The Entropy of Recursively-Indexed Geometric Distribution

Sangsin Na, Young Kil Kim, and Haing Sei Lee

Abstract

This paper proves by straightforward computation an interesting property of a recursive indexing: it preserves the entropy of a geometrically-distributed stationary memoryless source. This result is a pleasant surprise because the recursive indexing, though one-to-one, is a symbol-to-string mapping and the entropy is measured in terms of the source symbols. This preservation of the entropy implies that the minimum average number of bits needed to represent a geometric memoryless source by the recursive indexing followed by a good binary encoder of a finite input alphabet remains the same as that by a good encoder of an infinite input alphabet. Therefore, the recursive indexing theoretically keeps coding optimality intact. For this reason recursive indexing can provide an interface for a binary code with a finite code book that performs reasonably well for a source with an infinite alphabet.

I. Introduction

A recursively-indexed binary encoder is a two-stage binary encoder that has a recursive indexing followed by a usual binary encoder. The recursive indexing used in this fashion can be thought of as a preprocessor that controls the size of the input alphabet of the ensuing binary encoder.

The necessity of this kind of preprocessor arises when the source to be encoded has the alphabet of an infinite size, because ordinary fixed-to-fixed and fixed-to-variable length binary encoders assume a finite alphabet. For example, well-known algorithms by Huffman [1], and Shannon and Fano assume the finite alphabet size, as do arithmetic coding [2] and the Ziv-Lempel algorithm [3,4]. Previous work on encoding of an infinite alphabet size are reported in [5,6,7,8].

This paper reports an interesting property of a recursive indexing: it preserves the entropy of a geometrically-distributed stationary memoryless source (i.e., an independent identically distributed (iid) random sequence) when the entropy is measured in bits per this source symbol. This result is a pleasant surprise because the recursive indexing, though one-to-one, is a symbol-to-string mapping (i.e., fixed-to-variable length) and the entropy is measured symbolwise. (The preservation of the entropy is very natural for a symbol-to-symbol mapping.) This preservation of the entropy implies that the minimum average number of bits needed to represent a geometric iid random sequence by the recursive indexing followed by a good binary encoder of a finite input alphabet remains the same as that by a good encoder of an infinite input alphabet. Therefore, the recursive indexing theoretically keeps coding optimality intact. And it is unlikely that there are other examples of this kind.

Geometric distribution is a natural model for run-length distributions, which is often used as a theoretical base for sources for facsimile communications. Specifically, the run-length distribution of white pixels in digitized weather maps is strikingly close to and hence is well-modeled by a geometric distribution. For this reason the result of this paper provides a solid background for a potentially more efficient coding scheme than the conventional modified Huffman code in the CCITT facsimile standard [9, pp.465-468]. In fact, there is a report [10] that a Huffman code experimentally designed for recursively-indexed run-length distributions for white pixels for CCITT facsimile documents outperforms the modified Huffman code by 10 to 30% for documents that include figures and graphs, and performs approximately the same for documents that have only text. This report substantiates the following: recursive indexing can and does produce a

Manuscript received June 7, 1994; accepted August 10, 1995.

This work was partially supported by grants from Ajou University and the Korea Science and Engineering Foundation under Grant 92-21-00-08.

The authors are with Department of Electronics Engineering, Ajou University, Suwon, Korea.

more efficient code for documents, particularly those with figures and graphs; and such documents are well-suited for geometric distributions.

Section 2 introduces recursive indexing and presents related notions, Section 3 applies the recursive indexing to a geometrically-distributed random variable, reaching the conclusion that the entropy is preserved by the recursive indexing, while in Section 4 numerical examples are presented. Conclusion follows in Section 5.

II. Recursive Indexing

Recursive indexing is a mapping of a countable set to a collection of sequences of symbols from another set of finite size. Given a countable set $A=\{a_0, a_1, \ldots\}$ and a finite set $B=\{b_0, b_1, \ldots, b_{M-1}\}$ of size $M\geq 2$, the recursive indexing I of A by B is a mapping of A to the collection B^* of all sequences of symbols from B such that

$$I(a_i)=\underbrace{b_{M-1}b_{M-1}\ldots b_{M-1}}_{q-\text{ times}}b_r \text{ if } i=q(M-1)+r, \tag{1}$$

where q and r are, respectively, the quotient and remainder of i when divided by M-1, the largest index in set B. The set B is called the representation set and its elements representation symbols. A recursive indexing is just a way of representing a set of symbols as a string of symbols from a smaller set by using symbols of the latter set repeatedly--this repeated use is necessary to guarantee the desirable one-to-oneness of the recursive indexing. Defined as such, recursive indexing is a one-to-one mapping and as a code it is a symbol-to-variable length, M-ary, prefix-free code and therefore uniquely and instantaneously decodable.

It is then natural to consider the derived statistic of the representation symbols. Toward this goal we first compute the number of representation symbols needed to describe a typical source sequence $X_1X_2\ldots X_n$of length n from set A. This would be the case when an iid random sequence $\{X_k\}_{k=1}^{\infty}$ with alphabet A is to be encoded. Let $p_i=\Pr(X=a_i)$, where X is a random variable that is identical to X_k. And without the loss of generality it is assumed that these probabilities are ordered so that $p_i\geq p_{i+1}$ for all i. Then the recursive indexing I of A by B can be described as in Table 1. It is noted that the output of the recursive indexing is not iid any longer.

The number n_0 of the occurrences of symbol b_0 is computed as follows. Observe that it occurs once whenever $a_0, a_{M-1}, a_{2M-2}, \ldots, a_{k(M-1)}, \ldots$ occur. The frequencies of these symbols are given by $np_0, np_{M-1}, np_{2M-2}, \ldots, np_{k(M-1)}, \ldots$, respectively. Therefore,

$$n_0=n\sum_{k=0}^{\infty}p_{k(M-1)}.$$

Table 1. The recursive indexing I of A by B.

a_i	$I(a_i)$
a_0	b_0
a_1	b_1
$\vdots$	$\vdots$
a_{M-2}	b_{M-2}
a_{M-1}	$b_{M-1}b_0$
a_M	$b_{M-1}b_1$
$\vdots$	$\vdots$
a_{2M-3}	$b_{M-1}b_{M-2}$
a_{2M-2}	$b_{M-1}b_{M-1}b_0$
a_{2M-1}	$b_{M-1}b_{M-1}b_1$
a_{2M}	$b_{M-1}b_{M-1}b_2$
$\vdots$	$\vdots$

In a similar manner the number n_j of the occurrences of symbol b_j are found to be:

$$n_j=\begin{cases} n\sum_{k=0}^{\infty}p_{k(M-1)+j} & \text{for } j=0,1,\ldots,M-2, \\ n\sum_{k=0}^{\infty}\sum_{l=0}^{M-2}kp_{k(M-1)+l} & \text{for } j=M-1. \end{cases}$$

From these it is seen that on the average the number of representation symbols needed per source symbol is

$$\frac{\sum_{j=0}^{M-1}n_j}{n}=\frac{\sum_{j=0}^{M-2}n_j+n_{M-1}}{n}=1+\sum_{k=0}^{\infty}\sum_{l=0}^{M-2}kp_{k(M-1)+l}, \tag{2}$$

where the last equality follows from the fact that

$$\sum_{j=0}^{M-2}n_j=n.$$

It is convenient to define the expansion factor ε of the recursive indexing I to be the expression in (2), i.e.,

$$\varepsilon=1+\sum_{k=0}^{\infty}\sum_{l=0}^{M-2}kp_k(M-1)+l,$$

It is the factor by which one source symbol is expanded by I due to the repeated use of the representation symbols. Note that the expansion factor will be close to 1 if M is large.

The relative frequency q_j of representation symbol b_j then is computed as follows:

$$q_j=\frac{n_j}{\sum_{i=0}^{M-1}n_i}, \quad j=0,1,\ldots,M-1. \tag{3}$$

It can be considered to be the probability mass function of the random variable induced by I from X. For this reason it is so called in the remainder of the paper.

If an optimum symbol-to-variable length binary encoder after the recursive indexing is used, it takes one representation symbol at a time and produces the corresponding binary sequence from a set of variable length code words. It is designed, for example, using the Huffman algorithm. Then its rate R_I of the recursive indexing, the average number of binary digits per *representation symbol*, is bounded as follows:

$$H(B) \le R_I < H(B)+1,$$

where *H(B)* is the entropy of the random variable induced by *I*, i.e., $H(B) = -\sum_{j=0}^{M-1} q_j \log q_j$. (All logs are base 2.) In the design of an optimum symbol-to-variable length code by the Huffman algorithm it is observed in practice that rate R_I is almost equal to *H(B)*, the lower bound, when *M* is large.

Then overall rate *R* of the recursively-indexed binary encoder, the average number of binary digits per *source symbol*, then equals εR_I, and therefore is bounded by

$$\varepsilon H(B) \le R < \varepsilon (H(B)+1).$$

Note that in general $H(X) \neq \varepsilon H(B)$ even though the recursive indexing is one-to-one. The reason is that the source symbols are represented by *strings* rather than by *symbols* and that the probabilities of *symbols*, as opposed to those of the *strings*, are used to compute *H(B)*. But we have found an interesting example that for a geometrically-distributed X, $H(X) = \varepsilon H(B)$. It is concluded that the entropy of the output of the recursive indexing in terms of bits per symbol of the original source *X* is preserved for geometric distribution. This is presented in the subsequent section.

III. Recursive Indexing on Geometric Distribution

The recursive indexing presented in the previous section is applied to a geometric distribution. The result is that the entropy of a geometric source is exactly the same as that of the random variable derived from it by the recursive indexing, when the entropy is measured in bits per original source symbol.

Let *X* be a geometrically distributed random variable with the following probability mass function:

$$p_i = \Pr(X = a_i) = (1-p)p^i \quad \text{for } i = 0, 1, \ldots.$$

Then the entropy *H(X)* of *X* is given by

$$\begin{aligned} H(X) &= -\sum_{i=0}^{\infty} p_i \log p_i \\ &= -\sum_{i=0}^{\infty} (1-p)p^i \log (1-p)p^i \\ &= -\sum_{i=0}^{\infty} (1-p)p^i (\log(1-p) + \log p^i) \\ &= -(1-p)\log(1-p) \sum_{i=0}^{\infty} p^i - (1-p)\log p \sum_{i=0}^{\infty} ip^i. \end{aligned}$$

Since $\sum_{i=0}^{\infty} p^i = \frac{1}{1-p}$ and $\sum_{i=0}^{\infty} ip^i = \frac{p}{(1-p)^2}$, we have

$$H(X) = \frac{-p\log p - (1-p)\log(1-p)}{1-p} = \frac{H(p)}{1-p},$$

where $H(p) = -p\log p - (1-p)\log(1-p)$. Fig. 1 shows the entropy of a geometrically-distributed random variable as a function of *p*. Note that it diverges to ∞ as *p* approaches 1.

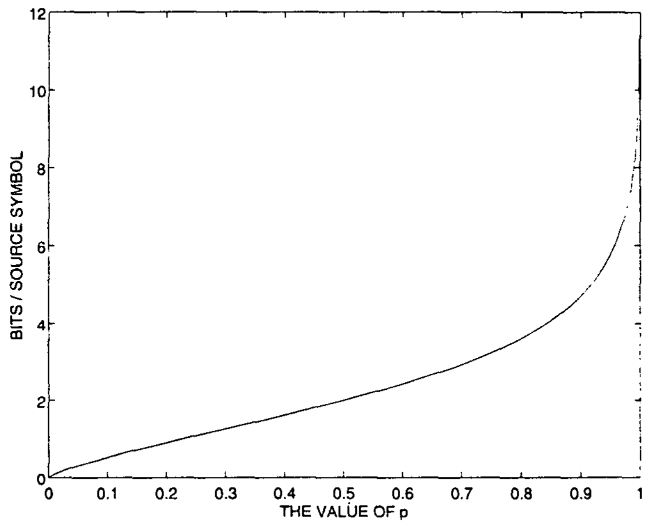


Fig. 1. The entropy of a geometrically-distributed random variable as a function of *p*. Note that it diverges to ∞ as *p* approaches 1.

Let f_j be the average number of representation symbol b_j used to represent one source symbol, i.e., $f_j = \frac{n_j}{n}$. Then the expansion factor ε in this case can be written as $\varepsilon = \sum_{j=0}^{M-1} f_j$. Then for $j = 0, 1, \ldots, M-2$, we have

$$\begin{aligned} f_j &= \sum_{k=0}^{\infty} p_{k(M-1)+j} \\ &= \sum_{k=0}^{\infty} (1-p)p^{k(M-1)+j} = \frac{(1-p)p^j}{1-p^{M-1}}. \end{aligned}$$

And for $j = M-1$

$$\begin{aligned} f_{M-1} &= \sum_{k=0}^{\infty} \sum_{l=0}^{M-2} p_{k(M-1)+l} \\ &= \sum_{k=0}^{\infty} \sum_{l=0}^{M-2} (1-p)p^{k(M-1)+l} \\ &= (1-p) \sum_{k=0}^{\infty} \sum_{l=0}^{M-2} p^{k(M-1)+l} \end{aligned}$$

$$
\begin{aligned}
&= (1-p)\sum_{k=0}^{\infty} kp^{k(M-1)} \sum_{l=0}^{M-2} p^{l} \\
&= (1-p)\sum_{k=0}^{\infty} kp^{k(M-1)} \cdot \frac{1-p^{M-1}}{1-p} \\
&= (1-p^{M-1})\sum_{k=0}^{\infty} kp^{k(M-1)} \\
&= (1-p^{M-1})\frac{p^{M-1}}{(1-p^{M-1})^2} = \frac{p^{M-1}}{1-p^{M-1}},
\end{aligned}
$$

where the first equality in the last line follows from the fact that $\sum_{k=0}^{\infty} kr^{k} = \frac{r}{(1-r)^2}$. Then the expansion factor ε is given by

$$
\varepsilon = \sum_{j=0}^{M-1} f_j = 1 + \frac{p^{M-1}}{1-p^{M-1}} = \frac{1}{1-p^{M-1}}.
$$

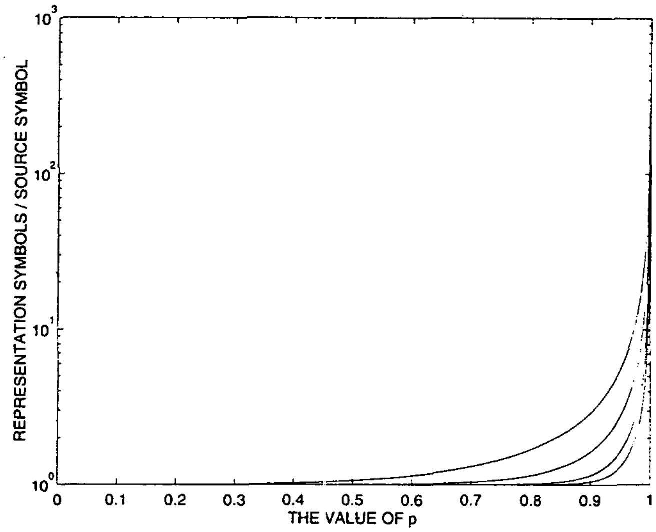


Fig. 2. The expansion factor ε of the recursive indexing for a geometrically-distributed iid source for various size of the representation set $M = 5, 10, 20, 30$ from the top curve to the bottom.

The last expression shows that for large M (i.e., for a representation set of a large alphabet) the recursive indexing has the expansion factor close to 1 and hence does not expand the source symbols very much. Also it can be seen that p is an indicator of the convergence rate of ε to 1---it determines how rapidly ε approaches 1. Fig. 2 shows the expansion factor for various values of M. As M gets bigger, the expansion factor stays close to 1 for a wider range of p. Note that, as p approaches 1, the expansion factor becomes arbitrarily large.

The probabilities q_j induced by the recursive indexing then are given by

$$
q_j = \frac{f_j}{\varepsilon} = \begin{cases} (1-p)p^{j} & \text{for } j = 0, 1, \ldots, M-2, \\ p^{M-1} & \text{for } j = M-1. \end{cases}
$$

Then $H(B) = -\sum_{j=0}^{M-1} q_j \log q_j$ is computed as follows:

$$
\begin{aligned}
H(B) &= -\Big(\sum_{j=0}^{M-2} q_j \log q_j + q_{M-1}\log q_{M-1}\Big) \\
&= -\Big(\sum_{j=0}^{M-2}(1-p)p^{j}\log((1-p)p^{j}) p^{M-1}\log p^{M-1} \\
&= -\Big\{(1-p)\log(1-p)\sum_{j=0}^{M-2} p^{j} + (1-p)\log p \sum_{j=0}^{M-2} jp^{j} \\
&\quad + p^{M-1}\log p^{M-1}\Big\}
\end{aligned}
$$

The summations in the first and second terms of the above can be rewritten as

$$
\begin{aligned}
\sum_{j=0}^{M-2} p^{j} &= \frac{1-p^{M-1}}{1-p}, \\
\sum_{j=0}^{M-2} jp^{j} &= \frac{p(1-p^{M-2})}{(1-p)^2} - (M-2)\frac{p^{M-1}}{1-p}.
\end{aligned}
$$

Then we have

$$
\begin{aligned}
H(B) = -\Big\{&(1-p^{M-1})\log(1-p) + \frac{1-p^{M-2}}{1-p} p\log p - (M-2)p^{M-1}\log p \\
&+ p^{M-1}\log p^{M-1}\Big\}.
\end{aligned}
$$

Further straightforward simplification yields

$$
\begin{aligned}
H(B) &= (1-p^{M-1})\frac{-(1-p)\log(1-p) - p\log p}{1-p} \\
&= (1-p^{M-1})\frac{H(p)}{1-p} = \frac{1}{\varepsilon}\cdot\frac{H(p)}{1-p}.
\end{aligned}
$$

Therefore, we have

$$
H(X) = \varepsilon H(B). \tag{4}
$$

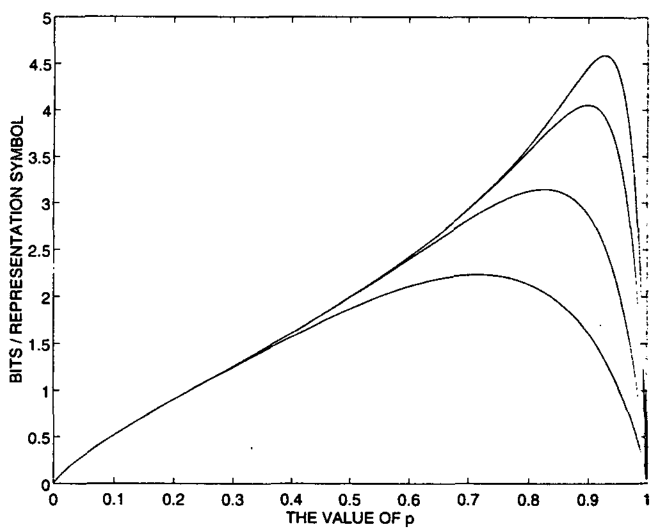


Fig. 3. The entropy $H(B)$ of the recursive indexing for a geometrically-distributed iid source for various size of the representation set $M = 5, 10, 20, 30$ from the bottom curve to the top.

Eq. (4) shows that for the geometric distribution the recursive indexing preserves the entropy. Fig. 3 depicts the entropy *H(B)* of the recursive indexing for geometric distribution for various values of *M*. As *M* gets bigger, the maximum entropy gets bigger and it is achieved at a bigger value of *p*. In interpreting (4) a little care must be taken: the right hand side is in bits per source symbol because the unit of ε is the number of representation symbols per source symbol and that of *H(B)* in bits per representation symbols. Therefore a careful interpretation would be that the recursive indexing preserves the minimum average length of representation of a geometrically-distributed iid source.

IV. Numerical Examples

In this section a numerical example of a geometric distribution is presented for recursive indexing and an ensuing binary code. This combination is so taken to point out that the resulting coding scheme can be truly optimum in the sense of the minimum average codeword length. Another example is presented for a Poisson distribution, which also has an infinite alphabet. However, the recursive indexing does not preserve the entropy in this case, thereby showing that geometric distribution is a special case where the recursive indexing preserves the source entropy.

A. Geometric Distribution

Let *X* be a geometric source with

$$p_i = \Pr(X=i) = \left(\frac{1}{2}\right)\left(\frac{1}{2}\right)^i, \text{ for } i=0,1,2,\ldots.$$

Table 2. The recursive indexing *I* of $A=\{0,1,2,\ldots\}$ by $B=\{0,1,2,3\}$.

$i \in A$	$I(i)$
0	0
1	1
2	2
3	30
4	31
5	32
6	330
7	331
⋮	⋮

This source has alphabet $A=\{0,1,2,\ldots\}$ and the entropy $H(X)=2$ bits/source symbol. Let the representation set $B=\{0,1,2,3\}$. Then *M=4* and the output of the recursive indexing must be composed of symbols from set *B*. The resulting recursive indexing of A by *B* then is a mapping described in Table 2.

Then applying the results of the previous section the induced probabilities q_i of the representation symbols are found to be

$$q_0=\frac{1}{2},\ q_1=\frac{1}{4},\ q_2=\frac{1}{8},\ q_3=\frac{1}{8}$$

with the expansion factor

$$\varepsilon=\frac{1}{1-\left(\frac{1}{2}\right)^{4-1}}=\frac{8}{7} \text{ representation symbols/source symbol}$$

This expansion factor shows that on the average one source symbol is represented by $\frac{8}{7}$ representation symbols. Note that the entropy $H(B)=\frac{7}{4}$ bits/representation symbol. Hence, $\varepsilon H(B)=\frac{8}{7}\cdot\frac{7}{4}=2=H(X)$, as expected. It must be noted that the equality always holds as long as $M\geq 2$, which is always satisfied if meaningful communication is intended.

This example reveals a further result: in some cases the optimum code for the *recursive indexing* is actually an optimum code for the *source* as well. The best fixed-to-variable length binary code C_q for the probabilities q_i can be designed by the Huffman algorithm, which gives

$b \in B$		$C_q(b)$
0	→	0
1	→	10
2	→	110
3	→	111

The rate R_I of code C_q is then

$$1\times\frac{1}{2}+2\times\frac{1}{4}+3\times\frac{1}{8}+3\times\frac{1}{8}=\frac{7}{4} \text{ bits/representation symbol.}$$

Then the overall code C_p for the source $\{0,1,2,\ldots\}$ is the cascade connection of the recursive indexing *I* and the binary code C_q for the representation set, i.e., $C_p(i)=C_q(I(i))$. The operation of the overall code is illustrated as follows: (note that the vertical bars '|' are used for the illustration purpose only to distinguish the individual symbols and their corresponding counterparts in the sequences) for a source sequence

0|1|2|0|8|0|0|1|0|7|4|3|2|···

The corresponding output of the recursive indexing *I* is:

0|1|2|0|332|0|0|1|0|331|31|30|2|···

And the output of the overall code is:

0|10|110|0|111111110|0|0|10|0|11111110|11110|1110|110|···

Note that for example the sequence 332 is encoded by C_q as

$$C_q(332) = C_q(3)\,C_q(3)\,C_q(2) = 111111110$$

The rate of the overall code C_p for the source then is

$$R = \varepsilon R_l = \frac{8}{7} \cdot \frac{7}{4} = 2 = H(X).$$

Since the code rate R achieves the source entropy, we know for sure that no other code can perform better than this code. However, it should be noted that unlike this fortunate case the optimality for the source is *not* in general guaranteed by the optimality for the recursive indexing. Nonetheless, we have the freedom of choice over the value M so that virtual optimality can be obtained by choosing a bigger M whenever it is desired.

A. Poisson Distribution

The second example is a Poisson distribution:

$$p_i = e^{-\lambda} \cdot \frac{\lambda^i}{i!}, \text{ for } i = 0, 1, 2, \ldots$$

Let $\lambda = 1$ for simplicity. Again we see that the source has an infinite alphabet $A = \{0, 1, 2, \ldots\}$ and the entropy $H(X) = 1.8825$ bits/source symbol. For the representation set $B = \{0, 1, 2, 3\}$ we obtain the same recursive indexing of A by B as in the previous example, i.e., the mapping described in Table 2. And the induced probabilities q_i of the representation symbols are found to be

$$q_0 = 0.3975, q_1 = 0.3546, q_2 = 0.1730, q_3 = 0.0748$$

with the expansion factor $\varepsilon = 1.0809$ representation symbols/source symbol. Then the entropy $H(B)$ is found to be 1.7773 bits/representation symbol. Hence, $\varepsilon H(B) = 1.9210 \neq H(X) = 1.8825$. Therefore, this examples serves as a distribution for which the recursive indexing does not preserve the source entropy.

V. Conclusion

A recursively-indexed binary encoder is presented as a preprocessor to reduce the size of representation symbols for the subsequent binary encoder. It is a symbol-to-string one-to-one mapping. In general the entropy of the output of the recursive indexing in bits per original source symbol is not the same as that of the source, because it is not a symbol-to-symbol one-to-one mapping. However, for a geometric distribution the entropy of the source is preserved by the recursive indexing. Because of its entropy preservation in such a case the recursively-indexed binary encoder can, by controlling the size of the representation set, control its rate without suffering the loss in coding optimality. Therefore, it seems to be a solution to the optimum coding of a geometrically-distributed iid source, for which no optimum symbol-to-string binary encoder can be designed without a compromise in optimality due to the infinite source alphabet.

References

[1] D. Huffman, "A method for the construction of minimum redundancy codes," *Pro. IRE*, pp. 1098--1101, Sept. 1952.

[2] I. Witten, R. Radford and J. Cleary, "Arithmetic coding for data compression," *Comm. ACM*, Vol. 30, pp. 520--540, June 1987.

[3] J. Ziv and S. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. on Inform. Theory*, pp. 337--343, May 1977.

[4] J. Ziv and S. Lempel, "Compression of individual sequence via variable-rate coding," *IEEE Trans. on Inform. Theory*, pp. 530-- 536, Sept. 1978.

[5] K.~Sayood and S. Na, "Recursive quantization," *24th Annual Asilomar Conference on Circuits, Systems, and Computers*, Nov. 1990.

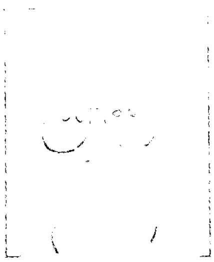
[6] K. Sayood and S. Na, "Recursive indexing and its application," *Proceedings of the 1991 International Symp. on Inform. Theory*, Budapest, Hungary, June 1991.

[7] K. Sayood and S. Na, "Recursively indexed quantization of memoryless sources," *IEEE Trans. on Inform. Theory,* pp. 1602--1609, Sept. 1992.

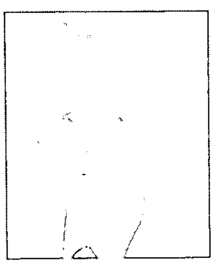
[8] K. Sayood and S. Na, "Recursively indexed differential pulse code modulation," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science Volume 14: Coding and Quantization*, pp.253--263, 1993.

[9] N. Jayant and P. Noll, *Digital Coding of Waveforms,* Prentice-Hall, New Jersey, 1984.

[10] J. Suh, private communication, 1995.



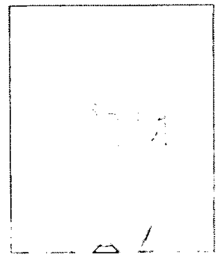
Sangsim Na was born in Illo, Chunnam, in 1959. He received the B.S. degree in electronics engineering from Seoul National University in 1982, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 1985 and 1989, respectively. From 1989 to 1991 he was at the Department of Electrical Engineering, the University of Nebraska, Lincoln. Since 1991, he has been with Ajou University and is currently an Associate Professor of Electronics Engineering. His research interests are in the area of information theory, data compression, digital communications, and signal processing.



Haing Sei Lee was born in Kunsan, Korea in 1943. He received the B.S. degree in electrical engineering from Chonpuk National University in 1966. He received the M.S. degree in electronic engineering from Seoul National University in 1972 and the Ph.D. degree in electronic engineering from Korea University in 1984. Since 1973, he has been with the department of electronic engineering at Ajou University, where he is currently a professor. From 1982 to 1983, he was a visiting professor at Columbia University, New York. From November 1987 to March 1988, he was a visiting professor at INRIA, Paris. From March 1992 to February 1994, he was the dean of Koje Junior College.

His main research interests include character recognition, speech recognition, neural network, digital signal processing, and man-machine interface.

Dr. Lee is a member of IEEE, the Korea Institute of Telematics and Electronics, and Acoustical Society of Korea.



Young-Kil Kim was born in Seoul Korea in 1954. He received the B.S. degree in Electronics Engineering from Korea University in 1978, the M.S. degree in Korea Advanced Institute of Science in 1980, and the Ph.D. degree in Electronic Engineering from the École Nationale Supérieure des Telecommunications of Paris, France, in 1984. Since 1980, he has been with the Department of Electronics Engineering at Ajou University, Suwon, where he is now a professor. His research interests are signal processing for underwater communications, biomedical instruments, and BISDN.

He is a member of the Korean Institute of Telematics and Electronics, the Korean Society of Medical and Biological Engineering, and Acoustical Society of Korea.