</table>
Bit Plane coding

Early bit planes have a biased distribution suggesting that resulting bit-stream can be coded further.
Arithmetic Coding

- Codes one bit at a time
- Relies on statistics to compress well
- Fewer bits for more probable symbols or sequences
- Statistics can adapt over time
Advantages of Arithmetic Coding with Bit-planes

- Our technique allows for progressive transmission, while MELPe currently is fixed.
- Does not require training, can adjust to characteristics of different languages.
- Arithmetic coding with bit-plane quantization has been shown better than VQ techniques in other coding applications.
Better Context

For good results, an arithmetic coder needs accurate statistics and contexts that bias the distribution of bits.

For example, distribution of bits in a bit plane for coefficients that are not yet significant is different than for those that are significant.
Current Results

Mean Square Error MELPe VQ vs ADCT

Avg
VQ: 0.0011335
ADCT: 0.0012087

Peak
VQ: 0.0099133
ADCT: 0.0047079

MELPe 2.4 Kbps

Current
Future work

- Find better contexts
  Current research coder has several groups of contexts for each bit plane:
  - insignificant, first refinement, refinement, sign
- Each block is relatively small
  - Need history from previous blocks to have good statistics
  - Modulate the history based on distance in time from current frame
- How to apply perceptual weighting to quantization in DCT domain?
Thanks!

- Eve Riskin
- Richard Ladner
- Amanda Askew
- Dane Barney