

低ビットレート用 MPEG-2 リアルタイムソフトウェアトランスコーダ「Trampeg」の開発

阿久津隆史[†] 木全 英明[†] 清水 淳^{††} 八島 由幸[†]
小林 直樹[†]

“Trampeg”——MPEG-2 Realtime Software Transcoder for
Low Bitrate Transmission

Takashi AKUTSU[†], Hideaki KIMATA[†], Atsushi SHIMIZU^{††}, Yoshiyuki YASHIMA[†],
and Naoki KOBAYASHI[†]

あらまし MPEG-2 で符号化されたコンテンツをインターネットや LAN 等の種々のネットワークで送信して視聴可能とするには、符号化レートを変換する、いわゆるトランスコーディングが必要となる。最も単純なトランスコーディングは、MPEG-2 を完全に復号して再度低ビットレートで符号化する方法であるが、符号化を行う際の演算量が大きく、汎用の PC 上でのリアルタイム処理は困難である。しかし、変換前の符号化情報を利用して演算量を抑えられれば、ソフトウェアでのリアルタイム変換が可能になる。今回 MPEG-2 を高品質にリアルタイム変換可能なソフトウェアトランスコーダ Trampeg (Transcoder for MPEG-2) を開発した。本論文では Trampeg の適用形態・設計指針及び処理アルゴリズムを示し、実際の PC 上で動作させた場合の評価結果から、ネットワーク経由で入力された MPEG-2 システムストリームをリアルタイム変換し、変換された低レートストリームを直接配信サーバに送出可能なことを確認した。

キーワード トランスコーダ, MPEG-2, リアルタイム変換, 動きベクトル, 画像符号化

1. ま え が き

デジタル放送の開始や DVD の普及などを機に、今後は大量のコンテンツが MPEG-2 で 1 元管理されることが想定される。MPEG-2 で管理されるコンテンツを種々のネットワークを用いて送信する場合には、符号化レートを変換するいわゆるトランスコーディングが必要となる。特にインターネットや LAN 等で視聴する場合、相応の低いビットレートへの変換が必須である。

トランスコーディングの方法として最も基本的な方法は、いったんすべてのフレームを画素領域に完全に復号し、復号された画像を必要に応じてサイズ縮小し、再度低ビットレートで符号化を行う方法である。しか

し、この方法では符号化を行う際の演算量が大きくなり、リアルタイム処理は困難である。リアルタイム処理を可能とするため、変換前の MPEG-2 ストリームに含まれる動きベクトル情報などを利用して、変換に要する演算量を抑える方法が提案されている [1]~[3]。今回、汎用の PC を用いて、MPEG-2 システムストリームを 20~512 kbit/s の低ビットレートストリームへ高品質にリアルタイム変換可能なソフトウェアトランスコーダ Trampeg (Transcoder for MPEG-2) を開発した [4], [5]。

本論文では、まず 2. でトランスコーダの適用形態と要求条件を明確にし、要求条件を満たすためのトランスコーディングアルゴリズムの設計指針を述べる。次に 3. では、MPEG-2 ストリームに含まれる動きベクトル情報を使って、低レート用の動きベクトルを推定するアルゴリズムを述べる。動きベクトル推定は Trampeg において演算量削減に最も大きな役割を果たす。続いて 4. では、実際の PC 上で画質・変換速度の両面から動ベクトル変換アルゴリズムの性能評価を

[†] NTT サイバースペース研究所, 横須賀市

NTT Cyber Space Laboratories, 1-1 Hikarino-oka, Yokosuka-shi, 239-0847 Japan

^{††} NTT 東日本, 東京都

NTT East Corporation, 4-5-6 Takebashi, Chiyoda-ku, Tokyo, 565-0456 Japan

行い、オーディオ部分も含めたシステム全体の評価結果を 5. で述べる。

2. システム要求条件とソフトウェアの基本構成

2.1 システム要求条件

想定する Trampeg の適用形態を図 1 に示す。大容量の専用線を利用するクライアントは、ハードウェア等で符号化された高品質の MPEG-2 コンテンツのストリームまたは蓄積ファイルを直接受信して視聴することができる。このとき、インターネット等でより低速な回線を利用するクライアントに対して、大容量の専用線上を送信される高品質の MPEG-2 コンテンツを一度ネットワーク上から Trampeg に入力し、リアルタイムに低ビットレートストリームに変換してから配信できれば、変換前の MPEG-2 コンテンツと同じ内容を低速回線上で同時に視聴可能となる。このためのシステム要求条件としては、以下の項目が必要である。

i) 標準テレビジョン信号に対する MPEG-2 ストリーム (SP@ML 及び MP@ML) では符号化ビットレートは規格により 15 Mbit/s 以下であり、また文献 [6] などから 4 Mbit/s で現行テレビジョン信号と同程度の画質が得られることが指摘されている。このことから入力ビットレートは 4~15 Mbit/s をターゲットとする。また変換後のビットレートについては、低ビットレートネットワークとして使用可能な、モデムユーザ (28.8 kbit/s), デジタル回線としては最も広く一般に普及している ISDN (64 kbit/s, 128 kbit/s), 及び DSL (384~512 kbit/s) を想定し、20~512 kbit/s に変換可能なこととする。

ii) 一定のビットレートのもとでは、画質 (画面サイズを含む) とフレームレートにはトレードオフの関係

があり、コンテンツによって最適値が存在する [7], [8]。このことから、トランスコードの際にユーザがフレーム数を重視するか、あるいは画質を重視するかを選択できる構成とする。画面サイズに関しては MPEG-2 からの変換の容易性を考慮し、H.263 や MPEG-4 のインターネット上の配信でよく用いられる SIF (352×240, 標準 TV の縦横 1/2) と QSIF (176×120, 縦横 1/4) を対象とする。ビットレートに関しては i) に示した ISDN の帯域である 100~120 kbit/s 程度、フレーム数に関しては文献 [8] から 5 frame/s を下回るとジャークネスが気になると指摘されているため、ソフト設計にあたっては 5~6 frame/s 以上を実現することを目標とする。

iii) i) と ii) の変換に加えて、変換後の低レートストリームの配信サーバへの送出 (多重化やパケット化) まで含めて、汎用の PC 上 (Pentium II, 500 MHz × 2 CPU 程度) でソフトウェアのみによるリアルタイム処理が可能であることとする。

iv) MPEG-2 ストリームをファイルから入力するだけでなく、ネットワーク経由での入力にも対応可能であることとする。

以上のうち、i) と ii) は入出力ストリーム自体に関する要求条件であるのに対して、iii) は出力低ビットレートストリームに関するソフトウェアのインタフェースに関する条件、iv) は MPEG-2 入力ストリームに関するソフトウェアのインタフェースに関する条件で、iii) と iv) はいずれも機能的な要求条件である。iii) と iv) により MPEG-2 ストリームを Trampeg 変換しながら配信サーバを用いてネットワーク配信することが可能となる。また、iii) に関して、ここでは一般に容易に入手可能な 2 CPU のハイエンドマシンを想定したが、二つの CPU へのリソース配分は後でも述べるように OS の管理下で最適に行われること、ソフトウェア構築に関しては PC の汎用性が考慮されるべきであることを考え、2 CPU 特有の処理は特に施さない。

次節ではこれらの要求条件に沿ってトランスコーダのアルゴリズムの基本構成を考える。

2.2 ソフトウェア基本構成

図 2 は、完全復号/再符号化 (タンデム接続) を基本としたトランスコーダのソフトウェアモジュールの構成である。タンデム接続では演算量が膨大となりリアルタイム処理はできない。そこで、リアルタイム処理のための高速処理を考えた場合、どの部分の演算量が問題となるかをまず考察した。

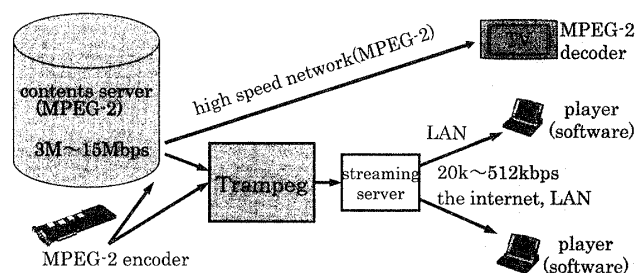


図 1 Trampeg の適用形態

Fig. 1 An example of Trampeg application.

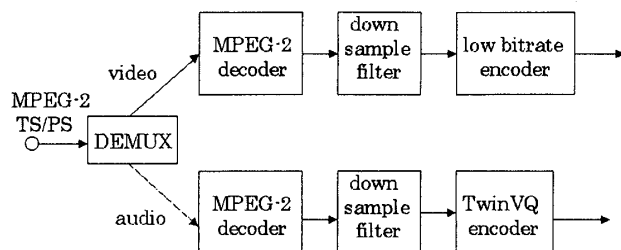


図2 タンデム接続の構成
Fig. 2 Diagram of tandem transcoder.

一般にオーディオ部分の演算量はビデオ部分に比較して小さい。MPEG-2の完全復号には、オーディオ・ビデオ合わせて400 MOPS以上を要するとされるが[9]、このうちオーディオ部分の復号に要する演算量の割合は、文献[9],[10]によれば10分の1以下となっている。またオーディオ再符号化に、低ビットレートでも高い品質を確保できるTwinVQ方式[11]を採用することを考えると、500 MHz×2 CPUのPCで11 kHz, stereoのリアルタイム符号化を行う場合、CPUの占有率は10%強である。以上より、オーディオ部分は完全復号してサブサンプリングしてから再度符号化しても、トランスコーディング全体の中での演算量の割合は低くなる。そこでオーディオに関しては、完全復号 + サブサンプリング + TwinVQ符号化の方式を採用することとした。

残る問題は、ビデオ部分の演算量をどのように削減するかとなる。文献[9],[12]をもとに、ビデオ部分の演算量を見積もったものを表1に示す。表1の値は、MPEG-2ビデオを完全復号し、サブサンプリングして画像を縦横1/2に縮小した後、動き補償(MC)及び離散コサイン変換(DCT)を基本としたH.263タイプの符号化でSIF, 30 frame/sの再符号化を行う場合である。再符号化の際の動きベクトル検出は階層探索として見積もったが、全探索の場合はこの16倍程度となる[12]。

表1によれば、低ビットレート符号化で行われる動き検出処理(ME)及びDCT, IDCTの割合が大きい。このためTrampegにおいては以下に述べる方針でソフトウェアの設計を行った。ME及びDCT, IDCTの演算量を削減するためには、下記の二つの方法が考えられる。

A. 各フレームの復号は完全に行うが、再符号化時に動きベクトル等のデータを利用する方法

B. 入力された符号化データをそのまま利用し、DCT

表1 タンデム接続の場合のビデオ部分の演算量(MOPS)
Table 1 Computation complexity of tandem transcoding from MPEG-2 video to lower bitrate stream (MOPS).

MPEG-2 decoder	200	VLD+IQ : 50
		IDCT : 30
		MC : 80
		Misc : 40
Filtering+subsampling	30	
Low bitrate encoder	730	ME : 500
		DCT, IDCT : 130
		Misc : 100
Video total	960	

やIDCT等の処理なしにDCT空間で再符号化する方法

A.の方法は動きベクトル探索処理等を省略でき、既にいくつかの検討例がある[1]~[3]。またB.の方法は、MPEG-2同士のレート変換[13],[14]やH.263, H.261同士のレート変換[15]のように、トランスコード前後でのストリーム構造が同じ場合には演算量が少なく有効である。しかし、MPEG-2のインタレース構造をもつ符号化データを、サイズ縮小・フレームレート削減を伴うプログレッシブ構造の低ビットストリームに変換する場合、DCT空間ですべて行うには複雑な演算が必要となる。

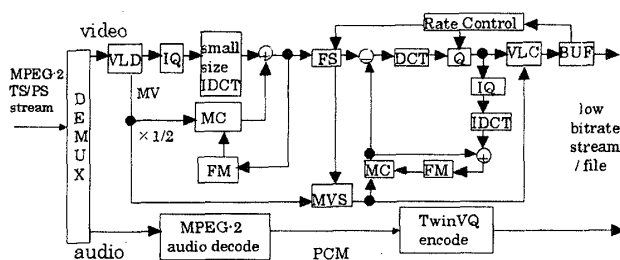
このような観点から、まずA.の方法を基本としてアルゴリズムを設計する。A.の方法では、表1の低ビットレートエンコード中の動き検出処理(ME)の演算量を0にできる。代わりに、変換前の動きベクトル等を利用して、高効率に低ビットレート用の動きベクトルを推定するアルゴリズムが必要となるが、ここでは動きベクトル追跡[16],[17]を基本とするアルゴリズムを用いることで解決を図る。この部分のアルゴリズムは次章で詳説する。このアルゴリズムは、30 frame/sのMPEG-2ストリームを、例えば6 frame/s程度のフレーム数に間引いて変換する場合にも適用できるので、低ビットレートエンコードの演算量は更に削減可能となる。

A.の方法を用いると、動きベクトル探索が不要となり演算量が削減できるが、MPEG-2復号処理については依然負荷が大きい。そこで、サイズ縮小のみをDCT空間で処理する方法を採用し、MPEG-2の復号処理の重さを削減する。すなわち、MPEG-2ストリームを完全には復号せず、DCT係数データの低周波成分のみを逆変換するハーフデコード方式により水平方

向の画像サイズを半分に (QSIF サイズの場合は更に水平垂直とも 1/2 倍に) 縮小する. そして, 復号結果中の片フィールド分のみを再符号化に用いる. これにより表 1 の MPEG-2 復号の IDCT と MC の演算量は, ほぼ半分にできる.

更に, SIF サイズへのリアルタイム変換として目標となる 10 frame/s 以下で, 変換対象が B フレームを含む MP@ML の場合, I, P フレームの画像のみを復号して処理を効率化することは有効である. 例えば, MP@ML としてよく用いられる $M=3$ (I, P フレーム間に B フレームを 2 枚ずつ含む) のストリームを 10 frame/s や 5 frame/s などで等間隔で変換するのは, 明らかに B フレームの復号画像なしでも可能である. それ以外のフレームレートで, 再符号化対象フレームが B フレームとなる場合, 時間的にそのフレームの前後で最も接近した I, P フレームで代用すれば B フレームの復号処理が削減できる. 今回は高速変換を実現する観点から, I, P フレームで代用する方式を採用する.

以上より Trampeg の基本構成を図 3 のようにした. Trampeg は MPEG-2 のシステムストリーム (TS または PS) を入力とし, 上記手法によりビデオデータを MC-DCT ベースの低レートビデオストリームに, オーディオデータを TwinVQ に変換する. ここで低レートビデオストリームは, グローバル動き補償 [18],



- FS : frame selection to be transcoded
 MVS : motion vector scaling
 VLD : variable length decoding
 VLC : variable length coding
 IQ : inverse quantization
 Q : quantization
 IDCT : inverse discrete cosine transform
 DCT : discrete cosine transform
 MC : motion compensation
 FM : frame memory
 BUF : buffer
 MV : motion vectors

図 3 Trampeg の基本構成
 Fig. 3 Configuration of Trampeg.

動きベクトルの最適符号割当て [19], 動き適応型フレームレート制御 [20] などを新たに導入して高効率化を図っている.

3. 動きベクトル変換アルゴリズム

動き探索等を省略しつつ高効率の圧縮を可能とする動きベクトルの算出アルゴリズムは, 前章で述べたように, 動きベクトル追跡 [16], [17] を基本とするアルゴリズムを採用し, 以下 i)~iv) の組合せとなっている (図 4).

- i) サイズ縮小とベクトル長変換 (MV scale)
- ii) I フレームを P フレームに変換する IP 変換 (IP trans) [21]
- iii) 動きベクトル追跡と P フレーム間引き (P drop) [16], [17]
- iv) 各縮小ブロック単位に算出されたベクトルの統合

以下, 各節で順にアルゴリズムを説明する. なお, 説明は SIF サイズへの変換の場合を例にとる.

3.1 画像サイズ縮小とベクトル長変換

まず, 変換前の MPEG-2 のインタレース構造による 720×480 サイズの符号化情報のうち, DCT 係数の水平方向低周波成分のみを用いたハーフデコードで水平方向を 1/2 にサイズ縮小し, 復号画像中の第 2 フィールドのみを抽出することで垂直方向も 1/2 とし, SIF サイズにする.

この際, サイズ縮小に対応して図 5 に示すように MPEG-2 の動きベクトルの長さ変換をする. 長さ変換係数は, 空間スケーリング係数 α と時間スケーリング係数 β で決まる. MPEG-2 の動きベクトルには, フレーム MC, フィールド MC (同パリティ), フィー

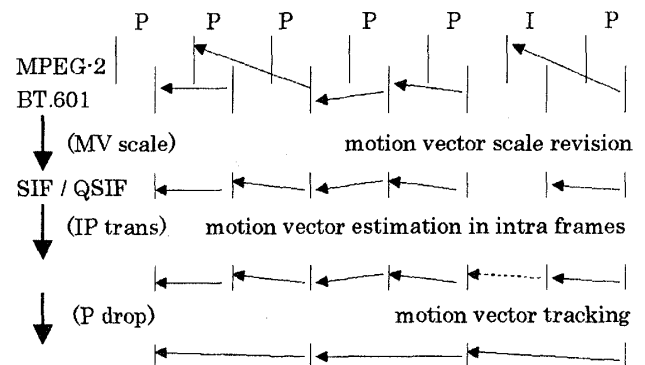


図 4 動きベクトル変換の流れ
 Fig. 4 Process outline of motion vector translation.

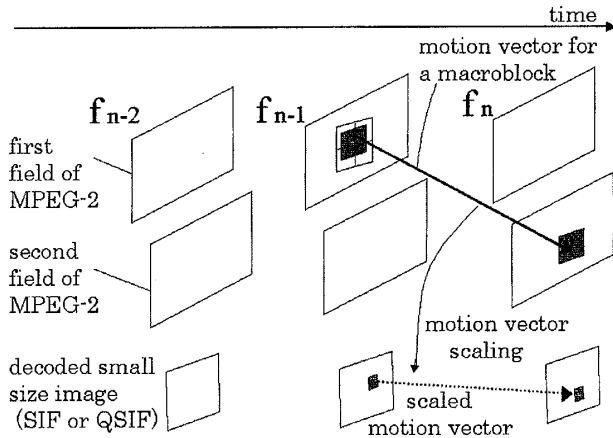


図5 動きベクトルの長さ変換方法

Fig. 5 Method of motion vector scaling.

表2 動きベクトルの長さ変換係数 (second field の復号結果を再符号化に用いる場合)

Table 2 Scaling coefficients of motion vectors.

MC type	scaling coefficients		
	α	β	$\alpha \times \beta$
frame MC	1/2	1	1/2
field MC (from same parity)	1/2	1	1/2
field MC (from different parity)	1/2	2/3	1/3

ルドMC (異パリティ) の3種があり、表2に示すように空間スケーリング係数 α は、画像サイズにより決まるのでどの場合も 1/2 となる。また時間スケーリング係数 β は、第2フィールドのみを抽出していることから、フィールドMC (異パリティ) の場合のみ 2/3 となる。最終的な長さ変換係数は、 α と β の積で与えられる。縮小復号された画像上では、MPEG-2 の 16×16 画素サイズのマクロブロックが 8×8 画素サイズのブロックとなるが、長さ変換されたベクトルは、それらの各ブロックに一つずつ対応づけられる。これらのベクトルを以下の処理で用いる。

3.2 Iフレームでの動ベクトル推定 (IP変換処理)

MPEG-2では0.5~1秒に1回の割合でIフレームが挿入された形で符号化されるのが普通だが、100 kbit/s以下の低レートでは符号化効率の観点からIフレームの間隔をより広く設定することが多い。このため、IフレームをPフレームにピクチャタイプ変換する必要がある。Iフレームには動きベクトル情報が存在しないので、これを前後のPフレームの動きベクトルを用いて推定するのがIP変換処理である。この処理は、変換するMPEG-2ストリーム中の再符号化対象フレームと参照フレームの間に、Iフレームが含まれる場合

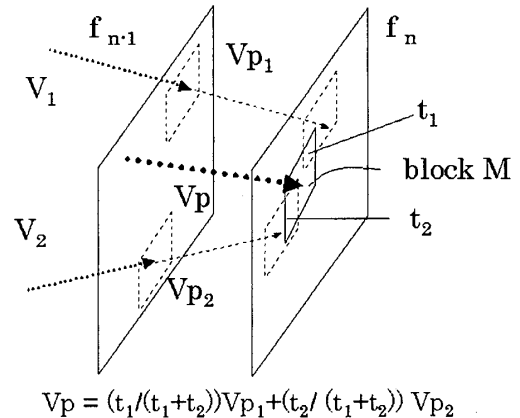

 t_1, t_2 : area of the intersection of block M and referenced block projected by V_{p1}, V_{p2}

図6 イントラフレームでの動きベクトル推定方法 (1)

Fig. 6 Method of motion vector estimation in intra frames (1).

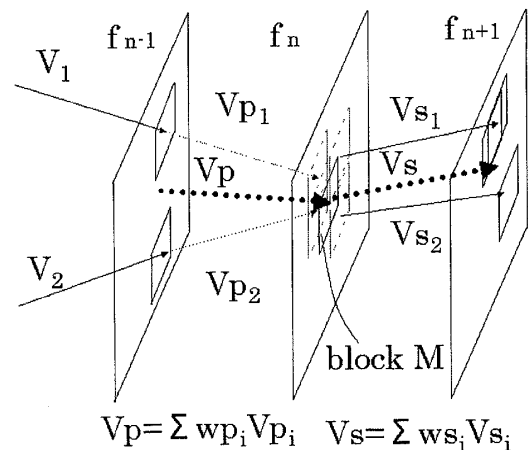


図7 イントラフレームでの動きベクトル推定方法 (2)

Fig. 7 Method of motion vector estimation in intra frames (2).

に必要となる。

図6のように、IP変換を行うIフレームを f_n とし、その直前のPフレームを f_{n-1} とする。そして、 f_{n-1} 上のブロックを、そのブロックのもつ動きベクトルと同じ動きベクトル (V_{p_i}) を用いて f_n 上へ投影する。 f_n 上のあるブロック M に対して、 f_{n-1} から投影されたブロックと共通部分をもつ動きベクトル V_{p_i} ($i = 1 \sim k_p$) をすべて取り出す。次に V_{p_i} を投影しているブロックとブロック M との共通部分の面積 (t_i) の比に応じて、投影された動きベクトル V_{p_i} を重み付けして平均動きベクトル V_p を算出する。図6は $k_p=2$ の場合の例である。

更に、直後のPフレーム f_{n+1} については、図7に

示すように、各ブロックのもつ動きベクトルにより f_n 上へ投影させる．そしてブロック M に対して、 f_{n+1} から投影されたブロックが共通部分をもつ動きベクトル V_{s_i} ($i = 1 \sim k_s$) をすべて取り出し、 V_p のときと同様に平均して動きベクトル V_s を算出する．

最後に、算出された動きベクトル V_p 、 V_s の向きがそろっているかどうかを判定する．向きがそろっているとき、すなわちある値 T に対して、

$$|V_p - V_s| < T \quad (1)$$

を満たすときのみ、ブロック M をインターブロック（フレーム間予測ブロック）と判定して動きベクトルを算出する（図 7）．M に対する最終的な動きベクトルは V_p 、 V_s を平均して算出する．式 (1) が成立しないときは、イントラブロックとして処理する．

3.3 P フレーム間引き処理（動きベクトル追跡）

低レートでも画質を維持するためには、再符号化するフレーム数を削減する必要がある．図 8 のように P フレームを 1 フレームスキップして参照する形で再符号化する場合を考える．P フレームが間引かれた場合、新たな参照フレームからの動きベクトルを求める必要があるが、この処理を 3.1 及び 3.2 で用意した動きベクトルを追跡することで実現する．

再符号化対象フレーム f_n の小ブロックの動きベクトル V を、

$$V = \sum w_{p_i} V_{p_i} + V_s \quad (2)$$

で算出する [17]．ただし、スキップするフレーム f_{n-1} への参照ベクトルを V_s 、 V_s により参照されている領域と重なる四つのブロックから f_{n-2} への参照ベクトルをそれぞれ V_{p_1} 、 V_{p_2} 、 V_{p_3} 、 V_{p_4} とする．また、 w_{p_i} は図 9 に示すように V_s の参照先と各ブロックとの共通部分の割合を表す係数である．式 (2) により参照する面積に応じて重み付けする際、対応する MPEG-2 のブロックがイントラブロックの場合はそのブロックに関する w_{p_i} は 0 とし、更に、イントラブロックの割合が半分以上となる場合は、再符号化対象フレーム f_n の小ブロックのベクトルは算出せずに、イントラブロックとして再符号化する．2 フレーム以上をスキップする場合は、上記の操作を繰り返す．

3.4 算出ベクトルの統合処理

前節までの処理により、縮小画像上の 8×8 画素の小ブロックごとにベクトルが算出される．再符号化時には 16×16 画素のブロック単位に動き補償を行うた

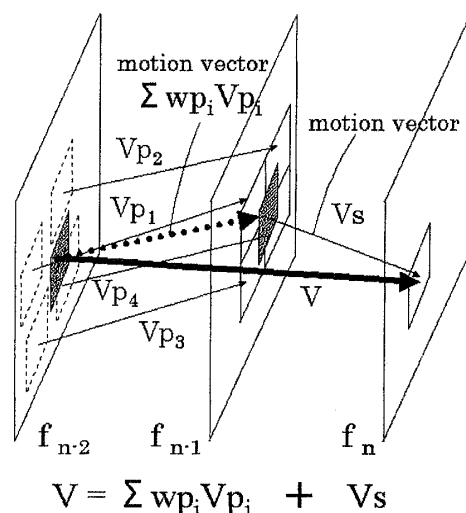


図 8 動きベクトル追跡方法 (1)
Fig. 8 Method of motion vector tracking (1).

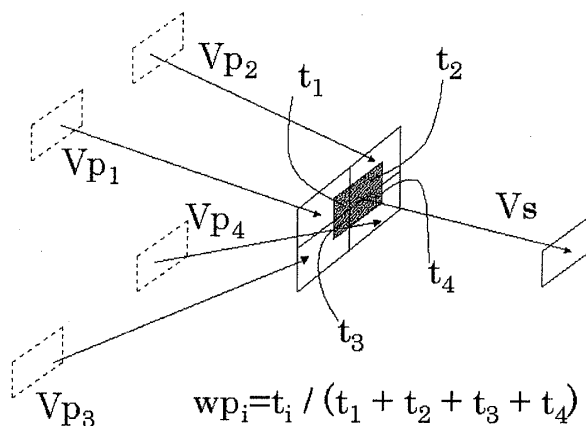


図 9 動きベクトル追跡方法 (2)
Fig. 9 Method of motion vector tracking (2).

め、算出された 8×8 画素単位のベクトルは各ブロック単位に四つのベクトルの平均がなされ、再符号化に用いられる．ただし、統合される 8×8 画素小ブロック中に、イントラ判定されベクトルをもたないものが三つ以上含まれる場合、 16×16 画素ブロックをイントラと判定する．また、イントラ判定された 8×8 画素小ブロックが二つ以下である 16×16 画素ブロックでは、ベクトルが算出された 8×8 画素小ブロックのみの平均を、統合後のベクトルとして再符号化に用いるものとする．

4. 動ベクトル変換アルゴリズムの性能評価

3. で述べた動きベクトル変換アルゴリズムの効果性を評価するため、次のような実験を行った．まず標準画像の Flower garden と Mobile and calendar との

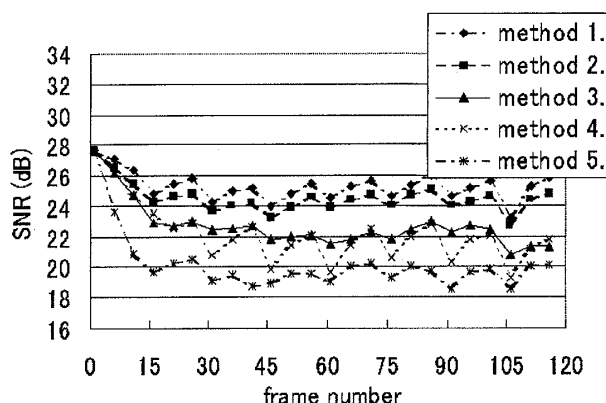


図 10 各処理方法による変換結果の画質の比較 (Flower garden)

Fig.10 SNR of transcoded images in each method (Flower garden).

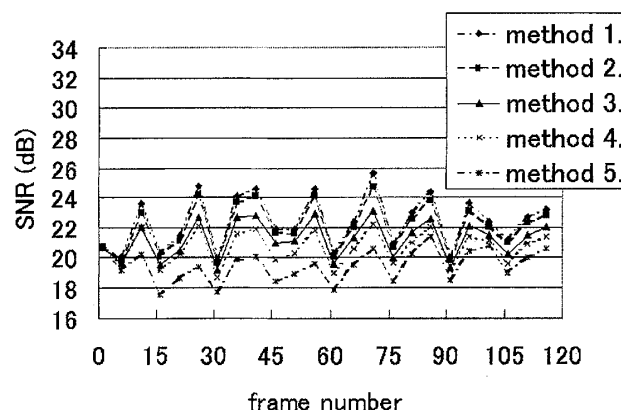


図 11 各処理方法による変換結果の画質の比較 (Mobile and calendar)

Fig.11 SNR of transcoded images in each method (Mobile and calendar).

それぞれ 120 フレーム (4 秒分) を MPEG-2 ビデオストリーム (SP@ML, N=15, 8 Mbit/s) に符号化した。動きベクトルの探索範囲は水平方向・垂直方向とも半画素単位で ± 63 とし、TM5 [22] の方式を用いた。この MPEG-2 ストリームを以下の方法 1～方法 5 によりトランスコードした。変換パラメータは、画像サイズ SIF, ビットレート 92 kbit/s, 毎秒フレーム数 6 frame/s とした。図 10 と図 11 は、MPEG-2 ストリームをサイズ縮小し復号した片フィールド画像 (すなわち低ビットレートエンコーダ入力画像、図 3 の FS の後のデータ) を基準として、方法 1～方法 5 によって得られる復号画像の SNR を比較したものである。

[方法 1] 再符号化時に半画素単位に全探索を行って求めた動きベクトルにより符号化

[方法 2] 再符号化時に整数画素単位に全探索を行って求めた動きベクトルにより符号化

[方法 3] 3. で述べた Tramepeg の変換アルゴリズムにより算出された動きベクトルを用いて符号化

[方法 4] Tramepeg 変換で IP 変換を用いずに、動きベクトルが算出できない部分では、「intra / inter (0 ベクトル)」の適応予測としたアルゴリズムで符号化

[方法 5] すべてのフレームに対して再符号化時に動きベクトル変換を行わず、「intra / inter (0 ベクトル)」の適応予測のみで符号化

図 10 と図 11 より以下の 2 点がわかる。

(1) 再符号化時に全探索した場合と Tramepeg との差は 1～2 dB 程度にとどまる。

(2) IP 変換の導入で 1～2 dB の画質改善効果がある。

なお、図 10, 図 11 では 15 フレームごとに SNR が下降する傾向が見られる。この理由を以下に挙げる。MPEG-2 デコード画像 (= 低ビットレートエンコーダ入力画像) は 15 フレームごとに I フレームとして符号化されたものであるが、TM5 [22] をはじめとする符号化制御によれば I フレームは P フレームや B フレームよりも高品質に符号化され、その性質が P, B フレームとは異なる。低ビットレートエンコーダでは 15 フレームごとの I フレームを P フレームとして符号化するため、時間的に先行する P フレームのローカルデコード画像から I フレームを予測することになるが、性質の異なる P フレームから予測することになるので結果として符号化効率が低下することになる。本件は低レートエンコーダで各 P フレームに同じ情報量を割り当てる制御を行っていることに起因している。本検討においてはこの SNR の低下に伴う視覚的な劣化は認められなかったが、IP 変換を行ったフレームへの情報割当て方法の最適化には検討の余地があるといえる。

一方表 3 は、方法 1～方法 5 の各場合の変換に要した時間を示している。測定に使用したのは 450 MHz の PentiumII Dual CPU の PC である。方法 1, 方法 2 の場合は演算量が大きくなり、リアルタイム処理は不可能であるが、方法 3～方法 5 はリアルタイム処理可能である。方法 1 と方法 2 においては、動き検出に全探索を用いているために、トランスコード全体の処理量に占める動き検出の処理量の割合は約 95% を占める [12]。動き検出の処理は符号化対象のフレーム間で行うため、処理量はそのフレーム数にほぼ比例す

論文/低ビットレート用 MPEG-2 リアルタイムソフトウェアトランスコーダ「Trampeg」の開発

表3 各変換方法による消費時間の比較 (単位: 秒, 変換シーケンスは 40 秒)

Table 3 Processing time for transcoding.

transition rate	460 kbit/s, 10 fr/s	92 kbit/s, 6 fr/s
method 1.	1761	1043
method 2.	361	214
method 3.	37	34
method 4.	37	34
method 5.	36	34

る。一方、変換後のビットレートは可変長符号化の処理量 (VLC) に影響するが、全体から見るとその割合はわずかであるため、460 kbit/s, 10 fr/s の場合と 92 kbit/s, 6 fr/s の場合の処理時間は変換後のフレームレートに大きく依存する。

一方、方法3, 方法4, 方法5では動きベクトルの検索をしないため、トランスコード全体の処理の中では、変換後のビットレートやフレームレートに依存しない MPEG-2 復号処理などの処理量の占める割合が大きくなり、変換後のフレームレートやビットレートによる影響は DCT, IDCT など及ぶものの、全体の処理量に与える影響は 10% 程度と少なくなっている。

また、同じビットレート同じフレームレートで比較すると方法3, 方法4, 方法5の処理量はほとんど変わらないこと、及び方法5はベクトル検出処理を全くしない方式であることから、方法3及び方法4で用いている 3. で提案した動きベクトル変換処理は、トランスコード全体から見れば処理時間は非常に小さいことがわかる。

図 10, 図 11 及び表 3 の結果から、Trampeg で用いている方法3の変換アルゴリズムは、画質的には再符号化時に半画素単位や整数画素単位の全探索をした場合をやや下回るが、リアルタイム処理を可能にするという条件下では最良の選択となっている。

5. ソフトウェア構築と性能評価

以上の議論に基づき、実際の PC 上で動作する Trampeg ソフトウェアを構築した。ソフトウェア仕様を表 4 に示す。Trampeg システムのリアルタイム性を評価するため、実際の PC 上でソフトウェアを動作させてその処理時間を測定した。測定には 450 MHz の PentiumII Dual CPU の PC を使用した。また、ソフトウェアは、MPEG-2 システム DEMUX 部・MPEG-2 映像デコーダ部・低レート映像符号化部・MPEG-2 音声デコーダ部・音声符号化部等を複数の処理単位

表4 Trampeg の諸元

Table 4 Specification of Trampeg.

input	MPEG-2 TS/PS	
(file /	video	SP@ML, MP@ML,
stream		frame structure, $M \leq 3$
from	audio	MPEG-2 Layer1, 2,
network)		MPEG-1 Layer2
decoding	video	half decode
method	audio	full decode
output	video	size : SIF(352 × 240),
(file /		QSIF(176 × 112)
stream		frame rate : 2~15 frame/s
to		bitrate : 20 kbit/s~512 kbit/s
network)	audio	format : TwinVQ
		bitrate : 8 kbit/s~80 kbit/s

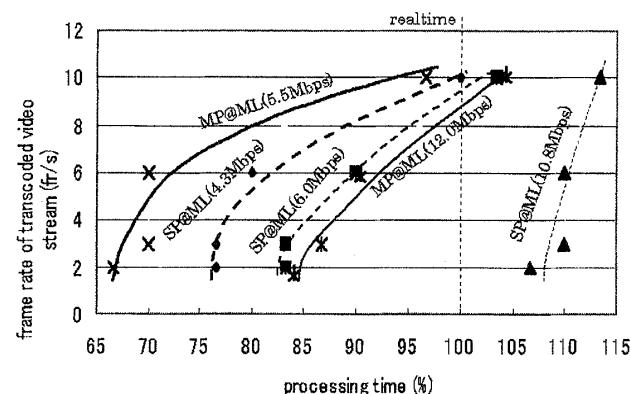


図 12 SP@ML 及び MP@ML の変換時間 (PenII 450 MHz × 2 CPU, () 内は MPEG-2 ストリームのビットレート)

Fig. 12 Processing time of Trampeg for MPEG-2 system stream (SP@ML and MP@ML).

(スレッド) に分割し、並列に実行処理が可能な構成となっている。図 12 に、入力ビットレートが 4.3, 6.0, 10.8 Mbit/s の SP@ML, 及び 5.5, 12.0 Mbit/s の MP@ML の MPEG-2 システムストリームをビデオ 92 kbit/s・オーディオ 20 kbit/s に変換した場合の変換時間の測定結果を示す。縦軸はトランスコード後のフレームレートを示しており、横軸はトランスコード対象の MPEG-2 ストリームの長さに対する、Trampeg 処理に要した時間の割合 (%) を示している。100% 以下の場合がリアルタイム処理が可能となる。これより、SP@ML 6.0 Mbit/s の入力を SIF, 92 kbit/s, 6 frame/s で高品質にリアルタイム変換可能であることがわかる。また、MP@ML の場合は B フレームの処理を省略できるため、処理速度は SP@ML の場合より速く、12 Mbit/s の MPEG-2 でもリアルタイム処理が可能である。更に、ネットワーク入力された

MPEG-2 システムストリーム (TS) を Trampeg によりリアルタイム変換し、ビデオ部分は SIF, 92 kbit/s, 6 frame/s, オーディオ部分は 20 kbit/s に変換されたストリームをダイレクトにサーバに送り、ネットワーク配信可能なことも確認できた。

6. む す び

MPEG-2 ストリームを低ビットレートストリームへリアルタイム変換可能なソフトウェアトランスコーダ Trampeg を開発した。本論文では、トランスコーダの適用形態と要求条件を明確にし、トランスコーディングアルゴリズムの設計指針を示した。また、変換アルゴリズム中で演算量削減に影響が大きい、MPEG-2 の動きベクトル情報を用いた再符号化時の動きベクトルの推定アルゴリズムを詳説した。更に、実際の PC 上でビデオ部分の Trampeg アルゴリズムの評価とオーディオ部分も含めたシステム全体の評価を行った。評価結果から、DCT 空間でのダウンサンプリングと、MPEG-2 ストリーム中の動きベクトルを適応スケールリングする処理を組み合わせた Trampeg アルゴリズムにより、画質が維持されつつ高速化が可能となり、リアルタイム処理の実現が確認された。これにより、1 元管理されている MPEG-2 コンテンツを、より帯域の狭いインターネットや LAN を介して簡易視聴することができ、広帯域、狭帯域での同時配信システムや、VOD におけるプレビューシステム等の構築を容易に行うことが可能になった。3. で示した変換アルゴリズムは、H.263 や MPEG-4 等の動き補償と DCT を基本にした符号化方式への変換にも適用可能である。

今後の課題として、普及が予想される HDTV 用の MPEG-2 ストリームへの対応、B フレームの復号画像を用いる高フレームレート送出機能の付加などを検討したい。

謝辞 本研究の機会を与えて頂いた、NTT サイバースペース研究所の木暮賢司所長、長谷雅彦プロジェクトマネージャに感謝します。また、石橋聡主幹研究員、渡辺裕早稲田大学教授はじめ、日ごろ御指導御助言頂いた皆様に感謝します。

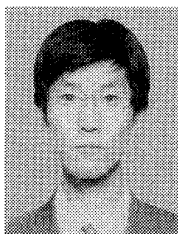
文 献

- [1] S.J. Wee, J.G. Apostolopoulos, and N. Feamster, "Field-to-frame transcoding with spatial and temporal downsampling," ICIP, vol.4, pp.271-275, 1999.
- [2] B. Shen, I.K. Sethi, and V. Bhaskaran, "Adaptive motion vector resampling for compressed video downscaling," ICIP, vol.I, pp.771-774, 1997.
- [3] J. Youn, M.T. Sun, and C.W. Lin, "Motion vector refinement for high-performance transcoding," IEEE Trans. Multimedia, vol.1, no.1, pp.30-40, March 1999.
- [4] 阿久津隆史, 木全英明, 清水 淳, 八島由幸, 小林直樹, "低ビットレート用 MPEG-2 リアルタイムソフトウェアトランスコーダ「Trampeg」の開発," 情処学研報, vol.AVM29-5, pp.25-30, 2000.
- [5] 辻 裕之, 阿久津隆史, 清水 淳, 八島由幸, 小林直樹, "低レート用 MPEG-2 トランスコーダ「Trampeg」の開発と性能評価," 信学技報, IE17-2, 2000.
- [6] 藤原 洋, 最新 MPEG 教科書, アスキー出版局, 1994.
- [7] 滝嶋康弘, 和田正裕, 村上仁己, "低レート動画像符号化における最適フレームレートの検討," 信学技報, IE56-7, 1991.
- [8] 林 孝典, 井合 知, "フレームレートの変化が動画像主観品質に及ぼす影響," NTT R&D, vol.46-8, pp.67-74, 1997.
- [9] M. Ikekawa, D. Ishii, E. Murata, K. Numata, Y. Takamizawa, and M. Tanaka, "A real-time software MPEG-2 decoder for multimedia PCs," ICCE, WAM1.1, pp.2-3, 1997.
- [10] T.H. Tsai, L.G. Chen, S.C. Huang, and H.C. Chang, "A low-cost architecture design with efficient data arrangement and memory configuration for MPEG-2 audio decoder," ISCAS, vol.4, pp.65-68, 1998.
- [11] T. Moriya, N. Iwakami, A. Jin, K. Ikeda, and S. Miki, "A design of transform coder for both speech and audio signals at bit/sample," ICASSP, vol.II, pp.1371-1374, 1997.
- [12] 大迫史典, 八島由幸, 小寺 博, 渡辺 裕, 島村和典, "動的演算量スケラブルアルゴリズムによるソフトウェア画像符号化," 信学論 (D-II), vol.J80-D-II, no.2, pp.444-458, Feb. 1997.
- [13] 笠井裕之, 富永英義, 花村 剛, 亀山 渉, "低遅延 MPEG-2 ビデオトランスコーダ符号量制御方式," 信学論 (B), vol.J83-B, no.2, pp.151-164, Feb. 2000.
- [14] P.A.A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," IEEE Trans. Circuits and Systems for Video Technology, vol.8, no.8, pp.953-967, Dec. 1998.
- [15] W. Zhu, K.H. Yang, and M.J. Beacken, "CIF-to-QCIF video bitstream down-conversion in the DCT domain," Bell Labs Technical Journal, pp.21-29, July-Sept. 1998.
- [16] 清水 淳, 嵯峨田淳, 渡辺 裕, 小寺 博, "フレームスキップによるビットストリームスケールリング," 画像符号化シンポジウム, PCSJ'96, pp.25-26, 1996.
- [17] 清水 淳, 嵯峨田淳, 渡辺 裕, 小林直樹, "参照フレーム変更によるビットストリームスケールリング," 信学論 (D-II), vol.J83-D-II, no.3, pp.967-976, March 2000.
- [18] 嵯峨田淳, 清水 淳, 渡辺 裕, 小寺 博, "インターネット映像伝送を考慮した映像符号化の効率改善," 1997 信学総大 (情報システム 2), D-11-87, 1997.

論文／低ビットレート用 MPEG-2 リアルタイムソフトウェアトランスコーダ「Trampeg」の開発

- [19] 清水 淳, 嵯峨田淳, 渡辺 裕, 小林直樹, “動きベクトルのノルム・方向成分表現による符号化方法の検討,” 画像符号化シンポジウム, PCSJ'97, pp.67-68, 1997.
- [20] 清水 淳, 嵯峨田淳, 上倉一人, 小林直樹, “量子化パラメータ情報削減による符号化性能の改善,” 1998 信学ソ大 (情報システム 2), D-11-6, 1998.
- [21] 阿久津隆史, 清水 淳, 八島由幸, 小林直樹, “MPEG-2 トランスコーディングの一検討,” 画像電子学会第 27 回年次大会予稿集, pp.29-30, 1999.
- [22] ISO-IEC JTC1, MPEG 93/N225b, “Test model 5/draft,” 1993.

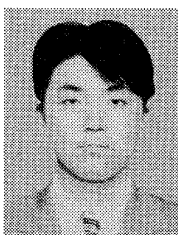
(平成 12 年 8 月 25 日受付, 12 月 22 日再受付)



阿久津隆史 (正員)

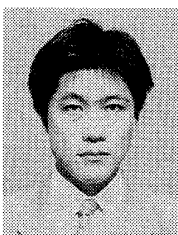
平 4 東大・理・数学卒. 平 6 同大大学院数理科学研究科修士課程了. 同年日本電信電話 (株) に入社. ロボットビジョン, 画像符号化の研究に従事. 平 10 本会学術奨励賞受賞. 現在, NTT サイバースペース研究所メディア通信プロジェクト研究主任.

情報処理学会, 日本ロボット学会各会員.



木全 英明

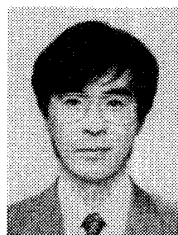
平 5 名大・工・応用物理卒. 平 7 同大大学院工学研究科修士課程了. 同年日本電信電話 (株) に入社. 画像符号化, エラー耐性の研究に従事. 映像情報メディア学会会員.



清水 淳 (正員)

平 2 日大・理工・電気卒. 平 4 同大大学院工学研究科修士課程了. 同年日本電信電話 (株) に入社. 画像符号化の研究に従事. 平 9 画像電子学会技術賞, 平 10 本会学術奨励賞各受賞. 現在, NTT 東日本法人営業本部主査. 映像情報メディア学会, 画像

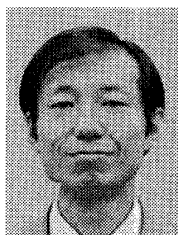
電子学会各会員.



八島 由幸 (正員)

昭 56 名大・工・電子卒. 昭 58 同大大学院工学研究科電子工学専攻修士課程了. 同年日本電信電話公社 (現 NTT) に入社. 以来, 画像信号の高効率符号化, 動画画像処理の研究に従事. 工博. 現在, NTT サイバースペース研究所メディア通信プロジェクト

主幹研究員. IEEE, 映像情報メディア学会, 情報処理学会各会員.



小林 直樹 (正員)

昭 54 東工大・理・応用物理卒. 昭 56 同大大学院物理情報工学修士課程了. 同年日本電信電話公社 (現 NTT) に入社. 以来, ファクシミリ, 画像入力, マルチメディアシステムの研究開発を経て, 現在サイバースペース研究所にて画像符号化の研究開発

に従事. 工博. 同研究所メディア通信プロジェクト画像符号化技術グループリーダー. TTC 第 5 部門第 2 専門委員長.