inter•noise 96
25th Anniversary Congress – Liverpool

INCE: 71.8

DATA COMPRESSION FOR SOUND LEVEL STORING

H Takinami, T Wakabayashi, N Hayashi & K Nakamura

RION Co Ltd, Kokubunji, Tokyo, Japan

1. INTRODUCTION

It is sometimes necessary to store sampled sound level data in order to check if there were unwanted data in the measurement duration. Since most of the conventional storing systems use fixed-bit-length binary code, it is difficult to store for a long duration when the sampling period is very short. For computer programs or data, it is usual to "compress" them for reducing the memory size. For time-varying sound level storing systems, we propose very simple and effective data compression procedures using variable-bit-length code. One method uses a procedure to convert the original sound level data to the differential data which were calculated between two consecutive data. Another method is a procedure using an inverse filter which converts the time constant in exponential time weighting. However, this method cannot be reconverted to the original data perfectly, the effect of compression is greater than the former method. In this report, the effectiveness of these data compression methods are described by application of an example of road traffic noise.

2. ENTROPY AND VARIABLE-BIT-LENGTH CODE

In this report, the term "data compression" is used to mean to reduce the total bit-length of the data for storing. The total bit-length depends on the number of the data, therefore, the mean bit-length (total bit-length/number of data) is used for evaluating the effect of compression. For evaluation, the fixed-bit-length of 10 bits is used for reference. A bit length of 10 bits is the minimum requirement for a 60-dB linearity range and 0.1 dB resolution sound level storing system with a fixed-bit-length binary code.
There are many kinds of data compression procedures. For sound level storing systems, it is very important to compress data in real time. This

means the compression procedure should be very simple and be able to installed in the system easily.

Variable-bit-length code. Using a variable-bit-length code is a very easy method to compress data. To store a data series $\boldsymbol{X} = \{x_i\}$, where i = 1 to n, by this method, first, the probability density of the occurrence of the data value in $\boldsymbol{X}$ is calculated, and then the shorter bit-length-code is allocated to the higher probability density data value. In this report, one of the variable-bit-length codes, Fibonacci code, is used.

Entropy. Entropy is considered to be the theoretical minimum mean bit-length. When a data series $\boldsymbol{X}$ is presented with m kinds of integer, the entropy of this data series $H(\boldsymbol{X})$ is calculated by

$$H(\boldsymbol{X}) = \sum_{j=1}^{m} P(v_j)\log_2 P(v_j) \qquad (1)$$

where v_j is the value of the integer and $P(v_j)$ is the probability density of the occurrence of v_j in $\boldsymbol{X}$. If the entropy of the sound level data themselves is small enough, it is easy to achieve great data compression using only the Fibonacci code.

Object for data compression. In this report, a series of sampled fast-exponential-time-weighted sound level $\boldsymbol{L} = \{l_i\}$, where i = 1 to 9000 and the number of sampling per one second is 30, is used as an example. Fig. 1 shows the level-time history of these data. The entropy of the data series $H(\boldsymbol{L})$ is 8.10. Therefore, it is not possible to compress the data enough using only the Fibonacci code. If $\boldsymbol{L}$ is stored using the Fibonacci code, the mean bit-length of $\boldsymbol{L}$ is 10.01 bits. There is no compression compared to the reference bit-length of 10 bits.

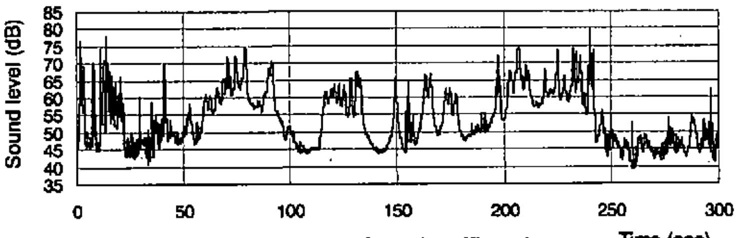

Figure 1. Level-time history of road traffic noise

3. CONVERSION TO DIFFERENTIAL DATA

To obtain sufficient effect of compression, the data series $\boldsymbol{L}$ has to be converted to another data series in which entropy is low compared to the entropy of $\boldsymbol{L}$. It is reported that the entropy of the sound level data series $\boldsymbol{L}$ can be reduced by conversion to the differential data series $\boldsymbol{D} = \{d_i\}$ by the following procedure[1],

$$d_1 = l_1$$
$$d_i = l_{i-1} - l_i \ (i = 2,3,....) \qquad (2)$$

The transient of exponential-weighted sound level is very rapid in rise up and comparably slow in decay. Therefore, most of the data of ***D*** have very small absolute values when the sampling period is shorter than the time constant of the exponential time weighting[1]. The entropy of the differential data series H(***D***) is 4.16 and the distribution of the data value of ***D*** (excluding d_1) is shown in Fig. 2.

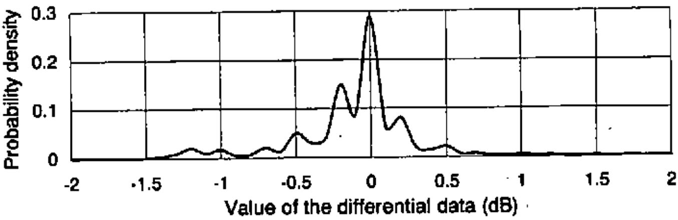

Figure 2. Distribution of the data value of ***D***

The small absolute value data (including zero) have high probability density of occurrence. Therefore, for allocating the codes to the data, it is not necessary to calculate the probability density of the occurrence of the data. Two types of coding based on the Fibonacci code were used. One is the Fibonacci code with a sign-bit which indicates positive or negative. However, two-bit-length code is not used in this coding, therefore it is not necessary to prepare any allocating tables for storing. Another is the modified Fibonacci code as shown in Table 1.

Table 1. Coding based on the Fibonacci code

Signed Fibonacci code		Modified Fibonacci code		Bit-length
Value of d	Code	Value of d	Code	
0	11	0	11	2
-----	-----	-1	011	3
0<abs(d)<= 1	011s	-2, +1	ff11	4
1<abs(d)<= 3	ff11s	-3, +2, +3	fff11	5
3<abs(d)<= 6	fff11s	-----	-----	6
6<abs(d)<=11	ffff11s	3<abs(d)<= 8	ffff11s	7
11<abs(d)<=19	fffff11s	8<abs(d)<=16	fffff11s	8
19<abs(d)<=32	ffffff11s	16<abs(d)<=42	ffffff11s	9

fff.. is a binary code which does not have any sets of consecutive '1'.
s is the sign bit which indicates positive or negative.

In this coding, the data with an absolute value less or equal to three are allocated to the normal Fibonacci code, and other data are allocated to the signed Fibonacci code. Almost all of the data of ***D*** are concentrated in the range between -3 to 3, and the bit-length shorter than the signed Fibonacci coding are allocated to these data.

The mean bit-lengths of the differential data series ***D*** with these coding are 4.89 bits and 4.31 bits, respectively. By comparing to the reference of 10 bits, they are compressed more than 50 %. And by comparing to the entropy of the data series, almost complete compression is achieved by these codings.

4. APPLICATION OF INVERSE FILTER

The entropy of the differential data series of the sound level and the sampling period of the data[1] are related as follows. When the time constant of the exponential time-weighting is fixed, the entropy decreases by decreasing the sampling period. And when the sampling period is fixed, the entropy decreases by increasing the time constant of the exponential time-weighting. Fig. 3 shows the mean bit-length of the differential data series with a different time constant stored by the modified Fibonacci code.

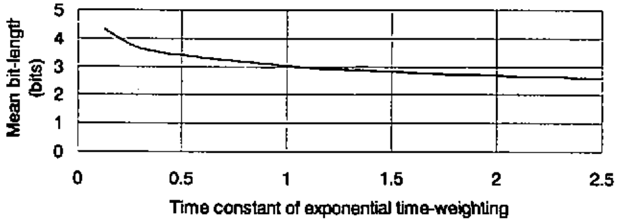

Figure 3. Mean bit-length of the differential data series

If the fast-exponential time-weighted (125-ms time constant) sound level data series can be converted from the longer time constant exponential time-weighted sound level data series, the sound level data series can be stored by a shorter mean bit-length. The exponential time-weighting is represented by a first order low-pass filter, therefore, the conversion of the time constant can be realized using a very simple digital processed inverse filter and low pass filter[2,3,4]. When the differential data series obtained from the slow exponential time-weighted sound level data (time constant is 1 s) is stored with the modified Fibonacci code, the mean bit-length of the data series is reduced to 3.01. The entropy of the data series is reduced to 2.19.

To examine the reproducibility of the conversion, the fast exponential time-weighted sound level data series ***D*** of road traffic noise is converted to the exponential time-weighted sound level data series ***C*** with the longer time constant (1 s), and that data series is reconverted to the fast exponential time-weighted sound level data series ***D'***. The data series ***C*** was rounded to a tenth of decibels. Fig. 4 shows duration of 60 seconds extracted from the data series ***D*** and ***D'***. In the reconverted data series ***D'***, many sharp peaks and dips appeared on the portion where the sound level decreases gradually. These peaks and dips appear because the data in ***C*** were rounded to a tenth of decibels. When the resolution of the data in ***C*** is not restricted, ***D'*** is identical to ***D***.

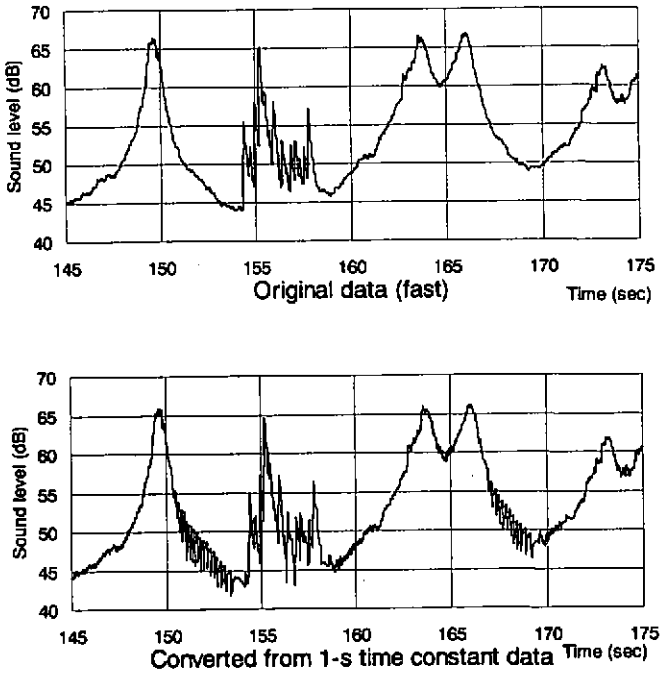

Figure 4. Comparison between the original and converted data

To evaluate the practicality of the converted data ***D'***, the maximum sound levels L_{max} and the percentile sound levels L_x calculated from ***D*** and ***D'*** are compared. The equivalent sound levels are obtained from the data series ***C*** directly and correctly, therefore, it is unnecessary to calculate them from the data series ***C***. The results obtained from each 60-s measurement period and the total measurement period of 5 min are shown in Table 2.

Table 2. The maximum and percentile sound levels in decibels

Period (s)	0 to 60		60 to 120		120 to 180		180 to 240		240 to 300		0 to 300	
Data	***D***	***D'***	***D***	***D'***	***D***	***D'***	***D***	***D'***	***D***	***D'***	***D***	***D'***
L_{max}	78.0	78.0	74.9	74.8	67.5	67.4	74.7	74.7	79.7	79.7	79.7	79.7
L_5	67.4	67.5	69.1	69.1	64.8	64.7	70.4	70.4	54.7	54.7	67.6	67.5
L_{50}	49.8	49.7	58.3	58.3	54.1	53.8	60.4	60.1	46.4	46.2	52.2	51.8
L_{95}	43.6	42.2	44.6	44.6	44.5	44.2	48.7	48.6	43.6	43.6	43.8	43.2

However, there are small differences at lower sound levels (L_{50} and L_{95}) because of rounding the longer time constant weighted data, the sound levels calculated from the data series ***D'*** are practically identical to those calculated from ***D***.

5. CONCLUSION

When the sampled sound level data is stored, it can be compressed by conversion to the differential data series and using the variable-bit-length code. And, by using the inverse filter and low pass filter which convert the time constant of the exponential time-weighting, a greater effect of compression can be obtained. Fig. 5 shows the mean bit-length and the entropy of the data series compared to the reference bit length and the fixed-bit length of 16 bit used in the conventional sound level data storing system.

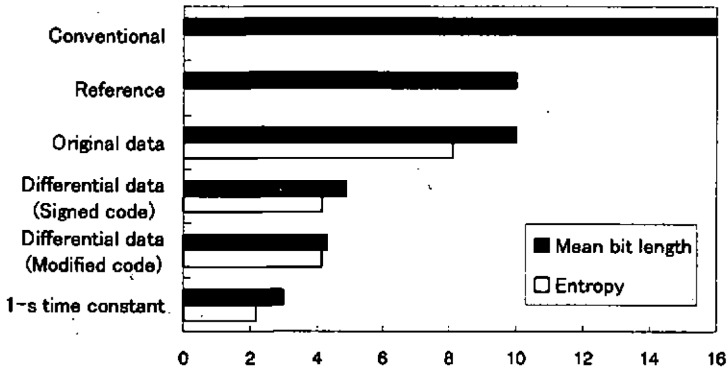

Figure 4. Mean bit-length and Entropy

The procedure which converts to the differential data series and store the data with the variable-bit-length code is very easy and can obtain great data compression effect. The procedure which converts the time constant of exponential time-weighting has a greater compression effect, however, the reconverted data series is not completely identical to the original data. Some appropriate procedures, such as a smoothing, should be considered to obtain better reproducibility.

References (in Japanese)

[1] Wakabayashi *et.al*, Acoust. Soc. Jpn, Spring 1993, pp.487-488.

[2] Iwahashi *et.al*, Acoust. Soc. Jpn, Autumn 1994, pp.641-642.

[3] Wakabayashi *et.al*, Acoust. Soc. Jpn, Autumn 1994, pp.643-pp.644.

[4] Takinami *et.al*, Tech. Rep. Noise Acoust. Soc. Jpn, N-95-19, 1995.