LETTER *Special Section on Letters Selected from the 1995 IEICE Spring Conference*

An Adaptive Coding-based Selection Scheme for a Communication Aid

Satoshi KOYAMA†, ***Member***

SUMMARY This paper discusses a coding-based selection approach to a communication aid for the severely motor disabled. Several approaches including row-column scanning are briefly described, then we propose a new selection scheme based on the theory of adaptive coding. They are compared each other with respect to average switch activations in generating some text samples.
key words: *human interface, motor disabled, communication aid, scanning, coding*

1. Introduction

It is difficult to generate text by operating a conventional keyboard for the severely motor disabled. In such cases, communication aids are widely used for the disabled, and row-column scanning is a typical selection scheme in those devices.

Buhr and Holte[1] and Damper[2] independently pointed out that row-column scanning can also be seen as coding. Huffman's algorithm gives an optimal solution to minimize average switch activations[1], in which statistics of the source are required to be known.

In this paper, several coding-based selection schemes including row-column scanning are summarized. Then we propose a new selection scheme based on the theory of dynamic Huffman coding[3], which requires no *a priori* information of the source. They are compared each other with respect to average switch activations in generating some text samples.

2. Selection Schemes Based on Coding

It is assumed in the following discussion that the user can control two switches correctly as intended. Of course, these switches (or sensors) must be appropriately customized for the user.

Suppose an alphabet is composed of M-letters $a_1, a_2, ..., a_M$. Each a_i occurs n_i times in a given text, and the size of the text is $N = \sum_{i=1}^{M} n_i$. The code length (or switch activation times) of a_i is denoted by l_i.

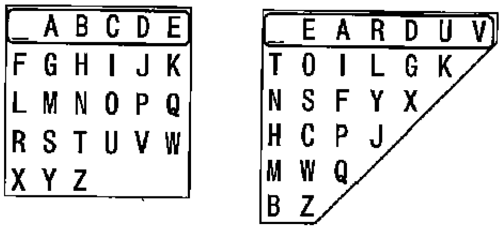

Fig. 1 The row-column scanning arraies.

(**Code A**) Alphabetically Ordered Row-column Scanning

Fig.1(a) shows a typical prompting array of row-column scanning, in which letters are alphabetically ordered. Prompting cursor starts from top-leftmost position each time, and two switches named `Next` and `Select` are used to valid a desired letter.

In order to access the letter '`H`' we first select second row by `Next-Select`, then select third column by `Next-Next-Select`. As a result, five switch activations are required. If '`0`' is assigned to `Next` and '`1`' is assigned to `Select`, the sequence to select '`H`' is "`01001`". That is to say, '`H`' is coded into "`01001`". Whole code set of the array of Fig.1(a) is shown in Table 1.

When some text is given, the average code length (average switch activations) is $\overline{l_A} = \frac{1}{N} \sum_{i=1}^{M} n_i l_{Ai}$.

(**Code B**) Statistically ordered row-column scanning

Letters placed on the upper-left corner of the array of Fig.1(a) require only a few switch activations. So frequently occurring letters should be placed there. Such modified row-column scanning array is shown in Fig.1(b), in which letters are statistically ordered[2]. Resulting code set is also shown in Table 1.

When some text is given, the average code length is $\overline{l_B} = \frac{1}{N} \sum_{i=1}^{M} n_i l_{Bi}$.

(**Code C**) Equi-length coding

Row-column scanning described above corresponds to a special class of binary coding in the sense that every

Manuscript received April 21, 1995.
Manuscript revised July 4, 1995.
†The author is with the Faculty of Education, Hirosaki University, Hirosaki-shi, 036-8560 Japan.

Table 1 Coding table in the case, $M = 27$: (**Code A**) alphabetically ordered row-column scanning, (**Code B**) statistically ordered row-column scanning, (**Code C**) equi-length binary coding, and (**Code D**) Huffman coding.

letter	Code A	Code B	Code C	Code D
_	11	11	00000	000
A	101	1001	00001	0100
B	1001	0000011	00010	0011011
C	10001	000101	00011	11100
D	100001	100001	00100	01100
E	1000001	101	00101	101
F	011	001001	00110	11011
G	0101	0100001	00111	001111
H	01001	00011	01000	1111
I	010001	01001	01001	1001
J	0100001	00010001	01010	0011010101
K	01000001	01000001	01011	0011010100
L	0011	010001	01100	01101
M	00101	000011	01101	11101
N	001001	0011	01110	0111
O	0010001	0101	01111	0101
P	00100001	0001001	10000	110100
Q	001000001	00001001	10001	0011010011
R	00011	10001	10010	1000
S	000101	00101	10011	1100
T	0001001	011	10100	0010
U	00010001	1000001	10101	001100
V	000100001	10000001	10110	001101000
W	0001000001	0000101	10111	110101
X	000011	00100001	11000	0011010010
Y	0000101	0010001	11001	001110
Z	00001001	00000101	11010	001101011

code-word has two `1`'s and terminate in `1`. Henceforth other binary coding schemes are examined.

First, equi-length binary coding provides a baseline for comparison. The code set is shown in Table 1, and the length of the code is $\overline{l_C} = \lceil log_2 M \rceil$.

(**Code D**) Huffman coding

Huffman's algorithm gives an optimal code set, which leads the average switch activations minimum[1]. Shorter code words are assigned to frequently occurring letters like **Code B**, in which statistics of the source are required to be known.

When some text is given, the average code length is

$$\overline{l_D} = \frac{1}{N} \sum_{i=1}^{M} n_i l_{Di} \ .$$

3. Selection Scheme Based on Adaptive Coding

As shown in the previous section, optimal row-column scanning is provided by **Code B**, and optimal binary selection is provided by **Code D**. In both cases, statistical model (letter-based i.i.d. source model with known parameters) is assumed. Although optimality will hold under this model, shorter representation of the text the user now generate is actually desired.

When Huffman coding is implemented in the file compression system, two-pass approach is often used, in which M-letters are counted in the first pass and compression is done in the second pass. But this approach cannot be available in a communication aid because we have no means foreseeing the text.

So it is meaningful to introduce an adaptive approach.

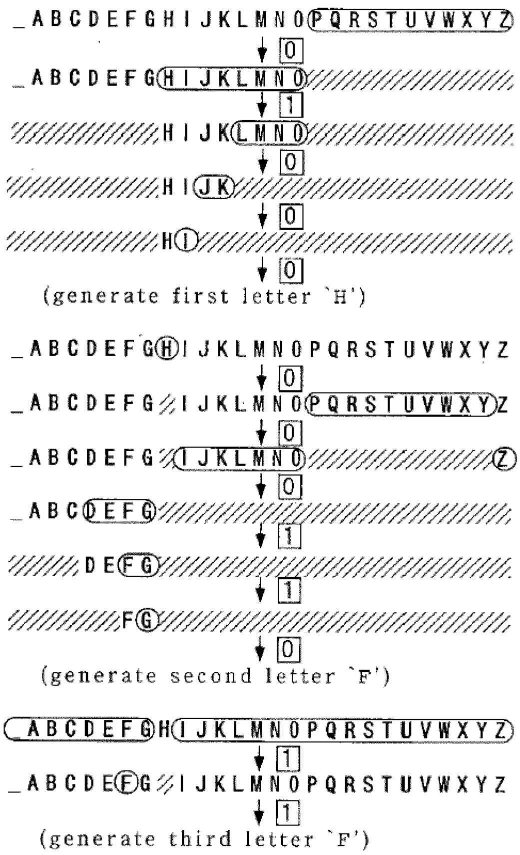

Fig. 2 An example of the adaptive selection scheme: prompting display and switch activations to generate "`HFF`".

(**Code E**) dynamic Huffman coding

We present a new adaptive selection scheme applied to a communication aid based on the theory of dynamic Huffman coding. The algorithm is based on [3].

As an example, prompting display and switch activations to generate "`HFF`" are illustrated in Fig.2, and corresponding coding trees and paths are also depictied in Fig.3.

Letters of the prompting display are divided into two groups at each stage. At every stage, the letters in the circles of Fig.2 belong to right-hand subtree of the current node and are selected when '`1`' is activated. The other letters belong to left-hand subtree and are selected when '`0`' is activated. Groups may be discriminated by colors instead. As selections proceed, the letters which were not selected are omitted (shaded in Fig.2).

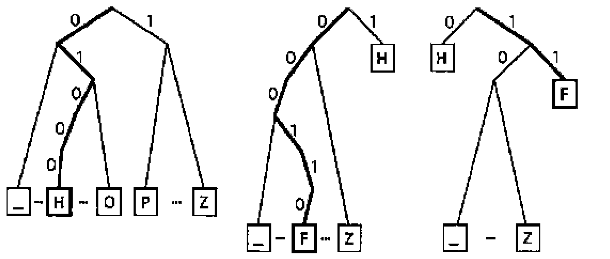

Fig. 3 Coding trees and paths to generate "HFF" corresponding to Fig.2.

Four or five operations are required for generating first letter. This is reasonable because we have no knowledge about occurrence of any letters at the initial stage. First letter 'H' is generated by the sequence "01000" as in Fig.2 and Fig.3(a). At this time, 'H' is a most frequently letter and one switch activation will generate 'H'(see Fig.3(b)). In our example, the second letter 'F' is generated by the sequence "000110" as in Fig.2 and Fig.3(b). At this time, 'H' and 'F' both are most frequently, so a few operations will generate each of those. In fact, the third letter 'F' is generated by the sequence "11" as in Fig.2 and Fig.3(c). Every time each letter is generated, coding tree is modified optimally according to the occurrence of letters in the text already appeared so far.

This method successfully learns the statictics of the text now generated. So there requires no knowledge about the source *a priori*, although letter-based i.i.d. model is still assumed.

When some text is given, the average code length is $\overline{l_E} = \frac{1}{N}\sum_{i=1}^{N} l^{(i)}$, in which $l^{(i)}$ is the code length of i-th letter of the text.

4. Comparison of selection schemes

In this section, several coding-based selection schemes described above are compared each other with respect to the average switch activations for generating three different types of text samples: CH1, NOVEL and PROG †.

Of each text, two alphabet sizes, $M = 97$ and $M = 27$, are exploited ††. Statistical parameters used in **Code B** and **Code D** refer to [2] in the case, $M = 27$, and [4] in the case, $M = 97$.

The result is shown in Table 2. The entropy $H = -\sum_{i=1}^{M} \frac{n_i}{N} log_2 \frac{n_i}{N}$ is also shown in the table, which is a goal of the letter-based i.i.d. model.

Table 2 Comparison of the average activations(average code length) in generating three text samples CH1, NOVEL, and PROG.

	$M = 97$			$M = 27$		
text	CH1	NOVEL	PROG	CH1	NOVEL	PROG
(size)	6680	11158	10666	6319	10685	7401
$\overline{l_A}$	11.01	11.27	9.59	5.10	5.20	5.00
$\overline{l_B}$	4.86	4.50	6.98	4.25	4.28	4.39
$\overline{l_C}$	7	7	7	5	5	5
$\overline{l_D}$	4.76	4.44	6.88	4.15	4.16	4.17
$\overline{l_E}$	4.67	4.46	5.30	4.08	4.15	4.13
H	4.57	4.37	5.21	4.03	4.09	4.06

Statistically ordered scanning of **Code B** is superior to alphabetically ordered scanning of **Code A**, and is further improved a little by adopting Huffman coding of **Code D**. However, the performance is not so good in the case of PROG, because program source text is not modeled so well by the parameters used here. Even such cases, dynamic Huffman coding of **Code E** goes well.

Meaningful differences between CH1 and NOVEL cannot be observed in all cases, which will be stated so long as letter-based model is used.

The significant improvement in the case, $M = 27$, is only observed between **Code A** (also **Code C**) and **Code B**. This is because frequencies of letters are not so widely distributed than the case, $M = 97$.

5. Conclusion

As we have seen, average switch activations of **Code E** are reduced to 40–80% of the conventional alphabetically ordered row-column scanning and almost achieve the entropy. This adaptive scheme goes well in generating any text. Advantage will be distinguished when alphabet size M is large and no knowledge about the source is given. Analitical considarations like this seems to be important for the future communication aid design.

†CH1 is an elementary text by which we first learn English language at junior high school in Japan. NOVEL is a part of some novel. PROG is a special text of some source program.

††There seems to be required at least 27-letters for human communication by text. The original text samples are converted by some rules in the case, $M = 27$.

References

[1] P.A. Buhr, and R.C. Holte, "Some considerations in the design of communication aids for the severely physically disabled," Medical & Biological Engineering & Computing, no.19, pp.725–733, 1981.
[2] R.I. Damper, "Text composition by the physically disabled: A rate prediction model for scanning input," Applied Ergonomics, vol.15, no.4, pp.289–296, 1984.
[3] J.S. Vitter, "Design and analysis of dynamic Huffman codes," J. ACM, vol.34, no.4, pp.825–845, 1987.
[4] T.C. Bell, J.G. Cleary, and I.H. Witten, "Text compression," Prentice-Hall, 1990.