

M-012

マルチコアプロセッサ向き MPEG 符号化の並列処理に関する検討 Parallel Processing of MPEG Encoding for Multi-Core Processor

三浦 康之†
Yasuyuki Miura

柏木 文徳‡
Fuminori Kashiwagi

1. はじめに

PC が高性能なものとなり、近年ではマルチコア CPU を搭載した PC も一般的になった。しかし、動画の圧縮処理には依然時間がかかり、動画の編集者の負担になっている。そのような問題を解決するための一つの方法として、マルチコア CPU の特性を利用して効率的な並列処理を行うことにより、MPEG 符号化の高速化を図ることが考えられる。

我々は、マルチコアの CPU を搭載したクラスタを用いた並列圧縮システムに向けた CPU 内並列化アルゴリズムの検討を行っている。今回は、ごく基本的な領域分割アルゴリズムを実装し、その特性について詳細な検討を行ったので、その検討経過について報告する。

2. 動画並列圧縮システム

2.1. 既存のソフトウェアにおける問題点

多くの動画編集ソフトでは、動画エンコーダは 1 個のプロセッサによる逐次処理が基本となっていた。そのようなソフトでは、動画の圧縮の際、圧縮の方法によっては再生時間の数倍の時間を有するものもあり、ユーザにとって大きな負担となる。

近年になってマルチコアプロセッサの普及に合わせてマルチコア向けの並列処理法に関する研究が進められ、一部市販されている[1]ものの、アルゴリズムの複雑さからコア数の増加に伴って並列化による速度向上が実装上難しくなるという問題がある。

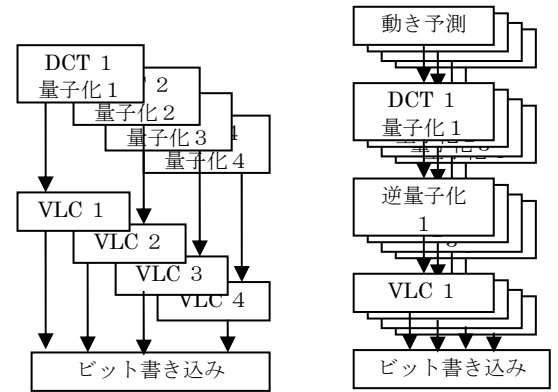
2.2. 提案システム

上記の背景を踏まえ、端末から離れた PC を用いて動画を高速に圧縮するシステムの開発を進めている[2]。このシステムは、通信網に特別線を使わず、それぞれのクラスタ、利用者の PC を Gigabit Ethernet で結合している。利用者は、クラスタマシンに符号化する動画を送り、クラスタマシンは、符号化処理が終了し次第、利用者の端末へ視聴可能な状態で戻す。今回、検討を行うのはクラスタ内の並列化である。

3. 並列処理アルゴリズム

3.1. 並列化アルゴリズム

まずは単純な空間分割の並列化アルゴリズムの実装を行った。空間分割の方法は、単純に画面を横に 4 分割し、それぞれを 1 つのコアに処理させるというものである。MPEG-4 における並列処理の流れを図 1 に示す。図 1 に示



a) I-VOP のみの場合 b) フレーム間予測を行う場合

図 1 空間分割による並列処理の流れ

すように、符号化の流れは I-VOP のみの場合と動き予測を含む場合で異なり、I-VOP のみの場合は DCT、量子化、VLC、符号化ビット列書き込みの順序で処理を行い、動き予測を含む場合にはそれらに加えて動き予測や IDCT、逆量子化の処理を含んでいる。これらの各処理について空間分割して順番に処理することになるが、ビット列の書き込みについては並列化が不可能なので、一つのプロセスに符号化ビット列を集めて逐次的に書き込みを行っている。また、空間分割による方法では AC/DC 予測符号化を実行することができないので、本アルゴリズムでは AC/DC 予測符号化を行わない設定としている。

3.2. 実装

PC 内で並列処理を行う代表的な方法としてマルチスレッドとマルチプロセスがある。前者は、スレッドの生成に伴うオーバーヘッドが少なく、メモリ領域の節約ができるが、多数のパラメータと大量のデータを扱う MPEG 符号化では実装になじまない方法であることから後者の方法を採用した。MPEG エンコーダは、VOP 枚ごとに繰り返し処理を行っているので、VOP の符号化を行うループの外でプロセスを生成することにより、プロセス生成に伴うオーバーヘッドの問題は解決される。

マルチプロセスにおけるプロセス間通信には、いくつかの方法がある。今回は、大容量のデータに対するランダムアクセスが可能な共有メモリと、同期が容易なパイプライン通信を組み合わせる方法を採用した。具体的には、取得したフレームや符号化後、P-VOP や B-VOP が参照するフレームは、共有メモリに格納する方法を採用し、は容量が小さく、かつプロセス間の同期が必要な符号化ビットストリームの通信はパイプライン通信を利用した。

† 湘南工科大学 工学部情報工学科, Department of Information Science, Shonan Institute of Technology

‡ 湘南工科大学 大学院工学研究科, Graduate School of Technology, Shonan Institute of Technology

4. 実験

4.1. 実験条件

空間を分割による並列化の効果を確認するため、MPEG-4 参照ソフトウェアを用いて3節で示した方法による並列化を行った。実験映像を図2に示す。図2のように、実験映像は樹木が激しく風に揺れるもので、空間による符号量に差が生じるという特徴がある。

30フレームすべてをI-VOPに符号化したものと、最初の1フレーム以外のすべてをP-VOPに符号化したものの2種類についての実験を行い、それぞれI-VOPとP-VOPの平均符号化時間を計測した。平均符号化時間を計測する際には、符号化時間全体とその内訳について計測した。

4.2. 実験結果

1フレームあたりの符号化時間を図3に示す。図3に示すように、I-VOPにおいてはビット列の書き込みによる負荷の集中が起こり、全体の速度向上が阻害されている。P-VOPにおいては、符号化時間に占めるビット列の書き込み時間の占める割合が低いため、このような現象は見られない。表1に、符号化ビットレートを変えて速度向上比を比較した結果を示す。表1のように、I-VOPでは4コアで概ね2.3~2.4倍、P-VOPでは3倍強の速度向上が見られる。

5. 並列処理アルゴリズムの改善

実験の結果から、プロセッサコアによって符号化時間にばらつきがあること、およびビット列の書き込みによる多くの時間を要することが並列化による速度向上を阻害していることが明らかになった。そこで、空間分割の方法を工夫することによって、各プロセッサコアに均等に負荷を割り振る方法が必要になることが分かる。

また、本稿の実験ではMPEG-4[3]について実験を行ったが、マクロブロックの大きさを適応的に選択できるH.264ではコアごとの符号化時間のばらつきがさらに増大することが予想される。また、今回実験を行った空間分割による方法ではAC/DC予測符号化を行えないことから、AC/DC予測符号化を行うための工夫が必要になる。これらの問題については後日の課題になる。

6. おわりに

本稿では、マルチコアのCPU向けのMPEG-4並列符号化手法として、空間分割による方法の性能を評価した。今後は、フレーム内予測符号化が可能で、かつ負荷分散を行えるアルゴリズムの検討を、H.264における評価を見据えて進める必要がある。

参考文献

- [1] TMPGEnc: <http://www.tmpegenc.net/>
- [2] 柏木文徳, 三浦康之, MPEG-4符号化の並列処理に関する検討, 第70回情報処理学会全国大会, 3Y-5, 2008.3.
- [3] ISO/IEC 14496, Final Draft International Standard MPEG-4, 1998.



図2 実験映像

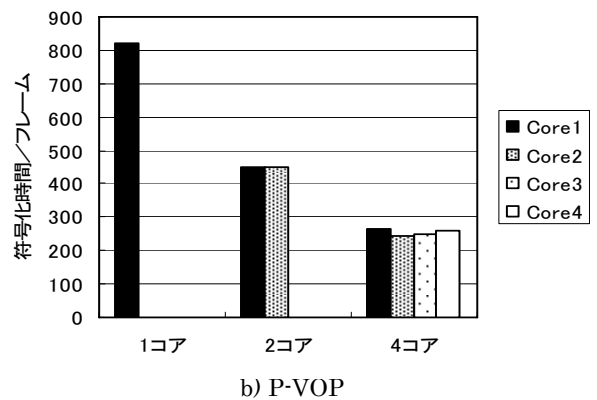
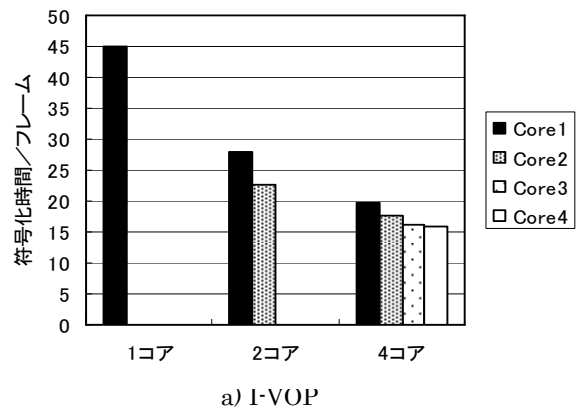


図3 1フレームあたりの符号化時間

表1 速度向上比の比較

	ビット数 (Kbits)	速度向上比	
		2コア	4コア
I-VOP	234.2	1.61	2.27
	145.9	1.72	2.41
P-VOP	67.9	1.81	3.12
	26.3	1.82	3.16