

スプライトを用いた MPEG-4 超高压縮符号化方式：VideoESPER

秦泉寺久美**，岡田重樹*，小林直樹*，渡辺裕+，米原紀子**

* 日本電信電話株式会社 NTT サイバースペース研究所

+ 早稲田大学国際情報通信研究センター

** NTTソフトウェア株式会社

あらまし

MPEG-4 では従来符号化法にはない，強力な符号化ツールが具備されている．筆者らはスプライトという符号化ツールに着目し，動画像をあらかじめ前景背景に分離し，前景にオブジェクト符号化，背景をスプライト符号化するスプライトモード[2]の開発を行ってきた．しかし，すべての動画像にスプライトモードが適応できるとは限らなかった．筆者らは，スプライト符号化を用いた汎用的な MPEG-4 動画像符号化システムである“VideoESPER (Efficient SPrte aided EncodeR)”を開発した．スプライトモードが優位なショットを自動的に抽出し，そのショットにはスプライトモードを適用し，そのほかのショットに対しては従来符号化法を踏襲するノーマルモードを適用するマルチモード符号化法を提案する．さらに，AVI データ等を逐次的に処理するための逐次処理方法，グローバルな動きの算出法，前景領域抽出法，スプライト生成法に改良を行った．また，符号化においては，前景背景に自動的に符号量を配分するためのマルチビデオオブジェクト(MVO: Multi Video Object)レート制御法を提案し，符号化効率を上げるために，スプライトに幾何変換を行う中間解像度変換法，スプライト回転法を提案する．また，実際に符号化実験を行って，従来符号化方式に対する VideoESPER の優位性を示した．一方で，VideoESPER における限界についても述べる．

VideoESPER: MPEG-4 Very Efficient Video Coding Scheme using Sprite

Kumi Jinzenji* Shigeki Okada*，Naoki Kobayashi*，Hiwoshi Watanabe+，Noriko Yonehara**

* NTT Cyber Space Laboratories, NTT Corporation

+Global Institute of Telecommunication and Information, Waseda University

**NTT Software Corporation

Abstract

MPEG-4, a new audio-visual coding standard has strong compression tools, such as sprite coding. We have focused on sprite coding and proposed “Sprite Mode”. In sprite mode, video sequence is divided into foreground object and background object. Foreground object is compressed using MPEG-4 object coding, while background object is compressed using sprite coding. Sprite mode does not always adopt any video sequence such that images having still camera motion can be compressed in enough quality in low bit-rate. So, in this paper, we developed new MPEG-4 video encoder using sprite coding, VideoESPER, which means Video Efficient Sprte aided EncodeR. In VideoESPER, multi-mode coding is newly proposed. Video shot suitable for conventional coding method is coded by normal mode (conventional method), while video shot suitable for sprite coding is coded by sprite mode. These two modes are automatic selected in any video sequences. We improved the processing flow using stuck system in order to consequently process video such as AVI format. We improved global motion detection algorithm, foreground object and background sprite generation algorithm. Then we proposed the simple rate-control for sprite mode and geometric sprite transforming method for more efficient sprite coding. Then, we have some experiment on MPEG-4 video coding. The result shows effectiveness of VideoESPER in low bit-rate coding. On the other hand, limitation of VideoESPER is also discussed.

1. はじめに

1999年3月に次世代オーディオビジュアル符号化方式である MPEG-4[1]が標準化された。MPEG-4 Visual においては、従来とは異なる符号化ツールが採用され、そのなかでも任意形状のビデオオブジェクト符号化は従来の符号化方式とは決定的に異なる方式であり、初期の MPEG-4 を特徴づけるものであった。筆者らは、この中のスプライトという符号化ツールに着目し、動画像をあらかじめ前景背景に分離し、前景にオブジェクト符号化、背景をスプライト符号化するスプライトモード[2]の開発を行ってきた。そして、前景領域が画像全体に対してある一定の比率以下の時に劇的な圧縮効率をもたらすことを示した。しかし、一方でこのことは、すべての動画像にスプライトモードが適応できるとは限らないことを示したものであった。また、これがゆえに汎用的な符号化システム構築までには至らなかった。また、文献[2]の方法では、以下の点で汎用的符号化器を構成する上で、特にオブジェクトの分離に大きな問題があった。ひとつは、スプライトを生成するのに必要であるグローバルな動き (GMV: Global Motion Vector) を、局所的な動きベクトル(LMV: Local Motion Vector)の特徴空間に分布の集中度から求めるボトムアップ的手法[2]で算出しているため、信頼性の低い LMV が多数算出される動画像ではカメラモーションを反映した GMV を算出することができない。また、前景領域の分離においては、あらかじめ前景比率を与える必要があったため、汎用性に欠いた方法であった。そこで、本稿では前景比率を与えないで算出する方法を提案する。スプライトの生成においては、前景領域を切り出した後の画像を背景画像とし、これを順次貼り合わせて(上書きして)スプライトを生成した。しかし、この方法では、前景が誤って抽出されなかった場合、スプライトに前景が残ってしまうという問題があった。

本稿では、文献[2]での知見を生かし、スプライト符号化を用いた汎用的な MPEG-4 動画像符号化システムである "VideoESPER (Efficient SPrte aided EncodeR)" を開発した。VideoESPER では、以下の新規な手法を提案する。スプライトモードが優位なショットを自動的に抽出し、そのショットにはスプライトモードを適用し、そのほかのショットに対しては従来符号化法を踏襲するノーマルモードを適用するマルチモード符号化法[3]を提案する。さらに、AVI データ等

を逐次的に処理するための逐次処理方法を提案する。また、文献[2]に比較して、高精度にオブジェクト分離を行うための、グローバルな動き (GMV: Global Motion Vector) の算出法と、前景動領域抽出、スプライト生成法に関して改良を行う。また、符号化においては、前景背景に自動的に符号量を配分するためのマルチビデオオブジェクト (MVO: Multi Video Object) レート制御法を提案し、符号化効率を上げるために、スプライトの縮小方式である中間解像度変換法[4]、スプライト反転方法を提案する。また、本稿では、実際に符号化実験を行って、従来符号化方式に対する VideoESPER の優位性を示すと共に、VideoESPER における限界についても述べる。

以下、2章では VideoESPER の概念、3章では VideoESPER のシステム構成、4章では改良を行った画像処理ルーチンの詳細、5章では符号化の詳細、6章では符号化実験による VideoESPER の評価、7章では結論を述べる。

2. VideoESPER の概念

2.1 MPEG-4 スプライト符号化

MPEG-4 以前の映像符号化では、フレーム毎に動き補償 (Motion Compensation: MC) を行って符号化を行っていた。スプライトモード[2]は、あらかじめ前景と背景を自動的に分離し、前景を MPEG-4 オブジェクト符号化、背景をスプライト符号化するものである。オブジェクト符号化では、任意形状のオブジェクトをテクスチャ、動き、形状によって表現できるという MPEG-4 で初めて標準採用されたツールである。スプライト符号化とは、一枚の静止画像からグローバルモーションによって該当する領域 (ここでは背景領域) を切り出してくるもので、従来動画像で表していた背景部分を静止画とフレーム分の GMV で表現することから、非常に高い圧縮効果が期待できる。

2.2 マルチモード符号化

スプライトモードが効果的に働く映像の条件として、(1)前景領域が画面全体に対して十分小さい領域であること[2]、(2)カメラモーションが十分大きい状態で撮影されていること [3]があげられる。ほとんどカメラモーションに変化が無い場合は、Not Coded MB が多数出現しやすいなどのことより、従来方式による符号化でも十分な画質を得ることが可能である。筆者らはスプライト符号化が効果的である映像ショットにはスプライトモードを、そうではないショットに対しては、ノーマルモード (従来法符号化方式を踏襲した MPEG-4 符号化法。) を自動で判定して符号化する

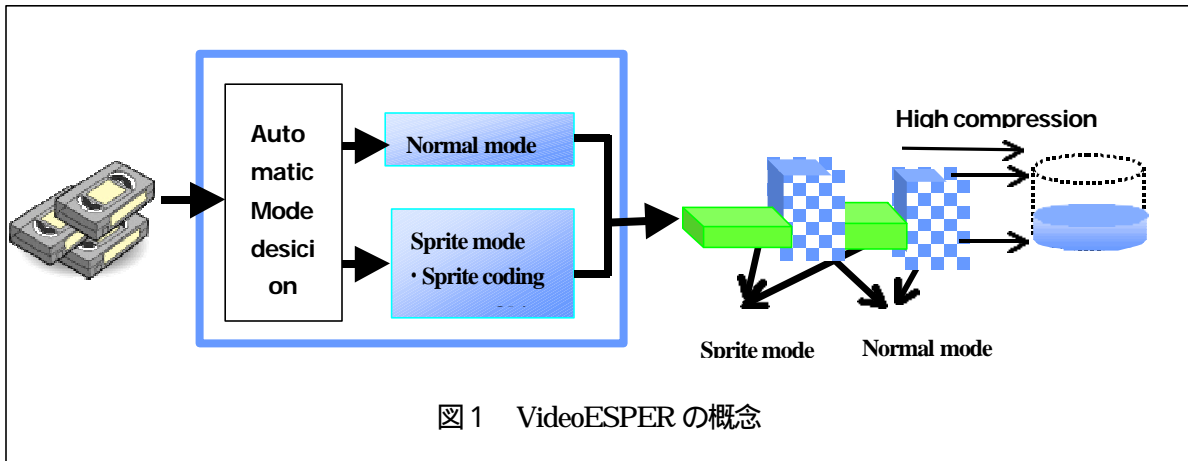


図1 VideoESPER の概念

るマルチモード符号化方式を提案する．図1にマルチモード符号化法の概念を示す．入力された映像は，符号化モード判定に基づき該当するビデオオブジェクトを算出し，ビデオオブジェクトが符号化される．スプライトモードに適している映像は，スポーツ映像や景観映像のように速いカメラの動きがあり，前景領域が十分小さい映像であることが予想される．

3 . VideoESPER のシステム構成

文献[3]では逐次処理が不可能な処理フローであった．よって AVI のように連続して入力される画像で処理可能にするために，スタック処理を取り入れる．図2に VideoESPER のシステム構成を示す．スタック処理部，ショット内処理部，符号化部，配信部の4つの構成をとる．画像処理部はスタック処理部とショット内処理部からなる．図4にスタック処理部の処理フロー，図5にショット内処理部のフローを示す．スタック処理部では，ある時間分の画像をスタックにいれ，逐次モードの判定を行い，判定済みのショットをスタックアウトして，ショット内処理部に画像データ等を受け渡す．ショット内処理部では，必要なビデオオブジェクトデータを生成し，符号化部へ受け渡す．これを繰り返して逐次処理を可能にする．

3 . 1 システム構成と逐次処理のためのスタック処理

スタック処理部における簡単な流れを以下に示す．

(1) T秒毎に区切り，ユニットに分割する．次のt秒($t < T$)以内にカット点がある場合は，ユニットの長さを $t + T$ 秒に延長し，カット点フラグを立てる．前ユニットがカット点で終わっていて，かつ，t ($t < T$)秒以内にカット点がある場合は，カット点

をまたぐT秒を強制Nショットとし，カット点フラグも立てる．

(2) n ユニットになったところで処理を停止する．
 (3) n ユニットにおいて，順方向 MV，順方向隣接フレーム間 GMV を算出する．
 (4) n ユニートをユニット毎に GMV による 1 次モード判定する．

(5) スタック内でカット点フラグが無く，S ショットに囲まれたNショットをSショットとする状態遷移を行う．ただし，強制Nショット(カット点を含むユニット)はそのままにする．

(6) S 候補において，ユニット毎にメディアンスプライト生成，前景抽出を行い，前景比率による2次モード判定を行う．

(8) 孤立するNおよびSショットをなくすために，判定結果に対し，一部 S->N への状態遷移を行う．
 (9) 1 ~ 4 ユニートをショット処理部へ送り，スタックをあける．

(10) スタックに空いた分のユニットを挿入する．
 また，以下(11)~(12)にショット内処理部における処理を記す．

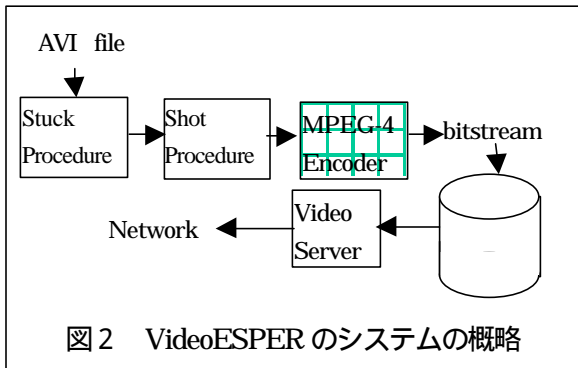
(11) S ショットは，任意の基準フレーム選定し，逆方向 LMV，逆方向隣接フレーム間 GMV を算出，スプライト生成を行う．

(12) 符号化部にてスプライトモードもしくはノーマルモードにて符号化を行う．

スプライトモードにおける前景オブジェクトはスタック処理部で算出され，背景スプライトはスタックアウトされた後の各ショットにおいて算出される．

3 . 2 GMV による 1 次モード判定

ユニット内にて，画像の中心座標のトラジェクトリを調べる．これは，いわゆる GMV における pan および tilt の値に相当する．トラジェクトリの大きさによってカメラの pan, tilt の有無を調べ、



閾値によって判定を行う。調べ方は以下のとおりである。

(1)各フレームにおいて、先頭フレーム対するGMVを算出する。

(2)各フレームにおける中心座標の値(S_{xi}, S_{yi})を算出する。

(3)各々の最大値、最小値 $SX_{max}, SY_{max}, SX_{min}, SY_{min}$ を算出し、その絶対値を算出する

$$SX = \text{abs}(SX_{max} - SX_{min});$$

$$SY = \text{abs}(SY_{max} - SY_{min});$$

(4)if($SX \geq TH \ || \ SY \geq TH$) mode=S; else mode=N;

ここで、THは以下の値をとる。

$$TH = \text{frames} * 2.0;$$

3.3 前景比率による2次モード判定

ユニット毎に前景オブジェクト(前景抽出法の詳細は次章で述べる。)および前景率を算出する。前景比率 FG_ratio は以下の式で表される。

$$FG_ratio = fg_size / frame_size$$

fg_size は前景の大きさ(単位:画素), $frame_size$ フレームの大きさ(単位:画素)である。さらに、以下のように前景比率 FG_ratio を用いて判定する。

(1)ユニット内の平均前景率 AVE_fg_ratio を算出。

(2) FG_ratio が0.2を超えるフレーム数をカウント(count)。

(3)以下の判定法で判定する。

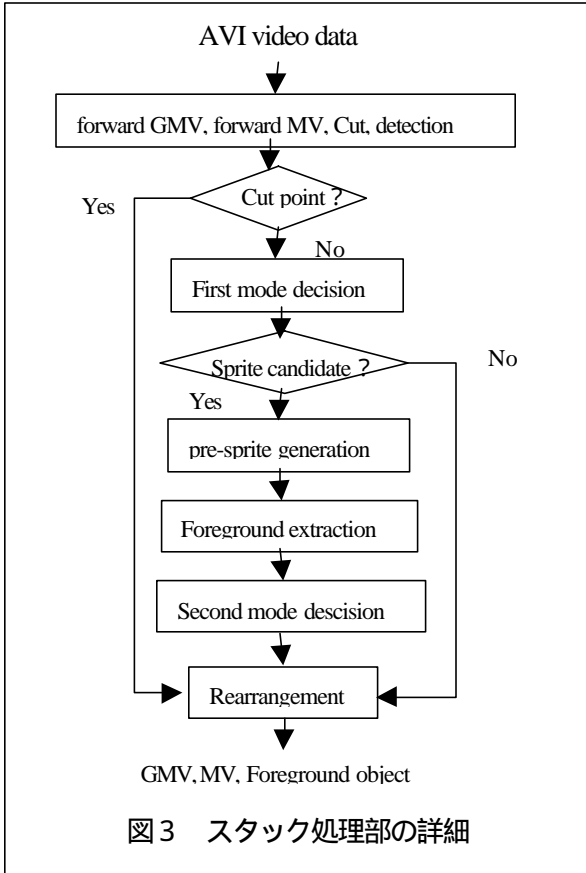
$$\text{if}(AVE_fg_ratio > 0.17) \text{ mode} = N;$$

$$\text{else if}(\text{count} \geq \text{frames} * 0.2) \text{ mode} = N;$$

$$\text{else mode} = S;$$

3.4 状態遷移およびスタッククリア

Sショットの一部をNショットに変更し、単独でSおよびNショットが存在しないようにする。その後、連続するSもしくはNショットをクリアし、ショット内処理部でビデオオブジェクトの生



成ならびに符号化が行われる。また、クリアされたスタックには新たにフレームが挿入される。

4. 画像処理の詳細

4.1 グローバルモーション算出

文献[2]の方法はLMVの特徴空間におけるクラスタリングからGMVを算出するボトムアップ手法である。よって、LMVの精度がGMVの精度に影響する。LMVは原画像と予測画像の差分がもっとも小さくなるものが採用される。テクスチャがあまり無い、フィールドにおける「芝」や「空」同じようなテクスチャの繰り返しになる「白線」のところはLMVが必ずしも正確な動きを反映するとは限らない。しかしながら、スプライトを生成するには、GMVがカメラの動きを正確に反映していることが必要である。一方、一般的なトップダウン手法[5]では画像全体の予測画像と原画像の予測誤差を最小化するGMVを算出する。本稿では、ボトムアップ手法とトップダウン手法を適応的に切り替える方法を新たに提案する。

(1)ボトムアップ手法[2]にて仮GMVを算出する。

(2)座標(i, j)における各MBにおいて、LMV(V_{x1}, V_{y1})に変換し、これと元のLMV(V_{x0}, V_{y0})との差分 $D(i, j)$ を算出する。差分 $D(i, j)$ は以下の式で与える。

$$D(i, j) = |Vx1 - Vx0| + |Vy1 - Vy0|$$

(3) $D(i, j) < Th_{GMV}$ の MB の個数をカウントする。

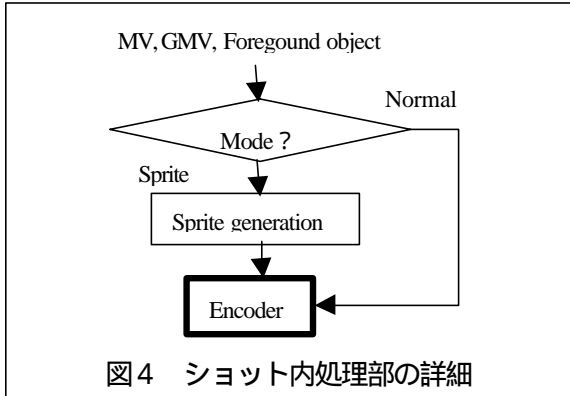


図4 ショット内処理部の詳細

(4) カウントが Th_{count} 以上である場合にトップダウン方式に切り替える。そのとき、直前フレームの GMV を中心に pan, tilt, zoom 方向に微小に変化させ、原画像と予測画像の差分を最小とする pan, tilt, zoom の組を GMV として採用する。

4.2 前景領域抽出

従来の動領域抽出技術は背景が固定である[1]、初期輪郭が必要であるなどの拘束条件があった。文献[2]の前景領域抽出法は、画面に対する前景の割合（前景比率）を与え、原画像とスプライトから切り出した背景画像との差分が大きいところから順番に指定した前景比率に達するまで選択していた。この方法は差分値の閾値を自動決定できるという利点があるが、あらかじめ前景比率を与える必要があり、前景の割合が未知である場合には適用できない。差分値の閾値と前景比率は比例の関係があるので、適当に差分閾値を決めれば前景の面積も決定されるが、差分閾値の与え方は一義的には決まらない。そこで、本稿では、MB 内の画素の分散、および、LMV と GMV の MB の座標における差分に着目し、フレーム内で差分閾値を動的に決定する方法を提案する。以下に前景抽出の概略を記す。

- (1) 差分閾値を MB 内の分散値が大きい領域と小さい領域用の 2 通りの閾値を設定する。基本はこの閾値にて前景 MB と背景 MB とに分類する。
- (2) さらに、MB 内の分散値が大きい領域で、かつ、LMV と GMV の差分が小さい MB は背景と MB とする。

ただし、この方法は前景領域を正確に切り出す必要がないという VideoESPER に特化した方法であり、画像の再利用などの用途には不向きであることを記しておく。

文献[2]のスプライト生成では、前景画像を切

り出した残りの部分（=背景画像）をそのまま張り合わせている。よって、前景が誤って抽出されない場合に背景に張り付いて視覚的に妨害するという問題があった。本稿では、この問題を回避するために、まず、画像を GMV によって時空間にマッピングした後に、各 XY 座標において時間方向にメディアンを算出して仮のスプライト（メディアンスプライト）を生成し、さらに着目する MB において、メディアンスプライトとの差分が最小になるフレームの MB 内のテクスチャを求めるスプライトの MB のテクスチャとする。

5. 符号化の詳細

VideoESPER は 2 レイヤのビデオオブジェクトを符号化するため、従来方法のレート制御法をそのまま適用することはできない。また、文献[6]で示されるマルチビデオオブジェクトレート制御法は動画像オブジェクトを対象にしているので、スプライトには適用できない。よって次節で簡単に 2 レイヤのビデオオブジェクトをレート制御する方法を示す。また、スプライト符号化の符号化効率を上げるためのスプライトに対する幾何変換（回転および縮小）処理について述べる。

5.1 2 レイヤ簡易レート制御

VideoESPER は 2 レイヤのビデオオブジェクトを符号化するため、従来方法のレート制御法をそのまま適用することはできない。VideoESPER では、前景オブジェクトと背景オブジェクトの符号量比率を最初に算出し、それに準じて符号量を割り当てる。具体的には以下の処理を行う。

- (1) QP=15 (QP: Quantization Parameter)にて前景背景を符号化し、その配分を前景背景比とする。
- (2) 全体のトータルビットからスプライトを QP 固定で符号化し、全項で求めた背景スプライト配分になる QP を探し出す。
- (3) 残りのビット数を計算し、前景オブジェクトの総ビット数とする。
- (4) 前景オブジェクトを残りのビット数を参照して VM17.0[7]で符号化する。ここで、前景のフレームレートが算出される。
- (5) スプライトをフレームレートに合わせて、QP 値は固定にて低遅延スプライト符号化モードで符号化する。

5.2 スプライトの回転

文献[4]にあるように、MPEG-4 では空間方向予測には AC/DC 予測が用いられる。これは上もしくは左に存在する係数が参照されるので、右から左に、あるいは下から上にパニングするような画像を低遅延スプライトモードで符号化する場合は AC/DC 予測が効果的に働かない。よって、こ

のようなスプライトを有する場合は、スプライトを180度回転し、かつ、GMVも回転後のスプライトから切り出せるように変換し、左から右、もしくは上から下にパニングするように変換した状態で前項の符号化を行う。

5.3 スプライトの中間解像度変換

文献[2]にあるように、静止画像においては、低ビットレートでは画像を縮小して符号化し、表示するときに拡大した方が符号化効率がよいことが示されている。VideoESPERでもこの原理を利用し、ビットレートが低いところ(64kbps以下等)では、スプライトに縮小処理をかけ、GMVを再計算して符号化を行う。

6. システムの評価

6.1 グローバルモーション算出における評価

前節で提案したグローバルモーション算出法の評価を行った。まず、LMVをブロックマッチングで算出し、それを基にGMVを算出する。画像は表1に示す5種類を用いた。比較対象は、以下の2手法である。

1. ボトムアップ[2]のみでGMVを算出
2. 提案手法

評価は各手法で求められた隣接フレーム間GMVにおいて、原画像とGMVによる予測画像の平均二乗誤差(MSE: Mean Square Error)によって行った。

表1 実験画像

Image	Frames	Image size	contents
horserace_a	999	SIF	pan, zoom
horserace_b	999	SIF	pan, zoom
skateboard	390	SIF	pan, tilt
soccer	999	SIF	pan, tilt, zoom
stefan	300	SIF	pan, zoom

表2 GMV算出法と平均MSEの比較

Image	Ave. MSE		Bottom-up ratio
	Bottom-up	Proposed	
horserace_a	342.571	333.566	0.796
horserace_b	171.534	171.540	0.997
skateboard	185.251	185.118	0.995
soccer	194.597	143.535	0.606
stefan	390.393	390.350	0.990

平均各手法における平均MSEと提案手法におけるボトムアップ法が採用された割合を表2に示す。また、顕著な改善の見られた“Soccer”の実験結果を図5(1-150フレームの結果を表示)に

示す。“Soccer”はフィールド内の芝領域が画面の大半を占めるなど、LMVの精度が不十分である。これらのところで多くトップダウン選択されることでMSEの向上をもたらしたことがわかる。基本的にトップダウン法は画像全体をアフィン変換するので、処理時間がボトムアップ法の数倍かかるというデメリットがある。提案手法はボトムアップ方式がGMVの誤抽出を起こすときにトップダウン方式を採用していることがわかる。図5で2方式とも突出しているMSEを算出しているところはカット点である。

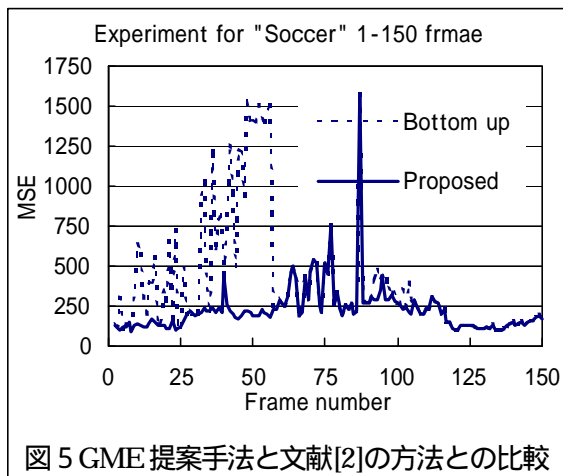


図5 GME提案手法と文献[2]の方法との比較

6.2 符号化実験

前述の5つの画像でVideoESPERの符号化実験を行った。比較対象は以下の2つである。

1. 従来符号化法によるシングルモード符号化(MPEG-4 main profile VM17.0[7])
2. VideoESPERによるマルチモード符号化(MPEG-4 main profile 準拠)

VideoESPERのノーマルモードはBピクチャを含まないものとする。(384kbps以下のbitstreamは、ほぼsimple profileと同じである。)目標ビットレートおよび目標フレームレートは、{128kbps, 15fps} (=test1)および{384kbps, 30fps} (=test2)で実験を行った。表3にモード判定結果の例を示す。表4に符号化効率、平均QP、平均SNR、フレームレートの比較を示す。図6にシーケンスにおけるSNRの遷移、図7にシーケンスにおける平均QPの遷移を示す。表5にスプライトモードの平均前景率とスプライトモードの選択率を示す。図8にスプライトショットの合成画像の例を示す。また、ユニット処理におけるショットの区切りはT=1秒、またユニット数n=5とした。表5より、“stefan”のみ、すべてのショットにてスプライトモードが選択された。表3

表3 モード判定結果の一例 (“skateboard”)

Start frame	End frame	First decision	Second decision	FG ratio	Final decision	Final shot
1	30	N	N		N	N
31	60	N	N		N	
61	90	N	N		N	
91	120	S	N	0.214	N	
121	150	S	S	0.092	S	S
151	180	S	S	0.101	S	
181	210	S	S	0.120	S	
211	240	S	S	0.063	S	S
241	270	S	S	0.097	S	
271	300	S	S	0.106	S	S
301	330	S	S	0.093	S	
331	360	N	N		N	N
361	390	N	N		N	

表5 平均前景率, スプライトモード選択率

Image	Sprite mode ratio	Foreground ratio
horserace_a	0.478	0.140
horserace_b	0.775	0.083
skateboard	0.538	0.096
soccer	0.516	0.125
stefan	1.000	0.094

より, カメラ操作のない部分や前景が大きくなるショットはノーマルモードが適用されていることがわかる.

表4, 図7よりより, test1, test2とも, 平均QPや達成したフレームレートはVideoESPERの方がよい結果になっており, 視覚的に高い品質の映像を実現している. 図8からも主観画質が高いことがわかる. しかし, 図6より VideoESPERの全体のSNRがシングルモードに対して低い値が出ているのは, 背景スプライトが各々のフレームを貼り合わせて生成されたものであるため, そこから切り出す背景画像は元の信号を忠実に再現してないことに起因する. しかし, 基本的にスプライトはカメラ操作がある部分で生成されないこと, また, 動きのある部分は人間の視覚特性が甘くなること, 信号処理的には再現性がないものの, 元の画像を貼り合わせていることにより, 極端なズームイン, アウトがない限り解像度はある程度保たれると予想できるなどのことから, 品質のよい画像が合成される. また, 前景画像に限って言えば, シングルモードよりも高いSNRを実現している.



VideoESPER

Single mode

図8 合成画像の例 “stefan”

6.3 VideoESPERの有効適用条件と限界

従来法にとって厳しい符号化条件である128kbps, 15fpsや, 384kbps, 30fpsにおいてはVideoESPERが従来符号化法を圧倒する結果となった. 一方で, 1画素あたりに十分な符号量を割り当てられる, 高いビットレートや目標フレームレートが十分低い場合には従来符号化法で十分な品質が得られると予想される. ここで, 従来法が優位となる場合を元の画像データから符号化することなし自動的に求める実験を行った.

符号化では必ず先頭フレームはIフレームとして扱い, フレーム内で閉じた符号化を行う. そこで, 最初のフレームをビットレート, フレームレート, 画像サイズをパラメータとするにいくつかの符号化条件においてVM17.0[7]にて符号化した場合の初期IフレームのQP(この場合, QPにリミッタをかけない)を測定する. 表6にその結果の例(画像は“Stefan”を使用)を示す. QP=15以下で符号化が可能である場合は, 十分な画質を与えることができると仮定すると, 表6から, 1画素あたりの符号量が0.3bit以上である場合はシングルモードでも十分な符号量が与えられることがわかる. また, 1画素あたりの符号量が0.3bit以下であっても, 平易な画像では十分な符号量が与えられる. また, ビットレートが高くなるにつれ, 背景スプライトでは再現できないような背景の細かな動きがシングルモード符号化では表現可能であることも予想される. VideoESPERの効果的な適用領域としては, 低レートで, かつ, カメラ操作のある画像ところに限定される.

VideoESPERにこの条件を科し, ゼロ番目のモード判定として, 採用することも可能である.

表4 符号化結果

Image	Mode	Object type	128kbps, 15fps			384kbps, 30fps		
			Ave. QP	Ave. SNR [dB]	framerate [fps]	Ave. QP	Ave. SNR [dB]	framerate [fps]
horserace_a	Multi	FG	19.99	25.35	15.06	10.51	29.45	30.00
		BG	20.42	19.76	15.06	10.31	19.82	30.00
		Norm.	30.10	24.80	10.88	23.77	25.90	30.00
		Total	---	23.24	12.88	---	24.21	30.00
	Single	---	30.37	24.97	8.83	25.12	20.20	30.00
horserace_b	Multi	FG	14.41	28.06	15.00	6.91	33.43	29.65
		BG	14.27	22.74	15.00	7.07	22.99	29.65
		Norm.	17.36	29.31	14.53	10.90	32.32	30.00
		Total	---	25.39	14.89	---	26.50	29.73
	Single	---	27.00	26.94	7.39	14.12	29.40	30.00
skateboard	Multi	FG	18.97	24.29	15.00	9.36	30.07	27.57
		BG	18.71	20.29	15.00	9.62	20.25	27.57
		Norm.	24.07	14.46	14.83	10.95	26.83	30.00
		Total	---	23.16	14.92	---	24.46	28.69
	Single	---	26.40	24.46	12.85	18.15	26.38	30.00
soccer	Multi	FG	12.17	29.87	15.00	4.93	33.83	30.00
		BG	11.88	23.08	15.00	4.03	23.11	30.00
		Norm.	24.22	28.54	14.00	7.59	30.09	30.00
		Total	---	26.63	14.52	---	27.73	30.00
	Single	---	24.35	28.74	13.65	14.00	30.64	30.00
stefan	Multi	FG	17.07	26.44	15.00	8.34	31.48	30.00
		BG	17.00	18.19	15.00	8.30	18.23	30.00
		Norm.	---	---	---	---	---	---
		Total	---	19.38	15.00	---	19.52	30.00
	Single	---	30.84	24.15	8.90	23.67	25.37	30.00

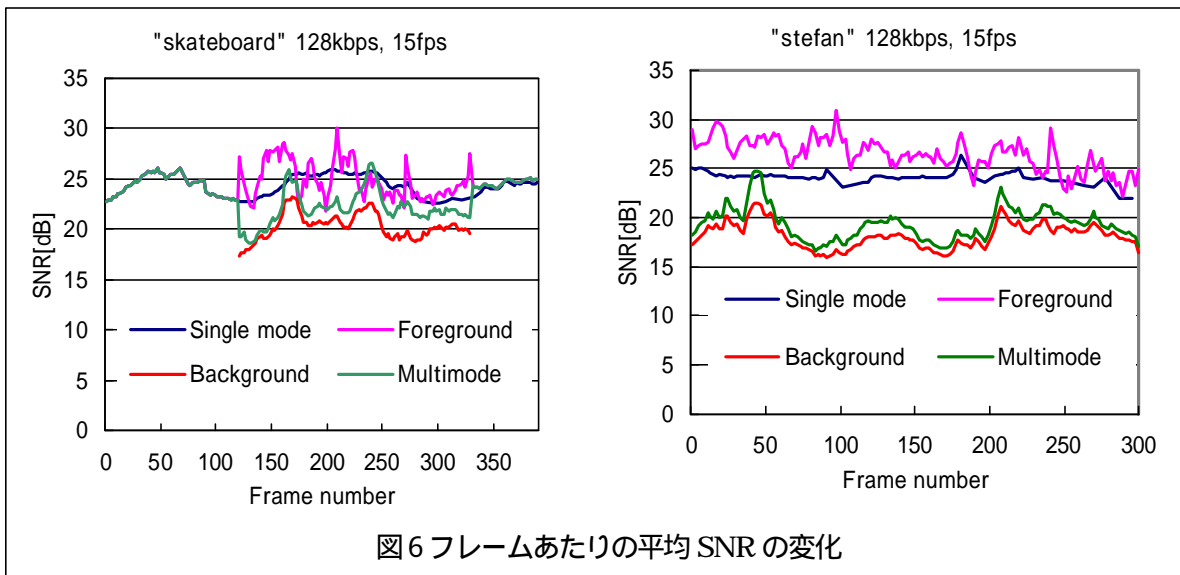


図6 フレームあたりの平均 SNR の変化

と結論づけられる。また、VideoESPER の適用領域においては、よりたくさんの画像をもちいて実験検証する必要がある。

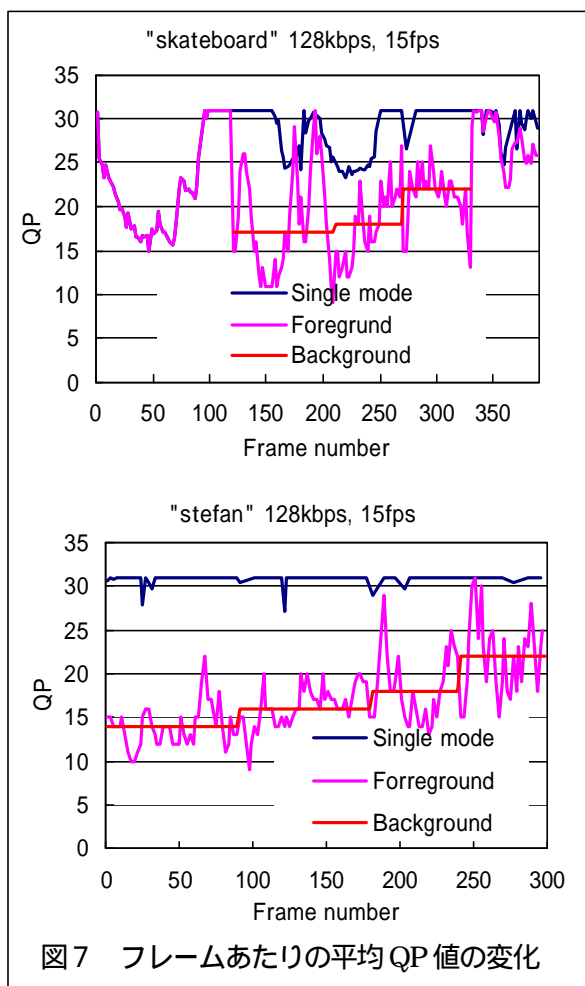


図7 フレームあたりの平均QP値の変化

表6 画素あたりの符号量とIフレームのQP値の関係(“stefan”)

Frame number	128kbps		384kbps		512kbps
	10fps	15fps	15fps	30fps	30fps
1	20	30	10	20	15
31	22	33	11	22	16
61	22	33	11	22	16
91	21	32	10	21	16
121	23	34	11	23	17
151	22	33	11	22	16
181	18	27	9	18	13
211	20	30	10	20	15
241	20	30	10	20	15
271	21	32	10	21	16

7. おわりに

筆者らは、スプライト符号化を用いた汎用的な

MPEG-4 動画像符号化システムである VideoESPER を開発した。スプライトモードが優位なショットを自動的に抽出し、そのショットにはスプライトモードを適用し、その他のショットに対しては従来符号化法を踏襲するノーマルモードを適用するマルチモード符号化法を提案した。さらに、AVI データ等を逐次的に処理するための逐次処理方法を提案した。また、グローバルな動きの算出法、前景領域抽出法、スプライト生成法に改良を行った。また、前景背景に自動的に符号量を配分するためのマルチビデオオブジェクトレート制御法を提案し、符号化効率を上げるために、スプライトの縮小方式である中間解像度変換法、逆走スプライト方法を提案した。また、実際に符号化実験を行って、低レートにおける従来符号化方式に対する VideoESPER の優位性を示した。一方で、レートが高くなると従来方式の画質があり、VideoESPER を適用することには限界があることを述べた。

謝辞

本研究を遂行するに当たって有意義なディスカッションをいただいた NTT サイバースペース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の皆様へ感謝いたします。

参考文献

- [1] “Information technology- Coding of audio-visual objects-Part2:Visual Amendment 1:Visual extensions ISO/IEC 14496-2
- [2] 秦泉寺久美, 渡辺裕, 岡田重樹, 小林直樹:「スプライト符号化を用いた MPEG-4 動画像超高压縮」, 電子情報通信学会和文論文誌 D-II, Vol. J84-D-II No.5 pp.758-768, May 2001
- [3] 岡田重樹, 秦泉寺久美, 渡辺裕, 小林直樹:「スプライト符号化を利用した MPEG-4 マルチモード符号化方式の研究」, PCSJ2000 pp.91-92 (2000年11月)
- [4] 岡田重樹, 高村誠之, 秦泉寺久美, 阪谷徹, 小林直樹:「スプライト符号化における解像度変換処理に関しての一検討」, 信学会総合大会 P.45 (2001年3月).
- [5] Michal Irani, P. Anandan, Jim Bergen, Rakesh Kumar, Steve Hsu, “Efficient Representations of Video Sequences and Their Applications,” Signal Processing: Image Communication Vol.8 pp.327-351, 1996.
- [6] Anthony Vetro, Huifang Sun, Yao Wang, “MPEG-4 Rate Control for Multiple Video Objects,” IEEE Trans. on Circuits And Systems for Video Technology, Vol.9, No.1, pp.186-199, February 1999.
- [7] “MPEG-4 Video Verification Model version 17.0,” ISO/IEC JTC1/SC29/WG11/N3515