マルチメディア通信と分散処理 110－5
（２００２．１１．２１）

実時間環境における MPEG-4 ビジュアル符号化のための実証実験

三浦康之　　勝本道哲
独立行政法人　通信総合研究所
{miu,katumoto}@crl.go.jp

概要

我々は、複数の画像・音声を入力とし、リアルタイムで MPEG-4 ビデオオブジェクト(VO),への符号化・編集・配信を一体的に行うリアルタイムコンテンツ編集システムを提案する。本システムは、離れた複数地点からのデジタルビデオによるライブ映像を用いたコンテンツを提供するこが可能である。
本システムにおける DV 復号および MPEG-4 VO 符号化は多大な CPU パワーを必要とするため、時間的制約の厳しい実時間環境下で低スペックなマシンを用いてライブ映像の符号化を実行する場合複雑な符号化は困難となる。そこで、CPU パワーに応じた符号化処理を行うことで、リアルタイムな MPEG-4 VO への符号化を実現する。
本稿では、Pentium4 プロセッサ上における libdv の DV デコーダおよび MPEG-4 参照ソフトウェアのビデオエンコーダの処理時間測定結果を報告し、今後の方針について述べる。

Experiments for Real-time MPEG-4 Visual Encoder

Yasuyuki Miura,　Michiaki Katsumoto
Communications Research Laboratory
{miu,katumoto}@crl.go.jp

Abstract

We propose a real-time contents edition tool that can input multiple audio-visual streams, encode to MPEG-4 video object (VO), edit and deliver at the same time. By this system, we can provide contents by using some live video from multiple points.
Because large amount of CPU power is needed to decode from DV data and encode to MPEG-4 VO, it is difficult to encode live videos complicatedly by using lower spec PC in real-time condition. Thus, we research for algorithms of MPEG-4 VO encoder that can accommodate by CPU spec.
In this paper, we report the execution time of DV decoder of libdv and video encoder of MPEG-4 in the reference software, and maintain the future work.

1　はじめに

ADSL や CATV など、近年の広帯域なネットワークの普及により、大容量の動画像コンテンツの配信が可能となっている[1]。また、今後 FTTH(Fiber To The Home)の普及が見込まれており、これによりユーザ同士で大容量の動画像を送受信することが可能になると予想される。
それにともない、広帯域ネットワークを通じて画像・音声を配信するサービスが数多く提案されている。また、広帯域を使用した高画質映像配信システムとして、Ruff Systems[2]が提案されている。Ruff Systems は、非圧縮の D1/HDTV 映像やビデオカメラから取得した DV ストリームを、TCP/IP 上で配信するシステムである。非圧縮画像や DV 画像を使用しているため、高画質の配信が可能だが、大きな帯域を必要とする。
我々はこれら個々のストリームを対象とした配信システムに対して、複数の画像や音声を入力とし、リアルタイムで編集・コード化して配信する作業を一体的に行うリアルタイムコンテンツ編集システムを提案する。本システムは、MPEG-4[3]のビデオオブジェクト（VO）

およびシーン記述言語の符号化および復号を分散環境において処理するもので、従来の配信システムに対して以下のような特長を有する。

① 離れた複数地点からのライブ映像を用いたコンテンツ編集作業が可能である

② 配信者側の編集方針に沿いつつ個々のユーザのニーズに沿ったコンテンツを提供できる

個々のユーザのニーズに沿ったコンテンツを提供するためには、多数の VO をユーザ側に送りコンテンツを構築するため、ユーザ側に多数の VO が集中する。そのため個々の VO に多くの帯域を割くことが困難となる。また、送信者が ADSL のような非対称な通信サービスを使用する場合、上り側の帯域幅は下り側のそれに比べて小さなものにとどまるため、やはり VO に多くの帯域を割くことが困難となる。そのため、高い圧縮率を持ち、さまざまな帯域に対応した MPEG-4 ビデオオブジェクト（MPEG-4 VO）等への符号化が必要になるが、MPEG-4 VO 符号化にはフレーム間圧縮に多大な CPU パワーを必要とするため、家庭用 PC などの低スペックなマシンを用いたライブ映像の複雑な符号化は困難となる。個人ユーザが自宅から、あるいは持ち運び可能な程度の機材を用いて出張先等から、ライブ映像を送信する等の場合、家庭用 PC 程度の機材しか使用できないことが多い。そこで、CPU パワーに応じた符号化処理を行うことで、リアルタイムな MPEG-4 VO への符号化を実現する。

本稿では、提案する実時間編集システムの概要を述べ、xdvshow[4]の DV デコーダおよび MPEG-4 参照ソフトウェア[5]のビデオエンコーダの処理時間測定結果を示し、実時間処理の可能性について議論する。

2 実時間符号化処理

ビデオカメラから取得した映像をその時点で配信するライブ配信を可能とするためには、符号化処理を実時間処理で行うことが求められる。実時間処理とは、外部からの連続的な入力に対し、定められた時間内に出力を返すことが要求されるシステムである。一秒間にｆフレームの動画を配信する実時間処理システムでは、１フレームに対する編集処理およびビットストリームのパケット化を平均１／ｆ秒以内に完了し、その場で配信しなければならない。

上記のような目標を達成するためには、以下のアプローチが有効である。

- 符号化処理を高速化し、制限時間以内に符号化処理を完了する。
- 性能の異なるマシンの上で正しく動作するために、プロセスの実行時間の監視を行い、監視の結果に基づいて複数の符号化アルゴリズムから適切なものを選択する。
- 符号化処理を段階的に行う。一定時間経過後に必要最小限の符号化を完了し、その後に解像度や圧縮率を高めた付加的な符号化処理を行う。制限時間がすぎた時点で付加的な符号化処理が完了しなかった場合、その時点で処理を打ち切り、直前の段階における符号化処理の結果を配信する。

3 実時間コンテンツ編集システム

1) 要求条件

実時間コンテンツ編集システムは、インターネット上に散在する、ライブ映像を含む多数の画像・音声を素材にした編集処理を行い、複数のユーザに対して配信するシステムである。本システムは、既存のさまざまな環境に柔軟に対応するため、各種回線・マシンによらず動作することを目指している。そのため、画像の符号化として、高い圧縮率を持ちさまざまな帯域に対応した MPEG-4 規格を用いる。さらに、複数の画像・音声の編集のために、シーン記述言語を使用する。

2) システム構成

図１に、構築するシステムの概念図を示す。本システムは、入力装置として複数のビデオカメラと、それらによる入力画像を符号化して配信する複数の配信モジュール、画像を受信して表示する受信モジュール、および一つの編集モジュールから構成される。

編集モジュールはオペレータからの指示に従い配信サーバに対する画像・音声のマルチキャスト要求を実際に行い、クライアントへ送信するシーン記述言語の生成およびマルチキャストを担当する。配信モジュールは、入力装置から DV 形式で取り込まれた画像・音声の MPEG-4 符号化を行い、複数のクライアントに対しインターネットを介して複数のマルチキャストグループによる階層化マルチキャストを行う。受信モジュールを持つクライアントでは、編集モジュールからのシ

ーン記述をもとに配信モジュールからの画像のレイアウトを決定し、表示する。ビデオカメラと編集装置は、IEEE1394 ポートを介して接続される。IEEE1394 ポートは、多くの DV 機器に備えられており、高速な上に一定周期でデータを送受信するアイソクロナス転送をサポートしているため、映像音声データの送受信に適している。

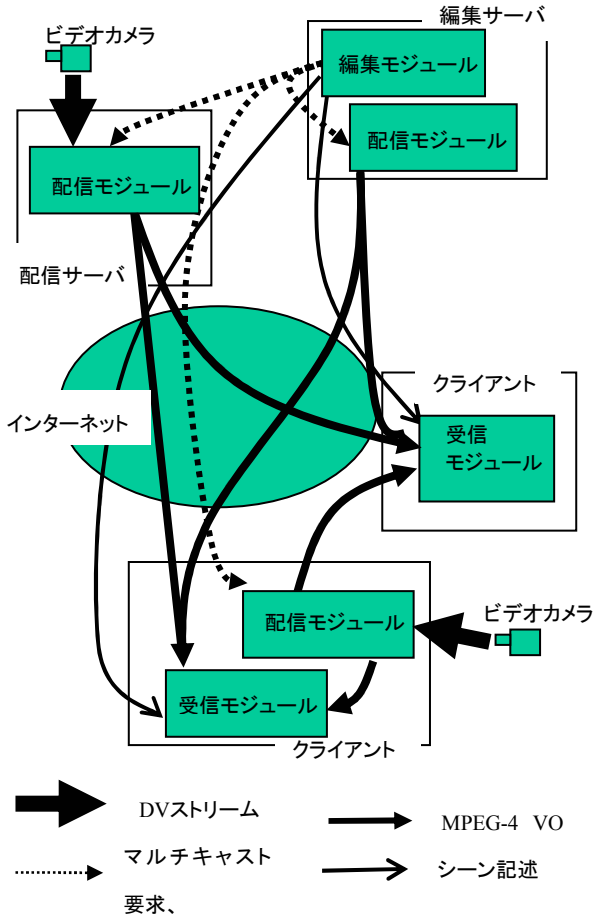

図１：リアルタイムコンテンツ編集システムの構成例

配信モジュール、受信モジュール、編集モジュールとしては任意の形態が考えられる。例えば、汎用のサーバマシン上のプログラムとして配置することもできるし、家庭用 PC 上で動作させても構わない。また、PC に対する組み込みシステムとして、専用の PC カードの形態を取ることも考えられる。

4　符号化時間計測

今回は、PC 上における DV コーデックおよびMPEG-4 ビジュアル符号化における問題点を抽出するため、既存のアプリケーションを用いた実行時間計測を行った。

1)　計測条件

DV のデコードには、DVTS に含まれる xdvshow を用いた。また、MPEG-4 ビデオ符号化には、MPEG-4 委員会作成の参照ソフトウェアの一つである MPEG-4 ビデオエンコーダを使用した。画像の大きさは 720×480 ピクセル、ハードは SONY のノート PC である PCG-GRX90/P(Pentium4 1.7GHz)を使用し 、OS は Vine Linux 2.5 を使用した。

2)　DV コーデック

4.2.1　アルゴリズム

図２に、xdvshow における DV データデコードの方法を示す。xdvshow は、DV コーデックのライブラリである libdv[6]をプログラム中で呼び出して DV ストリームのデコードを行っている。

DV フォーマットでは、DCT および可変長符号化を用いてフレーム内圧縮を行っている。720×480 ピクセルを持つ 525-60 システムにおける DV フォーマットの場合、１フレームが 1350 個のマクロブロックに分割される。１個のマクロブロックは 6 個の８×８ピクセル DCT ブロックにより構成される。うち、4 個は Y 成分、2 個はそれぞれ Cr 成分と Cb 成分を表している。符号化された DV データは、マクロブロックと同じ個数の DIF ブロックに区分される。5 個の DIF ブロックから 1 個のビデオセグメントが構成され、各ビデオセグメントを復号することにより 5 個のマクロブロックが生成される。

libdv では、各ビデオセグメントで可変長復号と各マクロブロックの復号を繰り返す。１フレームには 270 個のビデオセグメントが存在するため、「可変長復号」→「5 回のマクロブロック復号」のサイクルを合計 270 回繰り返すことにより、1 フレームが復号される。マクロブロック復号では、まず 6 個の DCT ブロックの逆量子化および逆 DCT が行われ、それにより生成した YCrCb 成分を RGB 成分に変換する。以上により、１フレームの復号には 270 回の可変長復号、8100 回の逆量子化および逆 DCT、1350 回の成分変換が行われることになる。

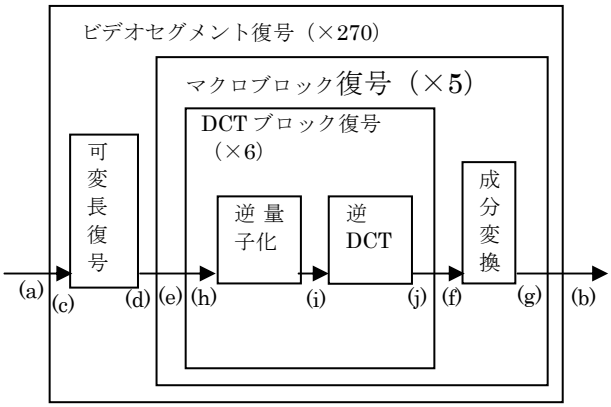

図２：DV デコーダの構成

4.2.2　測定方法

時間の計測には、gettimeofday システムコールを用いた。計測結果は毎フレームごとに集計して表示し、100 フレームの平均値を算出している。本方法では、計測自体のオーバーヘッドにより計測一回につき約１～２マイクロ秒程度の誤差が生じる。そこで、より正確な値を計測するため、計測点を限定した種別（１）から（３）の３種類の計測を行っている。

種別（１）は、フレームのデコード全体の実行時間のみを計測したもので、図２中における(a)-(b)間の実行時間を計測している。種別（２）は、全体の実行時間に加えて、可変長復号、およびマクロブロックの復号にあたる DCT ブロック復号と成分変換の実行時間を計測したもので、種別（１）に加えて(c)-(d)間、(e)-(f)間および(f)-(g)間を計測し、各区間について１フレームに要する実行時間の合計値を算出している。種別（３）は、種別（２）に加えて DCT ブロック復号内の逆量子化および逆 DCT の実行時間を計測したもので、二つ目に加えて(h)-(i)間および(i)-(j)間を計測し、１フレームの合計値を算出している。このような場合、(c)-(d)間で１フレームに 270 回、(e)-(f)間および(f)-(g)間で 1350 回、(h)-(i)間および(i)-(j)間で 8100 回の計測を行っているため、無視できないほどの誤差が生じる。測定に使用した画像を図３に示す。画像は 720×480 画素を持つ。測定用の画像では、木の葉の揺れによるフレーム間の若干の変化が起きているが、DV ストリームはフレーム間圧縮を行っていないため、実行結果に影響はない。

図３：DV デコーダの実行時間測定に使用した画像

4.2.3　実行時間

xdvshow における DV データ１フレームのデコードに要する時間を表に示す。

種別（１）の計測においてフレーム全体のデコードに要する時間は、およそ 38.4ms となっている。

これに対し、種別（２）の計測においてフレーム全体のデコードに要する時間は、およそ 47.4ms と増加している。種別（２）では、ビデオセグメントの復号中に１フレームにつき 270+1350+1350=2970 回の計測および結果の表示を行っているため、9ms 程度の誤差が生じる。可変長復号、DCT ブロックの復号および成分変換に、それぞれ約 17.3ms、9.4ms、11.6ms 程度の時間を要している。計測自体によって生じるオーバーヘッドのうち、実行時間の一部として実際に計測結果に参入される部分は１／２～1／３程度とそれほど大きくないため、誤差は可変長復号で 0.2～0.4ms、他の二つで 1~2ms 程度と予想される。

種別（３）の計測では、各 DCT ブロックの逆量子化および逆 DCT の実行時間の計測を行っている。それぞれの測定結果は 12.7ms、12.3ms 程度となっているが、うち半分近くは誤差と思われるため測定値そのものはほとんどあてにならない。ただし、双方とも同じ回数ずつの計測を行っていること、および種別（２）の結果から、DCT ブロックの復号自体に要する時間がほぼ 9ms～10ms と予想できるので、逆量子化および逆 DCT に要する時間は、双方とも 5ms 程度であると考えられる。なお、libdv version 0.98 はベクトル整数演算用命令セットを使用することによって、Pentium4 プロセッサにおいて逆 DCT を高速に実行

している。これらの命令セットを使用しなければ、実際にはより実行時間が長くなる。

表１：xdvshow における DV デコードに要する時間

種別	計測対象	実行時間 (ms)
（1）	フレーム全体	38.40
（2）	フレーム全体	47.40
	可変長復号	17.31
	DCT ブロック復号	9.41
	成分変換	11.59
（3）	フレーム全体	84.78
	可変長復号	19.08
	逆量子化	12.74
	逆 DCT	12.88
	成分変換	11.40

3)　MPEG-4 ビデオエンコーダ

4.3.1　符号化法

DV 符号化と MPEG-4 ビデオ符号化の大きな違いとして、フレーム間圧縮による符号化を行っている点が挙げられる。フレーム間圧縮のためには、前後の画像とのマッチングを行わなければならないため、一旦符号化した画像を再度復号して保持する必要がある。また、マッチング自体にも時間がかかるため、DV ストリームのデコードに比べて多くの計算量を要する。

VOP と呼ばれる MPEG-4 ビデオのフレームには、フレーム間圧縮を行わない I-VOP と、以前の VOP を利用したフレーム間圧縮を行う P-VOP、前後の VOP を利用した B-VOP の 3 種類に分類される。P-VOP や B-VOP は、I-VOP に比べてフレーム間のマッチングが必要なため、計算量が多くなる。また、I-VOP の符号化の際、他の P-VOP や B-VOP のフレーム間圧縮のために、一旦符号化した画像を再度復号する必要があるため、フレーム間圧縮を行わない符号化方式に比べて多くの計算量が必要となる。

MPEG-4 ビデオ符号化の過程では、DCT 変換の後に各 DCT 係数を量子化ステップ数で割ることによって符号量を圧縮することが可能である。その際、高い量子化ステップ数で量子化することにより圧縮率を高めることができるが、画質が劣化する。

4.3.2　測定方法

DV デコーダの場合と同様、gettimeofday システムコールを使用して計測部位の実行時間を測定した。今回は、90 フレームのエンコードを行い、1 VOP のエンコードにかかる実行時間の平均値を測定した。その際、I-VOP、P-VOP、B-VOP の 3 種類に分け、それぞれの平均値を出している。

今回の測定では、3 種類の符号化法を実行し、それぞれについて実行時間を測定している。一つ目は、全 VOP が I-VOP のみにより構成される符号化法である。このような方法を用いた場合、フレーム間圧縮を行わないため実行時間はそれほど大きくならない。二つ目は、I-VOP および P-VOP を含む符号化法である。エンコードを行う際、最初の VOP は I-VOP である必要があるため、今回は最初の VOP のみ I-VOP とし残りをすべて P-VOP とした。三つ目は、全種類の VOP を含む符号化法である。B-VOP は、通常二つの I-VOP または P-VOP の間に一定数が挟まる形で符号化される。今回は、二つの I-VOP または P-VOP の間に、二つの B-VOP が挟まる形にして符号化を行った。つまり、VOP の並びは、I-VOP のみを使用した場合が IIII.....、I-VOP および P-VOP を使用した場合が IPPPP.....、全種類を使用した場合が IBBPBBPB.... となる。

測定には、DV デコーダと同じ図３の画像を使用した

4.3.3　実行結果

MPEG-4 ビデオエンコーダにおいて、1 フレームの符号化にかかる平均実行時間を表２に示す。いずれの実験の場合にも、1 枚の VOP をエンコードするために、I-VOP では約 640～650ms、P-VOP では 900～1000ms、B-VOP では 2000ms 近く必要としている。なお、計測された符号化時間の他に、入力画像の読み込みや符号化後いったん復号した画像をディスクに書き戻す時間等が含まれるため、計測された時間よりも実際には 1 フレームあたり 100ms 程度多くの処理時間を必要とする。

これらの符号化時間は DV デコードの場合にくらべていずれも大きな値である。参照ソフトウェアとして作成された本ソフトウェアは、実時間で実行できることを意図していないため、エンコードに要する時間が大きくなる。例えば、MPEG-4 ビデオエンコーダでは１

個の DCT ブロックの逆 DCT に約 25 マイクロ秒の時間を要する。これを 1 フレームあたりに換算すると、約 200 ミリ秒かかる計算になる。同様のルーチンを、前述の libdv と同様の方法で行えば、所要時間は格段に少なくなるが、浮動小数点演算の代わりに整数演算を行っているため誤差が大きくなる。ライブ映像のエンコードへの可能性を検証するためには、まずはこれらの処理を効率化することから始める必要がある。

また、上記の実行時間のうち、動き補償に要する時間が、P-VOP でおよそ 300ms、B-VOP でおよそ 1300ms 程度となっているため、単純に実行時間を短縮する方法としては、P-VOP や B-VOP を使用しない方法は有力な候補の一つとなる。

表 2：MPEG-4 ビデオエンコーダにおける 1 フレームの実行時間

I-VOP のみ使用	I-VOP	642.5
I-VOP または P-VOP を使用	I-VOP	640.0
	P-VOP	900.5
全 VOP を使用	I-VOP	647.0
	P-VOP	1013.9
	B-VOP	1931.4

5. DV-MPEG 変換器

P-VOP や B-VOP を使用しない場合、フレーム間圧縮が行われないため、DV データから MPEG-4 VO への変換の流れを大幅に簡略化することができる。

MPEG-4 エンコーダのうち、フレーム間圧縮に関わる部分である動き補償が不必要になるため、直前にエンコードした画像が不必要になる。したがって、符号化後の画像に対する逆量子化や逆 DCT を省略することができる。

DV データから MPEG-4VO への変換は、通常はまず DV の復号を行い、次に MPEG-4VO への符号化を行う。その際に輝度データは DV データの逆 DCT を実行した直後に符号化のための DCT を行うため、逆 DCT→DCT の流れを省略することができる。

図 4 に、DV-MPEG 変換器の構成を示す。画素のうち、色差信号は 4:1:1 フォーマットから 4:2:0 フォーマットへの変換のための再構成が必要となる

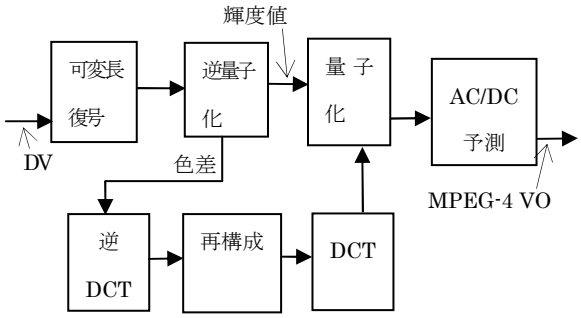

図 4：DV-MPEG 変換器

6. 今後の予定

本稿では提案する実時間編集システムの概要を述べ、xdvshow の DV デコーダおよび MPEG-4 参照ソフトウェアのビデオエンコーダの処理時間測定結果を示し、実時間処理の可能性について議論した。

今後、エンコーダの高速化を進める一方で P-VOP や B-VOP、形状情報も含めた MPEG-4 エンコーダの構築を行う。MPEG-4 エンコーダの全処理のうち、DCT および逆 DCT は、ベクトル演算用の命令セットを用いることで十分な高速化が可能である。他に、量子化や動き補償等に関しても順次高速化を行う。

また、実時間処理に対応した MPEG-4 エンコーダを構築するために、符号化処理を段階的に実行することを可能にする手法を検討する。現時点では、動き補償を階層化する方向での検討を行っている。

同時に、画質やビットレートに関する評価を行う。具体的には、P-VOP や B-VOP を使用した場合としない場合の違い、浮動小数点演算を使用した場合としない場合の違いなどについて議論する。

参考文献

[1] 「情報通信白書平成 14 年版」、総務省

[2] http://www.cnd.tel.co.jp/product/syo64.html

[3] ISO/IEC 14496, Final Draft International Standard MPEG-4, 1998

[4] http://www.sfc.wide.ad.jp/DVTS/

[5] ISO/IEC 14496-5, Final Draft International Standard MPEG-4: reference software, 1998

[6] http://libdv.sourceforge.net/