

低ビットレート用 MPEG-2 リアルタイムトランスコーダ「Trampeg」の開発

阿久津隆史* 木全英明* 清水淳** 八島由幸* 小林直樹*

*NTT サイバースペース研究所, **NTT 東日本会社

〒239-0847 神奈川県横須賀市光の丘 1-1

t_akutsu@nttvd.t.hil.ntt.co.jp

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 を完全にデコード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、ネットワーク入力された MPEG-2 システムストリームがリアルタイム変換され、変換された低レートストリームがそのままサーバを介して配信できることを確認した。本稿では Trampeg の技術概要を紹介し、その性能について述べる。

“Trampeg” — MPEG-2 Real-time Transcoder for Low Bitrate Transmission

*Takashi AKUTSU *Hideaki KIMATA *Yoshiyuki YASHIMA

*Naoki KOBAYASHI **Atsushi Shimizu

*NTT Cyber Space Laboratories **NTT East Corporation

Hikarinooka 1-1, Yokosuka, Kanagawa, 239-0847, JAPAN

t_akutsu@nttvd.t.hil.ntt.co.jp

The video delivery service supporting a wide variety of bitrates for the contents encoded to MPEG-2 demands some transcoding method to lower-rate compressed bitstreams. The most general method is to decode MPEG-2 stream completely and re-encode it to lower-rate one. But it doesn't enable real-time processing with software on a general PC system because of computational intensiveness. Required algorithm is for reducing the cost of motion estimation, complete decode and downsampling. Recently, a real-time high-quality software Transcoder for MPEG-2 named “Trampeg” has been developed recently, is introduced. MPEG-2 stream which is input through the network can be transcoded in real-time to lower-rate one which can be directly distributed via server. In this paper, technologies used in Trampeg and their evaluation result are introduced.

1. はじめに

放送局のデジタル化およびデジタル放送の開始、また DVD の普及などを機に、今後は大量の映像コンテンツが MPEG-2 で一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 等で視聴するためには、相応の低いビットレートへの変換が必須である。最も単純なトランスコーディングは、いったん各フレームを YUV 形式等に完全にデコードし、再度低ビットレートにて符号化を行う方法である。しかし、その方法では符号化を行う際の演算量が大きく、リアルタイム処理が困難である。より低コストで簡便な方法として、変換前の MPEG-2 ストリームの情報を用いて演算量を小さく抑え、汎用の PC を用いてソフトウェアで変換する方法がある。本稿では、MPEG-2 システムストリームを 20kbps～512kbps 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) の開発と、その技術内容について紹介する。

2. Trampeg の構成と変換アルゴリズム

MPEG-2 ストリームを汎用 PC でリアルタイム変換するためには、特に演算量の大きいビデオ変換部分の処理の効率化が必要である。ビデオ符号化データのトランスコーディングの実現方法としては、大きく分けて次の a.～c. が考えられる。

- 各フレームを YUV 形式等に完全にデコードし、再度低ビットレートで完全な符号化を行う方法
- 各フレームのデコードは完全に行うが、再符号化時に動きベクトル等のデータを利用する方法
- 入力された符号化データをそのまま利用し、IDCT や DCT 等の処理を行わずに、DCT 空間で再符号化する方法

a.の方法では演算量が大きいため、現状ではソフトウェアでリアルタイム処理を行うのは困難である。一方、b.の方法は動きベクトル探索処理等を省略できるため、a.より高速に処理可能であり、これまでにいくつかの検討例がある[1]～[3]。ただ、入力ストリームが MPEG-2 の場合はデコード処理が重くなるという欠点がある。また、c.の方法は条件次第では演算量が少なく有効であり、MPEG-2 どちらのレート変換[4],[5]や H.263,H.261 どちらのレート変換[6]に利用される。しかし、MPEG-2 のようにインタレース構造を持つ符号化データを、サイズ縮小・フレームレート削減をともなうプログレッシブ構造の低ビットレートストリームに変換する場合、フィールド DCT のフレーム DCT への変換をはじめ、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、Trampeg[7]では b.の方法を基本とし、サイズ縮小のみを DCT 空間で処理する方法を採用した。具体的には、以下の 1.および 2.に基づく変換を行う。

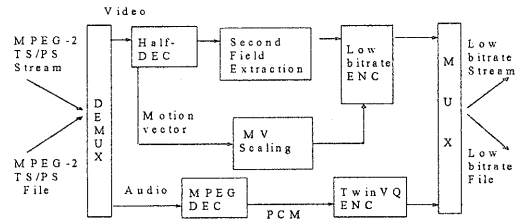


図 1: Trampeg の変換処理方法

表 1: Trampeg の主要諸元

入力ストリーム (file / network 入力)	MPEG-2 TS/PS Video : SP@ML, MP@ML, frame structure, M≤3 Audio : MPEG-2 Layer1,2 MPEG-1 Layer2
変換方式	Video : ハーフデコード方式 Audio : フルデコード方式
出力ストリーム (file / network 出力)	Video : size : SIF(352×240), QSIF(176×112) frame 数 : 2～15frame/s bitrate : 20kbps～512kbps (可変) Audio : format : TwinVQ 形式 bitrate : 8kbps～80kbps (可変)

- MPEG-2 ストリームを完全にはデコードせず、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で画像サイズを縮小する。
- 再符号化処理時に大きな演算量を要する動きベクトル探索を省略するため、変換前の MPEG-2 ストリーム中に存在する動きベクトルを利用し、低演算量での動き補償を実現する。

オーディオは処理量がビデオに比べて少ないので、いったん完全に PCM に復号してから再度 TwinVQ [8]に符号化する。

Trampeg の構成を図 1 に、諸元を表 1 に示す。Trampeg は MPEG-2 のシステムストリーム (TS または PS) を入力とし、前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[9]、動きベクトルの最適符号割り当て[10]、動き適応型フレームレート制御[11]などを新たに導入して高効率化を図った独自ストリームで、H.263 を超える高画質を提供できる[12]。Trampeg はこの変換をリアルタイムで行い、変換した低レートストリームをそのままネットワーク出力し、サーバを介して配信することができる。

3. 動きベクトル変換アルゴリズム

ビデオ変換部分で行われる処理を詳説する。処理は基本的に、以下 i)～iv) の組み合わせで行われ、v) は MP@ML の場合にのみ適用される。

- サイズ縮小とベクトル長変換
- 動きベクトル追跡と P フレーム間引き[13],[14]
- I フレームを P フレームに変換する IP 変換[15]
- 各縮小ブロック単位に算出されたベクトルの統合

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

	倍率
frame MC	1/2
field MC (from same parity)	1/2
field MC (from different parity)	1/3

v) B フレームの処理方法
各節で順に処理方法を述べる。

3.1 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 係数の低周波成分のみを用いたハーフデコードでサイズ縮小し、復号結果中の片フィールドのみを低レート化する。MPEG-2 の完全な復号は演算コストが大きいため、このサイズ変換により、処理時間を削減で

きる。また、画像サイズ縮小とノンインタレース化に対応して、MPEG-2 の動きベクトルも長さ変換する(表 2 および図 2)。そして、復号された片フィールド画像上の縮小された各ブロック(8×8 画素)に長さ変換後のベクトルを対応づけておく。これらのベクトルを以下の処理で用いる。

3.2 P フレーム間引き処理

3.1 節のように縮小デコード処理を行っても、再符号化時に MPEG-2 の全てのフレームを再符号化したり、通常の動きベクトル探索を行ったりすれば、リアルタイム処理は難しい。画質を維持しながら探索の演算量を削減するため、再符号化するフレーム数を削減する。これに伴い、3.1 節で用意した動きベクトルを用いて変換処理を行い、再符号化に用いる。

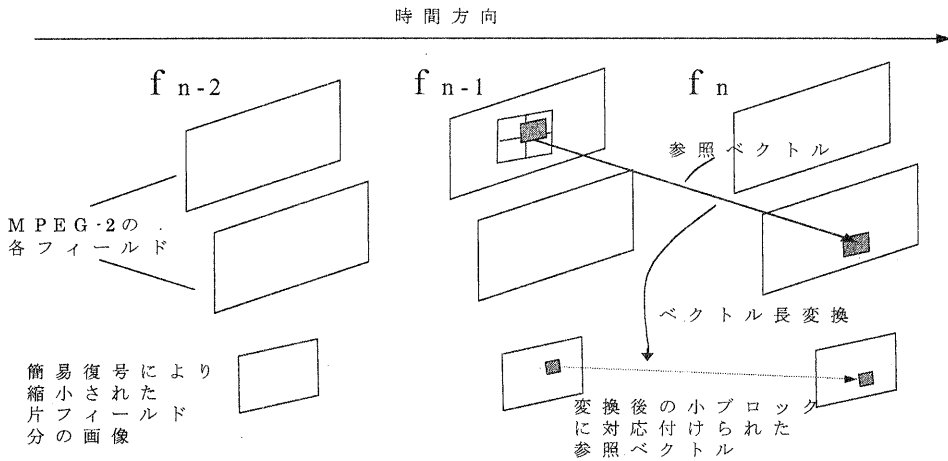


図 2 動きベクトルの長さ変換

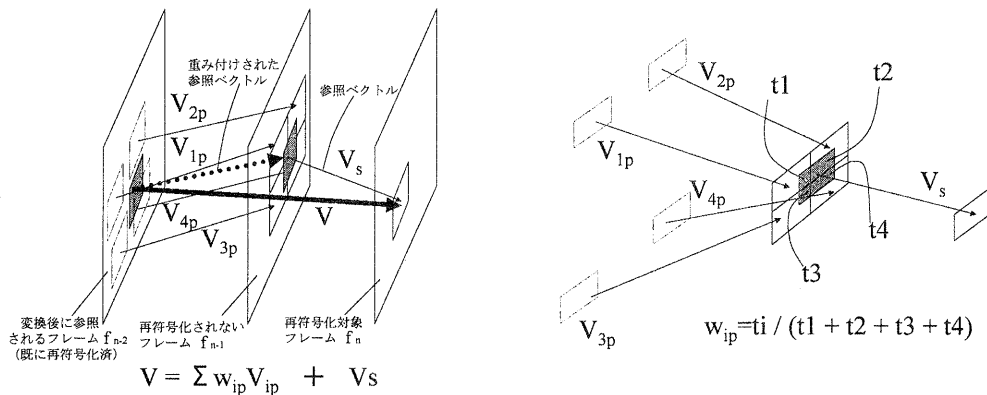


図 3 動きベクトル追跡による符号化用ベクトルの算出 (1 フレームの間引き処理を行う場合)

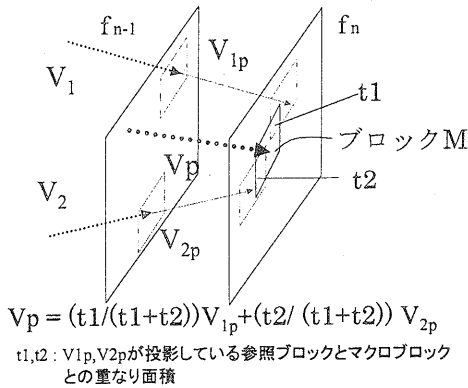


図4 IフレームをPフレームに変換する場合の参照ベクトルの算出方法

図3のように1フレームをスキップして参照する形で再符号化する場合、再符号化対象フレーム f_n の小ブロックの再符号化用の動きベクトル V は、

$$V = \sum w_{ip} V_{ip} + V_s$$

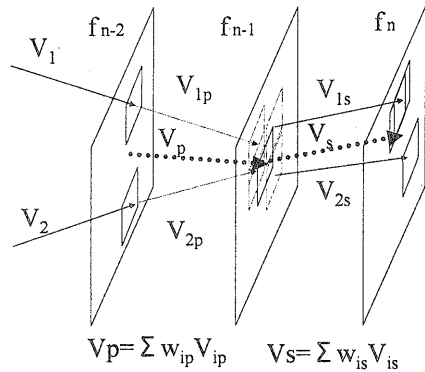
で算出される[14]。ただし、スキップするフレーム f_{n-1} への参照ベクトルを V_s 、 V_s により参照されている4つのブロックからの f_{n-2} への参照ベクトルをそれぞれ V_{1p} 、 V_{2p} 、 V_{3p} 、 V_{4p} とする。また、 w_{ip} は V_s の参照先と各ブロックとの共通部分の割合を表す係数とする(図3右図)。

上記の式により参照する面積に応じて重み付けする際、もとのMPEG-2のブロックがイントラモードの場合はそのブロックに関する w_{ip} は0とする。イントラモードのブロックの割合が半分以上となる場合は、ベクトルを算出せずにイントラブロックとして符号化する。

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。再符号化対象フレームと再符号化後に参照されるフレームとの間にIフレームが含まれる場合は、3.3節の方法でIフレームにIP変換処理を先に行った上で上記処理を適用する。

3.3 IP変換処理

変換するMPEG-2ストリーム中の、再符号化対象フレームと再符号化後に参照されるフレーム間にIフレームが含まれる場合、その前後のPフレームの動きベクトルを用いて、各小ブロックに対応する動きベクトルを算出する(IP変換処理)。まず、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} のブロックを、そのブロックの持つ動きベクトルと同じ動きベクトル ($V_i = V_{ip}$) を用いて f_n 上へ投影する。 f_n 上のブロック 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持つ動きベクトル V_{ip} ($i = 1 \sim k_p$) を全て取り出す。次に V_{ip} 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_{ip} を重み付けして平均動きベクトル V_p を算出する(図4は $i=2$ の例を示す)。さらに、直後のPフレーム f_{n+1} については、各ブロックの持つ動き



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持つ動きベクトル V_{is} ($i = 1 \sim k_s$) を全て取り出し、 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 の向きが揃っているかどうかを判定する。向きが揃っている時、すなわちある値 T に対して、

$$|V_p - V_s| < T$$

を満たす時のみ、 M をインターブロックと判定して動きベクトルを算出する(図4右)。 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。

3.4 算出ベクトルの統合処理

前節までに、各小ブロックに対応づけられたMPEG-2からの動きベクトルを変換し、 8×8 画素の小ブロックごとに再符号化用に算出する方法を示した。再符号化時には 16×16 画素のブロック単位に符号化するため、算出された小ブロック単位のベクトルは各ブロック単位に4ベクトルの平均が算出され、各ブロックの再符号化に用いられる。

3.5 Bフレームの処理方法

MPEG-2ストリームでは、I,P,Bフレームのいずれかとして各フレームは符号化されている。SP@MLの場合、各フレームはI,Pのいずれかとして符号化されている。この場合は、3.1節~3.4節までに記した方法で、再符号化用のベクトルを算出する。一方、変換対象がMP@MLでBフレームを含む場合、I,Pフレームのみを復号することで、さらに処理を効率化できる(Bフレームは動きベクトル情報のみを復号する)。一般にBフレームの画質はI,Pフレームに比べて低いため、Trampegでは再符号化対象フレームには用いない。符号化対象フレームがBフレームとなった場合は、その前後で最も接近したIまたはPフレームの復号結果を代用して再符号化する。MP@MLの変換処理では、IP変換時に前後のPフレームよりも接近したBフレームの持つ動きベクトルを利用して変換を行うことができる。

4. 動ベクトル変換アルゴリズムの性能評価

3章で述べた動きベクトル変換アルゴリズムの効果の評価するため、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, 8Mbps) 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 [16]の方式によるフィールド / フレーム適応予測 (DP 予測は off) を用いた。この MPEG-2 ストリームを以下の方法 1～方法 5 により変換した。変換パラメータは、画像サイズは SIF とし、92kbps, 6frame/s とした。図 5 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を基準として、方法 1～方法 5 の各場合の 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した場合
 方法 2: 再符号化時に整数画素単位の探索で求めた動きベクトルにより符号化した場合
 方法 3: 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した場合
 方法 4: Tramepeg 変換で IP 変換を off としたアルゴリズムにより符号化した場合
 方法 5: 再符号化時に、「イントラ / 動き補償なしの参照」の適応予測のみで符号化した場合

図 5 より以下の 2 点がわかる。

- 再符号化時に全探索した場合と Tramepeg との差は 1～2dB 程度にとどまる。
- IP 変換の導入は、1～2dB の画質改善効果がある。

なお、図 5 で 15 フレームごとに SNR が下降する傾向が見られるのは、MPEG-2 の簡易復号画像が 15 フレームごとにイントラフレームであるため、再符号化前の比較対象になる画質が向上するためと考えられる。

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 450MHz 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困難となっているが、方法 3～方法 5 はリアルタイム処理可能である。これらの結果から、Tramepeg で用いている変換アルゴリズム方法 3 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であるといえる。

5. 変換システムの性能評価

図 6 に、入力ビットレートが 4.3, 6.0, 10.8Mbps の SP@ML, および 5.5, 12.0Mbps の MP@ML の MPEG-2 システムストリームに対する Tramepeg の変換速度の測定結果を示す。これにより、SP@ML 6.0Mbps の入力を SIF, 92kbps, 6frame/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 の場合より速いことがわかる。さらに、ネットワーク入力された MPEG-2 システムストリーム (TS) を Tramepeg によりリアルタイム変換し、Video: SIF, 92kbps, 6frame/s, Audio: 20kbps に低レート変換されたストリームがそのままサーバを介して配信できることを確認した。

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 4 秒)

変換レート	方法 1	方法 2	方法 3	方法 4	方法 5
460kbps, 10fr/s	171	35	3	3	3
92kbps, 6fr/s	103	21	3	3	3

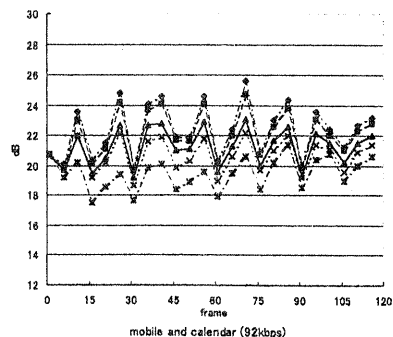
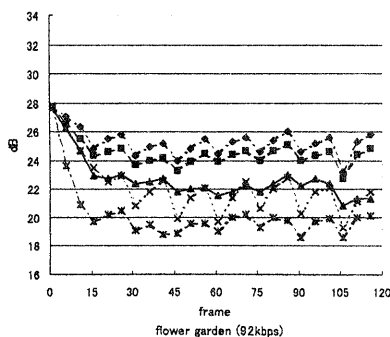


図 5 各処理方法による変換結果の画質の比較

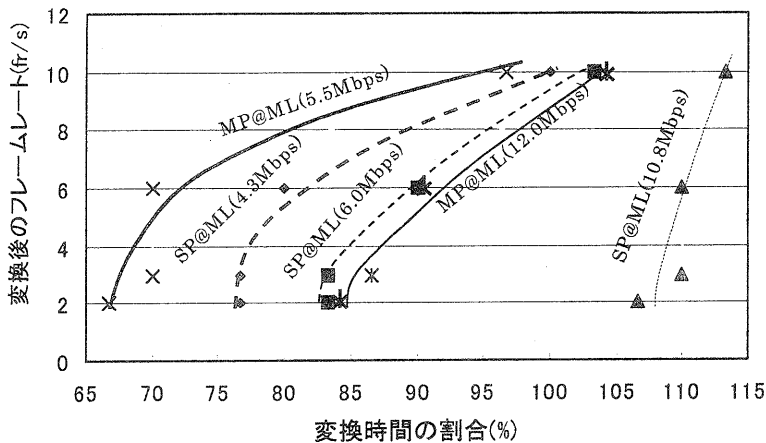


図 6 SP@ML および MP@ML の変換時間(PenII 450MHz×2)
(()内は MPEG-2 ストリームのビットレート)

6. まとめ

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。Trampeg では、DCT 空間でのダウンサンプリングと、MPEG-2 ストリーム中の動ベクトルを適応スケーリングする処理を組み合わせ、画質を維持しつつ高速処理を可能にした。これにより、一元管理されている 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る。本文で示した変換アルゴリズムは、H.263 や MPEG-4 といった動き補償と DCT を基本にした符号化方式への変換にも適用が可能であり、今後検討を進めたい。

参考文献

- [1] S. J. Wee, J. G. Apostolopoulos, and N. Feamster, "Field-to-Frame Transcoding with Spatial and Temporal Downsampling," ICIP, 1999.
- [2] B. Shen, I. K. Sethi, V. Bhaskaran, "Adaptive Motion Vector Resampling for Compressed Video Down-Scaling," ICIP, 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, Mar. 1999.
- [4] 笠井, 富永, 花村, 亀山: "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 Vol. J83-B-II NO.2, pp.151-164, 2000.
- [5] 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, Dec. 1998.
- [6] 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-September, 1998.
- [7] 阿久津, 清水, 八島, 小林: "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", 信学総大 D-11-147, 2000.
- [8] <http://sound.splab.ecl.ntt.co.jp/twinvq/j/>
- [9] 嵯峨田, 清水, 渡辺, 小寺: "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", 信学総大 D-11-87, 1997.
- [10] 清水, 嵯峨田, 渡辺, 小林: "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", PCSJ'97, pp.67-68.
- [11] 清水, 嵯峨田, 上倉, 小林: "量子化パラメータ情報削減による符号化性能の改善", 信学ソ大 D-11-6, 1998.
- [12] <http://www.netviker.com/>
- [13] 清水, 嵯峨田, 渡辺, 小寺: "フレームスキップによるビットストリームスケーリング", PCSJ'96, pp.25-26.
- [14] 清水, 嵯峨田, 渡辺, 小林: "参照フレーム変更によるビットストリームスケーリング", 信学論 Vol. J83-D-II NO.3, pp.967-976, 2000.
- [15] 阿久津, 清水, 八島, 小林: "MPEG-2 トランスコーディングの一検討", 画像電子学会'99 総会大会, pp.25-26, 1999.
- [16] ISO/IEC (MPEG-2), "Test model 5", April 1993.