

## 不均一誤り保護方式を用いたメディア処理システムの計算量削減手法

田島 直樹\* 清水一範\*\* 池永 剛\* 後藤 敏\*  
\*早稲田大学大学院情報生産システム研究科 \*\*早稲田大学基幹理工学研究所

### 概要

本論文では、不均一誤り保護方式による計算量削減メディア通信システムの提案を行う。提案システムでは、H.264/AVC 圧縮符号化技術と LDPC コードを結合している。H.264/AVC ビットストリームに誤りが発生すると、デコーダー側では誤りのあるスライスに対してエラーコンシールメントが行われる。その際、参照されるピクチャにも誤りがあるとエラーコンシールメントの精度が落ちてしまう。そこで、異なった符号化率によって不均一な強度で誤り訂正を施すことによって、参照される箇所にも誤りが発生することを防ぐ。このとき、均一の符号化率を用いるときと比べ計算時間を削減する。ソフトウェアシミュレーションによって提案手法は、不均一誤り保護方式を用いないシステムと比べて、最大で 28.24% の計算時間を削減できることが分かった。そのときの画質は PSNR からほとんどかわらないことがわかった。

## Low Complexity Video Compression System with Unequal Error Protection

Naoki Tajima\* Kazunori Shimizu\*\* Takeshi Ikenaga\* Satoshi Goto\*  
\*Graduate School of Information, Production and Systems, Waseda University, Japan  
\*\*Dept. of Computer Science, Waseda University, Japan

### Abstract

We proposed low complexity video compression system with unequal error protection(UEP). In this proposal, we combine H.264/AVC and LDPC codes. If H.264/AVC bit stream has some errors during transfer, the error concealment is done at the decoder side. If the slice which is referred to complement the lost part has some errors, quality of error concealment is become worse. We use UEP to avoid this situation. UEP is known as the method to reduce computational complexity. We reduce computational complexity by using UEP, while quality of video is kept. Our system can reduce 28.48% computational time compared with the system without UEP.

### 1. まえがき

マルチメディアデータの無線通信は、モバイル製品の増加および高機能化によって頻繁に行われるようになった。そこで重要な問題となるのが無線通信システムにおけるノイズである。ノイズが発生した場合、デコーダー側で誤りを訂正しなくてはならない。モバイル製品において、この処理はバッテリーの駆動時間に影響を与える。

したがって、誤りを訂正する際の処理を効率よく減少させることが重要となる。不均一誤り保護方式 [1], [2] は、重要度に応じて異なった強さの誤り訂正符号を適用することによって、均等に誤り訂正符

号を適用したときと比べ計算量を削減することができる。

私たちは誤り訂正符号化技術として LDPC [3] を用いた。また、動画像圧縮符号化技術として H.264/AVC [4], [5] を採用した。

この論文では、不均一誤り保護方式によるメディア通信システムの計算量削減手法を提案する。今回提案する不均一誤り保護方式は、H.264/AVC レファレンスソフトウェアデコーダー [6] に標準で装備されているエラーコンシールメント [7] に対して精度の観点から考慮したものである。エラーコンシールメントの際に誤り箇所を補完したときの画質がよくなるように、本来は重要度が同じであるスライスに対して、重要度の差をつける。これによって、補完

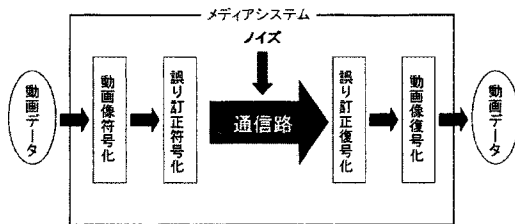


図1 対象メディアシステム

の際に参照される箇所の誤りをなくし、エラーコンシールメントの精度を向上させる。この際に画質の劣化を多少許すことにより、計算量の削減を行う。

ソフトウェアシミュレーションによって、不均一誤り保護を行わない手法より、最大で 28.24% の計算時間が削減できることがわかった。また、このときの動画像の視覚評価では、エラーがない場合とほとんどかわらない結果を得た。

本論文の残りは以下のように構成される。第2節では本研究のターゲットであるメディアシステム、エラーコンシールメントについて述べる。さらに、第3節では今回提案した計算量削減を狙ったメディアシステムについて述べる。第4節では提案手法に基づいたシミュレーション結果を示し、まとめを第5節で述べる。

## 2. メディアシステム

### 2.1 メディアシステム概要

本研究では、動画を動画圧縮符号化技術を用いて圧縮し、その後誤り訂正符号を付加し通信を行った後、受信側で誤りを訂正し、動画をデコードする一連のシステムの計算量削減を対象とする。このシステムを図1に示す。動画圧縮符号化技術としては、高い圧縮率を実現した H.264/AVC を採用した。また、誤り訂正符号はシャノンの限界に近い特性を持つ LDPC 符号を採用した。

### 2.2 LDPC による不均一誤り保護

LDPC による不均一誤り保護を実現するためには、符号化率、符号長、繰り返し回数、列重み、GIRTH の変更によるものがある。列重みによる不均一誤り保護方式が [1], [2] に紹介された。列重みによる不均一誤り保護方式では、重要データおよび非重要データに割り当てる列重みの区別が難しく、検査行列によって制限されるため能力に限界がある。しかし、

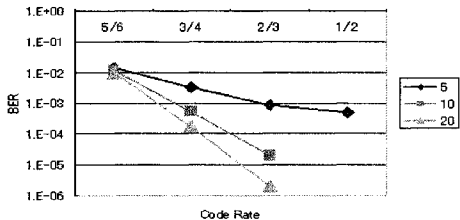


図2 符号化率とビットエラーレートの関係

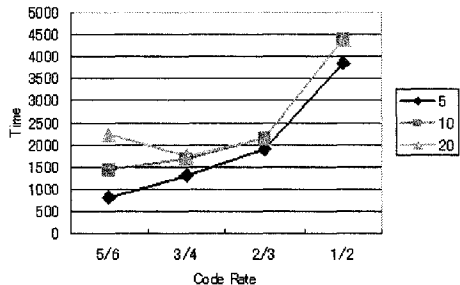


図3 符号化率と計算時間の関係

これらの不均一誤り保護方式の実現方法の中でも、符号化率、繰り返し回数は制御が比較的簡単で、検査行列による制限がない。そこで、今回は符号化率の変更を採用した。符号化率を変更したときの、ビットエラーレート(BER)とデコード時間をそれぞれ図2, 3に示す。このときの SNR は 3.0 とした。図2, 3にはそれぞれ繰り返し回数が 5 回, 10 回, 20 回のときの結果が示されている。図2において符号化率が 1/2, 繰り返し回数が 10, 20 回のときのビットエラーレートの値がないのは、誤りを完全に訂正できていることを表している。

これらの図から符号化率が小さくなると BER は改善されるが、計算時間が長くなっていることがわかる。

### 2.2 エラーコンシールメント

H.264/AVC においてデコーダ側では受信したビットストリームに誤りがある場合、誤りのあるスライスを捨て、捨てられたスライスは補完される [7]。補完は、マクロブロック単位で行われ、最初にモーションベクトルが上下左右のマクロブロックから推測される。図4に示されるように補完されるブロッ

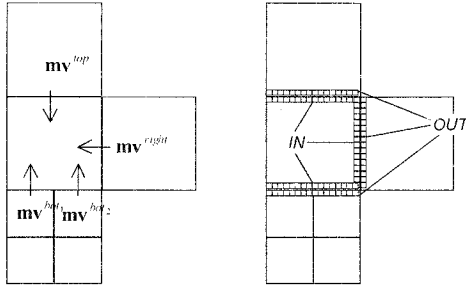


図4 モーションベクトルの補完候補

クのまわりのモーションベクトル全てが評価される。各モーションベクトルにおいて次の式が計算され、その値が最も小さいモーションベクトルが採用される。

$$\min_{\text{dire}\{top, bottom, left, right\}} \arg \left\langle d_{sm} = \left( \sum_{j=1}^N \left| \hat{Y}(mv^{dir})_j^{IN} - Y_j^{OUT} \right| \right) / N \right\rangle$$

また、上下左右のモーションベクトルとは別に、モーションベクトルが0の場合が常に評価され、動きの少ない動画については0モーションベクトルが最も有力な候補となる。

これらの補完アルゴリズムから、補完の際に補完されるマクロブロックの上下左右および参照ブロックに誤りがあると、補完の精度が落ちてしまう。

### 3. 提案システム

今回提案するメディアシステムは図5のようなになる。まず、動画を H.264/AVC エンコーダによって圧縮する。そのビットストリームをスライスグループごとに二つに分割し、それぞれに強度の違う LDPC コードを付加する。本研究では、LDPC の強弱は、符号化率の変更によって行った。通信後、LDPC により誤り訂正を行い、誤りのあるスライスを削除する。その後、再びビットストリームを結合し、H.264/AVC デコードを行う。この際エラーコンシールメントが行われる。

エラーコンシールメントの際に、補完されるマクロブロックの上下左右および参照ピクチャに誤りがあると、補完の精度が落ちてしまう。そこで、補完されるマクロブロックの上下左右のマクロブロック

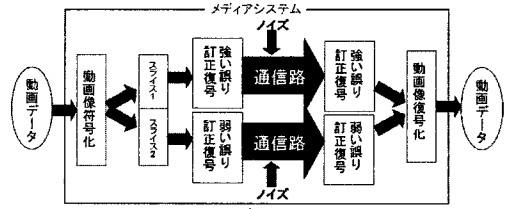


図5 提案システム

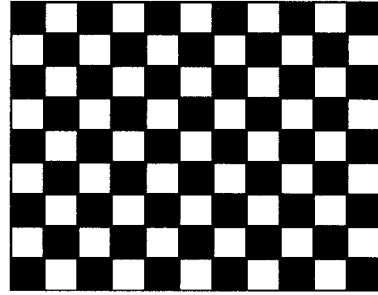


図6 ディスパース

は誤りが発生しないように強い誤り訂正符号で守られなければならない。そのために、各マクロブロックに上下左右に隣接するマクロブロックは中心のマクロブロックと違うスライスグループに属されている必要がある。これを実現するスライス分割はディスパースである。よって、本提案ではディスパースを採用した。ディスパースを図6に示す。

分割した各スライスは違う強度の誤り訂正符号によって守られる。図6において、一つの白いマクロブロックに注目する。白いマクロブロックに誤りがあると、その周りの上下左右の黒いマクロブロックを利用して白いマクロブロックの補完が行われる。そこで、白い部分を重要度の観点から非重要な部分とすると、黒い部分は重要な(エラーコンシールメント精度の観点から誤りが発生してはいけない)部分となる。そこで、白いスライスを符号化率 2/3 で誤り訂正を行うのに対して、黒い部分は符号化率 1/2 で誤り訂正を行い、誤りが発生することを防ぐ。このように、本来、データ自体に重要度の差がないものに対し、あえて重要度を定義し不均一な誤り訂正を施すことによって、計算量の削減を狙う。ただし、このときの重要度の与え方は、エラーコンシールメントの際に補完の精度を向上させる重要度定義としている。

	同じスライスにエラー	別のスライスにエラー
foreman	28.21854	28.492925
football	22.242486	22.715011
container	36.089604	36.369891
salesman	34.908611	35.578079
bus	23.914764	24.706577
canoa	24.328691	24.116798
table	26.199558	26.892287
CarPhone	31.598678	30.572055
mobile	24.16881	26.99137
headwithglasses	26.369891	27.796566

また、上下左右のマクロブロックと同様に、参照される部分も強く守られなければならない。参照される箇所は、一つ前のピクチャの同じ箇所が最も有力である。また、一つ前のピクチャの同じ箇所は補完されるマクロブロックの上下左右のモーションベクトルに関係なく、毎回評価に利用されるものである。表 1 2 枚連続でエラーが発生したときの PSNR 比較。そこで、一つ前のピクチャの同じ箇所のマクロブロックも強い誤り訂正符号で守ることによって、エラーコンシールメントの精度を高めることができる。そのため、図 6 において色分けされたように、白を重要度が非重要、黒を重要部分と全てのピクチャで決定するのではなく、一枚毎に白と黒の重要度を入れ替えることによって、参照ピクチャに誤りが発生することを防ぐ。図 6 のディスパースにおいて、黒スライスに 2 枚連続でエラーが発生した場合、1 枚目は黒スライスにエラーが発生し 2 枚目を白スライスにエラーが発生した場合、つまり、2 枚連続で同じスライスにエラーが連続で発生した場合、連続でピクチャに誤りが発生するが違うスライスにエラーが発生した場合のエラーコンシールメントの精度評価を表 1 に示す。10 個の動画について実験を行った結果、ほとんどの動画で違うスライスにエラーが発生したほうが PSNR がよいことがわかる。

また、I ピクチャに誤りが発生した場合、エラーコンシールメントによって補完されたピクチャは画質の劣化が大きい。I ピクチャと P ピクチャのエラーコンシールメントの結果を図 7, 8 に示す。図から明らかに I ピクチャのエラーコンシールメントの精度が悪いことがわかるが、PSNR を比較しても I ピクチャは 22.38, P ピクチャは 31.37 と、違いを簡単に発見することができる。よって、I ピクチャはスライスグループに関係なく、全て重要データとした。



図 7 I ピクチャのエラーコンシールメント  
PSNR=22.38



図 8 P ピクチャのエラーコンシールメント  
PSNR=31.37

#### 4. 実験結果

I ピクチャが 30 枚に 1 枚の割合で含まれる 3 つ動画について SNR を変更したときの、LDPC デコーダーの計算時間および PSNR をソフトウェアシミュレーションによって調べた。動画は foreman, football, container を用いた。

提案システムでは重要データを符号化率 1/2 で誤り訂正を行い、その他のデータを符号化率 2/3 で誤り訂正を行った。一方、不均一誤り保護方式を用いないシステム(NUEP)では符号化率は全て 1/2 で統一した。

提案システムと不均一誤り保護方式を行わないシステム(NUEP)の計算時間と PSNR を、foreman,

表 2 foreman における計算時間および PSNR

動画	foreman						
SNR	2	2.1	2.2	2.3	2.4	2.5	
計算時間(s)	提案	41.89	42.4	41.86	40.54	40.89	41.87
	NOUEP	55.06	59.03	55.77	57.22	55	56.05
PSNR(dB)	提案	29.42	29.42	29.48	34.98	34.98	35.06
	NOUEP	31.04	31.04	31.17	34.43	35.83	35.83
計算時間削減率(%)	23.92	28.18	24.94	29.15	25.65	25.3	
PSNR差(dB)	-1.62	-1.62	-1.69	0.55	-0.85	-0.77	
平均削減率(%)	26.19125226						
平均PSNR差(dB)	-1						



図 9 foreman 出力画像 (a)均一保護 (b)提案

表 3 football における計算時間および PSNR

動画	football						
SNR	2.7	2.8	2.9	3	3.1	3.2	
計算時間(s)	提案	232.5	231.6	232.4	230.8	231.5	231.2
	NOUEP	320.8	319.1	319.1	337.7	322.7	318.4
PSNR(dB)	提案	26.16	26.19	26.6	29.25	33	34.02
	NOUEP	26.49	26.49	27.32	30.21	34.02	34.02
計算時間削減率(%)	27.53	27.43	27.18	31.63	28.26	27.4	
PSNR差(dB)	-0.33	-0.3	-0.72	-0.96	-1.02	0	
平均削減率(%)	28.23997487						
平均PSNR差(dB)	-0.555						



図 10 football 出力画像 (a)均一保護 (b)提案

表 4 container における計算時間および PSNR

動画	container						
SNR	1.9	2	2.1	2.2	2.3	2.4	
計算時間(s)	提案	23.78	23.59	23.94	23.9	23.82	23.38
	NOUEP	27.24	27.36	27.11	27.03	27.41	27.43
PSNR(dB)	提案	35.8	35.8	35.83	35.83	36.27	36.32
	NOUEP	36.08	35.78	36.24	36.1	36.14	36.32
計算時間削減率(%)	12.7	13.76	11.7	11.57	13.08	14.75	
PSNR差(dB)	-0.28	0.02	-0.41	-0.27	0.13	0	
平均削減率(%)	12.92756325						
平均PSNR差(dB)	-0.135						



図 11 container 出力画像 (a)均一保護 (b)提案

football, container の 3 つの動画について表に示す。表 2 は foreman に対する結果を、表 3 は football, 表 4 は container の結果をそれぞれ示している。各 SNR について計算時間削減率, PSNR 差も示している。動きの少ない動画である container では、重要データ量が非重要データ量より大きいため、計算量の削減率が小さいが、動きの大きい foreman や football では最大で 28.24% の計算時間が削減できていることがわかる。

また、PSNR から、画質もほぼかわらないことがわかる。さらに、視覚特性評価の結果を図 9 から図 11 に示す。図 9 は foreman における不均一誤り保護方式を用いない場合の出力画像 (a) と提案システムにおける出力画像 (b) を示している。実験を行った最も劣悪な条件である SNR が 2.0 のときの結果を示す。同様に、図 10 は football の結果を示している。football は SNR が 2.7 のときの結果で、container は SNR が 1.9 のときの結果である。これらの比較から、提案システムにおい

て、誤り訂正能力の低い符号化率が大きなものを使用しているにもかかわらず、符号化効率の良いものだけを使用する従来のシステムとほぼかわらない画質の出力結果を得たことがわかる。

## 5. まとめ

H.264/AVC のビットストリームにおいてエラーコンシールメントの精度の観点から重要データを定義し、不均一誤り保護によって計算時間の削減を行った。重要データは非重要データの上下左右のマクロブロックとし、さらに非重要データの一つ前のピクチャの同じ箇所も重要データとした。これによって、誤り訂正の計算時間を削減しても、高いエラーコンシールメントの精度によって、画質の劣化をおさえ

た。PSNR 測定の結果、提案手法は不均一誤り保護方式を用いない場合とほぼ同じ画質を得た。また、視覚特性による評価においてもほぼ同様の結果を得た。その際、ソフトウェアシミュレーションによる計算時間は不均一誤り保護方式を用いない従来手法と比べて最大で約 28.24%削減することができた。

#### 謝辞

本研究は CREST, JST の支援による。

#### 参考文献

- [1]N. Rahnavard and F. Fekri, "Unequal Error Protection Using Low-Density Parity-Check Codes", ISIT 2004, Chicago, USA, June
- [2]C. Poulliat, D. Declercq, and I. Fijalkow, "Optimization of LDPC Codes for UEP Channels", ISIT 2004, Chicago, USA, June.
- [3]K. Shimizu, T. Ishikawa, N. Togawa, T. Ikenaga, S. Goto, "ASIC Implementation of LDPC Decoder Accelerating Message-Passing Schedule", Student Design Contest, IEEE International Solid State Circuits Conference(ISSCC), San Francisco, Feb. 5. 2006.
- [4]Joint Video Team, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC), Mar. 2003.
- [5]G. J. Sullivan, P. Topiwala and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions", SPIE Conf. on applications of digital image processing XXVII, Special session on advances in the new emerging standard: H.264/AVC., vol. 5558, pp. 53-74, Aug. 2004
- [6]Joint Video Team Reference Software, Version 11.0, <http://iphome.hhi.de/suehring/ttml/download/>
- [7]Viktor Varsa, etc. "Non-normalized error concealment algorithms", VCEG-N62.