実時間コンテンツ編集システムの動画像符号化における VOP 選択アルゴリズムの提案

三 浦 康 之† 勝 本 道 哲†

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実時間コンテンツ編集システムを提案している．本システムは，編集・配信・受信の 3 種類のモジュールにより構成され，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．相互に使用する環境の異なる配信モジュールにおいて高品質なライブ映像を配信するためには，環境に応じて適切な符号化を行うことが必要となる．特に，フレーム間圧縮を行うか否かを適切な方法で選択することは，符号化に要する時間や符号化後の符号量・画質を決める重要な要素となる．本稿では，実時間コンテンツ編集システムの概要および配信モジュールにおける DV-MPEG 変換器について説明し，DV-MPEG 変換器において符号化の対象とする各フレームがフレーム間圧縮を行うか否かを決定するためのアルゴリズムを述べる．提案するアルゴリズムに基づいて実証実験を行った結果，提案手法は適切な方法により符号化を行い，符号量の削減・画質の向上に寄与することが明らかになった．

VOP Selection Algorithm in Video Encoding of Real-time Contents Edition System

Yasuyuki Miura† and Michiaki Katsumoto†

We propose a real-time contents edition system that allows the input of multiple audio-visual streams and simultaneous encoding into MPEG-4 video objects, editing, and delivery. This system is designed to enable contents provision based on live video from multiple locations. Our system consists of editing, streaming, and display modules. To distribute high quality live video movies from streaming modules which have different environment each other, it is important to encode adaptively in various environment. In particular, it is important to select whether the inter-frame compression is done or not by each frame to reduce the number of encoded bits or improve the quality of movie. In this paper, we give an overview of the real-time contents edition system and the structure of DV-MPEG transducer in streaming side of the system as detail. We also describe an algorithm to decide whether the inter-frame compression is done or not by each frame. Also we implement and experiment a DV-MPEG transducer to prove the availability of our algorithm. Our results showed that our algorithm can select appropriate method of encoding, and the algorithm can contribute to reduce the number of encoded bits or improve the quality of movie.

1. はじめに

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]．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やツールが数多く提案されている．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]，DVTS[3] などが存在する．また，MPEG 動画や H.323 規格を利用し狭帯域に対応したシステムやツールは多数存在する[4)~6)]．これらはいずれも PC 上からカメラ画像をインターネット経由で受信することが可能であるが，単一の動画を対象としたアプリケーションであり，多数の動画を扱うには不向きである．

我々が提案するリアルタイムコンテンツ編集システム[7] 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行うものである．本システムは，MPEG-4[8] のビデオオブジェクト（VO）および編集用言語の符号化および復号を実時間分散環境において処理するものである．本システムではユーザ側に多数の VO が集中するため個々の VO に多くの帯域を割くことが困難となる．ま

† 独立行政法人通信総合研究所
Communications Research Laboratory (CRL)

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 VO 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，時間的制約の厳しい実時間環境下でスペックの低い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．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などから，ライブ映像を送信するなどの場合，家庭用 PC 程度の機材しか使用できないことが多い．

他方，比較的高性能なサーバマシンを使用した場合，スペックの低いマシンを使用する場合に比べ，フレーム間比較を行うなどの複雑な符号化が可能になる．このように，使用するサーバの性能により可能な処理の種類が変わってくる．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，デジタルビデオカメラからの DV ストリームを MPEG-4 VO に変換する DV-MPEG 変換器における適切な符号化処理を選択する方法を提案する．本稿ではまた，PC 上で提案アルゴリズムの実証実験を行い，その有効性を示す．

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処理である．1 秒間に R 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 $1/R$ 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．

- 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．

本稿の実時間コンテンツ編集システムでは，多くのライブ映像が実時間環境下で配信される．実時間環境では時間的制約があるため，配信のための適切なアルゴリズムを選択することが重要となる．上記の考え方に基づいて，本稿の実時間コンテンツ編集システムでは，システム内で VOP の符号化時間を測定し，与えられたフレームレートに基づいてシステムがフレーム間符号化を行うか否か決定する．

3. MPEG-4 システムおよび MPEG-4 ビデオ

MPEG-4 はマルチメディアに対応した画像・音声の国際標準規格である．シーン全体をシーン記述によって表現し，シーンを構成するビデオやオーディオなどの各 AV（Audio Visual）オブジェクトの時間的・空間的関係を木構造で表現する．各 AV オブジェクトは独立して符号化され，それぞれに最適な手法を選択できる．またシーン記述を動的に変更することにより，AV オブジェクトごとの操作/加工もできる．

MPEG-4 ビデオ符号化においては，個々の動画を VO（Video Object）と呼んでいる．さらに VO を構成する各フレームを VOP（Video Object Plane）と呼ぶ．これがビデオ符号化の基本となる．VOP には予測符号化の違いにより，I-VOP（Intra VOP），P-VOP（Predicted VOP）および B-VOP（Bidirectional Interpolated VOP）が存在する．I-VOP はフレーム内のみで符号化を行う VOP，P-VOP は 1 つ前の I-VOP または P-VOP を利用してフレーム間予測を行う VOP，B-VOP は前後の I-VOP や P-VOP を利用してフレーム双方向予測を行う VOP である．なお，AV オブジェクトの機能を実現するために，ビデオ符号化では任意形状の画像符号化への対応もされている．

本稿のシステムでは，TCP/IP 分散環境のもとで MPEG-4 VO の符号化，復号および合成を可能とするため，多くのライブ映像が TCP/IP 上を VO として流れる．複数の VO およびシーン記述が分散環境上に配信され，各ポイントで合成される．

4. 実時間コンテンツ編集システム

4.1 システム仕様

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

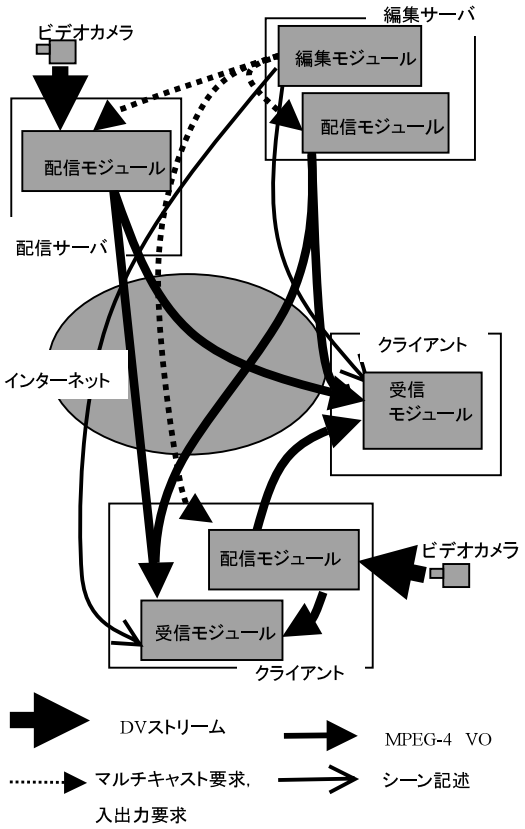

図 1 リアルタイムコンテンツ編集システムの構成例
Fig. 1 An example of real-time contents edition system.

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．また，多数の画像・音声を扱うため，システムの利便性を考え，システム稼動中に自由に送信者や受信者が参加・脱退が可能なものとすることが望ましい．

4.2 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 1 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モジュールに対する画像・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，複数の受信モジュールに対しインターネットを介してマルチキャストを行う．受信モジュールでは，複数の MPEG-4 画像音声データおよびシーン記述に基づいて動画像を表示する．ビデオカメラと各装置は，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，高速なうえ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．

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．たとえば，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 上で動作させてもかまわない．また，PC に対する組み込みシステムとして，専用の PC カードの形態をとることも考えられる．

5. 配信モジュールにおける DV-MPEG 変換器

5.1 要求条件

実時間処理の可能なコンテンツ編集システムでは，送信者側からデジタルビデオを使用して手持ちの PC を配信サーバとして画像を配信することも，職場の会議室などから高スペックのサーバマシンを使用して配信することも可能である．前者の場合，後者に比べて低スペックな PC を使用することが多く，複雑な符号化処理を行うのは困難である．後者の場合，より複雑な符号化処理が可能であるが，サーバマシンのスペックにより利用可能な符号化の種類が異なる．配信モジュールにおける MPEG-4 符号化には多大な処理を必要とするため，いかに適切なアルゴリズムを選択して必要な品質の動画を提供するかが重要となる．

人の顔画像のように動きの少ない動画像を送信する場合，比較的少ないフレームレートでもさほど問題にはならないが，スポーツ中継などのような動きの多い動画像を送信するような場合，動画質の維持の観点から一定以上のフレームレートを保つことが求められる．前者の場合，PC の処理能力が不足するような場合には各フレームの画質向上を優先してフレームそのものを間引くことが可能だが，後者のような場合は各フレームの符号化処理そのものを単純化することでフレームレートを維持する必要がある．そこで本システムでは R (flames/s) 以上のフレームレートを維持するという拘束条件が課せられるものとする．

I-VOP のみを使用して符号化を行う場合，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

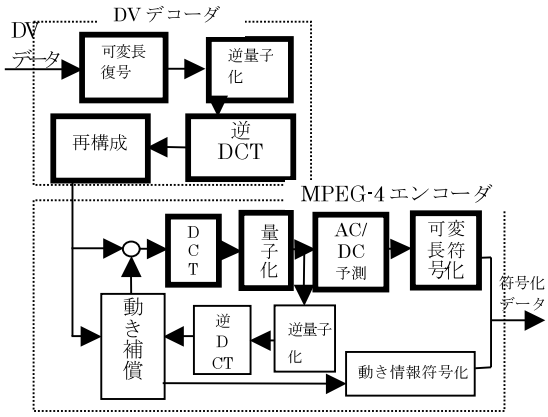

図 2 DV–MPEG 変換器の構成
Fig. 2 Construction of DV-MPEG transducer.

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ため，大幅に符号化時間を短縮することができる[7] が，符号化後の符号量が大きくなることが問題となる．MPEG-4 規格では量子化スケールを設定できるため符号量の調節が可能だが，量子化スケールを大きく設定するに従って画質が悪化するため，変換器を搭載するマシンの CPU パワーに余力がある場合にはフレーム間圧縮も行うことが望ましい．そのため，フレーム間圧縮の可能な VOP を含めた符号化の可能な変換器を構築する必要がある．

5.2 基本構成

I-VOP および P-VOP を用いたエンコードの可能な DV-MPEG 変換器の構成を**図 2** に示す．図中のすべての符号化処理要素は，基本処理と付加的処理に分類される．うち基本処理とは最低限必要な符号化処理要素，付加的処理は処理時間に余裕があるときにのみ行う処理要素である．図中，太枠で示された各処理は基本処理に該当する．これらは，全 VOP を I-VOP としたときに必要となる最小限度の処理である．対して，MPEG-4 エンコーダにおける「逆量子化」「逆 DCT」「動き補償」「動き情報符号化」は付加的処理と定義される．これらの処理は P-VOP および B-VOP によるフレーム間圧縮のために用いられる．

本稿の変換器は B-VOP を使用しない．後続のフレームに対するフレーム間圧縮のため，B-VOP を使用するとフレームの符号化の順序を変更する必要がある．変換器において B-VOP を使用することにより，フレームの順序を変える必要がある．このことは変換器におけるレイテンシを増大させる結果となる．

本稿で提案するアルゴリズムでは，各処理要素の処理時間を計測し，計測結果をもとに未来の符号化時間を予測し，最適な VOP の並びを決定する．類似の方法で MP3 プレイヤの復号方式などを決定する方法が提案されている[9] が，MPEG-4 ビデオ符号化においては複数の VOP にまたがる最適な符号法を決めるため，複数 VOP の合計符号化時間を求めたうえで，最適な方式を決定するという 2 段構えのアプローチが必要となる．

5.3 符号化処理時間の評価

各 VOP の符号化に必要な時間を評価する．評価に必要な各パラメータを以下のように定義する．

T_{min}：AC/DC 予測以外の基本処理の符号化に必要とされる時間

$T_{ACDC}(n)$：n 番目の VOP の AC/DC 予測に必要とされる時間

T_{IDCT}：MPEG-4 エンコーダの逆量子化および逆 DCT に必要な時間

$T_M(n)$：n 番目の VOP の動き補償および動き情報符号化に要する時間

上記のうち，$T_{ACDC}(n)$ と $T_M(n)$ を除く各数値は，途中で自由に処理を中断することができない．これに対し，P-VOP のフレーム間予測モードでは AC/DC 予測は用いられないため，AC/DC 予測に要する時間は VOP ごとに異なる．また，途中で動き補償の処理を中断しても中断結果を用いた符号化が可能なため，$T_M(n)$ は可変値をとることが可能である．具体的には，$T_M(n)$ は $0 \leq T_M(n) \leq T_{M-Pmax}$ のいずれかの値をとる．ここで，T_{M-Pmax} は P-VOP においてすべてのマクロブロックに対する動き補償を行う際に要する時間と定義される．上記の各パラメータは，図 2 に示される各処理要素の処理時間を計測することにより得られる．

ある VOP について，直後に I-VOP が来る場合，動き補償のための参照画像として使用することがないため逆量子化や逆 DCT を省略できる．したがって最短の時間で符号化することが可能である．直後に P-VOP が来る場合，動き補償のための参照画像を用意する必要があるため，逆量子化や逆 DCT が必要になる．したがって，直後に I-VOP が来るか P-VOP が来るかにより，VOP の符号化に必要な時間が異なる．以後の議論のため，前者を簡略化 I-VOP もしくは簡略化 P-VOP，後者を非簡略化 I-VOP もしくは非簡略化 P-VOP と呼ぶ．なお，これらは符号化の過程で参照画像を生成するか否かの違いにすぎないため，符号化後に両者が区別されることはない．

各 VOP の符号化時間は，以下のようになる．

- I-VOP の符号化
  - ・簡略化 I-VOP

$$T_{rI}(n) = T_{min} + T_{ACDC}(n) \quad (1)$$

・非簡略化 I-VOP

$$T_I(n) = T_{min} + T_{ACDC}(n) + T_{IDCT} \quad (2)$$

● P-VOP の符号化

・簡略化 P-VOP

$$T_{rP}(n) = T_{min} + T_{ACDC}(n) + T_M(n) \quad (3)$$

・非簡略化 P-VOP

$$T_P(n) = T_{min} + T_{ACDC}(n) + T_{IDCT} + T_M(n) \quad (4)$$

N 個の VOP 中に出現する簡略化 I-VOP，非簡略化 I-VOP，簡略化 P-VOP および非簡略化 P-VOP の数をそれぞれ N_{rIVOP}，N_{IVOP}，N_{rPVOP} および N_{PVOP} とおくと，N 個の VOP における符号化時間の合計は

$$\begin{aligned}
&T_{total} \\
&= \sum_{rIVOP} T_{rI}(n) + \sum_{IVOP} T_I(n) \\
&\quad + \sum_{rPVOP} T_{rP}(n) + \sum_{PVOP} T_P(n) \\
&= N_{rIVOP} T_{min} + \sum_{rIVOP} T_{ACDC}(n) \\
&\quad + N_{IVOP}\,(T_{min} + T_{IDCT}) + \sum_{IVOP} T_{ACDC}(n) \\
&\quad + N_{rPVOP} T_{min} + \sum_{rPVOP} (T_M(n) + T_{ACDC}(n)) \\
&\quad + N_{PVOP}\,(T_{min} + T_{IDCT}) \\
&\quad + \sum_{PVOP} (T_M(n) + T_{ACDC}(n)) \\
&= N T_{min} + (N_{IVOP} + N_{PVOP})\, T_{IDCT} \\
&\quad + \sum_{rPVOP} T_M(n) + \sum_{PVOP} T_M(n) \\
&\quad + \sum_{all-VOP} T_{ACDC}(n) \qquad (5)
\end{aligned}$$

となる．なおここで，$\sum_{rIVOP}$，$\sum_{IVOP}$，$\sum_{rPVOP}$，$\sum_{PVOP}$ および $\sum_{all-VOP}$ の各項目は，それぞれ簡略化 I-VOP，非簡略化 I-VOP，簡略化 P-VOP，非簡略化 P-VOP および全 VOP のすべての合計を意味する．

5.4 使用する VOP の決定

前節の結果から，符号化に使用する VOP の並びを決定する．解析の都合上，規則的な複数の VOP の並びを，VOP ブロックと定義し，VOP ブロックを組み合わせることによって VOP の並びを決定する．ここでは，以下の 3 種類の VOP ブロックを定義する．

IP ブロック：非簡略化 I-VOP の後に複数の非簡略化 P-VOP が並び，最後に簡略化 P-VOP が並ぶブロック（IPPP...Pp）

PI ブロック：非簡略化 I-VOP の後に一個の簡略化 P-VOP が並び，さらにその後に複数の簡略化 I-VOP が並ぶブロック（Ipii...）

I ブロック：簡略化 I-VOP 単独のブロック

なお，上記の（）内の記号は,「i」が簡略化 I-VOP,「I」が非簡略化 I-VOP,「p」が簡略化 P-VOP「P」が非簡略化 P-VOP をそれぞれ示している．

IP ブロックは P-VOP の数が多いブロックで，処理時間に余裕がある場合に用いられる．PI ブロックは，I-VOP の数が多く簡略化 I-VOP も含まれているため，IP ブロックに比べて処理時間の短縮が可能である．I ブロックは，処理時間の制約条件が非常に厳しく P-VOP がいっさい使用できない場合に用いられる．

式 (5) より，N 個の VOP 中に N_{IVOP} 個の非簡略化 I-VOP，N_{rPVOP} 個の簡略化 P-VOP および N_{PVOP} 個の非簡略化 P-VOP を使用して完全に動き補償を実行する際の合計符号化時間は

$$\begin{aligned}
T_{total} &= T_{min} N + T_{IDCT}\,(N_{IVOP} + N_{PVOP}) \\
&\quad + T_{M-Pmax}\,(N_{rPVOP} + N_{PVOP}) \quad (6)
\end{aligned}$$

となる．なお，P-VOP の ACDC 予測に要する時間の測定結果から他の VOP の ACDC 予測に要する時間を類推するのは困難であることと，T_{ACDC} が他のパラメータに比べて小さな値であることから，式 (6) 以降の VOP 決定アルゴリズムに関する議論では T_{ACDC} を省略し，基本処理の一部として取り扱う．

IP ブロックではすべての I-VOP が非簡略化 I-VOP となり，ブロック中最後の P-VOP のみが簡略化 P-VOP となるため，$N_{IVOP} = N_{rPVOP} = 1$，$N_{PVOP} = N - 2$ となる．したがって，式 (6) は

$$\begin{aligned}
&T_{total-IP} \\
&\quad = T_{min} N + (T_{IDCT} + T_{M-Pmax})\,(N-1) \quad (7)
\end{aligned}$$

となる．

PI ブロックでは，P-VOP の直前の VOP のみが非簡略化 I-VOP となり，残りの I-VOP は簡略化 I-VOP となる．また，ブロック内の唯一の P-VOP は簡略化 P-VOP となる．したがって，$N_{IVOP} = N_{rPVOP} = 1$，$N_{PVOP} = 0$ となる．したがって，式 (6) は

$$T_{total-PI} = T_{min} N + T_{IDCT} + T_{M-Pmax} \quad (8)$$

となる．

I ブロックは簡略化 I-VOP 単独のブロックなので，式 (6) は

$$T_{total-I} = T_{min} \quad (9)$$

となる．

ここで目標フレームレートを R とおき，N 個の VOP を含む 1 VOP ブロックの目標合計符号化時間を $N \times 1/R + T_{ad}$ とする．ここで T_{ad} を，時間補正値と定義する．この値は，直前までのブロックの目標符号化時間と実際の符号化時間の差に該当し，複数ブロックの平均符号化時間を目標値 $1/R$ に近い値になるように調整する目的で用いられる．N 個の VOP を持つ VOP を使用するための最低条件は，IP ブロックにお

いて

$$N/R + T_{ad} \geq T_{min}N + (T_{IDCT} + T_{M-Pmax})(N-1) \quad (10)$$

PI ブロックにおいて

$$N/R + T_{ad} \geq T_{min}N + T_{IDCT} + T_{M-Pmax} \quad (11)$$

I ブロックにおいて

$$N/R + T_{ad} \geq T_{min} \quad (12)$$

式 (10)〜式 (12) を変形して N についての条件を導出する．

IP ブロックにおいて

$$N \leq \frac{T_{IDCT} + T_{M-Pmax} + T_{ad}}{T_{min} + T_{IDCT} + T_{M-Pmax} - 1/R} \quad (13)$$

PI ブロックにおいて

$$N \geq \frac{T_{IDCT} + T_{M-Pmax} - T_{ad}}{1/R - T_{min}} \quad (14)$$

式 (13) により N の条件が求まるのは $T_{min}+T_{IDCT}+T_{M-Pmax} - 1/R > 0$ かつ $T_{IDCT} + T_{M-Pmax} + T_{ad} > 0$ の場合，式 (14) により求まるのは，$1/R - T_{min} > 0$ かつ $T_{M-Pmax} + T_{IDCT} - T_{ad} > 0$ の場合に限られる．これら個々の条件式が成り立たないケースは，それぞれ「すべての VOP を非簡略化 P-VOP と設定しても符号化時間に余裕があるケース」「IP ブロックでは目標符号化時間を達成できないケース」および「全 VOP を簡略化 I-VOP として設定しても符号化時間が不足するケース」「PI ブロックでは目標符号化時間を達成できないケース」に該当する．

式 (13)，(14) に基づいて使用する VOP ブロックの種類およびその長さを決定する．決定の流れは以下のとおりである．

① 配信モジュール起動時に IPi の順に入力画像の符号化を行って各 VOP について処理時間計測を行う．

② ① により測定された処理時間の値を用いて，VOP 判定アルゴリズムに従って，使用する VOP ブロックの種類および長さを決定する．

③ 決定された VOP ブロックの種類・長さに従って入力画像の符号化を行うと同時に，各 VOP について処理時間計測を行う．

④ ③ における VOP ブロック全体の符号化の終了後，② に戻り後続の VOP ブロックの種類および長さを決定する．

VOP ブロック決定アルゴリズムを**図 3** に示す．アルゴリズムの入力 $\boldsymbol{T}$ および R は，それぞれ各符号化処理の処理時間および目標フレームレートである．うち $\boldsymbol{T}$ は，すべての符号化処理時間を含み，$\boldsymbol{T} = \{T_{min}, T_{IDCT}, T_{M-Pmax}, T_{ad}\}$ で示される．これらの値のうち，時間補正値 T_{ad} は，直前のブロックまでの目標合計符号化時間（$1/R$×『直前のブロックまでの合計フレーム数』で示される）と，実際に測定された合計符号化時間の差として定義される．このような時間補正値を用いて直前のブロックまでの符号化時間に関する情報を利用することにより，平均符号化時間をより目標値に近づけることが可能となる．

```
VOP_Decision (T, R){

  nip= ⌊(TIDCT+TM-Pmax +Tad)/(Tmin+TIDCT+TM-Pmax-1/R)⌋
  npi=⌈(TIDCT + TM-Pmax—Tad) /( 1/R-Tmin)⌉

  if(1/R-Tmin <=0){          TYPE=I_block; len=1;}
  elseif(npi>npiMAX){        TYPE=PI_ block; len=npiMAX;}
  elseif(Tmin +TIDCT +TM-Pmax-1/R <=0){
                             TYPE=IP_ block; len=nipMAX;}
  elseif(nip>nipMAX){        TYPE=IP_ block; len=nipMAX;}
  elseif(npi>=3){            TYPE=PI_ block; len=npi;}
  elseif(nip>=3){            TYPE=IP_ block; len=nip;}
  elseif((nip=2) or(nip=2)) { TYPE=IP_ block; len=2;}

  return(TYPE,len);
}
```

図 3 VOP ブロック決定アルゴリズム
Fig. 3 A VOP decision-making algorithm.

アルゴリズム中では，式 (12)，(13) を満たす最も近い整数をそれぞれ仮の VOP ブロック長 N_{IP}，N_{PI} とし，求められた仮の値をもとにいくつかのケースに分けて VOP ブロックの種類および長さを決定する．決定の方針は以下のとおりである．

1) $1/R - T_{min} \leq 0$ となるケースでは符号化時間に余裕がないため，I ブロックを使用する．

2) N_{PI} が大きな値を持つ場合，VOP ブロックが長すぎて新たに VOP ブロックを決め直す機会が減少する．そこで VOP ブロック長の最大値をあらかじめ決めておき，計算結果が最大値を超える場合，あらかじめ決められた最大値を VOP ブロック長とした PI ブロックを使用する．

3) $T_{min} + T_{IDCT} + T_{M-Pmax} - 1/R \leq 0$ となるケースではすべての VOP を P-VOP と設定しても符号化時間に余裕があるため，2) のケースにならってあらかじめ決められた最大値を VOP ブロック長とし，IP ブロックを使用する．

4) N_{IP} が大きな値を持つ場合，2) のケースにならっ

てあらかじめ決められた最大値を VOP ブロック長とし，IP ブロックを使用する．

5) N_{PI} が 3 以上の値を持つ場合，ブロック長を nPI とした PI ブロックの平均符号化時間が目標平均符号化時間に近い値を持つことを示している．したがって，N_{PI} を VOP ブロック長とした PI ブロックを使用する．

6) N_{IP} が 3 以上の値を持つ場合，5) のケースと同様に N_{IP} を VOP ブロック長とした IP ブロックを使用する．

7) N_{PI} もしくは N_{IP} のいずれかが 2 の値を持つケースでは，非簡略化 I-VOP と P-VOP が 1 つずつ並ぶ VOP ブロックの平均符号化時間が目標平均符号化時間に近い値を持つことを示している．したがって，VOP ブロック長を 2 とした IP ブロックを使用する．

上記 1)〜7) のいずれにも該当しないケースは，N_{PI} と N_{IP} の双方が 1 以下の値を持つケースであるが，$N = 2$ で式 (13) と式 (14) は等価になるため，このようなケースは理論的には存在しない．

6. 動作実験

6.1 実験環境

前章に示したアルゴリズムを検証するため，MPEG-4 参照ソフトウェア[10] を用いて DV-MPEG 変換器を構築し，ライブ映像の変換を行った．実験用の変換器は，xdvshow[3] および MPEG-4 参照ソフトウェアを改造したもので，目標平均符号化時間 $1/R$ の条件下において以下のような動作を実行している．

① IEEE1394 ポートから DV データを取り込む．
② 取り込まれた DV データを YUV 形式の非圧縮データに変換する．
③ デコードされた YUV 形式のデータを MPEG-4 ビデオ形式に符号化する．
④ 符号化された MPEG-4 データをファイルに書き込む．

うち ② と ③ は，変換器の処理に該当する．① における入力画像のフレームレートは R frame/s である．画像を入力する際，R frame/s のフレームレートで入力し，入力画像をいったんキューに保存している．② 以降は，キューに保存された入力画像を 1 つずつ取り出して順次処理を行っている．使用した PC は VAIO ノート「PCG-VX7/BD」である．850 MHz の Pentium III プロセッサおよび 4 ピンの IEEE1394 ポートを 1 つ搭載したもので，ノート型 PC としては標準的なものである．本実験では，実行時間の短縮のため，DCT および IDCT の処理に Pentium シリーズのプロセッサに実装されている MMX 命令セットのベクトル演算命令を使用している．

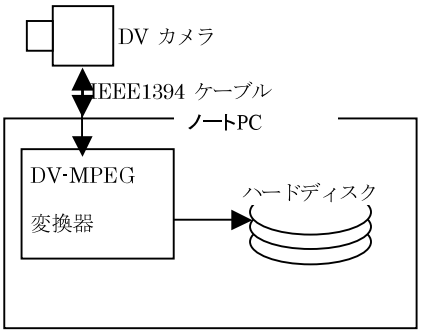

図 4 実験装置の構成
Fig. 4 Experimental environment.

実験環境を**図 4** に示す．実験では，IEEE1394 ケーブルを通してノート PC に DV カメラを接続し，DV カメラから取得した DV データを PC 内部で MPEG-4 ビデオ形式に符号化して，結果をファイルに書き込む．

本実験で使用する変換器は I-VOP および P-VOP の符号化が可能である．これらの VOP の並びは，5.4 節に述べられるアルゴリズムを使用している．ただし，ループ中に存在する処理要素の処理時間の計測が実際には困難である[7] ことから，処理時間の計測は以下に述べる方法を用いており，必要に応じて式の変更を行っている．

① 各 VOP について,「DV データの復号 →MPEG-4 VO への符号化 → ファイルへの書き込み」に要する実行時間全体を計測して記録する．
② VOP の種別ごとに，過去 5 回の実行時間の平均値を求め，それぞれ T_{ave-rI}，T_{ave-I}，T_{ave-P}，とする．過去の符号化処理回数が 5 回に満たない VOP については，過去すべての実行時間の平均値を求める．
③ 式 (1)〜(4) に基づいて各符号化処理要素の符号化時間を求め，式 (13)，(14) を変更する．

③ の具体的な手順は，下式のとおりである．

$$T_{min} = T_{ave-rI} \tag{15}$$

$$T_{IDCT} = T_{ave-I} - T_{ave-rI} \tag{16}$$

$$T_{M-Pmax} = T_{ave-P} - T_{ave-I} \tag{17}$$

これらを式 (13)，(14) に代入すると，それぞれ式 (18)，(19) のようになる．

$$N \leq \frac{T_{ave-P} - T_{ave-rI} + T_{ad}}{T_{ave-P} - 1/R} \tag{18}$$

$$N \geq \frac{T_{ave-P} - T_{ave-rI} - T_{ad}}{1/R - T_{ave-rI}} \tag{19}$$

本実験では，式 (13)，(14) の代わりに式 (18)，(19)

を用いた実験を行っている．この方法を用いた場合，細かく処理時間を計測する必要がないため時間計測によるオーバヘッドがほとんど存在しなくなる．その一方で，各 VOP の処理時間計測にタイムラグがあるため，OS 中で他のプロセスを実行中の場合のように処理時間そのもののばらつきが大きくなる場合，$T_{ave-P} > T_{ave-I} > T_{ave-rI}$ の順序関係が崩れる可能性がある．その場合，式 (18)，(19) そのものが成り立たなくなるおそれがあるため改めて各 VOP の実行時間を計測しなおす必要があるが，実際には過去 5 回の実行時間の平均値を求めることによりばらつきが平均化されるため，順序関係が崩れることはほとんどない．

本実験では，時間補正値 T_{ad} を用いた補正を行わない場合と補正を行う場合の 2 通りについての実験を行っている．補正を用いない場合，$T_{ad} = 0$ としている．補正を行う場合，$T_{ad} = 1/R \times$ (符号化開始時点からの合計フレーム数)−(符号化開始時点からの合計実行時間) と定義している．前者は，個々の VOP ブロック内において符号化時間を目標値に近づけるアルゴリズムであることに対して，後者は，直前までのブロックの符号化において余った（あるいは不足した）時間を持ち越すことにより，より目標符号化時間に近い時間による符号化を行うことを目指している．なお，本実験では VOP ブロック長の最大値を 10 としている．

エンコードに使用した画像を**図 5** に示す．図 5 (a) は，ビデオカメラのキャップを外さずに撮影した映像で，画面全体が黒一色で占められている．図 5 (b) は，カメラを移動させつつ建物を撮影した映像で，画面全体が上下左右に流れている．図 5 (c) は，カメラを固定して右側の列車が画面後方に進行する様子を撮影した映像である．

本実験では，各画像についてそれぞれ 100 フレームずつの符号化を行っている．入力画像のサイズは 720×480 ピクセルで，符号化の際に 368×240 ピクセルに縮小している．これは，入力画像サイズを縦横ともに半分の大きさにした画像であるが，画面の大きさをマクロブロックの大きさの整数倍にあわせるために画面の右端を黒色のピクセルで埋めている．目標平均符号化時間 $1/R$ は，100〜350 ms の可変値としている．1 ピクセルのサイズは，YUV いずれも 8 ビット，量子化タイプは H.263 を使用している．また，量子化スケールは 16 としている．

6.2　実験結果および考察

各画像における各 VOP の平均符号化時間を**表 1** に示す．表 1 から，画像 1 に要する I-VOP の符号化時間は他の画像より若干短くなっていることが分かる．これは，参照ソフトウェア内で画像 1 の可変長符号化

(a) 画像 1

(b) 画像 2

(c) 画像 3

図 5　実験画像
Fig. 5　Experimental movies.

表 1　各 VOP の平均符号化時間
Table 1　An average encoding time of VOPs.

画像	簡略化 I-VOP	非簡略化 I-VOP	簡略化 P-VOP	非簡略化 P-VOP
画像 1	79.2 ms	153.3 ms	261.9 ms	322.8 ms
画像 2	110.2 ms	190.3 ms	227.8 ms	299.5 ms
画像 3	105.8 ms	178.6 ms	269.4 ms	340.3 ms

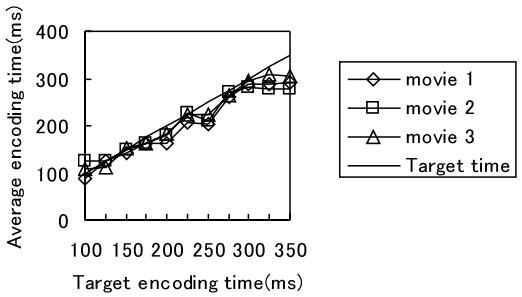

図 6 各画像の平均符号化時間（時間補正なし）
Fig. 6 Average encoding time (without adjustment).

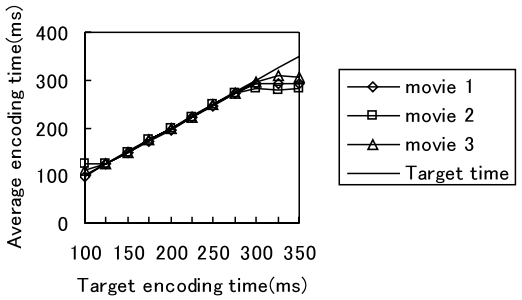

図 7 各画像の平均符号化時間（時間補正あり）
Fig. 7 Average encoding time (with adjustment).

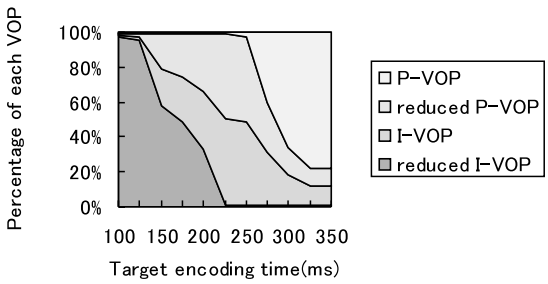

図 8 各 VOP の比率（画像 3，時間補正なし）
Fig. 8 Percentage of VOPs (Movie 3, without adjustment).

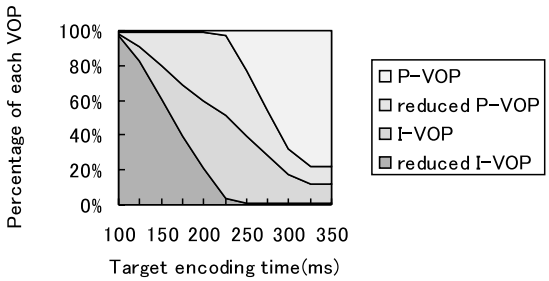

図 9 各 VOP の比率（画像 3，時間補正あり）
Fig. 9 Percentage of VOPs (Movie 3, with adjustment).

を短縮しているのが原因と考えられる．実験に用いた参照ソフトウェアでは，マクロブロック内の全ピクセルが黒一色のようなケースでは可変長符号化に要する時間が短くなる傾向がある．

各画像における，全 VOP の平均符号化時間を**図 6**，**図 7** に示す．ここで，平均符号化時間とは 1VOP あたりの「DV データの復号 → MPEG-4 VO への符号化 → ファイルへの書き込み」に要する実行時間の平均値を意味する．図 6 は時間補正を行わない場合，図 7 は時間補正を行った場合の結果である．すべてのケースにおいて，ほぼ目標符号化時間に近い値を示しているものの，時間補正なしの場合，200 ms 近辺と 250 ms 近辺で目標時間との誤差が大きくなっている．この近辺では，比較的長さの短い VOP ブロックを使用することが多いため，個々の VOP ブロックのみで目標平均符号化時間に近い時間で符号化処理を行うには限界がある．

このような問題は，図 7 に示されるように時間補正を行うことにより解決する．また，図 6，図 7 のいずれのケースでも 300 ms 以降で平均符号化時間に変化が見られなくなる．これは，P-VOP の符号化時間自体が 300 ms よりも短いため，大部分のフレームにおいて P-VOP を選択しても平均符号化時間が 300 ms を超えることがないためである．

各画像において各々の VOP が使用された割合を**図 8**，**図 9** に示す．図 8 は時間補正を行わない場合，図 9 は時間補正を行った場合の結果である．これらの図に示されるように，目標符号化時間が長くなるに従って符号化に多くの時間を要する P-VOP の数が増えてゆく．このように，提案手法では使用する VOP の数を調節しながらほぼ目標時間に近い値を得ている．時間補正を行わないケースでは，行うケースに比べて 200 ms や 250 ms の近辺で P-VOP の比率が低めになっている．そのため，目標符号化時間との間にずれが生じている．

時間補正を行った場合の各画像における平均符号量を**図 10** に示す．ここで，平均符号量とは 1VOP あたりの符号ビット数の平均値を意味する．画像 1 および画像 3 では，P-VOP が増加するに従い大きく符号量が減少している一方，画像 2 では目だった符号量の増減は見られない．このことから，画像 1 や画像 3 では P-VOP による大幅な符号量削減効果が得られることが分かる．画像 2 において符号量の変化が少ないのは，画面が移動しているため動き情報を利用しにくいためである．比較的フレームレートが高い（目標平均符号化時間が短い）条件では連続するフレーム間の一致度が比較的高いため，P-VOP の利用による若干の符号量削減効果が得られるが，フレームレートが下が

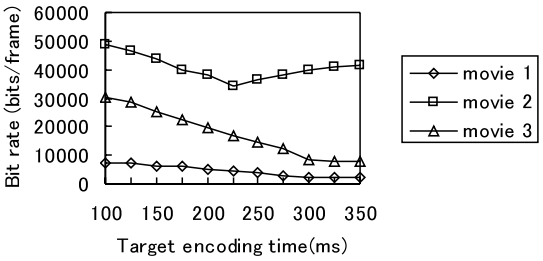

図 10　平均符号量（時間補正あり）
Fig. 10　Bit rate (with adjustment).

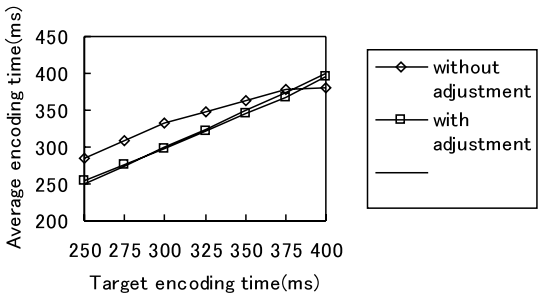

図 11　平均符号化時間（画像 3，他のプロセスが同時に稼動中）
Fig. 11　Average encoding time (Movie 3, with another process).

るに従って符号量削減効果が限定的になる．そのため，本実験では 200 ms 近辺で若干符号量が減っているが，それ以降は逆に符号量が増加している．

実験中に同一の PC 内で他のプロセスが稼動している状態で同様の実験を行った．プロセスは，for ループを繰り返すだけの単純なものである．他のプロセスが同時に稼動することにより，各 VOP の符号化時間が長くなるほか，各々の符号化時間のばらつきが大きくなる．他のプロセスが同時に稼動中の状態における画像 3 に対する全 VOP の平均符号化時間を**図 11** に示す．図に示すようにほぼ目標値に近い値が得られている．また，他のプロセスが同時に稼動している状況でもやはり時間補正を行うことによる平均符号化時間の改善が見られる．時間補正を行った場合の画像 3 における平均符号量を**図 12** に示す．図 12 では，他のプロセスが同時に稼動している状況と稼動していない状況での比較を行っている．図に示されるように，他のプロセスが同時に稼動している状況では P-VOP を使用する機会が少なくなるため，同じ目標符号化時間でも符号量が多くなる傾向がある．

同程度の符号量に対して P-VOP が増加することによる効果を確認するため，MPEG-4 参照ソフトウェアのレートコントロール機能を用いて比較を行った．参照ソフトウェアのレートコントロール機能は，量子化

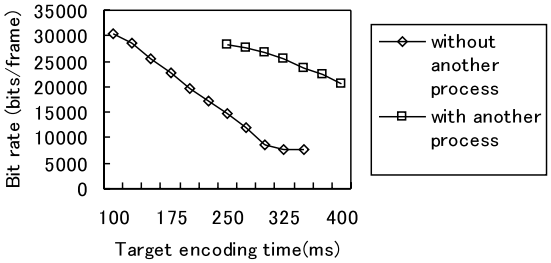

図 12　平均符号量（画像 3，時間補正あり）
Fig. 12　Average encoding time (Movie 3, with adjustment).

(a) I-VOP のみを用いた場合

(b) VOP 選択アルゴリズムを用いた場合

図 13　レートコントロールを用いた符号化後の画像
Fig. 13　Encoded movies with rate control.

スケールを変更して目標平均符号量を達成するための機能である．本実験では，参照ソフトウェア中の VM4 レートコントロール機能を用いて，30,000 bits/frame を目標フレームレートとした実験を行った．目標平均符号化時間を 350 ms とし，I-VOP のみで符号化を行った場合と提案手法を用いて P-VOP を併用した場合の比較を行った．符号化後の画像を**図 13** に示す．図 13 (a) は I-VOP のみで符号化を行った場合，図 13 (b) は提案手法を用いた場合である．前者の場合に比べて後者の場合の画質がはるかに良いものになっ

ていることが分かる．P-VOP を併用してフレーム間圧縮を行うことにより，個々の VOP に対する量子化スケールによって削減するビット数を抑えることができるため，提案手法では同程度の符号量に対する画質が向上する．

7. まとめ

本稿では，実時間コンテンツ編集システムの概要を述べ，うち配信モジュールにおける DV-MPEG 変換器の構成を詳細に示した．P-VOP と B-VOP の双方が使用されない場合，フレーム間圧縮を必要としないため符号化時間を短縮することができる一方，P-VOP のフレーム間圧縮機能を用いることで符号量の削減が可能なため，フレームレートやマシンスペックに合わせた適切な VOP の選択が必要になる．そこで本稿では，DV-MPEG 変換器における適切な VOP 選択アルゴリズムを提案し，提案手法の評価を行った．提案手法は，各 VOP の符号化時間計測の結果を用いて後続の VOP の符号化時間を予測して適切な VOP を選択する手法で，実験では複数の VOP をまとめた VOP ブロックごとに予測を行い，時間補正を行わない手法と，時間補正により，より目標時間に近づく VOP 選択手法について実験を行った．

実験の結果，双方とも目標符号化時間に近い結果を得られることが明らかになった．特に，時間補正を行うことにより，きわめて時間的誤差の少ない VOP 選択が可能となることが明らかになった．また，提案手法とビットレートの調整手法を併用することにより，より質の高い画像を配信することができることを示した．

参考文献

1) 情報通信白書平成 15 年版，総務省．
2) http://www.cnd.tel.co.jp/product/syo64.html
3) http:// www.sfc.wide.ad.jp/DVTS
4) ohphone. http://www.openh323.org/docs/ohphone_man.html
5) FFmpeg. http://ffmpeg.sourceforge.net/
6) Coriander. http://www.tele.ucl.ac.be/PEOPLE/DOUXCHAMPS/ieee1394/coriander/
7) 三浦康之，勝本道哲：実時間環境におけるビジュアル符号化のための実証実験，情報処理学会研究報告，2002-DPS-100, pp.25–30 (2002).
8) ISO/IEC 14496, Final Draft International Standard MPEG-4 (1998).
9) 金光恭治，本田晋也，高田広章：リアルタイム OS 上での QoS 制御手法の検討，情報処理学会研究報告，2003-SLDM-109, pp.35-40(2003).
10) ISO/IEC 14496-5, Final Draft International Standard MPEG-4: Reference Software (1998).

(平成 15 年 5 月 9 日受付)
(平成 15 年 12 月 2 日採録)

三浦 康之（正会員）

昭和 48 年生．平成 9 年東北大学工学部卒業．平成 14 年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了．この間，並列計算機の相互結合網に関する研究に従事．平成 14 年より独立行政法人通信総合研究所専攻研究員．現在，実時間処理の可能なコンテンツ編集システムの研究を行う．博士（情報科学）．

勝本 道哲（正会員）

平成 9 年東洋大学大学院博士課程修了．同年郵政省通信総合研究所（現・独立行政法人通信総合研究所）入所．平成 9〜11 年まで感性情報処理を用いたインターネットによる情報検索手法の研究開発に従事．平成 11〜13 年まで次世代インターネット技術である超高品質コンテンツ配信技術の研究開発に従事．平成 14 年よりインターネットアプリケーション技術の研究開発に従事する．現在，インターネットアプリケーショングループリーダー．情報処理学会 DPS 研究会運営委員．電子信学会 IA 研究会専門委員．平成 11 年度情報処理学会山下記念賞受賞．IEEE Computer Society，ACM 各会員．工学博士．