Multimedia Systems
Part 18

Mahdi Vasighi
www.iasbs.ac.ir/~vasighi

Department of Computer Science and Information Technology,
Institute for Advanced Studies in Basic Sciences, Zanjan, Iran
the predictor is based on $f_{n-1}$, $f_{n-2}$, ... . Therefore, the predictor must involve a memory.
Differential Pulse Code Modulation (DPCM) is exactly the same as Predictive Coding, except that it incorporates a quantizer step.

- can be uniform or non-uniform

- The original signal $f_n$,
- the predicted signal $\hat{f}_n$,
- the quantized, reconstructed signal $\tilde{f}_n$. 
Differential Pulse Code Modulation

How DPCM operates is to form the prediction $\hat{f}_n$, form an error $e_n$ by subtracting the $\hat{f}_n$ from the actual signal $f_n$, then quantize the error to a quantized version, $\tilde{e}_n$.

$$
\hat{f}_n = \text{function\_of}\ (\tilde{f}_{n-1}, \tilde{f}_{n-2}, \tilde{f}_{n-3}, \ldots)
$$

$$
e_n = f_n - \hat{f}_n
$$

$$
\tilde{e}_n = Q[e_n]
$$

transmit codeword($\tilde{e}_n$)

The main effect of the coder–decoder process is to produce reconstructed, quantized signal values

$$
\tilde{f}_n = \hat{f}_n + \tilde{e}_n
$$
Differential Pulse Code Modulation

The “distortion” is the average squared error and calculated as follows:

\[
\text{distortion} = \frac{\sum_{n=1}^{N} (\tilde{f}_n - f_n)^2}{N}
\]

One often sees diagrams of distortion versus the number of bit levels used.
Differential Pulse Code Modulation

- The predictor is always based on the reconstructed, quantized version of the signal.
- The box labeled "Symbol coder" in the block diagram simply means a Huffman Coder.
- We need to buffer previous values of $f\tilde{}$ to form the prediction.
• The predictor is always based on the reconstructed, quantized version of the signal
• The box labeled “Symbol coder” in the block diagram simply means a Huffman Coder
• we need to buffer previous values of $f^\sim$ to form the prediction
Differential Pulse Code Modulation

It helps us explicitly to understand the process of coding to look at actual numbers. Suppose we adopt a particular predictor as follows:

$$\hat{f}_n = \text{tru}_n\left(\frac{\tilde{f}_{n-1} + \tilde{f}_{n-2}}{2}\right)$$

$$e_n = f_n - \hat{f}_n$$

$$\hat{e}_n = Q[e_n]$$

Let us use the particular quantization scheme

$$\tilde{e}_n = Q[e_n] = 16 \times \text{floor \left[\frac{(255 + e_n)}{16}\right]} - 256 + 8$$
\[ \tilde{e}_n = Q[e_n] = 16 \times \text{fix}[(255 + e_n) /16] - 256 + 8 \]
Differential Pulse Code Modulation

As an example stream of signal values, consider the set of values:

\[
f_1 \ f_2 \ f_3 \ f_4 \ f_5
\]
\[
130 \ 150 \ 140 \ 200 \ 230
\]

\[
\hat{f}_n = \text{trunc} \ n \left( \frac{\tilde{f}_{n-1} + \tilde{f}_{n-2}}{2} \right)
\]

\[
e_n = f_n - \hat{f}_n
\]

\[
\hat{f} = [130, 130, 142, 144, 167]
\]

\[
e = [0, 20, -2, 56, 63]
\]

\[
\tilde{e} = [0, 24, -8, 56, 56]
\]

\[
\tilde{f} = [130, 154, 134, 200, 223]
\]

<table>
<thead>
<tr>
<th>(e_n) in range</th>
<th>Quantized to value</th>
</tr>
</thead>
<tbody>
<tr>
<td>-255..-240</td>
<td>-248</td>
</tr>
<tr>
<td>-239..-224</td>
<td>-232</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>-31..-16</td>
<td>-24</td>
</tr>
<tr>
<td>-15..0</td>
<td>-8</td>
</tr>
<tr>
<td>1..16</td>
<td>8</td>
</tr>
<tr>
<td>17..32</td>
<td>24</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>225..240</td>
<td>232</td>
</tr>
<tr>
<td>241..255</td>
<td>248</td>
</tr>
</tbody>
</table>
Differential Pulse Code Modulation

```matlab
predictor = [0 1];
partition = [-1:.1:.9];
codebook = [-1:.1:1];

t = [0:pi/50:4*pi];
x = sin(2*t);
encodedx=dpcmenco(x,codebook,partition,predictor);
decodedx=dpcmdeco(encodedx,codebook,predictor);
```
Delta Modulation (DM), a much-simplified version of DPCM often used as a quick analog-to-digital converter.

The idea in DM is to use only a single quantized error value, either positive or negative. (1-bit coder)

\[
\begin{align*}
\hat{f}_n &= \hat{f}_{n-1} \\
e_n &= f_n - \hat{f}_n = f_n - \hat{f}_{n-1} \\
\tilde{e}_n &= \begin{cases} 
+k & \text{if } e_n > 0, \text{ where } k \text{ is a constant} \\
-k & \text{otherwise,} 
\end{cases} \\
\hat{f}_n &= \hat{f}_n + \tilde{e}_n
\end{align*}
\]

prediction simply involves a delay.
let’s consider actual numbers. Suppose signal values are as follows:

\[
\begin{align*}
    f_1 &= 10, \quad e_1 = 11 - 10 = 1, \quad \tilde{e}_1 = 4, \quad \hat{f}_1 &= 10 + 4 = 14 \\
    f_2 &= 14, \quad e_2 = 13 - 14 = -1, \quad \tilde{e}_2 = -4, \quad \hat{f}_2 &= 14 - 4 = 10 \\
    f_3 &= 10, \quad e_3 = 15 - 10 = 5, \quad \tilde{e}_3 = 4, \quad \hat{f}_3 &= 10 + 4 = 14
\end{align*}
\]

We also define an exact reconstructed value \( \tilde{f}_1 = f_1 = 10 \) using a step value \( k = 4 \)
Delta Modulation

- Signal amplitude
- Analog input
- Staircase function
- Step size \( \delta \)
- Slope overload noise
- Quantizing noise
- Sampling time \( T_s \)
- Delta modulation output

Delta Modulation