

シームレスに映像切替え可能なトランスコーダ

桑原 直樹[†] 秦 淑彦[†] 野沢 俊治[†]

[†]三菱電機株式会社 先端技術総合研究所 〒661-8661 兵庫県尼崎市塚口本町 8-1-1

E-mail: [†]{Kuwahara.Naoki, Hata.Toshihiko, Nozawa.Toshiharu}@wrc.melco.co.jp

あらまし 多数のカメラの映像を高品質に符号化し高速ネットワークで配信する多カメラ監視システムにおいて、特別な計算能力を持たないパソコン上で複数映像を同時に表示する、或いはゲートウェイ接続された低速ネットワークに映像を中継配信する場合、トランスコーダにより低ビットレート映像に変換して配信する。監視では異常発生時に、瞬時にカメラを切替える、視覚的に途切れずに映像品質を変えることが重要であり、我々はこのようなシームレスな切替えが可能なトランスコーダを開発した。本稿ではその実現方式と RTSP を拡張した配信制御プロトコルについて述べる。プロトタイプによる評価の結果、従来よりも応答性良く切替えられることが確認できた。

キーワード 映像切替, 遠隔監視, トランスコーダ, RTSP

A video transcoder with seamless switching function

Naoki KUWAHARA[†] Toshihiko HATA[†] and Toshiharu NOZAWA[†]

[†]Mitsubishi Electric Corporation Advanced Technology R&D Center

E-mail: [†]{Kuwahara.Naoki, Hata.Toshihiko, Nozawa.Toshiharu}@wrc.melco.co.jp

Abstract In the video surveillance system with multiple cameras, a transcoder is used to reduce the bitrate of the videos to deliver videos over narrowband networks and to display multiple videos concurrently at a low performance computer. For surveillance use, it is important to switch the videos quickly and without a break visually, that is seamlessly, on emergency. Then we developed the transcoder that can switch videos and their quality seamlessly. This paper presents the novel scheme and the extension of RTSP for controlling the transcoder. The result of experiment showed effectiveness of the method.

Keyword Video switching, Remote monitoring, Transcoder, RTSP

1. はじめに

道路・河川や大規模施設における映像監視システムでは、数百台のカメラを設置し、IP(Internet Protocol)網やモバイル網を使って、いつでもどこでもどんな状況を確認できる多カメラ監視システムの導入が始まっている。このようなシステムを実現する一つのキーコンポーネントとして、高画質で符号化された監視映像を低ビットレートへと符号化変換して再配信するトランスコーダが提案されている[1]。トランスコーダは、低速回線網での映像閲覧を可能とするだけでなく、ネットワークや表示端末に過度の負担をかけずに複数カメラ映像を同時表示する上でも不可欠な技術となっている。

多カメラ監視システムでは全ての映像を同時に見ることではできないため、複数のカメラを定期的あるいはユーザ要求に応じて切替えて表示する。異常検知の際、対応するカメラに瞬時に切替える、映像から目を

離すことなく高品質な映像に切替える、といった迅速かつ視覚的に途切れのないシームレスな映像切替えが要求される。

トランスコーダのような中継機器での映像切替えは[2]において報告されているが、これはタイムスケジュールに沿って映像を切替えるものであり、監視のように突発的に切替え要求が生じるようなアプリケーションには対応できない。そこで我々は、多カメラ監視システムにおいてカメラおよび画質をシームレスに切替える機能を備えたトランスコーダを提案する。

本稿では、まず多カメラ監視システムおよび監視向けトランスコーダの概要について説明し、シームレス切替え機能の実現方法を述べる。続いて RTSP(Real Time Streaming Protocol)を拡張した本トランスコーダの制御プロトコルについて述べ、プロトタイプによる評価結果を示す。

2. 多カメラ監視システム

2.1. システム構成

図 1 にトランスコーダを用いた多カメラ監視システムの構成例を示す。数十～数百台規模のカメラが現場設置され、この映像を符号化して配信するエンコーダおよびこの符号化映像を蓄積し配信する映像サーバが光ファイバ等の高速ネットワークにより繋がれている。監視員は通常の監視業務では監視センタにある表示パソコンで、高速ネットワークを通じて受信した映像をモニタリングする。多数映像を同時に表示する場合には、高ビットレートである監視映像をトランスコーダで低ビットレートに変換して表示する。異常を発見した場合には図 2 のように視覚的に途切れずにカメラや画質を切替えて表示する。一方、トランスコーダにより低ビットレートへ変換した監視映像を、インターネットやモバイル網を介して個人端末に表示することで外出時や在宅時にも現場状況を確認できる。

2.2. 監視向けトランスコーダ

2.2.1. 内部構成

図 3 にトランスコーダの基本的な処理ブロック図を示す。トランスコーダは、(1)受信部、(2)分離部、(3)変換部、(4)多重化部、(5)送信部、から構成される。入力ストリームは映像あるいは音声が入力されたストリームである。変換部は復号部と符号化部とから構成され、入力と出力のビットストリームの符号

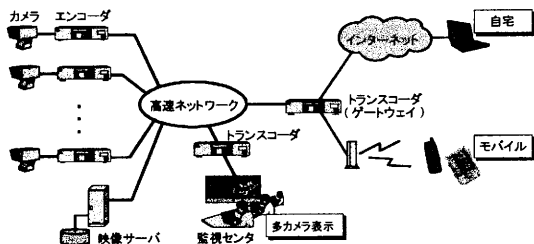


図 1: 多カメラ監視システム構成例

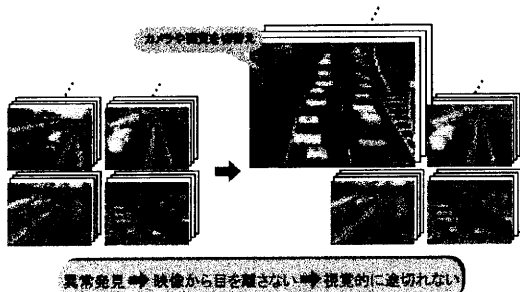


図 2: 多カメラ映像の表示例

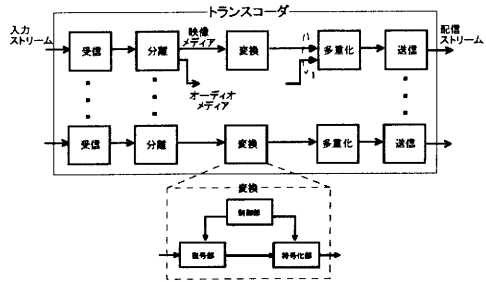


図 3: 処理ブロック図

化フォーマットによってその組合せが決まる。一般に、復号部と符号化部では画像を完全復号せずに途中の処理を省略し高速化を図っている[4][5]。

2.2.2. 符号化変換

低ビットレートへ変換するパラメータには様々なものがあり、これらをコンテンツや用途に応じて使い分ける。変換パラメータを表 1 に示す。

表 1: 変換パラメータ

変換パラメータ	概要
符号化フォーマット	コンテンツや用途に適した符号化フォーマットへの変換を行う 例) 変化の少ない映像は JPEG へ変換 動きを見る場合は MPEG-4 へ変換 表示端末が再生可能なフォーマットへの変換を目的に使用する場合もある
解像度変換	画素数を削減しビットレートを低減する 精細さよりも動きを重視する場合に利用
再量子化	量子化幅を拡大して再量子化する
フレーム間引き	フレームレートを低減させる 動きの少ないコンテンツに適した変換
ROI (Region of Interest)	
ROI の位置、大きさ	注目領域 (ROI) だけを高画質にする、或いは注目領域だけ切出すといった変換を行う。左記変換パラメータを領域毎に独立に設定できる。
縮尺率	
画質 (Q 値等)	
フレームレート	

3. シームレス切替え実現方式

トランスコーダでは、(1)カメラ、(2)符号化変換パラメータ、の 2 つの切替えをシームレスに行える必要がある。これらは異常時での切替えのほか、(1)は時分割でのカメラ切替えに、(2)は出力伝送網の状況に応じたレート制御においても有効な機能である。

3.1. カメラ切替え

従来のカメラ切替えは、表示中のカメラに対する制御コネクションを切断し、次のカメラに対する制御コネクションを確立するという処理を連続して行うことにより実現されていた。この手法では切替え処理の間、映像が停止する、映像切替わりの応答性が悪い、という問題があった。

そこで本トランスコーダでは、コネクションの切断

と再確立からなるカメラ切替えではなく、同じコネクションにおける「カメラ切替え」という機能を導入した。カメラ切替えの具体的な処理手順を図 4 に示す。カメラ切替えが要求されると、切替え先カメラのエンコーダ (ENC#n) とのコネクションをまず確立し、トランスコーダ内でカメラ切替えを行ってから前のエンコーダ (ENC#m) とのコネクションを切断する。カメラを切替えるタイミングは、切替え先の映像がフレーム内符号化されている場合には最初の画像フレームを受け取ったときに行い、フレーム間符号化されている場合には、次のイントラピクチャが検出されるまで待つてから切替えを行う。これにより映像が停止することなく連続性を持ってカメラを切替えることができる。一方、本手法では応答性が改善する確率も高くなる(図 5)。従来の処理時間(図内 t_1)と比べて、提案手法の切替え処理時間(同 t_2)は短くなるため、 $t_1 - t_2$ の区間にイントラピクチャ(同 X)が到着した場合には応答性が高まる。

システム基準時刻のリセット

MPEG-2 システム等では、エンコーダが発行するクロック情報(システム基準時刻)に基づき再現するクロックと、ビットストリーム内の画像フレーム毎に付

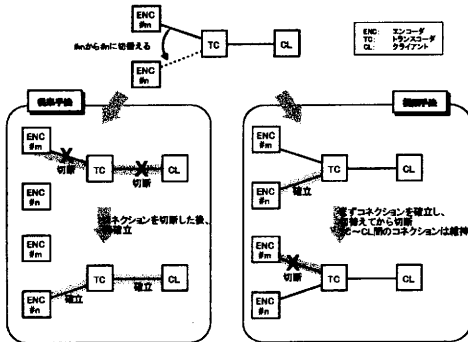


図 4: カメラ切替え方式

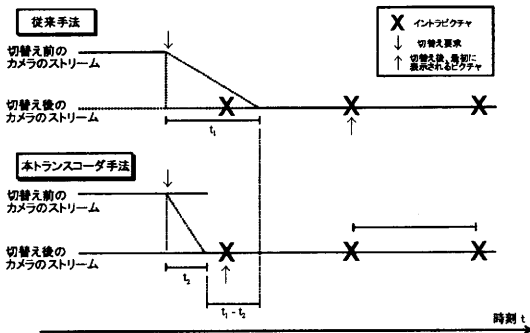


図 5: 従来と提案手法のカメラ切替えタイミング

与されたタイムスタンプとをデコーダが照合して適切なタイミングで映像を復号、表示する。

エンコーダが変わると異なるシステム基準時刻で符号化されるので通常はカメラ切替え前後ではタイムスタンプがかけ離れてデコーダが再生できずに映像が乱れるという症状が生じる。そこで例えば MPEG-2 PS に対してはビットストリーム内の切替え箇所にバックヘッダを挿入することで、デコーダのシステム基準時刻をリセットするようにする。

3.2. 符号化変換パラメータ切替え

3.2.1. パラメータ設定変更

本トランスコーダでは、表 1 に示した符号化変換パラメータをフレーム単位あるいはフレーム間の参照関係がリセットされる GoP(Group of Pictures)単位で設定変更できるよう構成した[1]。このため、遅くとも次のイントラフレームでは新しいパラメータ設定での符号化変換を開始できる。

3.2.2. 設定変更通知

符号化変換パラメータのうち、少なくとも符号化フォーマットが変更された場合はその変更をビットストリームに同期させて通知しないとクライアントでは映像をデコードできない。これを実現する一手法としてストリーム配信の packets にその通知を埋め込む方法がある。我々のプロトタイプではストリーム配信プロトコルに RTP[7](Real time Transport Protocol)を用いており、RTP ヘッダの PT(Payload Type)フィールドを使って伝達することとした。クライアント側ではこの PT フィールドの変化を検知してデコーダを切替える。

3.2.3. クライアント側準備

クライアント側では、上記通知(プロトタイプでは PT フィールド)で変更を検知してからデコーダの準備をしていてはプロセスの起動や初期化処理のために、映像が停止あるいは途切れてしまう。途切れなく表示を続けるためにクライアント側では必要な種類のデコーダを予め起動しておき、適切なデコーダを選択して使用する。

4. 配信制御プロトコル

4.1. 本システムにおける配信制御プロトコル

図 6 にシステム全体の配信制御プロトコルを示す。本システムではクライアントがトランスコーダに対しストリームの制御要求を出し、トランスコーダが該当するエンコーダ又は映像サーバへ制御要求を出すという仕組みになっている。制御メッセージには標準的な実時間データの配信制御プロトコルである RTSP[8]を用いるが、クライアント〜トランスコーダ間はトランスコーディングやシームレス切替えという RTSP にな

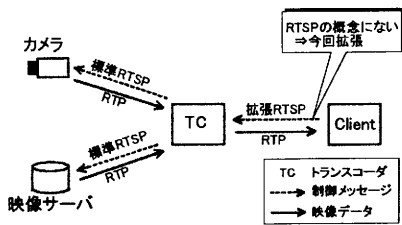


図 6：配信制御プロトコルの関係

い概念を含むため、RTSP の機能拡張を行った。

4.2. トランスコーダのモデル化

RTSP の拡張について述べる前にまずトランスコーダ機能をモデル化する。

4.2.1. 入出力モデル

想定するトランスコーダの入力と出カストリームとの関係を図 7 に示す。トランスコーダでは複数ストリームの同時入力、変換、出力を許し、表 2 のような入出力モデルを考える。

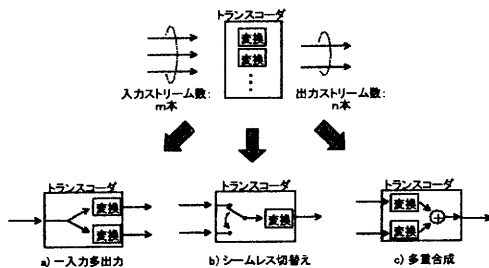


図 7：入出力モデル

表 2：入出力モデル

一入力多出力	一つの入カストリームを、異なるパラメータで変換・配信
シームレス切替え	セッションを切ることなく、映像・音声ソースを切替え
多重合成	複数ストリームを結合して、一つのストリームとして配信

4.2.2. 変換モデル

表 1 に示した変換パラメータを使用する。複数ストリームを扱う場合には、ストリーム毎にパラメータ設定でき、ROI 変換の場合は領域毎に画質設定ができるものとする。またセッションの途中でこれら符号化パラメータを切替え可能とする。

4.2.3. ストリーム制御モデル

ストリーム制御機能としては再生、逆再生、およびこれらの再生速度の指定、シーク、一時停止が可能であるものとする。

4.3. 拡張 RTSP 定義

上記モデルに基づき、トランスコーディングとシームレス切替え、映像と音声の多重化を実現するよう RTSP の拡張を検討した。

4.3.1. URL

本システムではシームレスカメラ切替えにより、同一 RTSP セッション中にソースが切替わることがあるため、URL には標準の RTSP とは異なり、アクセス先のソースを記述しないこととした。アクセス先のソースは後述する ANNOUNCE メソッドを使って指定する。セッションの識別はセッション ID でのみ行うこととした。なお URL にはアクセスするトランスコーダ名を記述する。

【URL】 `rtsp://transcoder`
`transcoder = <トランスコーダ名>`

4.3.2. メソッド

拡張 RTSP で使用するメソッドの一覧を表 3 に示す。

表 3：RTSP メソッド一覧

メソッド	概要
ANNOUNCE	セッション内に含めるストリームの指定
SETUP	セッション作成、送信先の指定
SET_PARAMETER	符号化変換パラメータの設定、変更、シーク
PLAY	再生、再生速度・方向の指定
PAUSE	一時停止
TEARDOWN	セッション終了

図 8 のようにメソッドとトランスコーダの内部処理には対応関係がある。例えば、SET_PARAMETER は符号化変換パラメータの指定にのみ使用する。SETUP は送信先の記述に使用する。また ANNOUNCE メソッドは、セッションで制御する映像・音声ソース群を指定する際に使用する。一方、PLAY、PAUSE は特定の処理ブロックに関するものではなく、ストリームの状態を制御するものである。

4.3.3. 配信ストリームの定義

入力ソースは、ANNOUNCE メソッドのメッセージボディに SDP (Session Description Protocol) [9] の "a=control:" 属性タグを使って記述することにより定義す

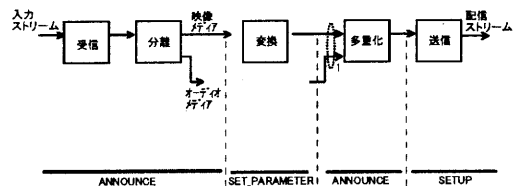


図 8：メソッドと処理ブロックの関係

る。複数のソースをセッション内に多重化する場合、複数の映像・音声ソースを列記する。この際、ソース毎にソース ID を割り振る。ANNOUNCE メソッドのメッセージ例をリスト 1 に示す。

一方、送信先は標準の RTSP と同様 SETUP の Transport ヘッダに記述する。マルチキャストの場合など、送信先とクライアントのアドレスは一致する必要はない。

4.3.4. トランスコーディング

SET_PARAMETER メッセージのボディに独自タグを用いて記述する。複数のソースが多重化されている

```
ANNOUNCE rtsp://transcoder1.melco.co.jp
Cseq:1
Session:12345678
Content-type:text/parameters

num_of_source:2 /* 入力ソースの数 */
source:1 /* 識別番号1のソース */
a=control:rtsp://yodogawa.go.jp/live/1 /* カメラ番号1のライブ映像 */

source:2 /* 識別番号2のソース */
a=control:rtsp://video_server.go.jp/movie/7 /* イベント番号7の蓄積映像 */

source:3 /* 識別番号3のソース */
a=control:rtsp://video_server.go.jp/audio/7 /* イベント番号7の蓄積音声 */
```

リスト 1: ANNOUNCE メソッドメッセージ例

場合はソース ID 毎にパラメータを記述する。また ROI 変換する場合には領域毎に画質指定する。設定例をリスト 2 に示す。なお、パラメータ設定を省略すると、デフォルト値が適用される。

4.3.5. シームレスカメラ切替え

```
SET_PARAMETER rtsp://transcoder1.melco.co.jp
Cseq:3
Session:12345678
Content-type:text/parameters

num_of_source:2 /* 入力ソース数 */
source:1 /* 識別番号1のソース */
format:MPEG4 /* 出力フォーマットはMPEG4 */
scaleX:1/2 /* 水平解像度を1/2に変換 */
scaleY:1/2 /* 垂直解像度を1/2に変換 */
framerate:10 /* 10フレーム/秒 */

source:2 /* 識別番号2のソース */
num_of_region:2 /* ROI変換を実施。ROI領域の数 */
regionID:1 /* 1つめの領域に対する設定 */
rectangle: /* 長方形のROI */
coordinate:0,0 /* 1つめの領域に対する座標情報 */
format:MJPEG /* 出力フォーマットはMJPEG */
framerate:2 /* 2フレーム/秒 */

regionID:2 /* 2つめの領域に対する設定 */
polygon:4 /* 四角形のROI */
coordinate:10,10 /* 2つめの領域に対する座標情報 */
coordinate:10,300
coordinate:200,300
coordinate:200,10
format:MJPEG /* 出力フォーマットはMJPEG */
framerate:15 /* 15フレーム/秒 */
```

リスト 2: SET_PARAMETER メッセージ例

新たな映像・音声ソースを定義した ANNOUNCE メソッドを再発行することにより行う。ストリーム数を減らす場合と区別するため、一部のみの変更であっても全てのストリームに関する属性タグを記述する。なおカメラ切替えの場合には、メッセージのヘッダ部分にセッション番号を記述する（リスト 1 参照）。

4.3.6. シーケンス例

本システムにおけるシーケンス例を図 9 に示す。まず ANNOUNCE メソッドを発行し、それから SETUP メソッドを発行する。これらはセッション対応付けのため、同一 TCP コネクションにて行うこととした。SETUP 後は SETUP 以外のメソッドを任意のタイミングで発行できる。

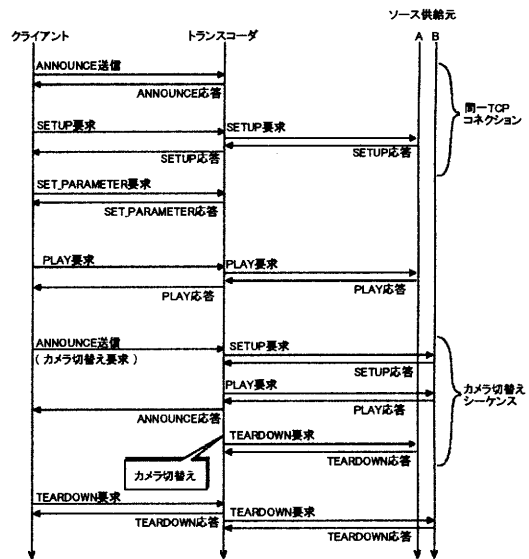


図 9: メッセージシーケンス例

5. 実験・評価

5.1. 測定項目

提案方式の有効性を評価するため、従来方法と本トランスコーダのプロトタイプそれぞれにおいて、(1) カメラ切替え、(2) 符号化変換パラメータの切替え、にかかる時間をそれぞれ測定した。測定に用いた機器を表 4 に示す。カメラ切替えに関しては映像サーバからエンコーダへの切替えにかかる時間を、符号化切替えに関しては MPEG-4 から MPEG-2 への符号化フォーマット切替えにかかる時間をそれぞれ 3~5 回測定し、その平均を求めた。いずれの測定においてもクライアント側で測定を行い、切替え要求を出してから、切替え

後の映像が表示されるまでの時間を Windows のシステムコール PerformanceCounter により測定した。なお、従来方式に対する測定では、TEARDOWN→SETUP→SET_PARAMETER→PLAY の処理を自動的に連続実行させて、その所要時間を測定した。なお、MPEG-2 データの GoP 間隔は 0.5 秒とした。

表 4：測定環境

エンコーダ	三菱電機製 MPEG-2 エンコーダ MPE
映像サーバ	Pentium4-1.6GHz, RAM-128MB, Win2000
トランスコーダ	Pentium4-2.0GHz, RAM-1GB, Win2000
クライアント	Pentium4-2.0GHz, RAM-512MB, WinXP, MPEG-2 ハードウェアデコーダボード
ネットワーク	Fast Ethernet Switching Hub

5.2. 結果

測定結果を表 5 に示す。測定結果からカメラ切替えの所要時間が飛躍的に短縮されたことがわかる。本測定結果には最大 GoP 間隔の誤差があるものの、提案方式では数百 ms 以内でカメラ切替えを実現できている。一方、符号化変換パラメータの切替えに関しては、所要時間は半分以下に短縮されたものの依然として 2 秒以上要している。詳細な測定が必要だが、切替え要求を出した後、切替え後の映像データが到着するまでは 200ms 程度に高速化されており、デコーダ内で時間を費やしている可能性が高い。

一方視覚的には、カメラ切替えの場合には途切れることなく映像が切替わることを確認できたが、符号化変換切替えの場合には、切替え直後に一瞬、デコーダ内バッファに残る前回画像が表示されるという症状が見られた。バッファをリセットしなかったために起こった症状であるが、リセットしたとしても、バッファがある程度充填されるまで何も表示されないことが予想され、映像が途切れてしまう可能性がある。

以上の結果から、符号化変換パラメータをシームレスに切替えるには、デコーダの選定も重要であることがわかった。特に前段に特別なバッファを必要としないソフトウェアデコーダの使用が有望と考えられる。

表 5：測定結果

	従来方式	提案方式
カメラ切替え	4531	223
符号化変換切替え	4531	2215

(ms)

6. まとめ

本稿では、シームレス切替え可能なトランスコーダについて述べた。シームレスなカメラ切替えと符号化変換切替えの実現方法およびその配信制御プロトコル

を示した。プロトタイプを用いての測定結果により提案方式の有効性を示した。今回の報告では、アクセス可能なソースや切替え後の符号化フォーマット等、クライアント側が予めこれらの知識を持っている場合を想定したが、これらの知識を持たない場合にトランスコーダ側からクライアントへ情報を通知するような仕組みについても今後は検討する予定である。

文 献

- [1] 秦, 桑原, 加藤, 関口, Vetro, “映像トランスコーダ”, 三菱電機技報, Vol.78 No.5, pp.34-38, May 2004
- [2] 今泉, 金子, 青木, 井口, 藤田, “ストリーミング放送における中継サーバ上での映像切替・合成制御方式”, 情報処理学会全国大会講演論文集, pp.5.101-5.104, March 2004.
- [3] 桑原, 秦, 野沢, 坪井, “マップトップ映像監視システム”, 電気学会道路交通研究会資料, RTA-01-36, pp.39-44, Dec. 2001.
- [4] 西村, 笠井, 高屋, 亀山, 榊, 花村, 富永, “リアルタイム MPEG-2 ビデオトランスコーダソフトウェアの開発”, 情報研報, AVM25-5, pp.25-30, 1999.
- [5] Vetro, Hata, Kuwahara, Kalva, “Complexity-Quality Analysis of MPEG-2 to MPEG-4 Transcoding Architectures”, IEEE International Conference on Consumer Electronics (ICCE), pp.130-131, June 2002.
- [6] 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, no.3, pp.967-976, March 2000
- [7] Schulzrinne, Casner, Frederick and Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, RFC1889, IETF, Jan. 1996
- [8] Schulzrinne, Rao and Lanphier, “Real Time Streaming Protocol (RTSP)”, RFC2326, IETF, April 1998
- [9] Handley and Jacobson, “SDP: Session description protocol”, RFC2327, IETF, April 1998