USING CONTEXT DEPENDENT DISTRIBUTIONS FOR CODING PREDICTION RESIDUALS OF COMPANDED AUDIO SIGNALS

Ioan Tabus, Florin Ghido
Department of Signal Processing
Tampere University of Technology
ioan.tabus@tut.fi, florin.ghido@tut.fi

Adriana Vasilache
Nokia Research Center
Tampere
adriana.vasilache@nokia.com

ABSTRACT

We propose a context conditioning scheme for encoding the prediction residuals when compressing files containing companded signals. Our scheme encompasses decompanding of the signals, performing linear prediction in the decompanded domain, and then companding back the predicted value into a companded prediction (CP) value, which will differ from the true companded value by an amount called companded prediction residual (CPR). The proposed context conditioning scheme for encoding the CPR, uses a probability distribution conditional on a context made up of two quantities: (1) the predicted value and (2) a scale parameter of the background probability distribution function assumed for the decompanded domain residuals. Various context building schemes and various storing strategies can be used to obtain the necessary conditional coding distribution of the CPR, to be used with an arithmetic coder or range coder. The implementation in fixed point precision can be done very efficiently and with very low memory requirements.

Index Terms— companding transforms, context modelling, lossless audio compression, G.711

1. INTRODUCTION

Companding of signals is a widespread technique for getting a more compact representation, typically when the necessary precision is not the same over the dynamic range of the measured signal. One important case is the speech signal transmitted in telephony, where a quasi-logarithmic companding function is applied before quantization, in order to represent the speech using 8 bits. We address mainly the case of speech signals in here, although the presented method is of interest for compressing any type of companded signals. Recently there was considerable interest in providing means for further compression of the G.711 [2] bitstreams, which are still a major part of narrowband PTSN/GSTN telephony traffic, resulting in a standardization process, G.711-LLC [4], for providing efficient encoding of G.711 bitstream while maintaining bit-exact reconstructions, with a number of possible voice over packet applications. For the case of these particular signals, our method will improve the compression of the solution selected in [4], if the applications will allow just a slightly higher computational complexity.

The compression obtained by G.711 is uniform, but being such a simplistic memoryless transform, the companded signal is still very redundant, since the audio signals are usually highly predictable due to the strong correlation between samples. To further compress the signal, the predictability of the signal should be exploited. A highly predictable signal in the linear domain is less predictable in the companded domain [1]. The compression scheme ought then to be constructed from a sequence of three stages [1]: a) decompanding, b) linear prediction, c) companding the predicted value, and d) encoding the companded domain residuals. In this paper we start from the above straightforward scenario and propose a new encoding scheme, which makes use of the remarkable structure of the probability distribution function \( P(\varepsilon_y) \) of the residuals \( \varepsilon_y \) in the companded domain. We derive the conditional pdf \( P(\varepsilon_y|\hat{y}) \) of the residuals in the companded domain, conditional on the value of the predicted value \( \hat{y} \) in the companded domain, based on an assumed parameterized pdf \( P(\varepsilon_x;\sigma_{\varepsilon_x}^2) \) of the residuals in the decompanded domain \( \varepsilon_x \) (e.g., having the variance \( \sigma_{\varepsilon_x}^2 \) as one parameter). Such conditioning is the key for achieving a very efficient encoding scheme for the residuals.

However, the most direct implementation requires a large memory space for storing one conditional distribution for each value of the possible companded prediction values (e.g., \( \hat{y} \in \{-128,\ldots,127\} \)) and for different values of the variance \( \sigma_{\varepsilon_x}^2 \). Therefore we propose a more refined scheme for parametric generation of the involved \( P(\varepsilon_y|\hat{y}) \), which will diminish considerably the necessary memory. Particularizing the scheme when \( P(\varepsilon_x;\sigma_{\varepsilon_x}^2) \) is a Laplacian distribution will lead to very effective and computationally attractive encoding scheme for the companded domain residuals \( \varepsilon_y \). The relative gains obtained over a simplistic model which uses an independent identically distributed Laplacian assumption for the residuals \( P(\varepsilon_y) \) in the companded domain (as opposed to Laplacian hypothesis for \( P(\varepsilon_x;\sigma_{\varepsilon_x}^2) \), which we propose to use) are, on average, typically in the range of 1 to 3%, with a very low additional computational effort.

2. PREDICTION AND CODING IN LINEAR AND COMPANDED DOMAINS

2.1. The companding transform

The compression scheme discussed in this paper takes as input a companded file and produces in the output a file which is a lossless compressed version of the companded file. First we review the process by which a companded signal is produced from the original signal. Each value \( z \in \{-R_1,\ldots,R_2\} \) of the original signal is quantized to \( \varepsilon = Q(z) \in C \), by the nonuniform quantizer \( Q \) having the codebook \( C \). For each codeword \( c \in C \) in the codebook there is a one to one correspondence with the integers \(-M_1 \leq m \leq M_2\), and for simplicity we denote \( c_m = \hat{f}(m) \), \( m = f^{-1}(c_m) \), where the function \( \hat{f} \) is nonlinear (e.g. approximating an exponential function in the case of G.711). The quantizer \( Q \) has decision levels \( d_m \) as follows:

\[ Q(z) = \begin{cases} c_{-M_1} & \text{if } -R_1 < z \leq d_{-M_1} \\ c_m & \text{if } d_{m-1} < z \leq d_m \\ c_{M_2} & \text{if } d_{M_2-1} < z \leq d_{M_2} = R_2. \end{cases} \]

978-1-4244-4296-6/10/$25.00 ©2010 IEEE 4694 ICASSP 2010
Each companded value \( y \) is thus obtained from an original value \( z \) as \( y = f^{-1}(Q(z)) \).

### 2.2. Distributions of various prediction residuals

In the compression scheme we first apply the decompanding transform, to get from the available companded domain value \( y(i) \) the values in the linear domain \( x(i) = f(y(i)) \), which is in general different from the unavailable original value \( z(i) \). With the available values, we can predict at time \( i \) the current values in the linear and companded domain, respectively, as

\[
\hat{x}(i) = a_1 x(i-1) + \ldots + a_k x(i-k)
\]
\[
\hat{y}(i) = f^{-1}(Q(\hat{x}(i)));
\]

where \( \hat{x}(i) \) is the linear domain or background prediction and \( \hat{y}(i) \) is the companded domain prediction. The quantization can also be expressed as \( Q(\hat{x}(i)) = f(\hat{y}(i)) \) and we further denote the quantization error in the linear domain

\[
\epsilon_Q(i) = \hat{x}(i) - Q(\hat{x}(i)) = \hat{x}(i) - f(\hat{y}(i)).
\]

The prediction errors are defined as

\[
\epsilon_x(i) = x(i) - \hat{x}(i) = x(i) - (a_1 x(i-1) + \ldots + a_k x(i-k))
\]
\[
\epsilon_y(i) = \hat{y}(i) - y(i) = f^{-1}(Q(f(y(i))) + \epsilon_x(i) + \epsilon_Q(i)) - y(i). \tag{1}
\]

We denote now \( \epsilon'_x(i) = \epsilon_x(i) + \epsilon_Q(i) \). Due to the link in (1) we can connect the distributions of \( \epsilon'_x(i) \) and \( y(i) \), given \( \hat{y}(i) \) as follows:

\[
P(\epsilon_y \leq n_1 | \hat{y} = n_2) = P(\epsilon_x \leq \hat{y} - y) = P(\epsilon_x \leq n_1 - n_2) = P(Q(\epsilon_x + \epsilon'_Q(i)) \leq c_{n_1} - c_{n_2}).
\]

We have the equivalent inequalities \( Q(c_{n_1} + \epsilon'_Q(i)) \leq c_{n_1} + \epsilon'_x(i) \leftrightarrow c_{n_1} + \epsilon'_x(i) \leq d_{n_1 + n_2} \) and thus

\[
P(Q(c_{n_1} + \epsilon'_Q(i)) \leq c_{n_1} + n_2) = P(\epsilon'_x(i) \leq d_{n_1 + n_2} - c_{n_2}).
\]

Now, if \( n_1 = -M_1, P(\epsilon_y = n_1 | \hat{y} = n_2) = P(\epsilon'_x(i) \leq d_{n_1 + n_2} - c_{n_2}) \), while for \( n_1 > -M_1 \)

\[
P(\epsilon_y = n_1 | \hat{y} = n_2) = P(\epsilon'_x(i) \leq d_{n_1 + n_2} - c_{n_2}) \quad \text{and} \quad -P(\epsilon'_x(i) \leq d_{n_1 + n_2} - c_{n_2}). \tag{2}
\]

These conditional distributions are illustrated in Figures 1 and 2 for the case of Laplacian distribution \( P(\epsilon'_x(i); \sigma^2_{\epsilon'_x}) \) with variance \( \sigma^2_{\epsilon'_x} = 600^2 \). The distribution of \( \epsilon'_Q(i) = \epsilon_Q(i) + \epsilon_Q(i) \) can be assumed to have a convenient form, e.g., Laplacian distribution, which is the case we will analyze in the Section 2.3. We also treat in Section 2.6 the slightly more complex case of a Laplacian distribution for the errors \( \epsilon_x \).

### 2.3. Method 1: Assuming Laplacian distribution for \( \epsilon'_x(i) \)

We will replace the Laplacian cdf

\[
P(\epsilon'_x < a) = \begin{cases} \frac{1}{2} e^{-a/\beta} & \text{if } a \leq 0 \\ 1 - \frac{1}{2} e^{-a/\beta} & \text{if } a > 0 
\end{cases}
\]

in (2) and use the inequalities of the quantizer \( d_{n_1+n_2-1} < c_{n_1+n_2} < d_{n_1+n_2} \). We take separately two cases. The condition \( d_{n_1+n_2-1} > c_{n_2} \) is satisfied when \( n_1 \geq 1 \), for which

\[
P(\epsilon_y = n_1 | \hat{y} = n_2) = \frac{1}{2} e^{\epsilon_x/\beta} (e^{-d_{n_1+n_2-1}/\beta} - e^{-d_{n_1+n_2}/\beta}).
\]

The condition \( d_{n_1+n_2} < c_{n_2} \) is satisfied when \( n_1 \leq -1 \), for which

\[
P(\epsilon_y = n_1 | \hat{y} = n_2) = \frac{1}{2} e^{-c_{n_2}/\beta} (e^{d_{n_1+n_2}/\beta} - e^{d_{n_1+n_2-1}/\beta}).
\]

The strategy of coding for a given \( \beta \) and \( \hat{y} \) is as follows. Split first the possible values of \( \epsilon_y \) in three ways: \( \epsilon_y < 0, \epsilon_y = 0, \epsilon_y > 0 \) and compute the corresponding probabilities. For the positive residuals we obtain:

\[
P(\epsilon_y > 0 | \hat{y} = n_2) = \frac{1}{2} e^{\epsilon_x/\beta} (e^{-d_{n_1+n_2}/\beta} - e^{-d_{n_1+n_2-1}/\beta}).
\]

Similarly, for the negative residuals we use the coding probability:

\[
P(\epsilon_y < 0 | \hat{y} = n_2) = \frac{1}{2} e^{-c_{n_2}/\beta} e^{-d_{n_2}/\beta}.
\]

and, finally for the null residual the coding probability is given by

\[
P(\epsilon_y = 0 | \hat{y} = n_2) = \lambda_0 = 1 - \lambda_1 - \lambda_{-1}.
\]

Now, we can use the three probabilities \( \lambda_1, \lambda_0, \lambda_{-1} \) to encode which of the three events \( \epsilon_y < 0, \epsilon_y = 0, \epsilon_y > 0 \) holds true.

After transmitting this event, we can condition on it to get the coding distribution for \( \epsilon_y \). When \( \epsilon_y < 0 \) holds true, we have

\[
P(\epsilon_y = n_1 | \hat{y} = n_2; \epsilon_y < 0) = e^{-d_{n_1+n_2-1}/\beta} (e^{d_{n_1+n_2}/\beta} - e^{d_{n_1+n_2-1}/\beta}), \tag{3}
\]

while when \( \epsilon_y > 0 \) holds true, we have

\[
P(\epsilon_y = n_1 | \hat{y} = n_2; \epsilon_y > 0) = e^{d_{n_2}/\beta} (e^{-d_{n_1+n_2-1}/\beta} - e^{-d_{n_1+n_2}/\beta}). \tag{4}
\]

When using arithmetic coding, cumulative distributions are needed. Thus, we are going to define the cumulative distribution corresponding to (4) when ordering the values of \( n_1 \) starting from the largest possible, 127, and going down to \( n_1 = 1 \),

\[
P(\epsilon_y > n_1 | \hat{y} = n_2; \epsilon_y > 0) = e^{d_{n_2}/\beta} e^{-d_{n_1+n_2-1}/\beta}
\]

and similarly, when using a cumulative distribution for (3)

\[
P(\epsilon_y < n_1 | \hat{y} = n_2; \epsilon_y < 0) = e^{-d_{n_2-1}/\beta} e^{d_{n_1+n_2}/\beta}.
\]

### 2.4. Introducing constraints on the maximum codelength

In order to keep the resulting codelength under a reasonable bound, we decide to transmit the residuals by the conditional distribution only when the codelength for each of the residuals in the frame is less than a selected threshold, \( Th \), otherwise we use an alternative method, e.g., encoding by Golomb-Rice codes the residuals for the full frame.

Therefore we can remove from the distribution used in arithmetic coding all the values \( n_1 \) for which \( P(\epsilon_y = n_1 | \hat{y} = n_2) < Th \), since such removal simplifies the computations and reduces the need of stored values or the needed precision of the stored values.
2.5. Fixed point computations

In order to implement in the most economical way from the point of view of memory and computation costs, we will introduce a new evaluation algorithm, based on properly defined look-up tables, which will occupy only about 4 kbytes. Since $\epsilon_{\alpha} = 2^{\alpha \log(2)}$, let us denote $g(n_1, n_2) = \frac{d_{n_1+n_2-1}-d_{n_2}}{\log(2)} \Delta I(n_1, n_2) + f(n_1, n_2)$, where $I(n_1, n_2)$ is integer and the fractional part is $f(n_1, n_2) \in [0, 1]$. For $\epsilon_{\alpha} = 2^\Delta, 2^{-2\Delta}, \ldots, 2^{-N_{\text{max}} \Delta}$, the worst error of this approximation is $\max_{\epsilon \in (0, 1)} |T_{\text{round}}(z) - 2^{-\epsilon}| \approx 2^{-11.54}$. 

For a quick "inverse" of $T_i$, we also tabulate the $N_{\text{max}} = 2048$ values $2^{-\Delta}, 2^{-2\Delta}, \ldots, 2^{-N_{\text{max}} \Delta}$. The coding distribution which will be used at both encoder and decoder is defined in the following. First evaluate the cumulative distribution $\hat{P}(\epsilon \geq n_1, n_2; \beta; \epsilon > 0)$, introduced by

\[ g(n_1, n_2) = \frac{d_{n_1+n_2-1}-d_{n_2}}{\log(2)} \Delta I(n_1, n_2) + f(n_1, n_2) \]

\[ P(\epsilon \geq n_1, n_2; \beta; \epsilon > 0) = 2^{-g(n_1, n_2)} = 2^{-I(n_1, n_2) - \beta f(n_1, n_2)} \]

\[ \hat{P}(\epsilon \geq n_1, n_2; \beta; \epsilon > 0) = 2^{-I(n_1, n_2)} T_{\text{round}}(\frac{\epsilon n_1}{\epsilon}) \]

followed by the evaluation of $\hat{P}(\epsilon \geq n_1, n_2; \beta; \epsilon > 0)$, where first we denote $\frac{d_{n_1+n_2-1}-d_{n_2}}{\log(2)} \Delta I(n_1 + 1, n_2) + f(n_1 + 1, n_2)$ and get

\[ \hat{P}(\epsilon \geq n_1 + 1, n_2; \beta; \epsilon > 0) = 2^{-I(n_1 + 1, n_2)} T_{\text{round}}(\frac{\epsilon n_1 + \epsilon n_2}{\epsilon}) \]

When forming the bitstream with an arithmetic coder we need only the above cumulative distributions. The arithmetic decoder will use the inverse table $R_i$ and only one companding operation to decode the value of $n_1$, being thus extremely fast.

2.6. Method 2: Assuming Laplacian distribution for $\epsilon_x(i)$

We are interested in a second method, which evaluates more precisely $P(\epsilon_x^2(i) \leq d_{n_2} - d_{n_2-1})$ when the distribution of $\epsilon_x(i)$ is exponential and that of $\epsilon_q(i)$ is uniform in the interval $[d_{n_2} - d_{n_2-1}]$. The distribution of the sum $\epsilon_x(i) + \epsilon_q(i)$ is given by the convolution

\[ P(\epsilon_x(i) \leq a) = \frac{1}{D_{n_2}} \int_{-d_{n_2-1}}^{d_{n_2}} P(\epsilon_x(i) \leq a - u) du, \]

where we denoted by $D_{n_2} = d_{n_2} - d_{n_2-1}$ the length of the Voronoi cell of the quantizer $Q(i)$ centered at $n_2$ and thus

\[ P(\epsilon_x(i) \leq d_{n_2} - d_{n_2-1}) = \frac{1}{D_{n_2}} \int_{d_{n_2-1}}^{d_{n_2}} P(\epsilon_x(i) \leq d_{n_2} - d_{n_2-1}) du, \]

When $d_{n_2}$ is negative and for the case Laplacian distribution of $\epsilon_x(i)$, we have

\[ P(\epsilon_x(i) \leq d_{n_1} + n_2 - d_{n_2}) \]

\[ = \frac{\beta}{2D_{n_2}} \left( e^{-\frac{d_{n_2}^2 - d_{n_2}^2}{2\beta^2}} - e^{-\frac{d_{n_2}^2 - d_{n_2}^2}{2\beta^2}} \right) \]

and similarly

\[ P(\epsilon_x(i) \leq d_{n_1} + n_2 - 1 - d_{n_2}) \]

\[ = \frac{\beta}{2D_{n_2}} \left( e^{-\frac{d_{n_2}^2 - d_{n_2}^2}{2\beta^2}} - e^{-\frac{d_{n_2}^2 - d_{n_2}^2}{2\beta^2}} \right), \]

resulting in the expression valid for $\epsilon_x(i) < 0$

\[ P(\epsilon_y = n_1 | y = n_2) = \frac{\beta}{2D_{n_2}} \left( e^{-\frac{d_{n_1} + n_2 - d_{n_2}^2}{2\beta^2}} - e^{-\frac{d_{n_2}^2 - d_{n_2}^2}{2\beta^2}} \right) \]

Similar expressions can be obtained for the values $\epsilon_y > 0$, and $\epsilon_y = 0$, which we omit here for brevity. Using the derived distributions with an arithmetic coder will provide a straightforward context coding of the residuals.

3. EXPERIMENTAL RESULTS

In Table 1 we show results of four algorithms for encoding the residuals, with an arithmetic coder will provide a straightforward context coding of the residuals.
Table 2. Overall compression results. Values represent compression in percentage (higher is better); columns are frame size in samples and Total. Tests are A1 (A-law and μ-law) - clean speech, A2 (A-law and μ-law) - noisy speech, and B (μ-law) - recorded speech.

<table>
<thead>
<tr>
<th>Test / Frame</th>
<th>40</th>
<th>80</th>
<th>160</th>
<th>240</th>
<th>320</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 A ITU</td>
<td>63.92</td>
<td>66.77</td>
<td>68.71</td>
<td>69.17</td>
<td>69.34</td>
<td>67.58</td>
</tr>
<tr>
<td>A1 A NEW</td>
<td>65.84</td>
<td>68.00</td>
<td>69.39</td>
<td>69.66</td>
<td>69.73</td>
<td>68.52</td>
</tr>
<tr>
<td>A1 A impr.</td>
<td>1.92</td>
<td>1.23</td>
<td>0.68</td>
<td>0.49</td>
<td>0.39</td>
<td>0.94</td>
</tr>
<tr>
<td>A1 μ ITU</td>
<td>55.61</td>
<td>58.43</td>
<td>60.71</td>
<td>61.2</td>
<td>61.41</td>
<td>59.47</td>
</tr>
<tr>
<td>A1 μ NEW</td>
<td>58.10</td>
<td>60.05</td>
<td>61.71</td>
<td>61.97</td>
<td>62.02</td>
<td>60.77</td>
</tr>
<tr>
<td>A1 μ impr.</td>
<td>2.49</td>
<td>1.62</td>
<td>1.00</td>
<td>0.77</td>
<td>0.61</td>
<td>1.30</td>
</tr>
<tr>
<td>A2 A ITU</td>
<td>59.94</td>
<td>62.99</td>
<td>65.05</td>
<td>65.5</td>
<td>65.69</td>
<td>63.83</td>
</tr>
<tr>
<td>A2 A NEW</td>
<td>61.96</td>
<td>64.23</td>
<td>65.72</td>
<td>65.99</td>
<td>66.09</td>
<td>64.80</td>
</tr>
<tr>
<td>A2 A impr.</td>
<td>2.02</td>
<td>1.24</td>
<td>0.67</td>
<td>0.49</td>
<td>0.40</td>
<td>0.97</td>
</tr>
<tr>
<td>A2 μ ITU</td>
<td>50.12</td>
<td>53.56</td>
<td>56.03</td>
<td>56.6</td>
<td>56.86</td>
<td>54.64</td>
</tr>
<tr>
<td>A2 μ NEW</td>
<td>52.91</td>
<td>55.37</td>
<td>57.15</td>
<td>57.48</td>
<td>57.59</td>
<td>56.10</td>
</tr>
<tr>
<td>A2 μ impr.</td>
<td>2.79</td>
<td>1.77</td>
<td>1.12</td>
<td>0.88</td>
<td>0.73</td>
<td>1.46</td>
</tr>
<tr>
<td>B μ ITU</td>
<td>46.45</td>
<td>49.83</td>
<td>52.18</td>
<td>52.69</td>
<td>53.02</td>
<td>50.83</td>
</tr>
<tr>
<td>B μ NEW</td>
<td>49.24</td>
<td>51.58</td>
<td>53.46</td>
<td>53.79</td>
<td>53.97</td>
<td>52.41</td>
</tr>
<tr>
<td>B μ impr.</td>
<td>2.80</td>
<td>1.75</td>
<td>1.29</td>
<td>1.09</td>
<td>0.95</td>
<td>1.58</td>
</tr>
</tbody>
</table>

4. CONCLUSIONS

We have presented a very efficient context method for encoding companded signals. We have shown how to utilize as a coding context the predicted value in the companded domain and we obtained results superior to those obtained when assuming the residuals in the companded domain to be identically distributed. One possible application is the encoding of companded speech signals, but the method can be utilized for compressing any predictable signal that is companded before discretization and transmission.

5. REFERENCES