

# リバーシブル符号とその誤り耐性を有する動画像符号化への応用

中條 健<sup>†</sup> 渡邊 敏明<sup>†</sup>

## Reversible Code and Its Application to Error-Tolerant Video Coding

Takeshi CHUJOH<sup>†</sup> and Toshiaki WATANABE<sup>†</sup>

**あらまし** 双方向に瞬時復号可能な可変長符号であるリバーシブル符号について検討する。リバーシブル符号は、ビットストリームの最初と最後から双方向に復号することができるので、もし、ビットストリームに誤りがあったとしても、復号できないビット数を減らすことができる。本論文では、まず、完全な可算無限アルファベットのリバーシブル符号が存在することを示し、符号語のハミング重みを利用した新しい完全な可算無限アルファベットのリバーシブル符号の構成方法を提案する。更に、提案した構成方法で設計したリバーシブル符号を誤り耐性を有する動画像符号化方式に適用し、その効果を示す。

**キーワード** リバーシブル符号, 双方向復号, 可変長符号, 動画像符号化, 伝送路誤り

### 1. まえがき

近年の移動体通信の普及はめざましく、今後は音声やデータばかりでなく、動画像の伝送の可能性も期待されている。そのためには、移動体通信のように伝送路の状態が厳しい条件においても、高品質の動画像伝送ができることが望まれる。ところで、動画像符号化のこれまでの標準方式である H.261, MPEG1, MPEG2 には、可変長符号が用いられる。可変長符号化の問題点としては、符号化したビットストリームに誤りが混入した場合、誤りが伝搬してしまうことがある。そのため、大量の正しいビットストリームが復号できなくなってしまう。

可変長符号の誤り対策としては、自己同期符号等がよく知られているが、ここでは、可変長符号の別の誤り対策として、双方向に瞬時復号可能な符号を用いて検討した結果について述べる。双方向に瞬時復号可能な符号とは、通常順方向に瞬時復号可能な性質に加えて、符号語を逆方向からも復号できる性質をもった符号のことである。本論文では、このような符号をリバーシブル符号と呼ぶこととする。

例えば、図 1 のようにリバーシブル符号ならば、ビットストリームを最初と最後の双方向から復号をす

ることができる。このことにより、誤りが混入したため、順方向からでは復号できないビットストリームも逆方向から復号することで、誤り部分を低減することができる。

リバーシブル符号の存在については、情報理論研究の初期の段階で既に Schützenberger [1] によって Anagrammatic Codes として示唆されており、また、Perrin [2] によって、Biprefix Code として研究されている。また、Fraenkel と Klein [4] は、双方向に瞬時復号可能な符号を Affix Code と呼び、与えられた符号語長の符号を探索するアルゴリズムの提案してお

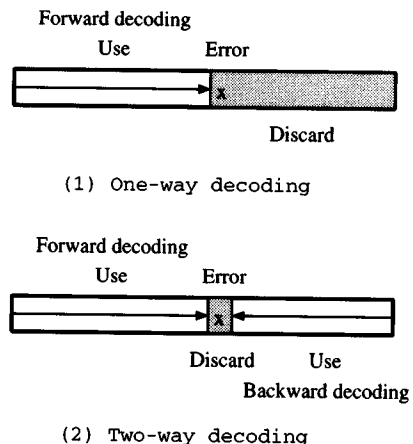


図 1 双方向復号による誤り耐性  
Fig.1 Error tolerance by two-decoding.

<sup>†</sup> (株) 東芝研究開発センター, 川崎市  
Research & Development Center, TOSHIBA CORPORATION,  
Kawasaki-shi, 210 Japan

り, Takishimaら [5]は, Reversible Variable Length Code という名称で, Huffman 符号の符号語を入れ換えて可能な限り双方向に瞬時復号をできるようにした上で, うまくいかない符号語には Suffix にビットを追加して符号を構成する方法などいくつかの手法を示している.

但し, 有限なアルファベット数において, 完全な符号木をもつリバーシブル符号の数が限られる [1], [4] ことから, 情報圧縮としての実用的な意味での研究は十分とは言えない.

本論文では, リバーシブル符号が可算無限アルファベットのときに数多く存在することに着目して, 符号語のハミング重みを利用する新しい符号構成方法を提案する. この手法は, 与えられた確率に対して, 最適であるという保証はないが, 実用的には符号の設計が容易であり, 圧縮性能としても, 従来の符号と比較して十分に効果がある手法であることを示す.

また, リバーシブル符号を動画像符号化の伝送路誤り対策として適用する. 具体的には, ITU-T H.263 [6] をベースとしてビットストリームをフレーム単位に階層化し, 伝送路誤り対策を施した動画像符号化方式 [12] の DCT 係数の可変長符号化にリバーシブル符号を適用して, 誤り耐性能力を実験する.

## 2. 基本的性質

瞬時復号可能な符号が, 次式のように Kraft の不等式を等号で満足する場合, その符号には冗長なビットパターンが存在しない. ここで,  $c(i)$  は, 符号語長  $i$  ビットの符号語数,  $m$  は最大符号語長を表している.

$$\sum_{i=1}^m c(i)2^{-i} = 1 \tag{1}$$

ここでは, このような符号を完全な符号と呼ぶこととする.

有限アルファベットの無記憶情報源に対する最適符号化である Huffman 符号は, 2元の場合, 必ず完全な符号になることが知られている.

そのため, リバーシブル符号の性質としても, 有限アルファベットで順方向からも逆方向からも双方向で完全な符号が望まれる. 例えば, 2 のべき乗のアルファベット数のときの固定長符号は, 明らかに双方向で完全な符号である.

しかし, 可変長符号で有限アルファベットで完全なリバーシブル符号の場合, その存在はかなり制限され

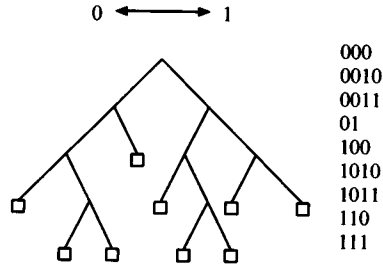


図2 完全なリバーシブル符号の例  
Fig.2 Example of complete reversible code.

る. 現状では, 効率的な作り方が存在しておらず, 計算機探索によって発見されているにすぎない. 例えば, アルファベット数 16 以下では, 図 2 に示す符号しか存在しないことが知られている.

これまでの研究において, いくつかの符号語長の分布の条件 [4], [9] が示されており, アルファベットのさまざまな生起確率分布に対応する符号の設計は困難であることは明らかである.

そこで本論文では, 次のリバーシブル符号の性質に着目する.

[定理 1] 次の式を満足するリバーシブル符号が存在する.

$$\sum_{i=1}^{\infty} c(i)2^{-i} = 1 \tag{2}$$

(証明) 次のような可算無限アルファベットの符号を考える.

0,11,101,1001,10001,...

この符号は, 図 3 で示すように瞬時復号可能な符号である. また, この符号は, 符号語のビットパターンが左右対称であることから, 順方向の符号木も逆方向の符号木も同一になることから, 双方向に瞬時復号可能なことがわかる. この符号の符号語長  $i$  ビット ( $i = 1, 2, \dots$ ) の符号語数は,  $c(i) = 1$  なので, これを式 (2) の左辺に代入すると,

$$\sum_{i=1}^{\infty} 2^{-i} = 1 \tag{3}$$

となる.

従って, 式 (2) を満足するリバーシブル符号が存在する. □

ここでは, 定理 1 を満足するリバーシブル符号のことを完全な可算無限アルファベットリバーシブル符号

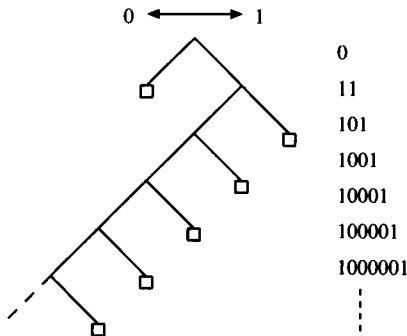


図3 完全な可算無限アルファベトリバーシブル符号  
Fig.3 Complete reversible code for countable infinite alphabets.

と呼ぶことにする。

定理1で述べていることは、式(1)のアルファベット数が有限であるという制約をゆるめて、可算無限個にしたときに、冗長なビットパターンがなくなるという性質をもつリバーシブル符号が存在するということである。実用的に有用なアルファベット数は有限であるから、このような符号は意味がないように考えられる。しかし、定理1の性質をもつリバーシブル符号の一部を使って、有限アルファベットの符号として使用した場合にも、従来の手法によるリバーシブル符号よりも有利な性能をもっている。なぜならば、アルファベット数が増えるにつれて、冗長なビットパターンがなくなっていくという性質をもっているため、冗長なビットパターンが比較的少ない符号が設計できるからである。また、有限アルファベットで完全なリバーシブル符号とは違い、次章で示す構成方法によって、さまざまな確率分布に対応することができることから、実用的に有用な符号を設計することが可能となる。

### 3. 可算無限アルファベトリバーシブル符号

ここでは、完全な可算無限アルファベトリバーシブル符号の構成方法について述べる。具体的には、復号する方向に依存しない値である符号語のハミング重みを利用して、双方向に瞬時復号可能な可変長符号を構成する。また、固定長符号と組み合わせ、リバーシブル符号を拡張する方法を示す。

#### 3.1 重み一定可変長符号

ハミング重みが一定になる2進系列を用いてリバーシブル符号を構成する方法を二つ提案する。

ハミング重みが一定になる2進系列を作成するア

ルゴリズムとしては、Schalkwijkのアルゴリズム[8]等が知られている。このアルゴリズムでは、ハミング重み  $w$  ( $0 \leq w \leq n$ ) で、 $n$  ビットの2進系列  $(u_0, u_1, \dots, u_{n-1})$  の順序値  $s$  の符号化、復号化ができる。

[重み一定可変長符号(1)の構成方法]

Step 1 符号語長  $n$  ( $n = w - 2, w - 1, w, \dots, w \geq 2$ ) の順に、ハミング重み  $w - 2$  の2進系列を作成する。  
Step 2 Step 1で作成した2進系列のPrefixとSuffixの両方に“1”を付け、最も短い符号語として、“0”を割り当てる。 □

なお、定理1の証明に用いた符号は、実は、この符号の  $w = 2$  の場合に相当する。

図4は、 $w = 3$  の場合の重み一定可変長符号(1)の符号語の構成を示している。まず、重み1 ( $= w - 2$ ) の2進系列を1ビットのものから、2ビット、3ビット...と作成する。次にその2進系列のPrefixとSuffixに“1”を付け、最も短い符号語として、“0”を割り当てている。

[定理2] 重み一定可変長符号(1)は、完全な可算無限アルファベトリバーシブル符号である。

(証明) この符号では、最も短い符号のときは、符号語は“0”、それ以外のときは、符号語は“1”から必ず始まり、“1”が全部  $w$  個出現したら、終わりがわかるような符号の構成になっている。符号語の“1”の数は、復号する方向に依存せず、また、同一の符号語長の符号語は、同じビットパターンにはならないので、重み一定可変長符号(1)は、双方向に瞬時復号可能である。

一番最初の符号語を除くと  $w$  を符号語のハミング重みとして、符号語長  $i + w$  ビット ( $i = 0, 1, \dots$ ) の符号語数  $c(i + w)$  は、符号語長  $i + w - 2$  ビットの中から  $w - 2$  個の“1”を選択する組合せの数になるので、

$$c(i + w) = \binom{i + w - 2}{w - 2} \quad i = 0, 1, 2, \dots \quad (4)$$

で与えられる。これを式(2)の左辺に代入すると、2項係数の母関数[10]の変形から、

$$2^{-1} + \sum_{i=0}^{\infty} \binom{i + w - 2}{w - 2} 2^{-(i+w)} = 1 \quad (5)$$

が成立するので、この符号は、可算無限アルファベットに拡張したときに Kraft の不等式を等号で満足する。

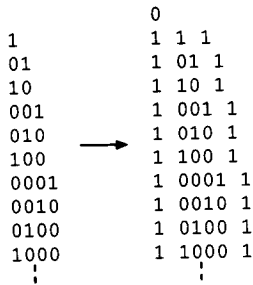


図4 重み一定可変長符号 (1) の構成 ( $w = 3$ )  
 Fig.4 Construction of constant weight variable-length code (1) ( $w = 3$ ).

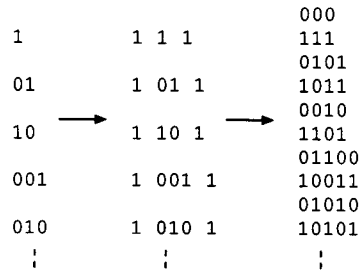


図5 重み一定可変長符号 (2) の構成 ( $w = 3$ )  
 Fig.5 Construction of constant weight variable-length code (2) ( $w = 3$ ).

従って、完全な可算無限アルファベットリバーシブル符号である。 □

[重み一定可変長符号 (2) の構成方法]

Step 1 符号語長  $n$  ( $n = w - 2, w - 1, \dots, w \geq 2$ ) の順に、ハミング重み  $w - 2$  の 2 進系列を作成する。

Step 2 Step 1 で作成した 2 進系列の Prefix と Suffix の両方に “1” を付ける。

Step 3 Step 2 の符号語のビット反転した符号語を追加する。 □

図 5 は、 $w = 3$  の場合の重み一定可変長符号 (2) の構成を示している。まず、重み 1 ( $= w - 2$ ) の 2 進系列を 1 ビットのものから、2 ビット、3 ビット、... と作成する。次にその 2 進系列の Prefix と Suffix に “1” を付ける。更に、この符号語のビット反転した符号語を追加している。

[定理 3] 重み一定可変長符号 (2) は、完全な可算無限アルファベットリバーシブル符号である。

(証明) この符号では、符号語の最初にあるシンボルが  $w$  個出現したら終わり、つまり符号語長がわかるような符号の構成になっている。

つまり、順方向からの場合でも逆方向からの場合でも、もし、最初のシンボルが “0” ならば、“0” が  $w$  個出現したら終わり、もし、最初のシンボルが “1” ならば、“1” が  $w$  個出現したら終わりがわかる。

符号語の “0” および “1” の数は復号する方向に依存せず、また、同一の符号語長の符号語は同じビットパターンにならないので、双方向に瞬時復号可能である。

この重み一定可変長符号 (2) は、 $w$  を符号語のハミング重みとして、符号語長  $i + w$  ビット ( $i = 0, 1, \dots$ ) の符号語数  $c(i + w)$  は、ビット反転した符号語を考慮すると、符号語長  $i + w - 2$  ビットの中から、 $w - 2$  個の “1” を選択する組合せの数の 2 倍になるので、

$$c(i + w) = 2 \binom{i + w - 2}{w - 2} \quad i = 0, 1, 2, \dots \quad (6)$$

で与えられる。これを式 (2) の左辺に代入すると、2 項係数の母関数 [10] の変形から、

$$\sum_{i=0}^{\infty} 2 \binom{i + w - 2}{w - 2} 2^{-(i+w)} = 1 \quad (7)$$

が成立するので、この符号は、可算無限アルファベットに拡張したときに Kraft の不等式を等号で満足する。

従って、完全な可算無限アルファベットリバーシブル符号である。 □

3.2 重み比一定可変長符号

符号語の “0” と “1” の数の比が 1 対 1 になることを用いた重み比一定のリバーシブル符号を構成する方法を提案する。

重み比一定の 2 進系列は、Schalkwijk のアルゴリズムと類似の方法によって作成することができる。具体的には、図 6 のような格子を考えて、(0, 0) 地点から出発して、右の経路を選択したときは、“0”，下の経路を選択したときは、“1”，を対応させ、(n, n) 地点 ( $n = 0, 1, \dots$ ) の黒丸に到達する最短経路に対応する 2 進系列を生成する。(n, n) 地点に到達する 2 進系列は、“0” と “1” の数の比は必ず 1 対 1 になることがわかる。(0, 0) 地点から出発して、各格子点 (i, j) に到達する経路数  $T(i, j)$  は、次の漸化式で定義することができる [11]。

$$\begin{aligned} T(i, 0) &= 1, \\ T(i, i + 1) &= 0, \\ T(i, j) &= T(i - 1, j) + T(i, j - 1) \\ 0 \leq j &\leq i \end{aligned} \quad (8)$$

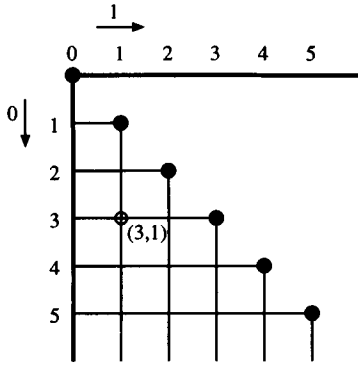


図6 重み比一定の2進系列の作成方法  
Fig.6 Construction method of constant weight proportion binary sequence.

この漸化式をもとに、(0, 0) 地点を出発して、(n, n) 地点に到達する 2n ビットの重み比一定の2進系列 (v<sub>0</sub>, v<sub>1</sub>, ..., v<sub>2n-1</sub>) は、次の式によって、順序値 s を一意に符号化、復号化することが可能となる。

$$s = \sum_{k=0}^{2n-1} v_k \cdot T(i-1, j)$$

$$0 \leq s \leq \frac{1}{n+1} \binom{2n}{n} - 1 \quad (9)$$

[重み比一定可変長符号の構成方法]

Step 1 符号語長 2n (n = 0, 1, 2, ...) の順に、ハミング重み比 1対1の2進系列を作成する。

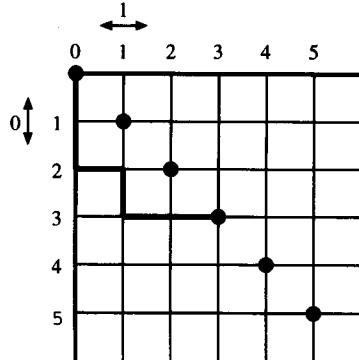
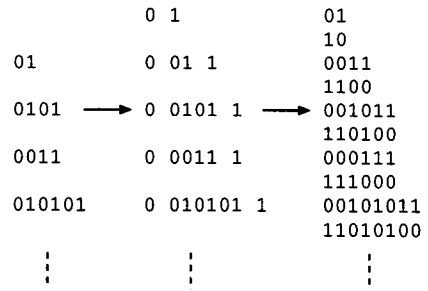
Step 2 Step 1で作成した2進系列のPrefixに“0”, Suffixに“1”を付ける。

Step 3 Step 2の符号語のビット反転した符号語を追加する。 □

[定理 4] 重み比一定可変長符号は、完全な可算無限アルファベットリバーシブル符号である。

(証明) この符号は、符号語の“0”と“1”の数が同じ数、重みの比が1対1になったら、符号語の終わりがわかる符号の構成になっている。図7でわかるように、(0, 0) 地点から出発して、“0”と“1”の数が同じになるのは、黒丸の地点しかなく、この符号の符号語は、黒丸から黒丸への最短経路で、かつ、途中の黒丸を通過しない経路に対応している。つまり、この符号語は、瞬時復号可能となる。また、黒丸への経路は、方向に依存しないので、この符号は、双方向に瞬時復号可能である。

この重み比一定可変長符号は、符号語長 2i + 2 ビッ



"001011"

図7 重み比一定可変長符号の構成  
Fig.7 Construction of constant weight proportion variable-length code.

ト (i = 0, 1, ...) の符号語数 c(2i + 2) が、

$$c(2i + 2) = \frac{2}{i + 1} \binom{2i}{i} \quad i = 0, 1, 2, \dots \quad (10)$$

で与えられる。これを式 (2) の左辺に代入すると、Catalan 数の母関数 [10] の変形から、

$$\sum_{i=0}^{\infty} \frac{2}{i + 1} \binom{2i}{i} 2^{-(2i+2)} = 1 \quad (11)$$

が成立するので、この符号は、可算無限アルファベットに拡張したときに Kraft の不等式を等号で満足する。

従って、完全な可算無限アルファベットリバーシブル符号である。 □

3.2.1 拡張方法

リバーシブル符号の拡張方法として、符号語の Suffix に 1 ビットを追加する方法がある [4]。この方法は、可算無限アルファベットリバーシブル符号にも適用することができる。具体的な符号の拡張方法としては、リバーシブル符号の Prefix あるいは、Suffix あるいは、

表 1 従来の符号との比較例  
Table 1 Comparison of reversible codes.

Alphabet	Probability	Huffman	Comp.RVLC [4]	Asym.RVLC2 [5]	ECW VLC (1)
E	0.14878570	001	01	001	000
T	0.09354149	110	100	110	001
A	0.08833733	0000	0000	0000	010
O	0.07245796	0100	1110	0100	011
R	0.06872164	0101	1111	1000	1100
N	0.06498532	0110	00010	1010	1101
H	0.05831331	1000	00011	0101	1110
I	0.05644515	1001	00110	11100	1111
S	0.05537763	1010	00111	01100	10100
D	0.04376834	00010	10110	00010	10101
L	0.04123298	00011	10111	10010	10110
U	0.02762209	10110	11000	01111	10111
P	0.02575393	10111	11010	10111	100100
F	0.02455297	11100	11011	11111	100101
M	0.02361889	11110	001000	111101	100110
C	0.02081665	11111	001010	101101	100111
W	0.01868161	011100	001011	000111	1000100
G	0.01521216	011101	101000	011101	1000101
Y	0.01521216	011110	101010	100111	1000110
B	0.01267680	011111	101011	1001101	1000111
V	0.01160928	111011	110010	01110011	10000100
K	0.00867360	1110100	110011	00011011	10000101
X	0.00146784	11101011	0010010	000110011	10000110
J	0.00080064	111010101	0010011	0001101011	10000111
Q	0.00080064	1110101000	1010010	00011010011	100000100
Z	0.00053376	1110101001	1010011	000110100011	100000101
Ave.codeword length		4.15572453	4.27795501	4.36068869	4.23658855
$\sum_{i=1}^{26} 2^{-l_i}$		1.00000000	1.00000000	0.87475586	0.98828125

その両方に固定長符号を付け、同一の符号語長の符号語数を拡大する方法である。固定長符号は、明らかにリバーシブル符号なので、固定長符号をリバーシブル符号の Prefix や Suffix につけても、やはりリバーシブル符号となる。

この拡張方法によって、n ビットの固定長符号を付けた場合、全体の符号語長は、n ビット増加するが、同じ符号語長の符号語の数が  $2^n$  倍にすることができる。次式でわかるように、この拡張方法を行っても、もとの符号が完全な符号ならば、拡張された符号は完全な符号となる。

$$\sum_{i=1}^{\infty} c(i)2^{-i} = \sum_{i=1}^{\infty} 2^n c(i)2^{-(i+n)} = 1 \quad (12)$$

図 8 は、リバーシブル符号の拡張化方法の例を示している。この例では、図 4 で作成した重み一定符号 (1) の Suffix に 2 ビットの固定長符号を付けることにより、全体の符号語長は 2 ビット増加するが、同じ符号語長の符号語数が 4 倍に拡大されている。

### 3.2.2 従来方法との比較例

英文のアルファベットの生起確率 [7] をもとに、前節

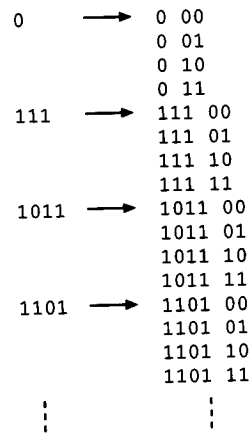


図 8 リバーシブル符号の拡張方法の例  
Fig. 8 Example of extended method of reversible code.

までで提案した構成方法を用いて、符号を設計してみた。この符号は、重み一定可変長符号 (1) で、 $w = 2$  のときの符号を 2 ビットの固定長符号で拡張して構成した符号 (Extended Constant Weight VLC (1)) である。

表 1 は、Huffman 符号とアルファベット数 26 のときの完全なリバーシブル符号 (Complete RVLC) と Asymmetrical RVLC2 [5] と、この符号の符号化効率を比較したものである。ここで、挙げたアルファベット数 26 の完全なリバーシブル符号は、最大符号語長 11 ビット以下で唯一存在すると言われている符号語長分布の符号 [4] である。また、Asymmetrical RVLC2 は、Huffman 符号をもとに符号語の入替えをして、可能な限り逆方向にも瞬時復号可能な符号語を増やした上で、それでもうまくいかない符号語はビットを追加して双方向に瞬時復号可能な符号を構成している。

我々の設計した符号は、Huffman 符号には劣るが、従来のリバーシブル符号と比較して短い平均符号長が得られている。その理由は、この符号は、有限のアルファベット数の場合、完全な符号ではないが、冗長なビットパターンが少ないということが挙げられる。冗長なビットパターンがどれくらいあるかを表す値としては、Kraft の不等式の左辺を計算してみればよいが、表 1 からわかるように、我々の設計した符号は、1 に近い値を示しており、冗長なビットパターンが少なく効率が良いことがわかる。

完全な可算無限アルファベトリバーシブル符号を使う手法は、これまで示したいくつかの符号構成方法と拡張方法によりさまざまな符号語長の符号が設計できることから、これらの中より最も平均符号語長の短い符号を選択することで、冗長度が小さく、確率分布にあった符号の設計が可能である。従って、有限アルファベットで適当な完全なリバーシブル符号が存在しない場合、この手法は有力である。

#### 4. 誤り耐性を有する動画像符号化への応用

超低レート用に開発した動画像符号化方式 [12] の DCT 係数可変長符号化部の誤り対策として、リバーシブル符号を適用した。この方式は、図 9 のように、符号化したビットストリームをフレームごとに階層的な構造をもたせて、ヘッダーやモード情報、動きベクトルのような誤りにより復号できないと画面が大きく破壊される情報は、上位の階層として強い誤り訂正符号で保護を行い、DCT 係数のように復号できなくてもある程度の画質を維持できる情報は、下位の階層として、相対的に弱い誤り訂正符号で保護を行っている。このようにすることで、全体の誤り訂正符号の冗長度を抑えている。更に、ヘッダーやモード情報、動きベクトルなどの重要な情報は 2 重化を行っている。リバーシ

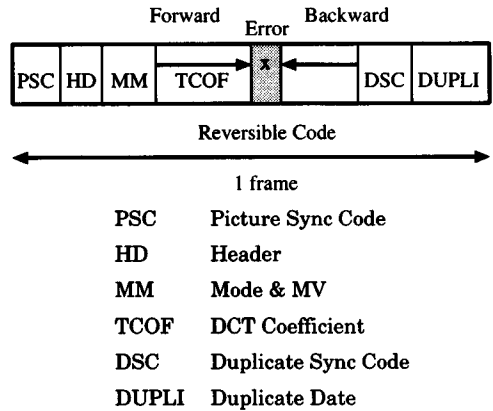


図 9 ビットストリームの階層構造  
Fig.9 Hierarchical structure of bits-stream.

ブル符号は、DCT 係数可変長符号化部の誤り訂正後の残留誤りの対策として用いている。

#### 4.1 符号化

符号化器では、H.263 と同様に DCT 後に量子化した変換係数をジグザグスキャンして、非零の最後の係数かを示す Last と零ランを表す Run と量子化係数値 Level の (Last, Run, Level) の組を可変長符号化する。

但し、

- (1) 非 Last 係数の INTRA
- (2) 非 Last 係数の INTER
- (3) Last 係数

の三つの符号語テーブルがあり、INTRA のときは、(1) と (3)、INTER のときは、(2) と (3) を使って符号化する。符号語テーブルは、重み一定可変長符号 (2) の  $w = 3$  を Level の正負で 1 ビット拡張した符号を用いていて、(3) を INTRA と INTER で共通化してあり、(1) と (3)、(2) と (3) で一つの符号になるように設計してある。表 2 と表 3 に符号語テーブルの一部を示す。

符号語テーブルに存在しない符号は、図 10 のように Run と Level の絶対値を固定長符号化を行い、先頭と末端に Escape 符号を付けて、Escape 符号の最終ビットで Last 係数と Level の正負を区別できるように符号化する。Escape 符号には、9 ビットの “11000010” と “11000011” の二つの符号語を割り振ってある。

また、誤りの検出能力を上げるために、マクロブロック単位に 1 ビットのチェックビットを付けている。

表2 非 Last 係数の符号語  
Table 2 Codeword for non last coefficient.

INTRA			INTER			Codeword
Last	Run	Level	Last	Run	Level	
0	0	1	0	0	1	000s
0	0	2	0	1	1	111s
0	0	3	0	0	2	0010s
0	0	4	0	2	1	1011s
0	1	1	0	3	1	1101s
0	0	5	0	0	3	01010s
0	1	2	0	1	2	00110s
0	2	1	0	4	1	10011s
0	3	1	0	5	1	10101s
0	4	1	0	6	1	11001s
0	0	6	0	7	1	001110s
:	:	:	:	:	:	:

表3 Last 係数の符号語  
Table 3 Codeword for last coefficient.

Last	Run	Level	Codeword
1	0	1	0100s
1	1	1	01100s
1	2	1	011100s
1	3	1	011010s
1	4	1	010110s
1	5	1	0111100s
:	:	:	:

If "t" is "0", this codeword is nonlast coefficient.  
If "t" is "1", this codeword is last coefficient.

Escape 1100001t 9 bits	Run xxxxxx 6 bits	Level xxxxxxx 7 bits	Escape 1100001s 9 bits
------------------------------	-------------------------	----------------------------	------------------------------

If "s" is "0", level is positive.  
If "s" is "1", level is negative.

図10 符号語テーブルにない TCOF の固定長符号のフォーマット

Fig.10 Format for fixed-length code of TCOF except codeword table.

### 4.2 復号化

復号化器では、順方向の復号は、通常の変長符号と全く同様に行う。

逆方向からの復号の場合は、一つ前のブロックの Last 係数の符号が復号されれば、ブロックの区切りがわかる。DCT 係数の符号化データの先頭のブロックは、ダミー符号を復号することで先頭がわかる。

誤りは、存在しないビットパターンが生じた場合、およびマクロブロック単位に付けられた 1 ビットのチェックビットによって検出される。誤りが検出され

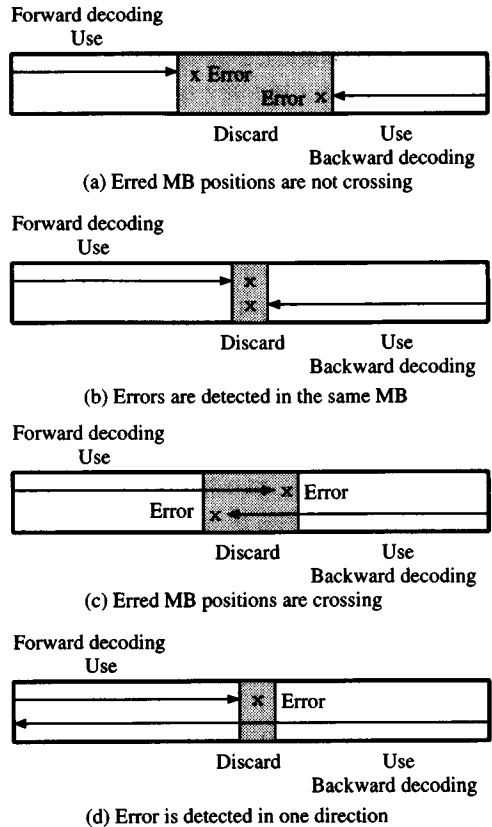


図11 双方向復号の時の誤り検出パターン

Fig.11 Error detection patterns for two-way decoding.

た場合、マクロブロック単位として次のような処理を行う。

(a) 誤り検出位置が交差しない場合

順方向で復号できたマクロブロックと逆方向で復号できたマクロブロックのうち、誤りの検出されなかったマクロブロックの復号結果のみを復号値として使用する。二つの誤り検出位置の間（図 11 (a) の灰色の部分）の復号結果は使用しない。

(b) 同一のマクロブロックで誤りが検出される場合

誤り検出位置までの順方向で復号できたマクロブロックと逆方向で復号できたマクロブロックの復号結果を復号値として使用する。誤り検出位置のマクロブロック（図 11 (b) の灰色の部分）の復号結果は使用しない。

(c) 誤りの検出位置が交差する場合

両方で誤りが検出されなかった復号結果を復号値として使用する。二つの誤り検出位置のマクロブロックの間（図 11 (c) の灰色の部分）の復号結果は使用し



ない。

(d) 片方向の復号結果にしか誤りが検出されない場合  
誤り検出位置までの順方向で復号できたマクロブ  
ロックと逆方向で復号できたマクロブロックの復号結  
果を復号値として使用する。誤り検出位置のマクロブ  
ロック (図 11 (d) の灰色の部分) は、復号値として使  
用しない。

復号値として使用しないマクロブロックは、INTRA  
の場合、DCのみを表示し、INTERの場合は、MCの  
みで表示を行っている。

### 4.3 実験結果

図 9 のビットストリームのうち、リバーシブル符号  
を用いた TCOF 部のみ、誤り訂正符号をかけないで、  
誤り耐性能力の実験を行った。図 12 に、リバーシ  
ブル符号による双方向復号の効果を示す。テスト画像  
は、QCIF にダウンサンプリングした四つの画像 (Akiyo,  
Container Ship, Hall Monitor, Mother & Daughter)  
を用いた。いずれの画像についても、24 kbit/s で 10  
秒間符号化したビットストリームの復号結果で、1.5  
秒から 10 秒までの間、 $10^{-5}$ 、 $10^{-4}$ 、 $10^{-3}$  の確率の  
ランダム誤りを挿入して復号した。結果の輝度信号の  
SNR の値は、それぞれのテストケースに対して、種  
を変えて 10 種類の異なる誤りパターンを挿入し、復  
号した平均の値である。図 12 でわかるように、双  
方向復号により、SNR の下落の傾向が押さえられて  
いる。この方式の場合は、 $10^{-4}$  の確率のランダム誤  
りまで、エラーがなかったときからの SNR の劣化が少  
なく、かつ、双方向復号の効果が現れていると考えら  
れる。

図 13 は、DCT 係数符号化部分以外は同一で、符  
号語テーブルのみを入れ換えて、リバーシブル符号と  
H.263 の可変長符号との比較したものである。条件は、  
フレームレート 7.5 Hz 固定で、ステップサイズ一定で  
24 kbit/s 程度になるように 10 秒間符号化したときの  
ビットレートである。すべての画像において、リバー  
シブル符号の方が符号量が少なく、H.263 の可変長符  
号と比較して同等以上の符号化効率が得られているこ  
とがわかる。リバーシブル符号の効率は、理論的には  
Huffman 符号以下であるが、実際の符号化では、適  
当な可算無限アルファベットリバーシブル符号を選  
択し、INTRA と INTER のテーブルを分けるなどモデル  
化を工夫することにより、実用上問題のない符号化  
効率をだすことができる。

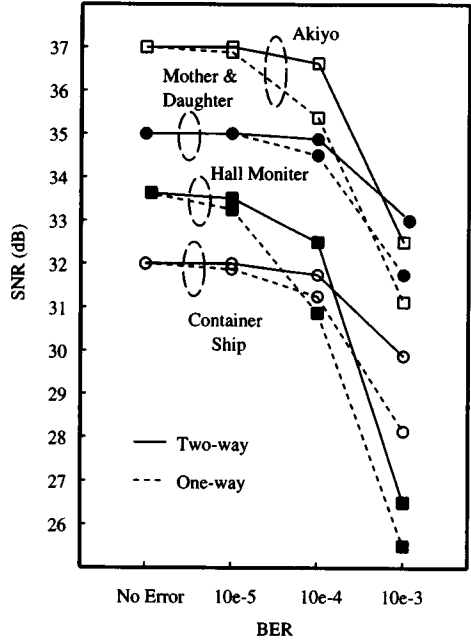


図 12 リバーシブル符号の効果  
Fig. 12 Comparison of reversible codes.

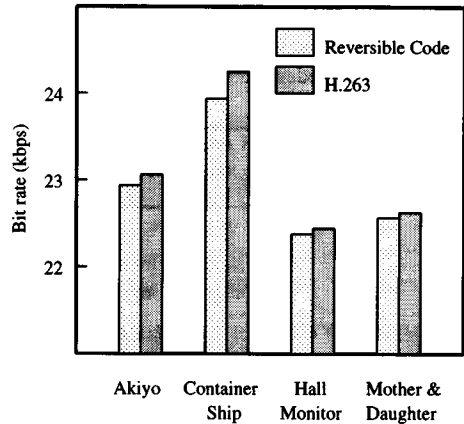


図 13 リバーシブル符号の符号化結果  
Fig. 13 Coding results of reversible code.

## 5. む す び

双方向に瞬時復号可能な可変長符号であるリバー  
シブル符号について検討を行った。可算無限アルファ  
ベットならば、完全な符号が存在することに着目して、  
符号語のハミング重みを用いた新しい符号の構成方法  
の提案をした。また、実用的な意味でのリバーシブル  
符号の応用として、動画像符号化の誤り対策として適

用し、その有効性を示した。

今後の課題としては、理論的な意味での最適なりパーシブル符号の存在を明らかにする必要がある。応用としては、可変長符号の誤り対策以外にも、双方向復号可能なことを利用して、他のアプリケーションについても考えてみたい。

## 文 献

- [1] M.P. Schützenberger, "On an application of semi-groups methods to some problems in coding," I.R.E Trans. Inf. Theory, vol.IT-2, pp.47-60, Sept. 1956.
- [2] D. Perrin, "Completing biprefix codes," Theoretical Computer Science, vol.28, pp.329-336, 1984.
- [3] K.M. Rose and A.H. Heiman, "Enhancement of one-dimensional variable-length DPCM image corrupted by transmission errors," IEEE Trans. Commun., vol.COM-37, no.4, pp.373-379, April 1989.
- [4] A.S. Fraenkel and S.T. Klein, "Bidirectional Huffman coding," The Computer Journal, vol.33, no.4, pp.296-307, 1990.
- [5] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable Length codes," IEEE Trans. Commun., vol.COM-43, no.2/3/4, pp.158-162, Feb./March/April 1995.
- [6] "Draft ITU-T recommendation H.263," ITU-T SG15, WP15/1, Q2/15, LBC-95-251, Oct. 1995.
- [7] J. Maxed and J.P. Robinson, "Error recovery for variable length codes," IEEE Trans. Inf. Theory, vol.IT-31, no.6, pp.794-801, Nov. 1985.
- [8] J.P.M. Schalkwijk, "An algorithm for source coding," IEEE Trans. Inf. Theory, vol.IT-18, no.3, pp.395-399, May 1972.
- [9] E.N. Gilbert and E.F. Moore, "Variable-length binary encoding," Bell System Technical J., vol.38, pp.933-967, July 1959.
- [10] R.L. Graham, D.E. Knuth, and O. Patashnik, "Concrete Mathematics," Addison-Wesley, 1989.
- [11] 野崎昭弘, "離散数学とはなにか," 数理科学, pp.5-9, Sept. 1991.
- [12] 渡邊敏明, 菊池義浩, 中條 健, 永井 剛, "MPEG4 対応低レート動画像符号化方式," 信学技報, IE95-145, March 1996.
- [13] Y. Miyakawa, Y. Saitoh, and H. Imai, "New binary encoding schemes of positive integers," IEICE Trans., vol.E74, no.9, pp.2504-2512, Sept. 1991.
- [14] 中條 健, 稻積宏誠, 西島利尚, 平澤茂一, "自然数表現の一般化について," 信学論 (A), vol.J75-A, no.1, pp.94-100, Jan. 1992.

(平成 8 年 5 月 30 日受付, 10 月 3 日再受付)



中條 健 (正員)

平 1 早大・理工・工業経営卒, 平 3 同大大学院理工学研究科修士課程了。同年 (株) 東芝入社。研究開発センターにて, 主として動画像符号化技術に関する研究開発に従事。



渡邊 敏明 (正員)

昭 58 北大・工・電子卒, 昭 60 同大大学院修士課程了。同年 (株) 東芝入社。研究開発センターにて, 動画像, 静止画像の圧縮符号化技術および信号処理技術に関する研究, 開発に従事。テレビジョン学会, IEEE 各会員。