

宇宙用誤り制御符号と画像圧縮方式の研究

Research on Error Control Coding and Image Compression for Spacecraft Data-Handling Systems

総合技術研究本部 エレクトロニクス技術グループ 金子晴彦
Haruhiko Kaneko, Spacecraft Electrical Engineering Group
Institute of Space Technology and Aeronautics

Abstract

Source coding and channel coding are essential techniques for efficient data processing in spacecraft. Although many of conventional coding techniques can be effectively applied to spacecraft data-handling systems, we need to develop some new source coding and channel coding techniques to design high-performance data-handling systems for spacecraft. Therefore, in our research, we have developed new coding methods suitable for spacecraft, that is, spotty byte error control codes for semiconductor memory systems, such as Solid-State Recorders (SSRs), and low-complexity video encoding method based on low-density parity check code for efficient compression of video data.

1 はじめに

本研究は宇宙機搭載データ処理システムに適した通信路符号化方式及び情報源符号化方式を構築するためのものであり、具体的には以下に示す2種の符号化方式について研究開発を行った。

(a) 宇宙機搭載半導体メモリ装置用誤り制御符号の研究：宇宙機搭載の半導体メモリ装置においては、陽子等の衝突によりSEU(Single Event Upset)が多く発生することから、これによるデータの誤りを効率的に訂正／検出するための誤り制御符号の開発と符号化／復号回路の実装を行った。本研究は、平成10年度より東京工業大学・大学院情報理工学研究科・藤原英二教授との共同研究により実施したものである。

(b) 低密度パリティ検査符号を用いた低計算量動画像圧縮方式の研究：既存のMPEG等の動画像圧縮方式は圧縮処理において動き予測を行うため、圧縮のために多くの計算処理を必要とする。宇宙機における圧縮処理のための計算量を大幅に削減するため、低密度パリティ検査(LDPC)符号を用いた動画像圧縮方式の基礎研究とシミュレーションを行った。本研究は、平成17年度より東京工業大学・大学院情報理工学研究科・藤原英二教授との共同研究により実施しているものである。

2 研究の概要

宇宙機搭載半導体メモリ装置用誤り制御符号の研究として、以下の項目を実施した。

- (a-1) 宇宙機において要求される誤り制御符号の機能を決定するため、MDS-1搭載SSRにおける誤り率及び誤りパターンの解析を行った。
- (a-2) 上記解析結果を基にスポットバイト誤り制御符号の機能と符号パラメータを決定した。
- (a-3) スポットバイト誤り制御符号及びReed-Solomon(RS)符号の符号化回路及び並列復号回

路の試作を行い、回路量、遅延、等の評価を行った。

低密度パリティ検査符号を用いた低計算量動画像圧縮方式の研究として、以下の項目を実施した。

- (b-1) 本圧縮法の圧縮アルゴリズム及び伸長アルゴリズムを構築した。
- (b-2) 上記圧縮アルゴリズム及び伸長アルゴリズムをソフトウェアにより実装し、画質(PSNR: Peak Signal-to-Noise Ratio)の評価を行った。

3 成果の概要

3.1 宇宙機搭載半導体メモリ装置用誤り制御符号の研究 [1, 2]

コンピュータシステムに対し効率的に誤り制御符号を適用するためには、一般に以下の手順により符号の開発と実装を行う必要がある[3, 4]。

- (i) 誤り率、誤りパターン、誤りの種類の解析。
- (ii) 符号パラメータ、符号機能の決定。
- (iii) 符号化／復号回路の設計と実装。

本研究では、MDS-1搭載のSSR(MDS-1/SSR)から取得したデータを用いて誤り率、誤りパターン、等の解析を行い、これに基づいて宇宙機搭載半導体メモリ用誤り制御符号のパラメータ、符号機能を決定した。また、FPGAを用いて符号化回路及び並列復号回路の実装を行った。本研究成果の概略を以下に示す。

3.1.1 MDS-1/SSRにおける誤り率／誤りパターン解析

MDS-1/SSRの構成をFig. 1に示す。MDS-1/SSRの記憶部は2枚のメモリスライス(MEMS1及びMEMS2)より構成され、それぞれのメモリスライスは6個のメモリモジュール(A,B,C,D,E,F)を搭載

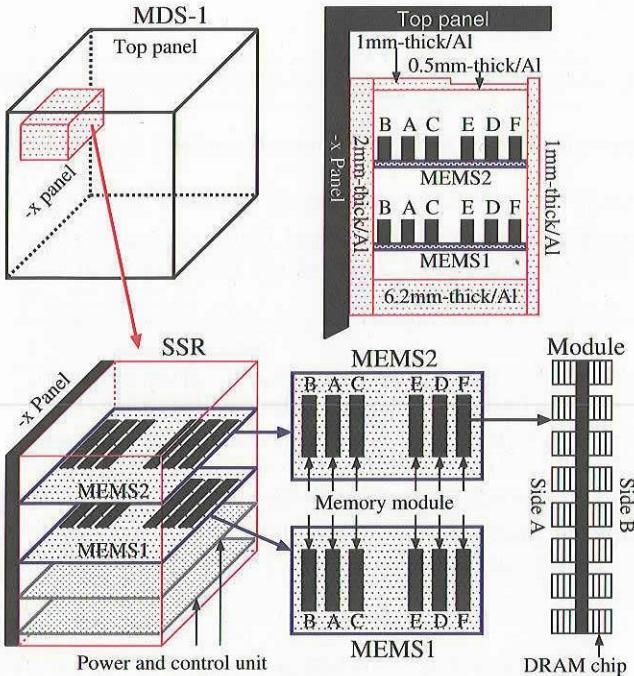


Figure 1: Configuration of the MDS-1/SSR.

する。また、各メモリモジュールは記憶容量 64Mb を有する DRAM 素子 64 個から構成される。

メモリモジュールの誤り率を Fig. 2 に示す。ただし、モジュール C 及び F の誤り率はシステムの設計上取得できていない。MEMS1 の上部は MEMS2 によって遮蔽されていることから、MEMS1 のモジュールの誤り率は MEMS2 と比較して低い。また、MEMS1においては衛星中心側のモジュール E 及び D が低い誤り率を有するが、MEMS2 においてはすべてのモジュールが同程度の誤り率を有する。

MDS-1/SSR は 1 アドレスに 48 ビットのデータを保持し、従ってデータの入出力は長さ 48 ビットを有する I/O ワード単位で行われる。MDS-1/SSR はデータ入出力幅 4 ビットの DRAM 素子を用いており、I/O ワードと DRAM 素子は Fig. 3 に示すような関係で対応付けられる。I/O ワード中の各ビットの誤り率を Fig. 4 に示す。I/O ワード中の各ビットの誤り率に大きな偏りがある場合、不均一誤り制御(UEC: Unequal Error Control)符号を用いた方が効率的に行える場合がある。しかし、UEC 符号適用のためには誤り率の偏りをより高い精度で解析する必要があり、本解析のみから UEC 符号を適用することは不可能である。

单一の DRAM 素子から入出力される長さ $b = 4$ ビットのデータブロックをバイトと称する。バイト誤りパターンと誤り率を Table 1 に示す。バイト内 1 ビット誤りがほとんどであるが、バイト内 2 ビット誤り及びバイト内 3 ビット誤りも発生していることから、宇宙機搭載半導体メモリ装置に対してはこのような誤りを訂正／検出可能な符号の適用が必要である。以下では、長さ b ビットを有するバイト中の $t (\leq b)$ ビット誤りを t/b 誤りまたはスロッティバイト誤りと称する。MDS-1/SSR から取得したデータによると、单一の I/O ワード中に複数のスロッティバイト誤りが発生する場合がある。I/O ワード中の

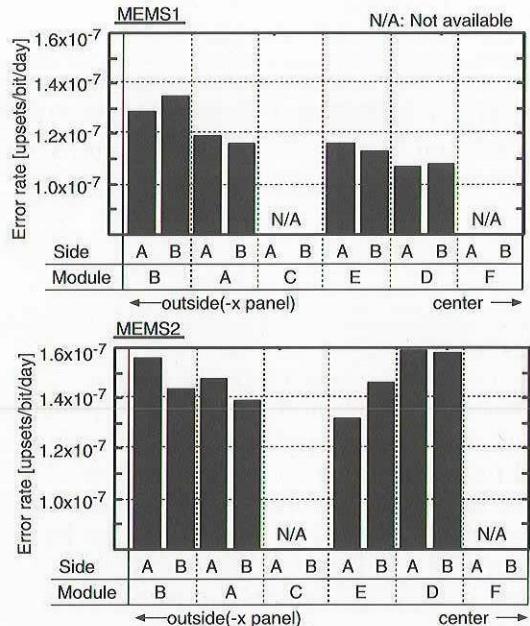


Figure 2: Error rates of memory modules.

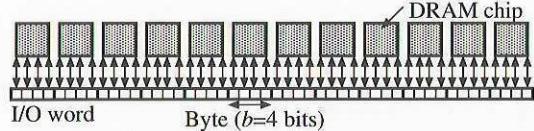


Figure 3: I/O word structure.

誤りパターンの例と誤り率を Table 2 に示す。

3.1.2 誤り制御符号の機能とパラメータの決定

上記のデータ解析より、宇宙機搭載半導体メモリ装置に対してはスロッティバイト誤りを訂正／検出可能な誤り制御符号を適用する必要がある。本研究では、この条件を満足する符号として、Table 3 に示す 3 種のスロッティバイト誤り制御符号、及び 3 種の RS 符号を選択した。Table 3 に示す 6 種の誤り制御符号は線形符号であり、検査行列 \mathbf{H} の零空間として定義できる。ガロア体 $GF(2^{16})$ 上の短縮 RS 符号(ii) の検査行列の例を以下に示す [3]。

$$\mathbf{H} = \begin{bmatrix} I & I & I & I & \dots & I \\ I & T & T^2 & T^3 & \dots & T^{10} \\ I & T^2 & T^4 & T^6 & \dots & T^{20} \end{bmatrix}$$

ただし、 T は原始多項式 $p(x) = x^{16} + x^{12} + x^3 + x + 1$ により定義される 16×16 隅伴行列である。

スロッティバイト誤り制御符号の構成法は文献 [5], [6] 及び [7] において詳細に述べられている。例として、(160,128) $S_{4/16}EC-D_{4/16}ED$ 符号の検査行列を以下に示す。

$$\mathbf{H} = \begin{bmatrix} I & I & I & I & \dots & I \\ \gamma^0 H' & \gamma^1 H' & \gamma^2 H' & \gamma^3 H' & \dots & \gamma^9 H' \\ \gamma^0 H' & \gamma^2 H' & \gamma^4 H' & \gamma^6 H' & \dots & \gamma^{18} H' \end{bmatrix}$$

ただし、 I は 16×16 単位行列、 $\gamma \in GF(2^8)$ は $GF(2)$ 上の 8 次の原始多項式 $p(x) = x^8 + x^7 + x^2 + 1$ の根

Table 3: Functions and parameters of the selected error control codes.

	Code	Function	Correctable error	Detectable error	Code parameters				
					Code length	Information bit length	t	b	Check bit length
(i)	RS	DbEC	Double byte errors	—	192	128	—	16	64
(ii)	RS	S _b EC-D _b ED	Single byte errors	Double byte errors	176	128	—	16	48
(iii)	RS	S _b EC	Single byte errors	—	160	128	—	16	32
(iv)	Spotty	[T _{1/b} EC] ₂ -S _b ED	Triple-bit errors confined in two bytes	Single byte errors	160	129	—	16	31
(v)	Spotty	S _{1/b} ED-D _{1/b} ED	Single t/b errors	Double t/b errors	160	128	4	16	32
(vi)	Spotty	S _{1/b} ED	Single t/b errors	—	144	131	2	16	13

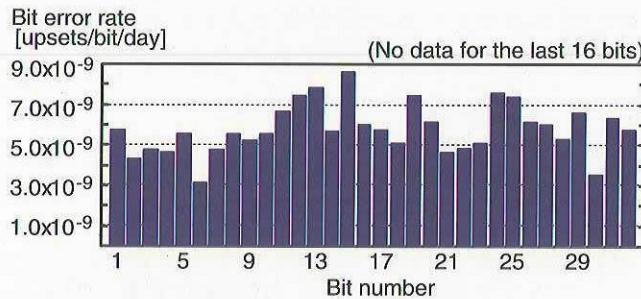


Figure 4: Error rates in 48-bit I/O word.

Table 1: Byte error patterns and error rates.

1-bit errors	☒☒☒ 5.51x10 ⁻⁹	☒☒☒ 5.63x10 ⁻⁹	☒☒☒ 4.43x10 ⁻⁹	☒☒☒ 5.53x10 ⁻⁹	☒☒☒ 0	☒☒☒ 0	☒☒☒ 0	☒☒☒ 0	☒☒☒ 0					
	☒☒☒ 0	☒☒☒ 0			☒☒☒ 0	☒☒☒ 0	☒☒☒ 0	☒☒☒ 0	☒☒☒ 0					
2-bit errors	☒☒☒☒ 0	☒☒☒☒ 0	☒☒☒☒ 2.52x10 ⁻¹¹	☒☒☒☒ 1.26x10 ⁻¹¹	☒☒☒☒ 0	☒☒☒☒ 1.26x10 ⁻¹¹	☒☒☒☒ 0	☒☒☒☒ 0	☒☒☒☒ 0					
	☒☒☒☒ 0	☒☒☒☒ 0												
3-bit errors	☒☒☒☒ 0	☒☒☒☒ 0	☒☒☒☒ 1.26x10 ⁻¹¹	☒☒☒☒ 1.26x10 ⁻¹¹										
	☒☒☒☒ 0	☒☒☒☒ 0												
4-bit error	☒☒☒☒ 0	☒☒☒☒ 0												

□: Error-free bit ☒: Erroneous bit Error rate: events/byte/day

である。また、

$$H' = [\gamma^0 \gamma^1 \gamma^2 \gamma^3 \gamma^4 \gamma^5 \gamma^6 \gamma^7 \\ \gamma^{14} \gamma^{164} \gamma^{235} \gamma^{182} \gamma^{140} \gamma^{147} \gamma^{248} \gamma^{96}]$$

は任意の4列ベクトルが線形独立な 8×16 行列である。

$$\gamma^i H' = [\gamma^{i+0} \gamma^{i+1} \gamma^{i+2} \gamma^{i+3} \gamma^{i+4} \gamma^{i+5} \gamma^{i+6} \gamma^{i+7} \\ \gamma^{i+14} \gamma^{i+164} \gamma^{i+235} \gamma^{i+182} \gamma^{i+140} \gamma^{i+147} \gamma^{i+248} \gamma^{i+96}]$$

は 8×16 行列である。ここで、 $\gamma^i \in GF(2^8)$ は長さ8を有する列ベクトルとして表現する。

3.1.3 符号化回路及び並列復号回路の実装と評価

半導体メモリ装置においては高速な復号が要求されることから、線形フィードバックシフトレジスタ(LFSR: Linear Feedback Shift Register)を必要としない並列復号法[8]を用いて回路を実装した。FPGAにより実装した符号化／復号回路のブロック図をFig. 5(a)に、並列復号の流れをFig. 5(b)に示す。

符号化及び復号回路はVerilog-HDLにより記述し、ALTERA Cyclone EP1C20で実装した。実装の結果、6種の誤り制御符号の符号化／復号回路はTable 4に

Table 2: Error counts in an I/O word.

Error pattern	Example of error pattern	Count [events]
No error	☒☒☒☒☒☒☒☒	23,673,844
Single 1/4-error	☒☒☒☒ 1☒☒☒	1,674
Single 2/4-error	☒☒☒☒☒☒☒☒	1
(1/4-error)+ (3/4-error)	☒☒☒☒☒☒☒☒	1
Triple 2/4-error	☒☒☒☒☒☒☒☒	1

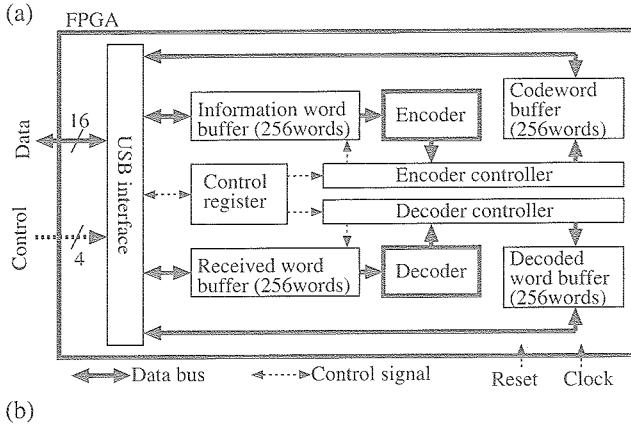
Table 4: Gate count.

Code	Encoder			Decoder		
	Codeword generation	Control circuit	Total	Syndrome generation	Error detection/correction	Control circuit
(i)	12,720	15,120	27,840	7,248	259,200	33,336
(ii)	5,040	15,504	20,544	6,024	25,968	33,240
(iii)	1,080	15,336	16,416	1,632	10,152	33,312
(iv)	3,024	15,144	18,168	3,912	54,912	33,432
(v)	3,504	15,144	18,648	3,648	20,400	33,912
(vi)	1,344	14,904	16,248	72,456	33,528	105,984

示すゲート数を有することが明らかになった。また、実装した回路の遅延と動作周波数をTable 5に示す。実装した回路は50MHz以上の動作周波数を有するが、これを高速なメモリシステムに適用するためには、さらなる処理の高速化が必要である。このためには、処理の並列化、パイプライン化、等の回路の最適化が必要である。また、並列復号において、誤り位置及び誤りパターンを導出するための処理で用いる部分行列の重みを最小化して、回路遅延を削減する必要がある。

3.2 低密度パリティ検査符号を用いた低計算量動画像圧縮方式の研究 [9, 10]

従来のMPEG等の一般的な動画像圧縮方式においては動き予測を用いてデータ圧縮を行うため、圧縮処理における計算量が非常に多い。一方、衛星搭載カメラや携帯型テレビ電話等においては消費電力等のリソースに制約が存在することから、圧縮のための計算量が少ない動画像圧縮方式が必要である。このような要求に応える圧縮方式としてTurbo符号を用いた「distributed video coding」が従来提案されている[11]。しかし、本手法は受信側から送信側へのフィードバックループが必要であるため、必ずしも実用的ではない。本研究では、低密度パリティ検査(LDPC)符号[12, 13]を用いて、圧縮処理における計算量が非常に少ない実用的な動画像圧縮方式を提案し、シミュレーションによる画質の評価を行った。



(b)

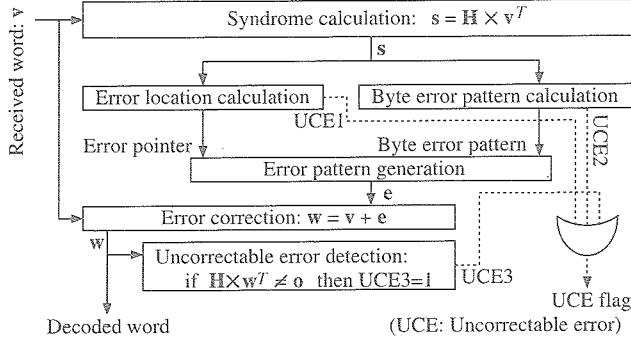


Figure 5: (a) Block diagram of the developed FPGA. (b) Flow of parallel decoding.

Table 5: Clock frequencies and delays.

Code	(i)	(ii)	(iii)	(iv)	(v)	(vi)
Delay [ns]	17.623	16.160	15.096	19.523	19.863	17.459
Frequency [MHz]	56.74	61.88	66.24	51.22	50.34	57.28

3.2.1 圧縮手順

本圧縮法において、動画像は I ピクチャ及び L ピクチャの 2 種のピクチャより構成され、I ピクチャは L ピクチャの間に一定間隔で挿入する。I ピクチャは JPEG と同様の手順でフレーム内符号化を行う。L ピクチャの圧縮手順は以下のとおりである。

- (1) L ピクチャの大きさを $H\sigma \times W\sigma$ とし、これを $\sigma \times \sigma$ の大きさを有する $N = HW$ 個のマクロブロックベクトル L_0, \dots, L_{N-1} に分割する。マクロブロックベクトルを $\mathbf{L} = (L_0, \dots, L_{N-1})$ とおく。

- (2) 2 元 ($N, N - M$)LDPC 符号の検査行列を

$$\mathbf{H} = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \dots & a_{M-1,N-1} \end{bmatrix}$$

とおく。行列 \mathbf{H} を用いて、長さ N を有するマクロブロックベクトル \mathbf{L} を、長さ $M (< N)$ を有するシンドロームブロックベクトル $\mathbf{S} = (S_0, \dots, S_{M-1})$ に変換する。ただし、

$$S_i = \sum_{j \in \{J | a_{i,j}=1\}} L_j$$

は $\sigma \times \sigma$ シンドロームブロックである。

- (3) シンドロームブロック \mathbf{S} に対し JPEG と同様の非可逆圧縮 (DCT, 量子化, ジグザグスキャン, ランレングス符号化, 可変長符号化) を適用し、圧縮データを得る。

上記の圧縮手順では動き予測を用いないことから、従来の動画像圧縮法と比較して非常に少ない計算量で実行できる。また、上に示すように L ピクチャの圧縮時にはフレーム間の相関を用いないが、伸長時にフレーム間の相関を利用することから、Motion JPEG 等の従来のフレーム内符号化方式よりも高い圧縮率を得ることができる。

3.2.2 伸長手順

I ピクチャは JPEG と同様の手順で伸長する。L ピクチャの伸長手順は以下のとおりである。

- (1) 圧縮手順 (3) により圧縮したシンドロームブロックベクトル \mathbf{S} を JPEG と同様の手順で伸長し、 $\mathbf{S}' = (S'_0, \dots, S'_{M-1})$ を得る。ここで、JPEG は非可逆圧縮であるから $\mathbf{S} \simeq \mathbf{S}'$ である。
- (2) L ピクチャの前方及び後方に位置する伸長後の I ピクチャ I_1 及び I_2 を用いて、動き補償フレーム補間 (MCFI) により L ピクチャの予測画像 L' を構成する。画像 L' のマクロブロックベクトルを $\mathbf{L}' = (L'_1, \dots, L'_{N-1})$ とする。また、動き予測と同時に L' の各ピクセルにおける予測値の「不確かさ」を求め、これよりエラーブロックベクトル $\mathbf{E} = (E_1, \dots, E_{N-1})$ を構成する。ただし、 E_j は L'_j に対応するエラーブロックである。
- (3) 予測画像 L' には予測誤りが存在することから、シンドロームブロックベクトル \mathbf{S}' を用いて LDPC 符号の反復復号を行い、 L' に存在する予測誤りを訂正する。すなわち、すべての $i \in \{0, \dots, M-1\}$ に対し $S'_i = \sum_{j \in \{J | a_{i,j}=1\}} L'_j$ を満足するように、エラーブロックベクトル \mathbf{E} を用いて L' を反復更新し、復号後の L ピクチャ $\hat{\mathbf{L}}$ を構成する。以下に L' の反復復号手順を示す。マクロブロック L'_j 、エラーブロック E_j 及びシンドロームブロック S'_i をそれぞれ以下のような $\sigma \times \sigma$ 行列で表現する。

$$L'_j = \begin{bmatrix} l'^{0,0}_j & l'^{0,1}_j & \dots & l'^{0,\sigma-1}_j \\ \vdots & \vdots & \ddots & \vdots \\ l'^{\sigma-1,0}_j & l'^{\sigma-1,1}_j & \dots & l'^{\sigma-1,\sigma-1}_j \end{bmatrix}$$

$$E_j = \begin{bmatrix} e^{0,0}_j & e^{0,1}_j & \dots & e^{0,\sigma-1}_j \\ \vdots & \vdots & \ddots & \vdots \\ e^{\sigma-1,0}_j & e^{\sigma-1,1}_j & \dots & e^{\sigma-1,\sigma-1}_j \end{bmatrix}$$

$$S'_i = \begin{bmatrix} s'^{0,0}_i & s'^{0,1}_i & \dots & s'^{0,\sigma-1}_i \\ \vdots & \vdots & \ddots & \vdots \\ s'^{\sigma-1,0}_i & s'^{\sigma-1,1}_i & \dots & s'^{\sigma-1,\sigma-1}_i \end{bmatrix}$$

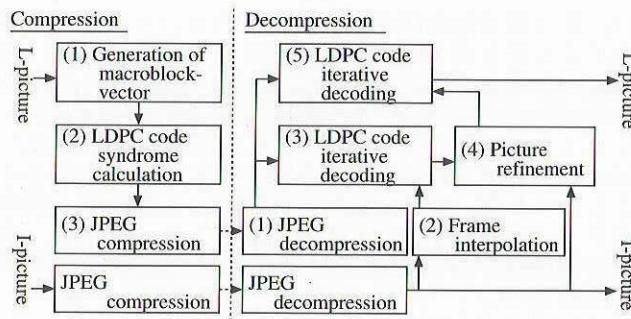


Figure 6: Compression / decompression process.

すべての $(p, q) \in \{(p, q) \mid 0 \leq p, q \leq \sigma - 1\}$ に対して、それぞれ独立に以下の処理を行う。

- (i) すべての $i \in \{0, 1, \dots, M - 1\}$ に対し、シンドロームブロックベクトル \mathbf{S}' とマクロブロックベクトル \mathbf{L}' の不一致度 $x_i^{p,q}$ を求める。

$$x_i^{p,q} = s_i'^{p,q} - \left(\sum_{j \in \{j \mid a_{i,j}=1\}} l_j'^{p,q} \right)$$

- (ii) エラーブロックベクトル \mathbf{E} を用いて、すべての $(i, j) \in \{(i, j) \mid a_{i,j} = 1\}$ に対し、重み付き不一致度 $y_{i,j}^{p,q}$ を求める。

$$y_{i,j}^{p,q} = x_i^{p,q} \times \frac{e_j^{p,q}}{\sum_{k \in \{k \mid a_{i,k}=1\}} e_k^{p,q}}$$

- (iii) すべての $j \in \{0, 1, \dots, N - 1\}$ に対し、次式により $l_j'^{p,q}$ の値を更新する。

$$l_j'^{p,q} := l_j'^{p,q} + \frac{\sum_{i \in \{i \mid a_{i,j}=1\}} y_{i,j}^{p,q}}{w_j}$$

ただし、 w_j は \mathbf{H} の第 j 列の Hamming 重みである。

- (4) 上記の LDPC 符号の復号においては誤訂正が生じる場合があることから、 I_1, I_2 , 及び \hat{L} を用いて動き補償ノイズ除去をおこない \hat{L} の補正を行う。
- (5) 上記の画像補正において、誤った補正を行う場合があることから、手順 (3) と同様に LDPC 符号の反復復号を用いてこの誤りを訂正する。

手順 (4) 及び (5) を繰り返して行うことにより画質の向上が可能である。圧縮及び伸長手順を Fig. 6 に示す。テスト画像「mobile」[14] を提案手法により圧縮し、これを伸長するときの例を Fig. 7 に示す。

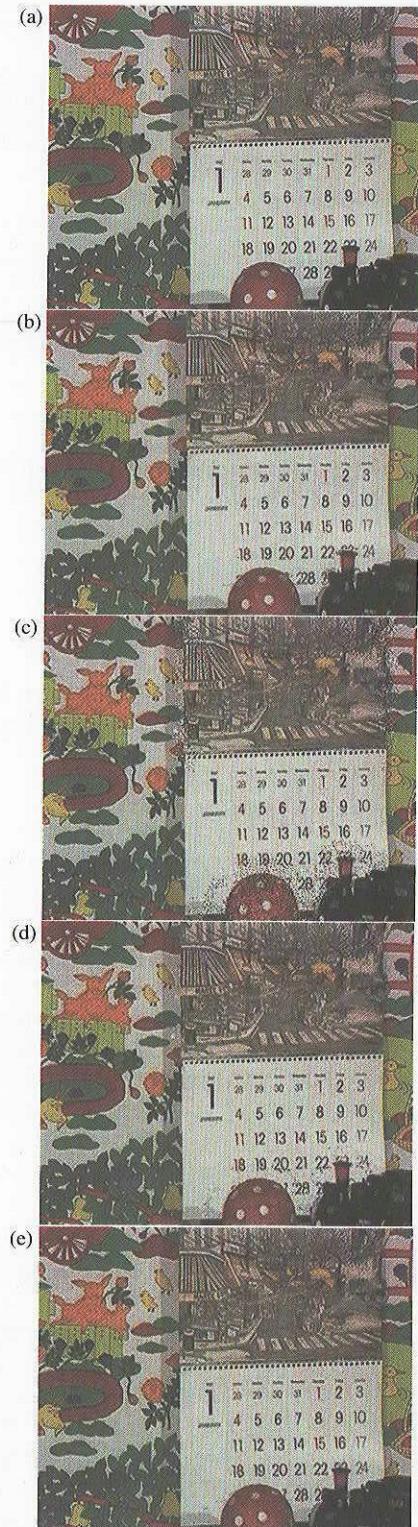


Figure 7: Example of decompression process: (a) Original image. (b) Estimated image by step (2). (c) Decoded image by step (3). (d) Refined image by step (4). (e) Final output of the decompression process with four iterations of steps (4) and (5).

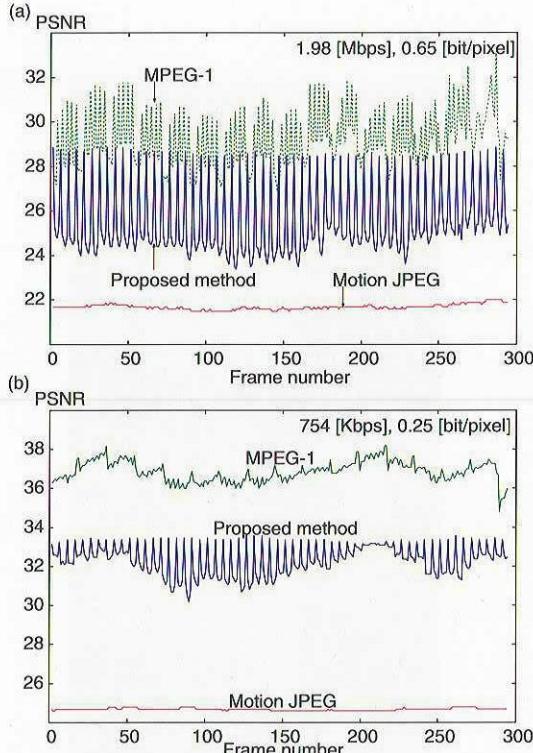


Figure 8: PSNR of the proposed method, MPEG-1, and Motion JPEG: (a) mobile. (b) students.

3.2.3 提案手法の評価

提案手法の画質評価として、テスト画像「mobile」及び「students」に対するPSNRをそれぞれFig. 8(a)及びFig. 8(b)に示す。また、比較としてMPEG-1及びMotion JPEGのPSNRを同図に示す。ただし、MPEG-1に対する評価はフリーソフトウェア[15]を用いて行い、GOP(Group of pictures)はIBBPBBPBPBPPBBBである。本評価より、提案手法のPSNRはMotion JPEGよりも高く、MPEG-1よりも低いことが明らかになった。

4 まとめ

本稿では、宇宙機用データ処理システムのための通信路符号化／情報源符号化方式の研究成果として、以下の報告を行った。

(a) 宇宙機搭載半導体メモリ装置用誤り制御符号の研究として、MDS-1搭載半導体レコーダ(SSR)における誤り率／誤りパターンの解析を行い、これらの誤りを訂正するためのスポットバイト誤り制御符号及びReed-Solomon符号の機能及び符号パラメータを決定した。また、FPGAを用いて符号化回路及び並列復号回路の試作を行い、回路量、遅延、等の定量的評価を行った。本研究により宇宙機搭載半導体メモリ装置に対してはスポットバイト誤り制御符号が優れた誤り制御機能を与えることが明らかになった。今後は、符号化／復号回路の高速化、誤り制御符号を用いた場合のユーザBERの導出、等を行う必要がある。

(b) 宇宙機において動画像圧縮にかかる計算処理の負荷を軽減するため、低密度パリティ検査符号を用

いた低計算量動画像圧縮方式の研究を行った。本圧縮法により従来のMPEG等と比較して1/10から1/20程度の計算量で動画像の圧縮が可能であることを明らかにした。本圧縮法の画質の定量的評価として、シミュレーションによりPSNRを求め、本圧縮法がMotion JPEGよりも高い画質を有することを明らかにした。提案した圧縮法は宇宙機のみでなく、携帯型テレビ電話等への適用が期待できる。今後は、画質の向上及び伸長速度の高速化、等に関する研究が必要である。

参考文献

- [1] H. Kaneko, "Error Control Coding for Semiconductor Memory Systems in the Space Radiation Environment" (to appear), Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Oct. 2005.
- [2] 金子晴彦、市川倫、笛田武志、"宇宙機搭載半導体メモリ装置における誤り率解析と誤り制御符号化／復号回路の実装," 電子情報通信学会技術研究報告, FIIS-05-159, June 2005.
- [3] T. R. N. Rao and E. Fujiwara, "Error Control Coding for Computer Systems," Prentice Hall, 1989.
- [4] <http://fujiwara-www.cs.titech.ac.jp/res/>
- [5] K. Suzuki, T. Kashiyama, and E. Fujiwara, "A General Class of M-Spotty Byte Error Control Codes," Proc. Asia-Europe Workshop on Information Theory, pp.24–26, Oct. 2004.
- [6] T. Kashiyama and E. Fujiwara, "A General Class of Byte Error Control Codes for S-Spotty Byte Errors," Proc. of 2004 International Symposium on Information Theory and Its Applications, pp.1297–1302, Oct. 2004.
- [7] 清水護、鈴木一克、櫻山俊彦、藤原英二、"誤りバイト数を制限したm-スポットバイト誤り訂正符号," 2005年電子情報通信学会総合大会講演論文集, D-10-9, March 2005.
- [8] 北神正人、小林鉄太郎、藤原英二、"バイト誤り訂正符号における並列復号法," 1995年電子情報通信学会ソサイエティ大会論文集, A-114, Sep. 1995.
- [9] 金子晴彦、"動画像圧縮符号化装置、圧縮データ伸長装置、動画像圧縮符号化及び／又は圧縮データ伸長プログラム、動画像圧縮符号化方法、及び、圧縮データ伸長方法," 特願2005-029117.
- [10] 金子晴彦、"LDPC符号を用いた低計算量ビデオ圧縮法" (掲載予定), 2005年電子情報通信学会ソサイエティ大会講演論文集, A-6-5, Sep. 2005.
- [11] B. Girod, et al., "Distributed Video Coding," Proc. IEEE Special Issue on Advances in Video Coding and Delivery, Vol.93, No.1, pp.71–83, Jan. 2005.
- [12] R. Gallager, "Low-Density Parity-Check Codes," IEEE Trans. Information Theory, Vol.8, No.1, pp. 21–28, Jan. 1962.
- [13] D. J. C. Mackay, "Good Error-Correcting Codes Based on Very Sparse Matrices," IEEE Trans. Information Theory, Vol.45, No.2, pp.399–431, March 1999.
- [14] <http://ftp3.itu.ch/av-arch/video-site/sequences/>
- [15] <http://www.tmpgenc.net/>