J_011

低密度パリティ検査符号と多元Sum-Productアルゴリズムを用いた動画像圧縮法

Video Coding Based on Low-Density Parity-Check Codes and Nonbinary Sum-Product Algorithm

金子晴彦*
Haruhiko Kaneko

1 まえがき

動画像圧縮符号化方式として，従来より動き予測に基づくフレーム間符号化方式が広く用いられている．MPEG等に代表されるフレーム間符号化方式は，フレーム内符号化方式と比較して高い圧縮率を有するが，圧縮処理において動きベクトルの検索を行なうことから，圧縮のための計算量が多いという特徴を有する．一方，携帯型テレビ電話，画像センサーネットワーク，人工衛星，等においては，電源容量や搭載可能な回路規模に制約があることから，計算量の少ない動画像符号化方式が適している．

近年，圧縮処理における計算量が少ない動画像符号化方式として，Distributed Video Coding (DVC) が提案されている [1]．DVCにおいては誤り制御符号を用いることにより，動き予測を行なわずに効率的に動画像圧縮を行なうことができる．すなわち，圧縮処理においてはフレーム内符号化を行ない，伸長処理においてフレーム間相関を利用した復号を行なう．本稿では，著者が従来提案した低密度パリティ検査 (LDPC) 符号を用いたDVC[2] を拡張し，多元Sum-Productアルゴリズムを用いた新たなDVCを提案する．

2 圧縮アルゴリズム

以下に圧縮アルゴリズムの概略を示す．動画像はIピクチャ(Intra-coded picture) 及びLピクチャ(LDPC-coded picture) の2種のピクチャより構成し，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．以下にLピクチャの圧縮手順を示す．

(1) 縦 $H\sigma$ 画素，横 $W\sigma$ 画素を有するLピクチャを行列形式で $\mathbf{L} = [l_{i,j}]_{H\sigma\times W\sigma}$ と表現する．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 $L_0,\ldots,L_{N-1}$ に分割し，マクロブロックベクトルを $\mathrm{MBV}(\mathbf{L}) = (L_0,\ldots,L_{N-1})$ とおく．図1に $\mathbf{L}$ と $\mathrm{MBV}(\mathbf{L})$ の関係を示す．

(2) 2元 $(N, N-M)$LDPC 符号の検査行列を $\mathbf{H} = [h_{i,j}]_{M\times N}$ とする．行列 $\mathbf{H}$ を用いて，長さ N ブロックを有するマクロブロックベクトル $\mathbf{L}$ を，長さ $M(<N)$ ブロックを有するシンドロームブロックベクトル $\mathbf{S} = (S_0,\ldots,S_{M-1})$ に変換する．ただし，$S_i = \left(\sum_{j\in\{j|h_{i,j}=1\}} L_j\right)$ は $\sigma\times\sigma$ シンドロームブロックである．ここで，Σ は行列の各要素についてそれぞれ整数環上で和をとることを示す．

(3) シンドロームブロックベクトル $\mathbf{S}$ に対しJPEGと同様の非可逆圧縮を適用する．

上記の手順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，Lピクチャは LDPC符号の検査行列を用いることにより，圧縮符号化するマクロブロック数を削減していることから，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が得られる．

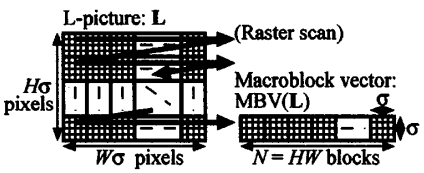

図1 Lピクチャとマクロブロックベクトルの関係

3 伸長アルゴリズム

Iピクチャは JPEGの復号アルゴリズムに基づいて伸長する．伸長処理対象のLピクチャ $\mathbf{L}$ が，2枚のIピクチャ $\mathbf{I}^0$ 及び $\mathbf{I}^1$ の間に存在すると仮定する．ただし，図2に示すように $\mathbf{I}^0$ と $\mathbf{L}$ の間には λ_0 枚のLピクチャが存在し，$\mathbf{L}$ と $\mathbf{I}^1$ の間には λ_1 枚のLピクチャが存在するものとする．伸長手順の概略を図3に示す．伸長アルゴリズムは以下のとおりである．

3.1 準備 縦 $H\sigma$ ピクセル，横 $W\sigma$ ピクセルを有するフレーム $\mathbf{I}^0, \mathbf{I}^1$ 及び $\mathbf{L}$ を以下のように行列で表現する．

$$\mathbf{I}^0 = [\iota^0_{i,j}]_{H\sigma\times W\sigma},\ \mathbf{I}^1 = [\iota^1_{i,j}]_{H\sigma\times W\sigma},\ \mathbf{L} = [l_{i,j}]_{H\sigma\times W\sigma}.$$

任意の2枚のフレーム $\mathbf{A} = [a_{i,j}]_{H\sigma\times W\sigma}$ 及び $\mathbf{B} = [b_{i,j}]_{H\sigma\times W\sigma}$ において，$\mathbf{A}$ の点 a_{i_a,j_a} と $\mathbf{B}$ の点 b_{i_b,j_b} を中心とする平均絶対値誤差 (Mean Absolute Difference)

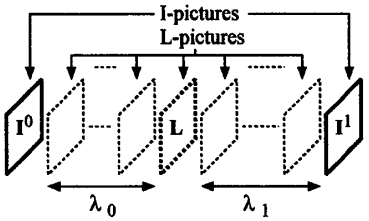

図2 フレーム列の例

*宇宙航空研究開発機構, JAXA

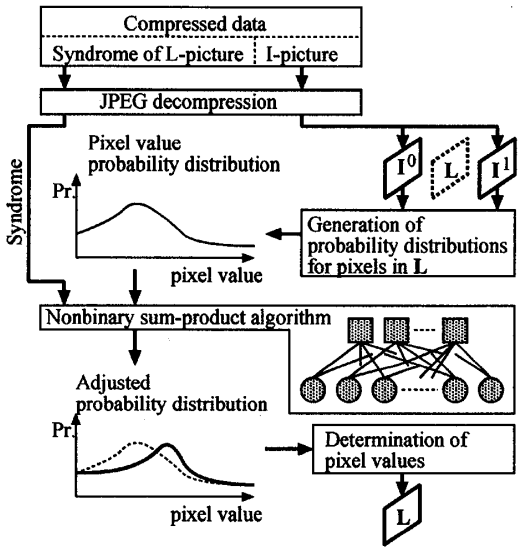

図 3　伸長手順の概略

を以下のように定義する．

$$\mathrm{MAD}_R(\mathbf{A},\mathbf{B};i_a,j_a;i_b,j_b)=\frac{\sum_{s=-R}^{R}\sum_{t=-R}^{R}|a_{i_a+s,j_a+t}-b_{i_b+s,j_b+t}|}{(2R+1)^2}.$$

ただし，参照ウィンドウサイズは $(2R+1)\times(2R+1)$ である．

3.2 画素値確率分布の生成　ここで，Iピクチャは JPEG に基づいて既に伸長されているから $\mathbf{I}^0$ 及び $\mathbf{I}^1$ は既知であり，Lピクチャ $\mathbf{L}$ は未知である．Lピクチャ $\mathbf{L}$ の前方及び後方のIピクチャ $\mathbf{I}^0,\mathbf{I}^1$ が与えられたとき，$\mathbf{L}$ 上の画素 $l_{i,j}$ が明るさ w を有する条件付き確率を

$$R_w^{i,j}=\Pr(l_{i,j}=w\mid\mathbf{I}^0,\mathbf{I}^1)$$

とあらわす．多元 Sum-Product アルゴリズムを実行する準備として，各画素 $l_{i,j}$ における画素値確率分布 $\mathbf{R}_{i,j}=(R_0^{i,j},R_1^{i,j},\ldots,R_{\Lambda-1}^{i,j})$ を求める．ただし，$\Lambda-1$ は画素値の最大値である．一般の動画像に対して正確な画素値確率分布を求めることは困難であるが，本稿では以下の値を用いる．

$$R'^{i,j}_w=\max_{\substack{(X_a,Y_a;X_b,Y_b)\\ \in\mathbf{W}_w^{i,j}}}\left(\frac{\Lambda-1}{\mathrm{MAD}(\mathbf{I}^0,\mathbf{I}^1;i+X_a,j+Y_a;i+X_b,j+Y_b)}\right)$$

ただし，$\mathbf{W}_w^{i,j}$ は以下に示す検索ベクトルの集合である．

$$\mathbf{W}_w^{i,j}=\{(X_aY_a;X_bY_b)|X_a=(\lambda_0{+}1)x,Y_a=(\lambda_0{+}1)y,\ X_b=-(\lambda_1{+}1)x,Y_b=-(\lambda_1{+}1)y,-\rho_v\le x\le\rho_v,\ -\rho_h\le y\le\rho_h,(\iota^0_{i+X_a,j+Y_a}=w)\vee(\iota^1_{i+X_b,j+Y_b}=w)\}$$

ここで，検索ウィンドウサイズは $(2\rho_v+1)\times(2\rho_h+1)$ である．分布 $(R'^{i,j}_0,R'^{i,j}_1,\ldots,R'^{i,j}_{\Lambda-1})$ を正規化した分布 $\mathbf{R}_{i,j}=(R_0^{i,j},R_1^{i,j},\ldots,R_{\Lambda-1}^{i,j})$ を画素値確率分布とする．

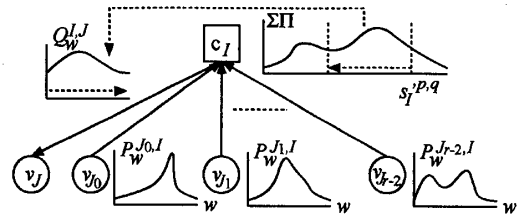

図 4　C ノードにおける確率分布計算

3.3 多元 Sum-Product アルゴリズム　上記で求めた画素値確率分布 $\mathbf{R}_{i,j}$ 及び，伸長したシンドロームブロックベクトル $\mathbf{S}'=(S'_0,S'_1,\ldots,S'_{M-1})$ を用いて，多元 Sum-Product アルゴリズムを実行することにより，Lピクチャの各画素の推定値を決定する．ただし，$S'_I=[s'^{p,q}_I]_{\sigma\times\sigma}$ は $\sigma\times\sigma$ シンドロームブロックである．従来の2元 Sum-Product アルゴリズム [3] は，以下のように容易に多元上のアルゴリズムへ拡張できる．ただし，以下のアルゴリズムは，すべての $(p,q)\in\{(p,q)|0\le p\le\sigma-1,0\le q\le\sigma-1\}$ に対してそれぞれ独立に実行する．

(1) 初期化　Tanner グラフを $G=(V,E)$ とおく．ただし，$V=\{v_0,v_1,\ldots,v_{N-1},c_0,c_1,\ldots,c_{M-1}\}$ は点集合，$E=\{(c_i,v_j)|h_{i,j}=1\}$ は枝集合である．V ノード v_J における初期画素値確率分布を

$$(P_0^J,P_1^J,\ldots,P_{\Lambda-1}^J)=(R_0^{i,j},R_1^{i,j},\ldots,R_{\Lambda-1}^{i,j})$$

とおく．ここで J と (i,j) の関係は，図1より，$i=\lfloor J/W\rfloor\times\sigma+p$, $j=(J\mod W)\times\sigma+q$ で与えられる．枝 $(c_I,v_J)\in E$ 上のメッセージ $\mathcal{M}(v_J\to c_I)$ を確率分布 $(P_0^{J,I},P_1^{J,I},\ldots,P_{\Lambda-1}^{J,I})=(P_0^J,P_1^J,\ldots,P_{\Lambda-1}^J)$ として初期化する．

(2) C ノードにおける更新　C ノード C_I に隣接する r 個の V ノードの集合を $\{v_J,v_{J_0},v_{J_1},\ldots,v_{J_{r-2}}\}$ とおく．メッセージ $\mathcal{M}(c_I\to v_J)=(Q_0^{I,J},Q_1^{I,J},\ldots,Q_{\Lambda-1}^{I,J})$ を以下の式により生成する．

$$Q_w^{I,J}=\sum_{(x_0,\ldots,x_{r-2})\in\Gamma_w}\left(\prod_{t=0}^{r-2}P_{x_t}^{J_t,I}\right)$$

ただし，$w\in\{0,1,\ldots,\Lambda-1\}$,

$$\Gamma_w=\left\{(x_0,\ldots,x_{r-2})\left|\sum_{t=0}^{r-2}x_t=(s'^{pq}_I-w),x_t\in\{0,\ldots,\Lambda-1\}\right.\right\}$$

である．すべての枝 $(c_I,v_J)\in E$ において上記の式によりメッセージ $\mathcal{M}(c_I\to v_J)$ を生成する．

C ノードにおける確率分布計算を図 4 に示す．

(3) V ノードにおける更新　V ノード v_J に隣接する c 個の C ノードの集合を $\{c_I,c_{I_0},c_{I_1},\ldots,c_{I_{c-2}}\}$ とおく．メッセージ $\mathcal{M}(v_J\to c_I)=(P_0^{J,I},P_1^{J,I},\ldots,P_{\Lambda-1}^{J,I})$ を以下の式により生成する．

$$P_w^{J,I}=kP_w^J\prod_{t=0}^{c-2}Q_w^{I_t,J}$$

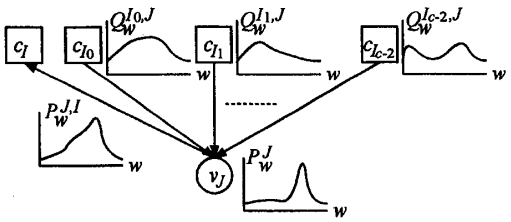

図5　Vノードにおける確率分布計算

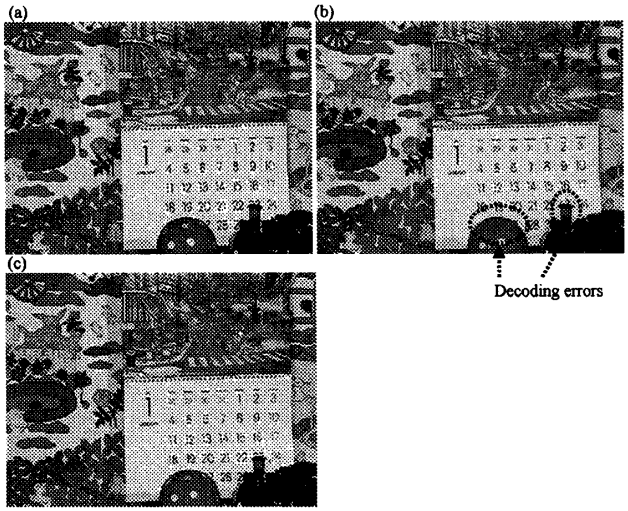

図6　評価画像による例．(a) 原画像．(b) 伸長後の画像 (多元 Sum-Product アルゴリズムの反復0回)．(c) 伸長後の画像 (多元 Sum-Product アルゴリズムの反復5回)．

ただし，k は正規化のための係数である．Vノードにおける確率分布計算を図5に示す．

上記手順の (2) と (3) は一定回数反復して実行する．

(4) **画素値の決定** Vノード v_J に隣接する c 個のCノードの集合を $\{c_{I_0}, c_{I_1}, \ldots, c_{I_{c-1}}\}$ とおく．V_J における画素値 l_J を以下の式により定める．

$$l_J = \arg\max_w \left(P_w^J \prod_{t=0}^{c-1} Q_w^{I_t,J} \right)$$

以上より，v_J に対応するLピクチャの画素値を得る．

4　評価

主観的評価　多元 Sum-Product アルゴリズムの効果を視覚的に示すため，図6に評価画像の例を示す．ここで，(a) は原画像，(b) は多元 Sum-Product アルゴリズムの反復回数を0回とした場合の伸長後の画像，(c) は多元 Sum-Product アルゴリズムの反復回数を5回とした場合の伸長後の画像，である．図6(b) に存在する誤りが，図6(c) では消去できていることから，提案した多元 Sum-Product アルゴリズムの有効性を確認できる．

PSNR　評価用動画像「mobile」[4] を用いて，提案手法，Motion JPEG 及び MPEG-1 の PSNR(Peak Signal-to-Noise Ratio) を評価した結果を図7に示す．ただし，提案手法における Sum-Product アルゴリズムの反復回数は5回である．提案手法の評価においては圧縮率向上のために，圧縮処理において非常に単純な動き予測を行っている．なお，このために必要な計算量は MPEG-1 と比較して10分の1以下であり，提案手法の有効性は失われない．各符号化方式とも圧縮率は約 65:1 である．PSNR の平均値を比較すると，提案手法が 29.48[dB] であるのに対し，Motion JPEG は 25.05[dB]，MPEG-1 は 32.54[dB] であり，今後は画質のさらなる向上が必要であることが明らかになった．

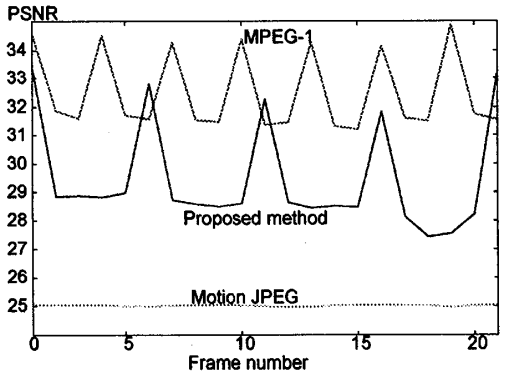

図7　PSNR の評価（圧縮率 65:1; 多元 Sum-product アルゴリズムの反復回数5回）

圧縮・伸長計算量　提案手法の圧縮処理における計算量は，単純な動き予測処理を含めても Motion JPEG と同程度である．一方，伸長処理においては多数の浮動小数点演算が必要であることから，大型計算機レベルの処理が必要であり，今後は伸長処理計算量を削減するための研究が必要である．

5　まとめ

本稿では，LDPC 符号と多元 Sum-product アルゴリズムに基づく DVC を提案した．提案手法について画質の評価を行い，Motion JPEG よりも高い PSNR が得られることを明らかにした．

謝辞

本研究は稲盛財団研究助成金による補助を受けて実施したものである．

参考文献

[1] 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.

[2] H. Kaneko, "Low Complexity Video Compression Based on Low-Density Parity Check Codes," *IEICE Trans.*, Vol.E89-A, No.1, pp.340–347, Jan. 2006.

[3] 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.

[4] http://ftp3.itu.ch/av-arch/video-site/sequences/