

画像サイズ変換と動きベクトルの再利用を用いた MPEG-2/H.264 トランスコード手法の検討

Study of MPEG-2/H.264 transcoding using the reuse of motion vectors and the image size conversion

土野裕馬[†]
Yuma Tsuchino

植山康平[†]
Kouhei Ueyama

吉留健[†]
Takeshi Yoshitome

三柴数[†]
Kazu Mishiba

1. はじめに

現在、映像の圧縮方式として MPEG-2 と H.264 がよく使われている。H.264 は MPEG-2 より圧縮効率が良く、低ビットレート環境にも対応できる。しかし、圧縮率の低い MPEG-2 で符号化されたコンテンツがすでに多数存在することから、これらを H.264 へのトランスコードすることが必要となっている。H.264 は高性能である反面、処理が非常に複雑であり時間がかかるため、高速化が求められている。

そこで、本研究では、MPEG-2 から H.264 へ変換を行うトランスコードの処理時間の短縮を図る。処理時間を短縮させる手法の研究は様々行われている。例えば、MPEG-2 の 8×8 DCT ブロックを H.264 の 4×4 整数 DCT へ数学的に直接行列変換することで MPEG-2 デコード処理と H.264 エンコード処理を不要にする手法 [1] や、H.264 で選択可能な多数のマクロブロック (MB) モードすべてを試すのではなく MPEG-2 符号化時に選択したモードを参考に適用する H.264 モードを限定したり、あるいは良い評価値が得られた時点で打ち切ることによって処理時間の短縮を図る手法が提案されている [2] [3] [4]。また、MPEG-2 での動きベクトル情報を用いることで H.264 エンコード処理の一部を省略し、トランスコード時間の短縮を可能にする手法が存在する。しかし、この手法でも映像の高精細化による演算量の増大が原因でトランスコードに時間がかかる。そこで、従来の画面サイズを変えないトランスコードではなく、画面サイズを縮小してトランスコードするとともに、MPEG-2 符号化で得られた動きベクトルを再利用することで、演算量の削減をめざす。

2. 従来手法

MPEG-2 から H.264 へのトランスコードは、MPEG-2 のデコーダと H.264 のエンコーダを直列に繋げれば実現可能である。従来手法である MPEG-2/H.264 トランスコードの手順を図 1 に示す。まず、MPEG-2 で圧縮されたビットストリームを MPEG-2 のデコーダに通じて復号を行う。次に復号により得られた映像をそのまま H.264 のエンコーダへ入力、その後、符号化を行うことで H.264 のビットストリームを得ることができる。しかし、H.264 のエンコードは様々なブロックサイズを使うなど複雑な処理を行っているため、処理量が膨大になる。そのため、この手法ではトランスコードに時間がかかるという問題点がある。

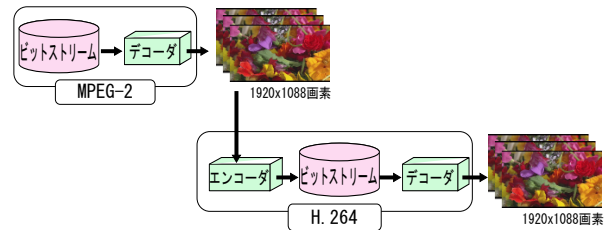


図 1: 従来手法の処理手順

3. 提案手法

3.1. 提案手法の概要

提案手法は、映像の縮小と、MPEG-2 の動きベクトル情報を用いて H.264 エンコードにおける動き探索を省略する方法を組み合わせた手法である。動き探索は H.264 エンコードにおける処理の約 60% 以上を占める膨大な処理となっているため、これを省略することで大幅な処理量の削減が見込まれる。

図 2 に具体的な提案手法の処理手順を示す。提案手法では、まず MPEG-2 のビットストリームをデコーダに入力しデコード映像を得る。ここで従来手法ではそのままの映像サイズで H.264 エンコーダに入力していたが、提案手法ではデコード映像を横方向に 1/2、縦方向に 1/2、縦横共に 1/2 のいずれかに縮小し、その後 H.264 エンコーダに入力する。それによって、提案手法は H.264 エンコーダにおける処理を縮小した分だけ削減できる。さらに、MPEG-2 の動きベクトル情報を再利用して、H.264 エンコードにおける動き探索を省略することで大幅に処理量を減らし高速化できる。これらの方法によって高速にエンコードを行い、H.264 のビットストリームを得ることができる。再生する際はデコーダに入力して、デコード映像を拡大することで元のサイズで映像を再生する。

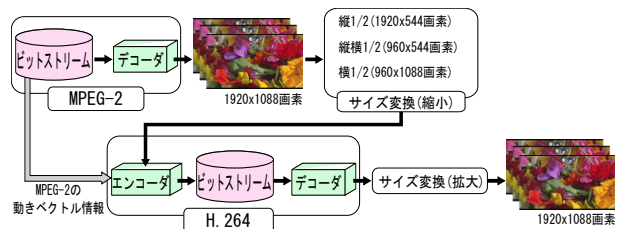


図 2: 提案手法の処理手順

[†]鳥取大学 大学院 工学研究科

3.2.MPEG-2 の動きベクトルの再利用法

MPEG-2 の動きベクトルを H.264 で再利用する際、双方で動きベクトルの精度が異なっているため、補正する必要がある。動きベクトルの精度は MPEG-2 が 1/2 画素精度、H.264 が 1/4 画素精度となっており、動きベクトルの値が異なっている。そのため、H.264 で使用する際は 1/2 画素精度から 1/4 画素精度に変換するために動きベクトルの値を 2 倍にしている。さらに、提案手法では、サイズ変換と動きベクトルの再利用を組み合わせているので、サイズ変換前後で MB の数が異なる、また動きベクトルの参照位置がずれる問題もある。これらを解決するために、動きベクトルの補正、H.264 の特徴である様々なブロックサイズを利用する。図 3 の a) のように縦方向に 1/2 縮小する場合 16×8 画素ブロックを使用し、さらに動きベクトルの y 成分の値を 1/2 に補正すると元サイズでの処理と同じにすることができる。H.264 には MB を 16×8 画素ブロック、または 8×8 画素ブロックに分割するモードが存在するため、横方向に 1/2 縮小する場合は 16×8 画素ブロック、縦横共に 1/2 縮小する場合は 8×8 画素ブロックを使用することでサイズ変換前後で情報を一致させる。

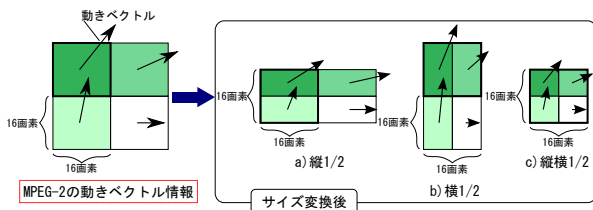


図 3: MPEG-2 の動きベクトルを再利用する方法

4. 評価実験

4.1. 実験環境

縦方向、横方向に 1/2 縮小および縦横共に 1/2 縮小をそれぞれ用いた提案手法における符号化効率を比較するシミュレーション実験を実施した。拡大、縮小ともに補間方法として、双三次補間 (Bicubic 補間) を使用する。フレーム数は 30 枚、GOP 構造は N=15、M=1、MPEG-2 における量子化ステップは 16 で固定、H.264 での量子化ステップの値を 16, 20, 22, 26, 28, 32, 34, 36, 40 の 9 通りに変化させ RD 特性を作成する。MPEG-2[5] と JM9.0[6] のソフトウェアを用いて 1920×1088 画素の様々な映像に対しトランスコードを行う。また、映像の走査方式としてプログレッシブ方式とインターレース方式の二種類について考察を行う。

4.2. インターレース方式の映像を入力した場合

本実験では、インターレース方式の映像を入力し、提案手法の有効性を確認する。テスト映像には、図 4、図 5 に示す二種類の映像 [7] を使用した。Flower Basket は動きがほとんどなく、Crowded Crosswalk は動きの大きい映像である。

図 6、図 7 は、それぞれ Flower Basket と Crowded Crosswalk を入力とした場合の従来手法と提案手法の RD 特性である。図 6、図 7 から、どちらも横方向に 1/2 縮小を行った場合の提案手法の RD 特性と従来手法の RD 特性が近似している。つまり、両方の映像に対しても横方向に 1/2 縮小を用いた提案手法が従来手法と同程度の性能を持つことが分かる。また、縦方向に 1/2 縮小を用いた提案手法は、従来手法に比べ画質が大きく劣化している。図 6 と図 7 を比較すると、動きの大きい映像である図 5 のテスト映像を使用したとき、RD 特性の劣化が大きいことが分かる。縦横共に 1/2 縮小を使用した提案手法も、縦方向に 1/2 縮小と同じ位置に存在している。このことから、縦縮小の影響を大きく受けていると考えられる。



図 4: Flower Basket (動きの小さい映像) 図 5: Crowded Crosswalk (動きの大きい映像)

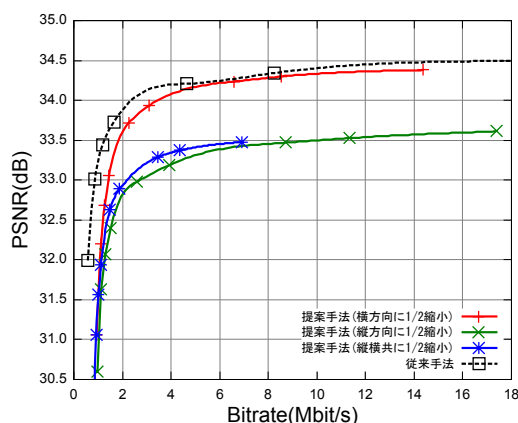


図 6: Flower Basket における RD 特性の比較

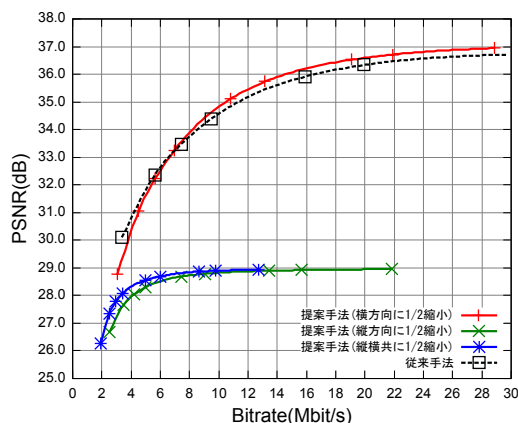


図 7: Crowded Crosswalk における RD 特性の比較

4.3. プログレッシブ方式の映像を入力した場合

入力映像として、ビデオカメラで撮影した映像を使用した。このカメラにて、プログレッシブ方式で撮影した映像をMPEG-2でエンコードし、出力されたストリームを入力としてシミュレーションを行った。今回使用した映像は、中央の物体を対象としてズームアウトしていく映像A(図8)と、ブラインドを対象にカメラを移動させる映像B(図9)である。映像Aは中央の物体が細かい縦方向のエッジを持っているという特徴がある。また、映像Bは細かい横方向のエッジが多いという特徴を持つ。図10、図11にそれぞれの映像をトランスコードを行い作成したRD特性を示す。映像Aにおいて、提案手法の中で最も従来手法のRD特性に近いのが、縦方向に1/2縮小を用いた場合である。その他の縮小方法を使った場合、縦縮小を行った場合に比べ劣化している。一方、映像Bでは、横方向に1/2縮小した場合が最も従来手法の性能に近いことが分かる。

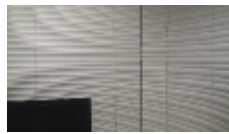


図 8: 映像 A(縦縞が多い) 図 9: 映像 B(横縞が多い)

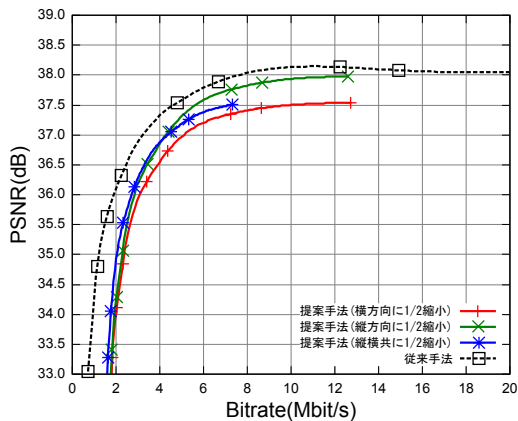


図 10: 映像 A における RD 特性の比較

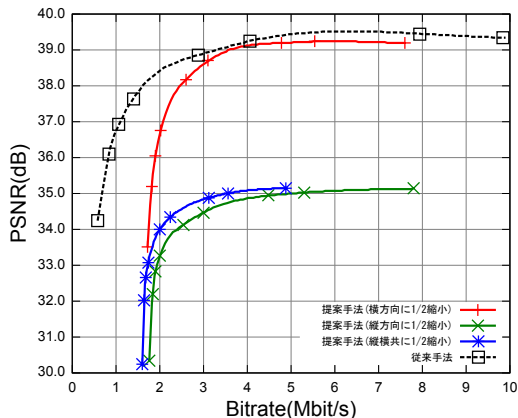


図 11: 映像 B における RD 特性の比較

4.4. 走査方式の違いによる影響

走査方式の違いが、どのように提案手法に影響しているか確認するため、ビデオカメラにて走査方式の違いほとんど同じ映像(図12)を作成し、それを入力としたトランスコードを行いRD特性を作成、確認する。図13に示すインターレース方式の映像を入力した提案手法において、縮小方向の違いのみで大きくRD特性に差が出ており、横縮小を行った手法が有効であることが分かる。一方、図14に示すプログレッシブ方式の映像を入力した場合、縮小方向の違いでRD特性ともそれほど差がないことが分かる。これらの原因として、インターレース方式に発生するノイズが考えられる。このノイズは動く物体、または背景に発生し、動きが激しいほど大きなノイズとして現れる。また、ノイズは細かい横縞として現れる特徴がある。縦方向に縮小を行うと細かい横縞は維持されないため、画質が大きく劣化する。そのため、インターレース方式では横縮小を用いた提案手法が最も良い性能を発揮する。



図 12: 比較に使用した映像

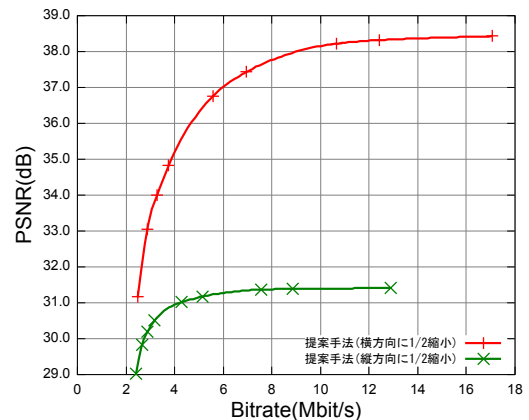


図 13: インターレース方式の映像における RD 特性の比較

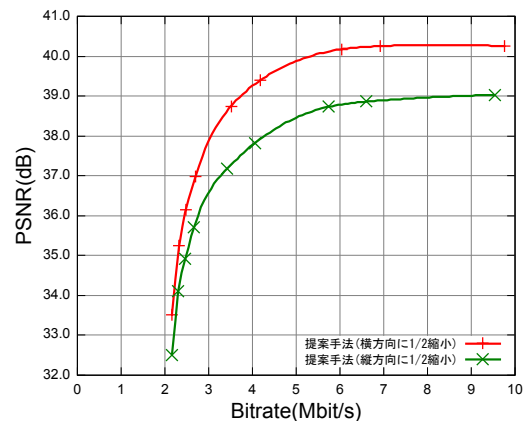


図 14: プログレッシブ方式の映像における RD 特性の比較

4.5. 縦横共に1/2縮小を用いた提案手法の検討

提案手法に用いる縮小方向が画質に大きな影響を与えるが、縮小を縦横どちらも1/2にした場合、どのような条件において有効であるか検討を行う。使用したテスト映像を図15に示す。この映像は、歩行する中央の人物に注目して撮影されたものであり、カメラも動いているため少量の動きを持っている。

図16に示すRD特性から、すべての提案手法におけるRD特性が従来手法のものに近い位置に存在することが分かる。また、低ビットレートにおいて、縦横共に1/2縮小を用いた提案手法が従来手法とほぼ同じ位置に存在し、その他の縮小方法を用いた提案手法より優れた性能を持っていることが分かる。これはテスト映像の符号化による劣化が大きく、それにより縮小による画質劣化がRD特性にほとんど影響しなかったことが原因だと考えられる。



図15: シミュレーションに使用した映像

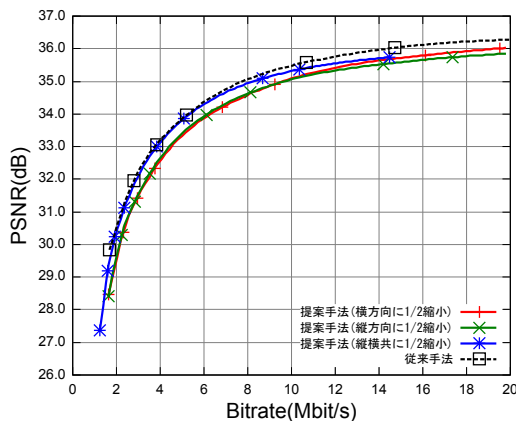


図16: 従来法と三種類の提案法のRD特性による比較

5. おわりに

本研究では、トランスコードの処理時間の削減するために、サイズ変換と動きベクトル再利用を併用したMPEG-2/H.264 トランスコードを提案した。実験結果より、提案手法に用いる縮小方向を入力映像に応じて適応的に選択することで、従来手法に匹敵する符号化効率を得られることが分かった。処理量はMPEG-2の動きベクトル情報の再利用による動き探索の省略、また縮小による画素数の減少により大幅に削減できる。以上より符号化効率の劣化を抑えてトランスコードの処理時間を短縮することができる。なお本研究はJSPS 科研費 24560460 の助成を受けたものである。

参考文献

- [1] J.-K. Lee and K.-D. Chung. Quantization/dct conversion scheme for dct-domain MPEG-2 to H.264/AVC transcoding. IEICE Trans. Comm., vol.E87-B, no.7, pp.2856-2863, July 2004.
- [2] 筑波健史, 永吉功, 花村剛, 富永英義. MPEG-2 から H.264 への再符号化におけるイントラ予測モード決定手法に関する検討. 社団法人電子情報通信学会, 第 2005-AVM-49 巻, pp.37-42, June 2005.
- [3] X. Lu, A.M. Tourapis, P. Yin, J. Boyce. Fast mode decision and motion estimation for H.264 with a focus on MPEG-2/H.264 transcoding. Proc. IS-CAS 2005, vol.2, no.7, pp.1246-1249, May 2005.
- [4] G. Chen, Y. Zhang, S.Lin, and F.Dai. Efficient block size selection for MPEG-2 to H.264 transcoding. Proc. 12th Annual ACM International Conference on Multimedia, vol.2, no.7, pp.300-303, Oct 2004.
- [5] MPEG-2 Software Simulation Group (MSSG)
- [6] Joint Video Team(JVT) "Reference Software JM9.0"
- [7] 映像情報メディア学会「ハイビジョン・システム評価用標準動画像」