Appl. Math. and Comp. Sci., 1997, Vol.7, No.3, 659–675

UNIVERSAL VOICE COMPRESSION ALGORITHMS BASED ON APPROXIMATE STRING MATCHING

ILAN SADEH*, ALEXANDER KEIZELMAN*
MICHAEL ZAK*

Two practical voice-coding schemes based on approximate string matching are proposed. The first one is an approximate fixed-length string-matching data compression, and the other is a Lempel-Ziv-type quasi-parsing method by approximate string matching. The performance of the two algorithms and their sub-optimal versions are evaluated. The main advantages of the proposed methods are the asymptotic sequential behaviour of the encoder and the simplicity of the decoder. In particular, a sub-optimal tree-structure algorithm is proposed and its real-time advantages are demonstrated.

1. Introduction

Although it seems strange, audio data are remarkably hard to compress effectively. Let us make a short overview of widely-used speech compression algorithms:

- *ADPCM (Adaptive Delta Pulse Code Modulation)* public standards for voice compression.
- *Shorten* performs compression of waveform files such as audio data. A simple predictive model of the waveform is used followed by Huffman coding of the prediction residuals. But when we try to achieve better compression rates using an acceptable quantisation error in bits, speech quality is decreased dramatically fast.
- An *LPC (Linear Predictive Coding)* coder fits speech to a simple, analytic model of the vocal tract, then throws away the speech and ships the parameters of the best-fit model. An LPC decoder uses those parameters to generate synthetic speech that is usually more or less similar to the original one. The result is intelligible, but sounds like a talking machine.
- *CELP (Code-Excited Linear Prediction)* is one of the most frequently used coding techniques operating at low bit rates. However, CELP vocoders still have a drawback: unnatural speech rate quality is rapidly increasing as the bit rate is decreased.

* Computer Science Department, Natural Science Faculty, Ben-Gurion University of the Negev, Beer-Sheva, Israel, e-mail: sade@math.tau.ac.il.

We propose two practical source-coding schemes based on approximate string matching which entertain all specific properties of the human auditory system.

Approximate string matching is a technique with considerable theoretical merits and practical promise. Our major motivation comes from data-compression theory, although we believe that our results might have a wide range of applications in pattern-recognition theory, artificial intelligence, biological research and, in particular, the Genome Project. Its algorithmic complexity and data structure were studied by Landau and Vishkin (1986), Galil and Giancarlo (1988) and Storer (1990). However, we focus our attention on the probabilistic aspect of the problem, in particular on universal source coding. Universal source-coding algorithms that achieve the theoretical bound, $R(D)$, have been proposed by Sadeh (1993a; 1993b; 1994). The most appropriate application we had in mind is in multimedia, where voice is transmitted on the same channels and the decoder in a universal receiver must be cheap. Consequently, bandwidth reduction and cheap decoding procedures are the first necessary step to bring multimedia closer to reality.

The first practical universal source coding were proposed by Lempel and Ziv (1977; 1978) for compression of sequences by a lossless encoder. Their algorithms are based on a parsing procedure which creates a new phrase as soon as a prefix of the still unparsed part of the string differs from all preceding phrases. A fast implementation was given by Welch (1984). The asymptotic behaviour of its implementation through suffix trees was studied by Szpankowski (1993). A suffix tree construction algorithm was presented by McCreight (1976) and a simple algorithm for sorting the suffixes of a string was given by Manber and Myers (1993). Digital search trees were used in (Jacquet and Szpankowski, 1995) to obtain second-order properties of the LZ parsing scheme. LZ parsing algorithms play a crucial role in applications such as efficient transmission of data (Lempel and Ziv, 1977; 1978), estimation of entropy (Ziv, 1978), discriminating between information sources (Gilbert and Kadota, 1992), test of randomness, estimating a statistical model of individual sequences (Plotnik *et al.*, 1992) and so forth.

Berger (1971), Blahut (1987), Gray (1975), Ornstein and Shields (1990), Sadeh (1995) and others proved the existence of optimal codes subject to a fidelity criterion. These works vary in the assumptions on the class of sources, the fidelity criterion and the type of convergence of the rates to the rate distortion function $R(D)$. In most cases, the construction of a code involves the following step: Given a block of source symbols, find the code in an exponentially large class of codes, which performs best for this block. This task is prohibitively complex even for modest code-book sizes and makes these schemes impractical for implementation. Most of the schemes postulate that an *a priori* distribution of the source or some statistical information be known at both ends of the data link. Such postulates are inappropriate in the design of a universal machine.

Sadeh (1993a; 1993b; 1994) has proposed two practical source-coding schemes based on approximate string matching. The first one is an approximate fixed-length string-matching data compression method combined with a block coder based on the empirical distribution, and the other is an LZ-type quasi-parsing method by approximate string matching. The output process, which is also a data base, is also assumed

to be stationary and ergodic. It has been shown by Sadeh (1993a; 1993b; 1994) that in the former algorithm the compression rate converges to the theoretical bound, $R(D)$, as the string length tends to infinity. We note that the decoding algorithm is very simple. The encoding algorithm is exponential at the beginning, but it is implementable. As time elapses, the encoding algorithm tends to be asymptotically linear with respect to the size of the data-base string. The algorithm of Wyner and Ziv (1989) is obtained as a special case of no-distortion $D = 0$.

A similar result holds for the latter algorithm in the limit of the long data base generated by the former one, as proved by Sadeh (1993a; 1993b; 1994). The algorithm of Lempel and Ziv is obtained as a special case of no-distortion for the quasi-parsing method by approximate string matching.

We follow the terminology and some results of Sadeh (1993a; 1993b; 1994) who adopted the terminology of Wyner and Ziv (1989), and Ornstein and Weiss (1993) in their work concerning lossless algorithms.

There is some literature regarding the probabilistic analysis of problems of approximate pattern matching (Arratia and Waterman, 1989; Arratia *et al.*, 1990; Luczak and Szpankowski, 1997; Sadeh, 1993a; Steinberg and Gutman, 1992). In particular, we would like to mention two references related to a probabilistic analysis of pattern matching in the context of computer science (Knuth, 1981) and combinatorial probability (Pittel, 1985).

We implement Sadeh's algorithms (1993a; 1993b; 1994) and propose a suboptimal tree-structure algorithm for real-time voice coding. The results seem to be promising in comparison with other known methods.

2. Physiological and Psychological Acoustics

This section covers the basics of psychoacoustic modelling. For more details, we refer the interested reader to (Meyer and Neumann, 1972; Kinsler *et al.*, 1982).

2.1. Human Speech

The acoustic energy associated with speech originates in the chest muscles which, by contraction, force air from the lungs up trough various components of the vocal mechanism. This steady stream of air may be looked on as a carrier of energy that must be modulated in its velocity and corresponding pressure in order to produce sounds. The requisite modulation is accomplished in two basic ways, leading respectively to voiced and unvoiced sounds.

Voiced sounds include the vowels of ordinary speech as well as tones characteristic of the singing voice. The primary modulating agent for voiced sounds is the larynx, across which the vocal cords are stretched. The vocal cords are two membrane-like bands forming a diaphragm with a slit-like opening that modulates the air stream as it vibrates open and shut. The length of this opening and the tension to which the vocal cords are stretched determine the fundamental frequency of this modulation. The shorter and lighter vocal cords possessed by most women vibrate almost twice

as rapidly as those possessed by men. This accounts for the higher pitch of most women's voices. The action of the vocal cords produces a sawtooth pressure waveform containing many harmonics.

The resonating cavities and orifices of the nose, mouth, and throat form an acoustic filtering network altering the relative amounts of harmonics. Many of this are controllable at will and thereby a wide variety of voiced sounds may be produced.

It is also possible for the voice mechanism to produce unvoiced sounds. For instance, a steady forcible exhaling of the breath will produce a hissing sound caused by turbulences set up in the flow of air through the numerous irregularities along the vocal tract. Here the sounds produced by modulating the airstream with the lips, teeth, or tongue. As with the voiced sounds, conscious control of the tongue and lips alters the resonances of the cavities and orifces, producing a wide variety of unvoiced sounds so that recognizable ones can be generated (whispering). A spectrum analysis of the unvoiced sounds reveals a band of practically continuous frequency coverage largely confined to the upper portion of the audible range.

The loudness of human voice is dictated by the stream of air forced trough the vocal cords from the lungs. The frequency of the human voice is controlled by the elasticity and vibration of the vocal cords, while the resonators govern the quality of the sound produced.

Intelligibility of speech is an indication of how well speech is recognized and understood. This depends on acoustic power delivered during the speech, speech characteristics, hearing acuity and ambient noises.

The relative speech power depends on the frequency, for both men and women. Much of the acoustic power of speech is concentrated in the frequency range of 100 to 1000 Hz. Despite this fact, higher frequencies are of great importance, because a higher frequency band of speech makes the consonants of speech intelligible.

2.2. Ear Characteristics

2.2.1. Frequency Range and Sensitivity of Hearing

The human ear is capable of hearing sound waves having frequencies from about 20 Hz to 20 kHz; the upper frequency limit becomes markedly lower with increasing age.

The threshold of hearing is the minimum perceptible free-field intensity level of a tone that can be detected at each frequency over the entire range of the ear. The tone presented to both ears should have a duration of about 1 s. For tones shorter than 0.3 s the apparent loudness increases with increasing tone duration, and for tones longer than 3 s "fatigue" sets in and the apparent loudness diminishes with time.

As the intensity of the incident acoustic way is increased, the sound grows louder and eventually produces a tickling sensation. This level is less dependent on the frequency than the threshold of hearing and is called the threshold of feeling. As with the lower threshold, it varies somewhat from individual to individual, but not to a great extent. As the intensity is still further increased, the tickling sensation gives rise to pain at about 140 dB.

The frequency of maximum sensitivity is near 4 kHz. Below this, the threshold rises with decreasing frequency, the minimum power required to produce an audible sound at 30 Hz being nearly a million times as great as at 4 kHz. Thus for high frequencies the threshold rises rapidly to a cut-off.

2.2.2. Loudness

The threshold of hearing is very much frequency-dependent. In general, two tones of equal sound pressure but of different frequencies are not heard as equally loud. Therefore, to characterize a sound by its subjective effect, it is not sufficient to state its sound intensity; the characteristics of the ear must be taken into consideration. In addition to the physical quantity, i.e. the sound pressure, a physiological quality, i.e. a loudness level, has been introduced.

Considerably higher sound pressures are required to produce an equal-loudness impression at low frequencies than those needed at high frequencies. At the upper end, the lines of constant loudness agree roughly with the lines of constant sound pressure. The threshold of hearing forms the lowest curve of equal loudness.

The loudness level measurement by a subjective listening comparison is difficult to carry out in practice. To obtain the loudness level from the sound pressure level, one must first consider the frequency dependence of the sensitivity of the ear. The loudness level meter contains therefore electrical filters that evaluate various frequency components according to their intensities.

3. Principal Lemma of Approximate String Matching

We present an extension of Kac's Lemma (Kac, 1947), proved by Sadeh (1993a; 1993b; 1994) and based on ideas of L.H. Ozarow and A.D. Wyner (Wyner and Ziv, 1989).

Consider a finite-valued stationary infinite sequence v defined on an alphabet V. Let v_i^j denote a sample sequence between positions i and j in the sequence v. Let B be any set of strings of length l taken from the space of all possible strings of length l, defined on V, i.e. $B \subseteq V^l$ such that $\Pr(B) > 0$. We denote by Y_n a string of length l starting from position n, $Y_n = v_n^{n+l-1}$, and we say that two strings Y_n and Y_m are approximately matched with respect to B if $Y_n \in B$ and $Y_m \in B$. We denote the conditional probability that an approximate match with respect to B occurs for the first time at step k by

$$Q_k(B) = \Pr\left\{Y_k \in B; \quad Y_j \notin B; \quad 1 \leq j \leq k-1 \mid Y_0 \in B\right\}$$

The average recurrence time to B is defined by

$$\mu(B) = \sum_{k=1}^{\infty} kQ_k(B)$$

The event that in the realizations of v we can find members of B is

$$A = \left\{Y_n \in B \quad \text{for some } n, \ -\infty < n < \infty\right\}$$

Lemma 1. (The Extended Kac Lemma) *Under the above assumptions, we have*

$$\Pr\{A\} = \Pr\{Y_0 \in B\}\mu(B)$$

In particular, for stationary ergodic processes, we obtain

$$1 = \Pr\{Y_0 \in B\}\mu(B) = \Pr\{B\}\mu(B)$$

4. Algorithms and Convergence to $R(D)$

4.1. Definitions

Let u be an ergodic finite-valued stationary sequence. Let $\bar{u}$ denote a sample sequence or the block (consecutive symbols in the sequence). The notation u_i^j stands for the block between positions i and j in the sequence u. The sequence u_{-n}^{-1} is our data base.

Definition 1. Let L be the first index such that the string $u_0 \dots u_{L-1}$ is not a substring of the data base u_{-n}^{-1}. That L is denoted by $L_n(u)$, i.e. for $n = 1, 2, \dots$ $L_n(\bar{u})$ is the smallest integer $L > 0$ such that

$$u_0^{L-1} \neq u_{-m}^{-m+L-1}, \quad L \leq m \leq n$$

$$L_n(u) = \inf\left\{L > 0 : \; u_0^{L-1} \neq u_{-m}^{-m+L-1}, \; L \leq m \leq n\right\}$$

Definition 2. The random variable $N_l(\bar{u})$ for $l > 0$ is the smallest integer $N \geq l$ such that

$$u_0^{l-1} = u_{-N}^{l-1-N}$$

$$N_{\bar{u}} = \inf\left\{N \geq l : u_0^{l-1} = u_{-N}^{l-1-N}\right\}$$

Let $\rho(\bar{u}, \bar{v})$ denote the average value of the 'per-letter' distortions for the letters that comprise the block $\bar{u}$, i.e.

$$\rho(\bar{u}, \bar{v}) = \frac{1}{l}\sum_{k=1}^{l} d(\bar{u}_k, \bar{v}_k)$$

where the pair $(\bar{u}_k, \bar{v}_k)$ denotes the letters at the k-th position at the source and the user, respectively. The distortion is assumed to be memoryless.

Definition 3. For each sample sequence $\bar{u}$ of length l, taken from a sequence u, we define the set

$$D - \text{Ball}(\bar{u}) = \{\bar{v} \mid \; \rho(\bar{u}, \bar{v}) \leq D\}$$

Definition 4. For each sample sequence $\bar{u}$ we define the random variable

$$DL_n(\bar{u}, v_{-n}^{-1}) = \max_{\bar{v}:\rho(\bar{u},\bar{v})\leq D} L_n(\bar{v}, v_{-n}^{-1})$$

In other words, using the sequence v_{-n}^{-1} as our data base, we continue the sequence $u_0, u_1 \ldots$ until there is no L-string $\bar{v}$ in the D – Ball surrounding the L-string $\bar{u} = u_0 \ldots u_{L-1}$ such that $\bar{v}$ is a substring of the data base v_{-n}^{-1}. We define $DL_n(\bar{u}, v_{-n}^{-1}) = L$.

Definiton 5. For each sample sequence $\bar{u}$ we define the random variable

$$DN_l(\bar{u}, v_{-n}^{-1}) = \min_{\bar{v}:\rho(\bar{u},\bar{v})\leq D} N_l(\bar{v}, v_{-n}^{-1})$$

In other words, we choose all the strings of length l taken from V^l that are neighbours of $u_0^{l-1} = \bar{u}$. From this set we select a string $\bar{v}$ with the smallest $N_l(\bar{v})$. The selected $\bar{v}$ is the element in D – Ball $(\bar{u})$ with minimal first repetition.

Speech is represented by a sequence of symbols (frames), each of which contains N sounds. In order to compare two audio frames, we use the Discrete Fourier Transform for mapping from the audio to the frequency representation domain.

According to specific properties of the human auditory system, we assign a weight to each frequency.

Definition 6. Let u and v be two different audio frames. Let U_1^N and V_1^N be the Fourier coefficients of $DFT(u)$ and $DFT(v)$, respectivelly. Then we define the distance between the two frames as

$$\text{distance}\,(u, v) = \max_{50\text{ Hz}\leq\text{frequency}\,(i)\leq 5000\text{ Hz}} \text{abs}\,(U_i - V_i) * \text{weight}_i$$

i.e. the maximum for all indices which correspond to real frequencies from 50 to 5000 Hz.

Definition 7. Let u and v be two different audio frames. They are equal under our comparison function if and only if

$$\text{distance}\,(u, v) < \text{Approximation Ratio}$$

For implementation of a search tree we need a characteristic function of the audio frame (a key of the audio frame) which maps from the frequency to the real-number representation domain.

Definition 8. Let u be an audio frame. Let U_1^N be the Fourier coefficients of the $DFT(u)$. The key of the audio frame is defined by

$$\text{key}\,(u) = \sum_{\text{frequency}\,(i)=50\text{ Hz}}^{5000\text{ Hz}} \text{abs}\,(U_i) * \text{weight}_i$$

i.e. similar frames have identical keys.

4.2. Duality Lemma

We present a duality between the two random variables of Definitions 4 and 5, as proved by Sadeh (1993a; 1993b; 1994). This duality implies the duality between the proposed algorithms.

Lemma 2. *For each sample sequence $\bar{u}$, a data base v_{-k}^{-1}, and any positive integers $n \leq k$ and $l \leq n$, we have*

$$\left\{DN_l(\bar{u}, v_{-k}^{-1}) > n\right\} = \left\{DL_n(\bar{u}, v_{-n}^{-1}) \leq l\right\}$$

4.3. Approximate Fixed-Length String Matching—Data Compression

The sequence u_{-n}^{-1} is called the "past". The sequence $u_0, u_1 \ldots u_{l-1}$ is a new block. It is assumed that the optimal data-compression algorithm has already been applied to the history in order to create this data base of length n. We also assume that the data base was "empty" before the encoding procedure has begun.

The new sequence can be compressed by assigning indices to blocks since each symbol in the data base is assigned an index.

The data compression scheme is as follows:

Approximate Fixed-Length String—Data Compression Scheme:

1. Choose a string $\bar{u} = u_0^{l-1}$ of length l.
2. If u_0^{l-1} can be approximately matched up to the tolerance D by a substring of v_{-n}^{-1}, encode it by specifying $DN_l(\bar{u}, v_{-n}^{-1})$. Append $v_{-DN_l}^{l-1-DN_l}$ to the data base.
3. Otherwise, append the string v_0^{l-1} satisfying $\rho(u_0^{l-1}, v_0^{l-1}) = 0$ to the data base.
4. Add a bit as a flag to indicate whether or not there is a match.
5. Repeat the process from Step 1, with string $\bar{u} = u_l^{2l-1}$ of length l and data base v_{-n+1}^{l-1}. Shift the indices by l to the appropriate values.

Approximate Fixed-Length String—Data Decompression Scheme:

1. Handle the coming string according to the 'flag' If the flag indicates 'Match', copy the substring of v_{-n}^{-1} specified by the pointer $DN_l(\bar{u}, v_{-n}^{-1})$ to the decoder data base. Append the string $v_{-DN_l}^{l-1-DN_l}$ to the data base in the decoder at position 0.
2. If the flag indicates that there is no match, append the string received in the input buffer at the beginning of the decoder data base.
3. Shift the indices by l to the appropriate values. Update n to $n+l$. Repeat the process from Step 1 with a new input.

Theorem 1. *Consider a data base $v_{-\infty}^{-1}$ generated by the Approximate Fixed-Length String Data Compression Scheme from a stationary ergodic process u. We assume*

that $v_{-\infty}^{-1}$ *is a stationary and ergodic process. Then, for all* $\beta > 0$,

$$\lim_{l\to\infty} \Pr\left\{\left|\frac{\log DN_l(\bar{u}, v_{-\infty}^{-1})}{l} - R(D)\right| > \beta\right\} = 0$$

and the average compression ratio attains the bound $R(D)$.

This theorem has been proved by Sadeh (1993a; 1993b; 1994) by using Lemma 1.

4.4. Quasi-Parsing Method by Approximate String Matching

The machine is based on a quasi-parsing procedure which creates a new phrase as soon as a prefix of the still unparsed part of the input sequence differs from all substrings in the data base by more than D per letter. The data base is the concatenation of all the preceding phrases. The encoding of each phrase consists of the pointer N to the last approximately matched string, the string length DL_n and the last reproducing letter with zero distance from the last input symbol.

Quasi-Parsing Scheme:

1. Let $l = 1$.
2. Choose a string u_0^{l-1} of length l.
3. If u_0^{l-1} can be approximately matched up to the tolerance D by a substring of v_{-n}^{-1}, store a pointer N to that substring and increment 1. Go to Step 2.
4. Otherwise, append the string v_{-N}^{l-2-N} to the data base track at position zero and further, and append the letter v_{l-1} (the reproducing letter which satisfies $d(u_{l-1}, v_{l-1}) = 0$). The encoding is done by the pointer to the string v_{-N}^{l-2-N}, the length $DN_n(u)$ and the last reproducing letter associated with the last source letter.
5. Repeat the process from Step 1, where the data base is appended with the chosen string denoted by $v_0^{DL_n}$. The data base contains now $n + DL_n(u)$ reproducing symbols. Shift the indices to adopt to the algorithm.

Theorem 3. *Let* u *be a stationary ergodic process defined on an alphabet* U. *Consider a suffix* v_{-n}^{-1} *taken from the data base* $v_{-\infty}^{-1}$ *generated by an encoder-decoder pair as described in the Approximate Fixed-Length String Data Compression Scheme. Suppose that at time zero we switch to the Quasi-Parsing Scheme. Assuming that the scheme preserves stationarity, as the memory size* n *tends to infinity, for the new sample sequence* $\bar{u}$ *encoded from the input* u *by the Quasi-Parsing Scheme, we have in probability*

$$\lim_{n\to\infty}\left\{\frac{\log n}{DL_n(\bar{u}, v_{-n}^{-1})}\right\} = R(D)$$

5. Implementation of the Search Tree

Each frame in the data base has the corresponding node in the search tree, which consists of the key, index in the data base and pointers to the left and right sons. The search is executed by one path on the tree.

To search for the target we first compare the currently processed fragment with the corresponding fragment in the data base with some distortion. If it is not the same, we go to the left or the right subtree as appropriate and repeat the search in that subtree. If we find the fragment, the procedure succeeds. Otherwise, we continue searching until we hit an empty subtree. The search works in time $O(log(n))$.

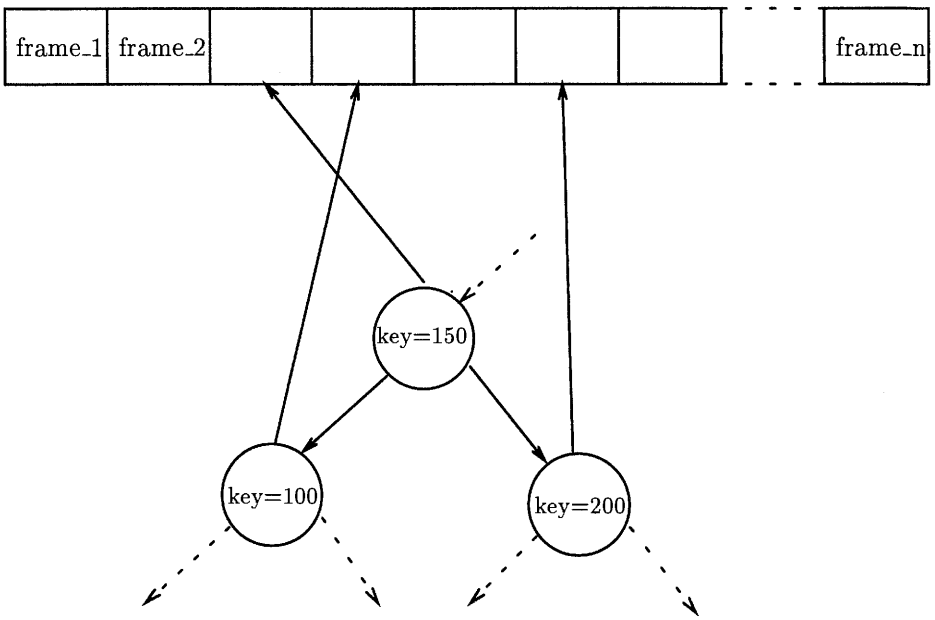

Fig. 1. The search tree.

According to (Kruse, 1987), the first case, which inserts a node into an empty tree, is easy. We only have to make the root point to the new node. If the tree is not empty, then we must compare the key with the one in the root. If it is less, then the new node must be inserted into the left subtree; if it is greater, then it must be inserted into the right subtree.

The keys which are inserted in the tree constitute a sequence of random numbers, and therefore the tree is balanced.

6. Compression Performance

6.1. Results

The sample rate 8 kHz was accomplished. The overall speech quality was found to maintain its distinctness, continuity and high intelligibility even at a frame size 256 bytes.

The graphs show the performance of the Approximate Fixed-Length String-Matching Method and that of the Quasi-Parsing Method which are implemented with linear and tree search. The original file size is 100 KB.

The compression performance for the Approximate Fixed-Length (length = 4) String-Matching Method with linear search implementation is presented in Fig. 2.

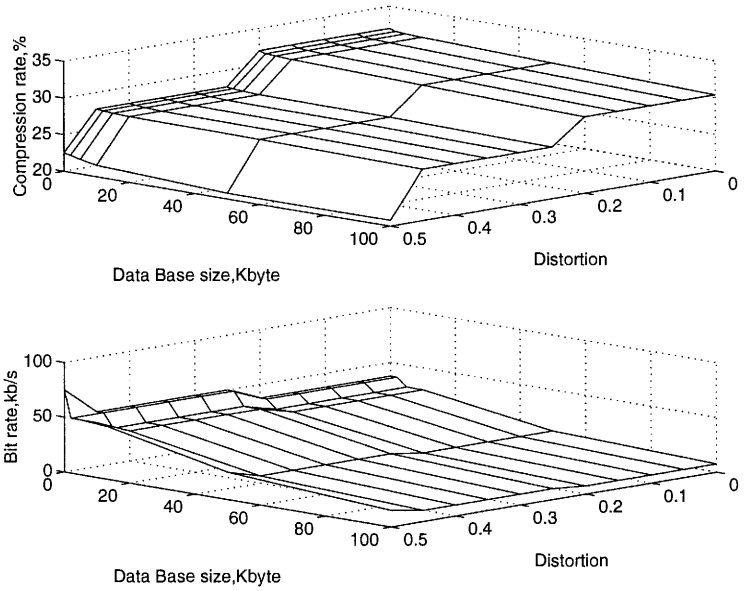

Fig. 2. Distortion vs. data-base size and compression rate/speed.

The compression performance for the Approximate Fixed-Length (length = 4) String-Matching Method with tree search implementation is presented in Fig. 3.

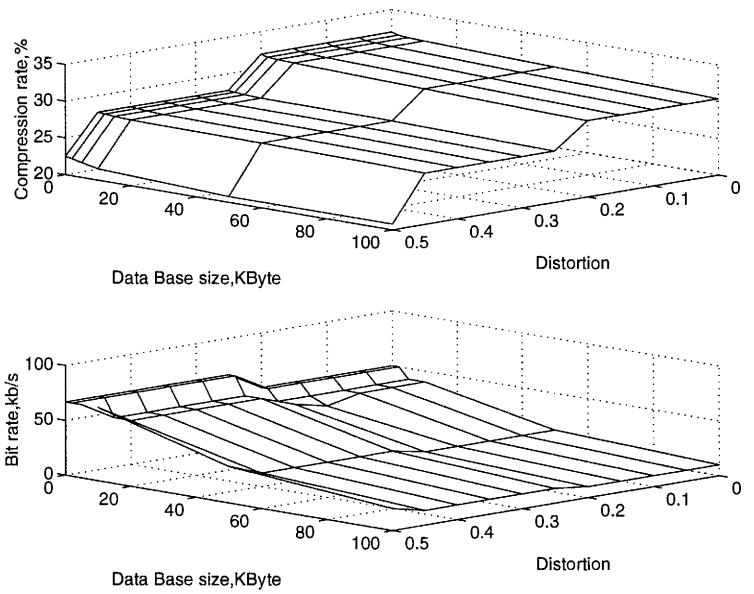

Fig. 3. Distortion vs. data-base size and compression rate/speed.

The compression performance for the Quasi-Parsing Method with linear search implementation is presented in Fig. 4.

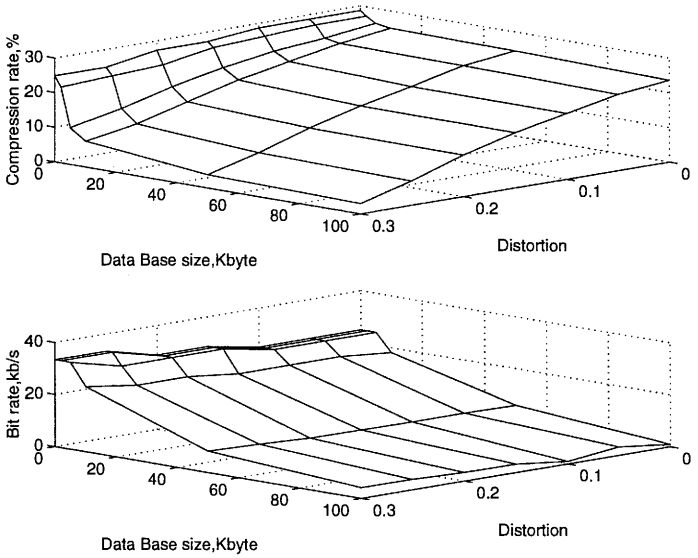

Fig. 4. Distortion vs. data-base size and compression rate/speed.

The compression performance for the Quasi-Parsing Method with tree search implementation is presented in Fig. 5.

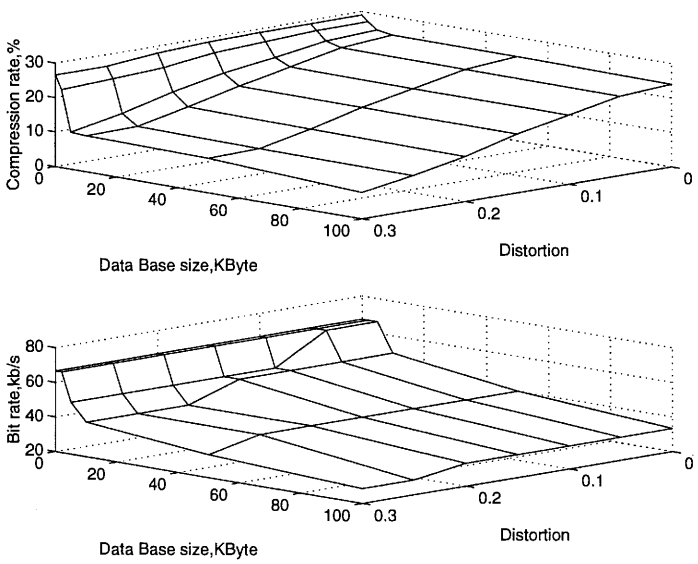

Fig. 5. Distortion vs. data-base size and compression rate/speed.

6.2. Analysis of Results

1. Approximate Fixed-Length String Matching vs. Quasi-Parsing Method: We can see that the Quasi-Parsing Method takes higher compression rates and better quality than the Approximate Fixed-Length String Matching.
2. Linear search vs. tree search: The tree search works in time $O(\log(n))$ and achieves a compression rate approximately equal to that of the linear search.

Consequently, the best choice for the speech compression is the Quasi-Parsing Method, which is implemented with tree search.

6.3. Comparison with Other Tools

Table 1 compares our technique with general and speech-purpose compression utilities commonly available. The table shows that a speech-specific compression utility can achieve a considerably better compression than the majority of general tools.

Tab. 1. Compression rates vs. quality and speed (the original file size 65 KB).

program	% rate	quality	speed kB/s
compress	73	lossless	65
zip	64	lossless	65
gzip	64	lossless	65
shorten default	60	lossless	65
shorten with quantization error 1	52	bad	65
shorten with quantization error 2	45	very bad	65
aiffccomoress	52	lossless	65
ADPCM	50	lossless	8
aucompress default	38	high	50
aucompress with distortion 0.1	37	good	50
aucompress with distortion 0.2	34	good	50

The program runs faster than real-time ones on most SUN SPARCstations.

7. Conclusions

The important properties of our implementation are as follows:

1. A real-time fast compression is accomplished by using a search tree.
2. High compression rates can be achieved by using an initial data base and setting distortion.
3. A good comparison function for two frames is obtained by considering specific properties of the human auditory system.
4. One of the features is applying an acoustic filter for reducing noise.

Sadeh's algorithms (Sadeh, 1993a; 1993b; 1994) are thought to be the first universal asymptotically sequential algorithms that attain the bound $R(D)$. But we emphasize that all of Sadeh's results are obtained after infinite time of operation and based on the accumulated data. It is still better than blockcoding because we send only the necessary messages and not all the pre-calculated codebook. The weak points are the slow convergence to $R(D)$, the initial complexity encoding procedure and the requirement (in practical cases) for a good and long initial data base. However, in practical cases we believe that the rate of convergence is faster. In the real life there exist properties such as Markovity, periodicity and/or some *a priori* information about the signal. In such cases the algorithm is paving the way for practical solutions. The most important issues are the asymptotical complexity of the encoder and that the decoder in both algorithms is very simple. It is based on copying strings either from a specified location in the data base or from the input data. In practice, the low-complexity of the decoder is usually an important advantage. It reduces the overall price of communication systems. The transmitter can be implemented in a costly way, but the receivers must be cheap.

Other important properties of the algorithms are as follows:

1. Optimality is obtained for a general stationary ergodic source.
2. Optimality is obtained for all memoryless distortion measures.
3. They are easily adapted to multimedia applications.
4. The Lempel-Ziv algorithm (CCITT Standard) is recovered as $D = 0$.
5. A realization with relatively low complexity and implementation with dynamic dictionary.
6. An appropriate definition of the distortion measure makes it possible to reduce the information content of a voice record while keeping a minimal audio distortion.
7. Sub-optimal tree-structure algorithms are proposed and demonstrated in this paper.
8. Benefits of noise reduction.

The most commonly-used methods for lossy compression are based on a predictive coding (DPCM), a transform (such as DFT, DCT, DST, WHT, wavelet, Haar and others), and vector quantization, with possible hybrid combinations. Most of them lack any general proof of optimality. Most of them use some kind of MSE criteria or data compression is achieved by coding only "high-energy" coefficients in their transform. Moreover, almost all of these techniques are not real-time algorithms. All transform-coding methods are performed after the speech signal is received and only then transformed and processed. Thus, it is not real-time computation. Our algorithms achieve an almost optimal compression performance with tolerable resources. However, the potential of approximate string matching can further be enhanced through its use in a subband coding system (Pearlman, 1994). Although the proposed coding methods are optimal as the data grow large, the performance achieved may not be the asymptotic limit, since voice samples are finite in extent. For several coding methods, it

has been shown both mathematically and experimentally that a subband or wavelet decomposition can achieve gains over direct coding of the original (full-band) source. Recently, it has been proved that for Gaussian sources the coding-rate gain is equal to the improvement in the convergence of the rate-distortion function (Padmanabha Rao and Pearlman, 1991). For Gaussian and non-Gaussian sources, this improvement in the convergence of the rate-distortion function in subbands is related directly to a measure of composite memory, which is markedly smaller in subbands than in the original fullband source (Nanda and Pearlman, 1992). This means that the approach to the rate-distortion function is potentially closer when using a subband decomposition of the source. Therefore, using approximate string matching in a subband decomposition with appropriate allocation of the rate among the subbands will result in a better rate-distortion performance than with direct coding, because the results obtained will be closer to the rate-distortion limit. This circumstance also holds true for zero distortion, so that there is also a rate advantage for lossless coding of source subbands. Our future objective is to seek algorithms which achieve a nearly optimal universal compression performance with limited resources in real time. The theory, which has been recently proved in (Padmanabha Rao and Pearlman, 1991; Sadeh, 1993a; 1994), strongly encourages us to look for a combination of subband coding and approximate string matching.

An important property is that our algorithm decompresses much faster than it compresses. Recall that the aim of the communications system is to convey the information generated by a sender or a source to a receiver at a reasonable cost. The encoder could be a costly device that transforms the source output into a form that can be transmitted over the channel. But the decoder that converts the channel output into a form that can be interpreted by the receiver must be cheap. This implies that the methods are attractive for practical applications.

References

Arratia R. and Waterman M. (1989): *The Erdos Renyi strong law for pattern matching with given proportion of mismatches.* — The Annals of Probability, Vol.17, pp.1152–1169.

Arratia R., Gordon L. and Waterman M. (1990): *The Erdos Renyi law in distribution for coin tossing and sequence matching.* — The Annals of Statistics, Vol.18, pp.539–570.

Berger T. (1971): *Rate Distortion Theory: A Mathematical Basis for Data Compression.* — Prentice-Hall.

Blahut R.E. (1987): *Principles and Practice of Information Theory.* — Addison-Wesley.

Galil Z. and Giancarlo R. (1988): *Data structures and algorithms for approximate string matching.* — J. Complexity, Vol.4, pp.33–72.

Gilbert E. and Kadota T. (1992): *The Lempel Ziv algorithm and the message complexity.* — IEEE Trans. Inform. Th., Vol.IT-38, pp.1839–1842.

Gray R.M. (1975): *Sliding block source coding.* — IEEE Trans. Inform. Th., Vol.IT-21, pp.357–368.

Jacquet P. and Szpankowski W. (1995): *Asymptotic behaviour of the Lempel Ziv parsing scheme and digital search trees.* — Th. Comp. Sci.

Kac M. (1947): *On the notion of recurrence in discrete stochastic processes.* — Bull. Am. Math. Society, Vol.53, pp.1002–1010.

Knuth D.E. (1981): *The Art of Computer Science. Seminumerical Algorithms.* — Reading: Addison-Wesley, Vol.2.

Kruse R.L. (1987): *Data structures & Program Design.* — New York: Prentice-Hall.

Landau G.M. and Vishkin U. (1986): *Introducing efficient parallelism into approximate string matching and a new serial algorithm.* — Proc. 18th Ann. ACM Symp. *Theory of Computing*, Berkeley.

Lempel A. and Ziv J. (1977): *A universal algorithm for sequential data compression.* — IEEE Trans. Inform. Th., Vol.IT-23, pp.337–343.

Lempel A. and Ziv J. (1978): *Compression of individual sequences via variable-rate coding.* — IEEE Trans. Inform. Th., Vol.IT-24, pp.530–536.

Luczak T. and Szpankowski W. (1997): *A lossy data compression based on an approximate pattern matching.* — (submitted).

Manber U. and Myers E.W. (1993): *Suffix trees: A new method for on-line string searchers.* — SIAM J. Computing, Vol.22, No.5, pp.935–948.

McCreight E.M. (1976): *A space economical suffix tree construction algorithm.* — J. ACM, Vol.32, No.2, pp.262–272.

Meyer E. and Neumann E.G. (1972): *Physical and Applied Acoustics.* — New York: Academic Press.

Nanda S. and Pearlman (1992): *Tree coding of image subbands.* — IEEE Trans. Image Proc., Vol.1, pp.133–147.

Ornstein D.S. and Shields P.C. (1990): *Universal almost sure data compression.* — The Annals of Probability, Vol.18, pp.441–452.

Ornstein D.S. and Weiss B. (1993): *Entropy and Data Compression Schemes.* — IEEE Trans. Inform. Th., Vol.39, No.1.

Padmanabha Rao R. and Pearlman W.A. (1991): *On entropy of pyramid structures.* — IEEE Trans. Inform. Th., Vol.37, No.2.

Pearlman W.A. (1994): *Personal communications.*

Pittel B. (1985): *Asymptotic growth of a class of random trees.* — The Annals of Probability, Vol.13, pp.414–427.

Plotnik E., Weinberger M. and Ziv J. (1992): *Upper bounds on the probability of sequences emitted by finite state sources and on the redundancy of the Lempel Ziv algorithm.* — IEEE Trans. Inform. Th., Vol.38, pp.66–72.

Sadeh I. (1993a): *On approximate string matching.* — Proc. Data Compression Conference *DCC' 93*, Snowbird, Utah.

Sadeh I. (1993b): *Data compression in computer networks.* — Ph.D. Dissertation, Tel-Aviv University.

Sadeh I. (1994): *On lossy universal source coding.* — (submitted).

Sadeh I. (1995): *Operational rate distortion theory.* — Appl. Math. and Comp. Sci., Vol.5, No.1, pp.139–169.

Steinberg Y. and Gutman M. (1992): *An algorithm for source coding subject to a fidelity criterion, based on sring matching.* — IEEE Trans. Inform. Th., Vol.39, pp.877–887.

Storer J.A. (1990): *Lossy on Line Dynamic Data Compression.* — Springer-Verlag.

Szpankowski W. (1993): *A generalized suffix tree and its unexpected asymptotic behaviours.* — SIAM J. Computing, Vol.22, pp.1176–1198.

Welch T.A. (1984): *A technique for high performance data compression.* — IEEE Trans. Computers, Vol.17, No.6, pp.8–19.

Wyner A.D. and Ziv J. (1989): *Some asymptotic properties of the entropy of a stationary ergodic data source with applications to data compression.* — IEEE Trans. Inform. Th., Vol.35, pp.1250–1258.

Ziv J. (1978): *Coding theorem for individual sequences.* — IEEE Trans. Inform. Th., Vol.IT-24, pp.405–412.

Received: 18 September 1995
Revised: 25 February 1996