

DCT 領域での多地点映像合成処理についての検討

瀧口 昌彦[†] 加藤 晴久[‡] 中島 康之[‡] 半谷 精一郎[†]

[†] 東京理科大学工学部 〒162-0825 東京都新宿区神楽坂 1-3

[‡] 株式会社 KDDI 研究所 〒356-0003 埼玉県上福岡市大原 2-1-15

E-mail: [†] takiguchi@tsm.kddilabs.jp, hangai@ee.kagu.tus.ac.jp [‡] {hkato, nakajima}@kddilabs.jp

あらまし 利用者端末での遅延を最小限に抑えるため、ネットワーク上で複数の映像を 1 画面に再変換して出力する TV 会議システムの構築について、その高速化を検討する。ベースバンド変換では、システム内での伸長、合成再圧縮において、DCT 係数からピクセル値への変換、逆変換を行うため処理負荷が大きい。本研究では、伸長、合成再圧縮時に処理負荷の大きい DCT 変換を行わず DCT 領域で逆動き補償 (IMC)、解像度変換、動き補償 (MC) をすることで高速化を図る。また入力シーケンスの動き予測情報を再利用することで、動き予測処理を省略する。これらの処理方式を採用した、DCT 領域で全ての処理を行う映像合成方式の有効性について、シミュレーション実験によりベースバンド方式と比較検討する。

キーワード DCT, 逆動き補償(IMC), 動き補償(MC)

A Study on The DCT Domain Multipoint Video Transcoding

Masahiko TAKIGUCHI[†] Haruhisa KATO[‡] Yasuyuki NAKAJIMA[‡] and Seiichiro HANGAI[†]

[†] Faculty of Engineering, Tokyo University of Science 1-3 Kagurazaka, Shinjuku-ku, Tokyo, 162-0825 Japan

[‡] KDDI R&D Laboratories Inc. 2-1-15 Ohara, Kamifukuoka-shi, Saitama, 356-0003 Japan

E-mail: [†] takiguchi@tsm.kddilabs.jp, hangai@ee.kagu.tus.ac.jp [‡] {hkato, nakajima}@kddilabs.jp

Abstract In this paper, we study compressed domain multi-point video transcoding method. In the conventional MCU(Multi-point Control Unit) for multipoint video conferencing, baseband domain transcoding is used to output multipoint video in a single video stream. Since decoding of multiple video stream and re-encoding are involved in the course of transcoding, computational complexity is usually very high. Here, we propose the method of the inverse motion compensation (IMC), the resize, and the motion compensation (MC) in DCT domain. These methods enable to build the DCT domain multipoint video transcoding. We investigate the effectiveness of the DCT domain processing videoconferencing system by comparing it with the base-band domain transcoding.

Keyword DCT, Inverse Motion Compensation (IMC), Motion Compensation(MC)

1. はじめに

TV 会議システムは、これまで ISDN 等のデジタル回線を用い普及してきたが、近年のブロードバンドの普及ならびに PC の能力向上に伴い、さまざまなネットワーク、機器環境で利用されるようになった。これに伴い、多地点接続して利用するニーズも高くなりつつあるが、多地点映像を 1 つの映像に変換する MCU (Multipoint Control Unit) は多数の圧縮映像を復号して 1 つの映像にまとめて再圧縮するため、非常に処理負荷が高く、装置も高価なものになっている。

また、TV 会議参加者が遅延を感じることなく TV 会議システムを快適に利用するためには、送信端から受信端までに許容される遅延は 100 - 150ms 程度といわれている。ただし利用者端末での映像の圧縮および伸長における遅延や、ネットワーク伝送による遅延が存在するため、MCU 上の TV 会議システムによる処理は 10ms 以内の遅延に抑えなければならない[1]。

本稿では、DCT 係数上で逆動き補償 (IMC)、解像度変換、動き補償 (MC) の処理を行うことにより高速に映像合成する方式について検討を行ったので報告する。

2. TV 会議システムに利用できる従来の DCT 領域処理技術

2.1 従来の DCT 領域での解像度変換

筆者らはこれまでに行列の対称性を利用した DCT 領域の高速解像度変換について提案してきた[2]では、4 組の 8x8DCT 係数(マクロブロック)から 縦横 1/2 に縮小した 1 組の 8x8DCT を出力することを検討した。入力 DCT 係数のそれぞれ左上 4x4 成分以外の成分を強制的に 0 にし、16 x 16 行列 U としたとき、ベースバンド変換では 8x8IDCT、平均をとって縦横半分にする解像度変換、8x8DCT の処理を行うことによって、解像度変換された DCT 係数を得る。一方[2]方式は U に前と後ろから掛ける 2 つの行列を予め計算してリサイズ行列 S_L と S_R とする。解像度変換処理は、リサイズ行列の対称性を利用した行列掛算 $S_L U S_R$ によって行われる。

この方式は、ベースバンド解像度変換方式と比較して約 86.62%処理時間の削減を実現した。8x8DCT 係数の左上 4x4 成分に対して、強制的に 4x4DCT 変換を行うことで解像度変換を行う方式[3]と比較し、演算数は同じだが画質の向上を実現している。

2.2 従来の共通演算部を利用した DCT 領域での逆動き補償

DCT 領域での逆動き補償は、参照フレームの参照領域を含む最大 9 組の 8x8DCT 係数 R_{mn} を 8x8DCT 係数上で扱う。計算量の削減は、ブロック内にある参照部分をまとめて演算し、切り出すことで実現する[4]。途中に用いられるシフト計算は単位行列と 0 行列の組み合わせによって行われるので処理負荷はかからない。ベースバンドにおいてブロック単位で 16 回計算していたものが、[4]方式では 9 回に減る。この処理にかかる演算数(乗算回数と加算回数の和)は 10144 回である。しかし Chen の高速 DCT を用いたベースバンド領域での逆動き補償にかかる演算数は 6048 回であり、従来方式は演算数が増えてしまい、高速化できていない。

3. 提案する DCT 領域処理の概要

ベースバンドの TV 会議システムでは、ベースバンド領域で合成縮小処理を行うために入力映像の伸長、出力映像の圧縮をするため処理量が多かった。そこで、DCT 係数上で図 1 のような手順で処理を行うことにより高速化を図る。以下に図 1 における各処理の簡単な説明をする。

可変長復号(VLD) 4 組の入力 MPEG 動画から量子化済みの予測誤差係数および動き予測情報を抽出する。

逆量子化(IQ) 量子化済み予測誤差係数を逆量子化し、量子化誤差を含んだ予測誤差情報(8x8DCT 係数)を逆動き補償部(IMC)に出力する。

逆動き補償(IMC) 1 フレーム分の参照 8x8DCT 係数をフレームメモリから取り出す。IQ からの処理対象フレームの差分 8x8DCT 係数および動きベクトルを入力し、DCT 係数上で逆動き補償し、処理対象フレームの完全なる 8x8DCT 係数を導出する。

解像度変換およびフレーム合成 完全なる 8x8DCT 係数を、縦横半分のサイズに DCT 係数上で解像度変換し、ひとつのフレームに合成する。

動き補償(MC) まず復号部(VLD)より入力される動きベクトル情報を用いて適当な動きベクトルを合成フレームの各マクロブロックに割り当てる。フレームメモリから得られた 1 フレーム分の参照 8x8DCT 係数、および合成部からの処理対象フレームの完全なる 8x8DCT 係数から、DCT 係数上で動き補償し、対象フレームの予測誤差情報(8x8DCT 係数)を導出する。

なので、予め計算して格納しておくことで演算回数を削減することができる。

さらに式(1)の定数行列の積は式(6)に示すような対称性が現れる。

$$\begin{pmatrix} T_8 & 0 \\ 0 & T_8 \end{pmatrix} V_j \begin{pmatrix} T_8' & 0 & 0 \\ 0 & T_8' & 0 \\ 0 & 0 & T_8' \end{pmatrix} = \begin{pmatrix} A_j & A_j' & 0 \\ 0 & A_j & A_j' \end{pmatrix} \quad (6)$$

$$\begin{pmatrix} T_8 & 0 & 0 \\ 0 & T_8 & 0 \\ 0 & 0 & T_8 \end{pmatrix} H_i \begin{pmatrix} T_8' & 0 \\ 0 & T_8' \end{pmatrix} = \begin{pmatrix} A_i' & 0 \\ A_i'' & A_i' \\ 0 & A_i'' \end{pmatrix}$$

ここで、 A_i, A_i' は 8×8 行列を表す。式(1)に式(6)を適用すると DCT 領域で動き補償された DCT 係数行列 Y は式(7)で導出される。

$$Y = \begin{pmatrix} A_j & A_j' & 0 \\ 0 & A_j & A_j' \end{pmatrix} R \begin{pmatrix} A_i' & 0 \\ A_i'' & A_i' \\ 0 & A_i'' \end{pmatrix} \quad (7)$$

さらに式(7)を展開すると式(8)のようになる。

$$Y = \begin{pmatrix} (A_j R_{00} + A_j' R_{10}) A_i' + B_j A_i'' & B_j A_i' + (A_j R_{02} + A_j' R_{12}) A_i'' \\ (A_j R_{10} + A_j' R_{20}) A_i' + C_j A_i'' & C_j A_i' + (A_j R_{12} + A_j' R_{22}) A_i'' \end{pmatrix} + D \quad (8)$$

$$B_j = (A_j R_{01} + A_j' R_{11}) \quad (9)$$

$$C_j = (A_j R_{11} + A_j' R_{21})$$

式(8)の括弧内の式は、それぞれ共通する部分があり、式(9)の計算した結果を保持することで演算量を削減することができる。また、解像度変換行列は行列 A_i, A_i' だけなので、格納に必要なメモリ量も小さく抑えることができる。

3.1.2 IMC 行列の対称性を利用した逆動き補償の高速化

また、一般に部分行列 A と A' の間には密接な関係がある。例えば $i=1$ の場合、 8×8 行列の A_1 と A_1' の関係を図3に示す。ただし、 A_1 と A_1' で要素が完全に一致する箇所を“+”，正負符号が反転する箇所を“-”，全く異なる箇所を“ ”で表している。例えば A_1 の左上の要素を a_{00} とすると、

$$a_{00} \neq a'_{00}$$

$$a_{01} = -a'_{01}$$

$$a_{31} = a'_{31}$$

のような関係がある。よって、式(10)のように A と A' で加算、減算をした行列には要素のなかに 0 が増え、式(8)に適用することで演算量の削減が可能となる。

$$\begin{cases} Z_i = A_i + A_i' \\ W_i = A_i - A_i' \end{cases} \quad (10)$$

	-	-	-	-	-	-	-
-			+		+		+
-				-		-	
-	+				+		+
-		-		-		-	
-	+		+				+
-		-		-		-	
-	+		+		+		

図3 A_i と A_i' の関係

式(8)に式(10)を代入すると、

$$Y = \begin{pmatrix} \{\alpha_j + \beta_j\} Z_i' + \{\alpha_j - \beta_j\} W_i' / 2 & \{\beta_j + \gamma_j\} Z_i' + \{\beta_j - \gamma_j\} W_i' / 2 \\ \{\delta_j + \varepsilon_j\} Z_i' + \{\delta_j - \varepsilon_j\} W_i' / 2 & \{\varepsilon_j + \lambda_j\} Z_i' + \{\varepsilon_j - \lambda_j\} W_i' / 2 \end{pmatrix} + D \quad (11)$$

$$\alpha_j = \{Z_j (R_{00} + R_{10}) + W_j (R_{00} - R_{10})\} / 2$$

$$\beta_j = \{Z_j (R_{01} + R_{11}) + W_j (R_{01} - R_{11})\} / 2$$

$$\gamma_j = \{Z_j (R_{02} + R_{12}) + W_j (R_{02} - R_{12})\} / 2 \quad (12)$$

$$\delta_j = \{Z_j (R_{10} + R_{20}) + W_j (R_{10} - R_{20})\} / 2$$

$$\varepsilon_j = \{Z_j (R_{11} + R_{21}) + W_j (R_{11} - R_{21})\} / 2$$

$$\lambda_j = \{Z_j (R_{12} + R_{22}) + W_j (R_{12} - R_{22})\} / 2$$

式(11)の括弧内の式は、それぞれ共通する部分があり、式(12)の計算した結果を保持することで演算数を削減できる。

上記は $i=j=1$ の場合を例に挙げたが、 i によって対称性の分布が異なり、 i が奇数より偶数、偶数より4の倍数のほうが対称な値を持つ要素が多く、計算量削減効果も大きい。よって i と j が異なる数値を持つ場合、2の因数が多い成分から抽出処理を行うことで全体の計算量を削減することができる。式(11),(12)は前からの乗算を先にしたほうが良い場合である。後ろからの乗算を先に計算したほうが良い場合、式(8)は式(10)を代入した後に Z_i, W_i について括弧で括り、式(13),(14)のようになる。

$$Y = \begin{pmatrix} \{Z_j (\alpha_i + \beta_i) + W_j (\alpha_i - \beta_i)\} / 2 & \{Z_j (\beta_i + \gamma_i) + W_j (\beta_i - \gamma_i)\} / 2 \\ \{Z_j (\delta_i + \varepsilon_i) + W_j (\delta_i - \varepsilon_i)\} / 2 & \{Z_j (\varepsilon_i + \lambda_i) + W_j (\varepsilon_i - \lambda_i)\} / 2 \end{pmatrix} + D \quad (13)$$

$$\alpha_i = \{(R_{00} + R_{10}) Z_i' + (R_{00} - R_{10}) W_i'\} / 2$$

$$\beta_i = \{(R_{01} + R_{11}) Z_i' + (R_{01} - R_{11}) W_i'\} / 2 \quad (14)$$

$$\gamma_i = \{(R_{02} + R_{12}) Z_i' + (R_{02} - R_{12}) W_i'\} / 2$$

$$\delta_i = \{(R_{10} + R_{20}) Z_i' + (R_{10} - R_{20}) W_i'\} / 2$$

$$\varepsilon_i = \{(R_{11} + R_{21}) Z_i' + (R_{11} - R_{21}) W_i'\} / 2$$

$$\lambda_i = \{(R_{12} + R_{22}) Z_i' + (R_{12} - R_{22}) W_i'\} / 2$$

なお、本提案方式によると演算数は小さく抑えられるが、ベースバンド変換方式と代数的には等価であるため、全く同一の画質を得ることができる。逆にベースバンド変換方式では逆 DCT、逆動き補償などの処理ごとに結果を整数に格納するため、不必要な丸め込み操作が何度も発生して僅かながら画質に影響するが、本方式では一括して処理するため丸め処理も最小限に抑えられる。

3.1.2 高周波情報削減による逆動き補償の高速化

逆動き補償後の解像度変換処理において高周波情報が失われることに注目し、 R_{mn} の 8x8 DCT 係数の低周波 4x4 成分以外の要素を省略するため、強制的に 0 にする。このとき、式(11)または式(13)の処理に必要な演算数(乗算数、加算数の和)は i, j の値に応じて変化する。このときの演算数を表 1 にまとめる。半画素精度の動きベクトルの場合、最大演算数は $i=j=1$ の場合と等しい。

表 1 提案方式の演算数

$j \backslash i$	1 or 7	2 or 6	3 or 5	4
1 or 7	2056	2008	2056	1552
2 or 6	2008	1976	2008	1520
3 or 5	2056	2008	2056	1552
4	1552	1520	1552	1216

(回 / macroblock)

3.2 動き補償に用いる動きベクトル(MV)の選択

動き補償をするとき、解像度変換して合成されたマクロブロック(MB)に対し、動きベクトル(MV)を与える必要がある。ここで、入力 MB に割り当てられていた MV を再利用することで動き予測処理を省略する。

縦横 1/2 に解像度変換した MB に対して入力 MB は 4 つ存在し、MV は MB のモードによって最大 4 つ存在する。これらの MV の中で、2 番目に小さい MV を縦横半分にして合成 MB の MV として採用する。ただし入力にイントラ MB が 2 つ以上ある場合はイントラ MB として扱う。結果として、入力動きベクトルが 1 つか 2 つの場合はイントラ MB になり、3 つ以上の場合は MV を割り当てて動き補償を行う。この MV 選択の有効性はシミュレーション実験で検証する。

4. DCT 領域処理のシミュレーション実験

4.1 各方式の PSNR による評価

4 つの MPEG1 ストリーム (VGA) を合成・縮小して 1 つ

の MPEG1 ストリーム (VGA) を生成する実験を以下の 4 つの方式について行った。

- (1) 提案方式 (IP 符号化, P ピクチャでのイントラ符号化含む)
- (2) ベースバンド方式
- (3) 参照方式 (提案方式, IP 符号化, P ピクチャでのイントラ符号化含まない)

なお、(3) についてはイントラブロック選択の有効性を検証するために用いた。また、PSNR 測定のための画像は原画像を平均値処理で縮小し合成したものをを用いた。

図 1 のような DCT 領域処理を実装し、PSNR と処理時間の計測を行った。実験に用いた PC は Pentium (933MHz), 256MB SDRAM, OS は Windows2000 である。入出力させる MPEG 動画のパラメータを表 2 に示す。入力は同一各パラメータで符号化された動画データを用いた。

PSNR 実験結果を図 4 に示す。提案方式では、ベースバンド変換方式と比較して最大で約 1.55dB ほど画質が劣化した。これは、提案方式では高周波を強制的に 0 にしていること、および割り当てた MV がベースバンドの動き予測で得られた MV よりサーチ精度が低下していることによる画質の劣化であると考えられる。また提案方式と従来方式を比較した場合、提案方式で高周波を強制 0 にしていること以外は代数的には等価である。そのため、提案方式が最大で約 0.15dB PSNR が劣化しているのは強制 0 の影響のみと考えられる。高周波の影響が少なくなる低ビットレートになるほど PSNR の差はなくなる。

次に提案方式と、参照方式(動き予測 MB のみの提案方式)を比較してみると、通常のほうが 0.15dB 前後 PSNR が高い。イントラ MB が現れるのはほとんどが動きの大きな映像部分

表 2 入出力符号化条件

	入力 MPEG	出力 MPEG
フレームサイズ	640x480	640x480
フレームレート	29.97fps	29.97fps
フレーム数	30 フレーム	30 フレーム
ビットレート	5Mbps	3Mbps
GOP 構造	M=1, N=15	M=1, N=15

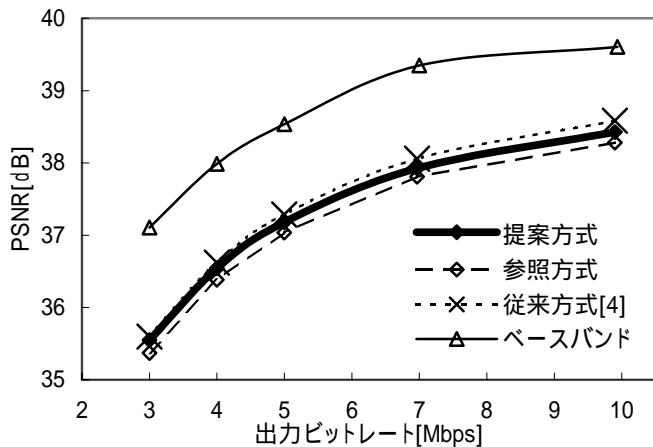


図4 各方式のPSNR

であり、ベースバンド方式の動き予測でもイントラ MB として符号化されているものが多い。イントラ MB の割り当ては妥当であると考えられる。

4.2 各方式による処理速度評価

演算回数について従来方式[4]と比較した結果を表3、図5に示す。なお、現在のプロセッサでは乗算と加算の処理負荷にそれほど差がないため、演算数は乗算回数と加算回数の和を用いた。

表からもわかるように IMC についてはベースバンド方式に比べ20 - 30%に削減しており、従来方式に比べると1/5-1/8程度で処理を行うことができた。また、全体処理時間は、ベースバンド方式に比べて約67%の処理で行うことができるため、減った。

提案動き補償方式は、ベースバンド方式のDCT、IDCTおよび動き補償と等価である。しかし提案方式では高周波情報削減による高速化ができない。そのため演算量は15840回/macroblockとなり、ベースバンド方式の6048回/macroblockの約2.6倍となる。しかし提案方式は動き予測をしないことで高速化が図れる。よって動き予測も行うベースバンド方式に比べて、提案動き補償方式は約51%の処理時間で行うことができた。

表3 各IMC方式にかかる演算数

	提案方式	従来方式[4]	ベースバンド
演算数	1216 ~ 2056 (20.1%~34.0%)	10144 (167.7%)	6048 (100%)

(回 / macroblock)

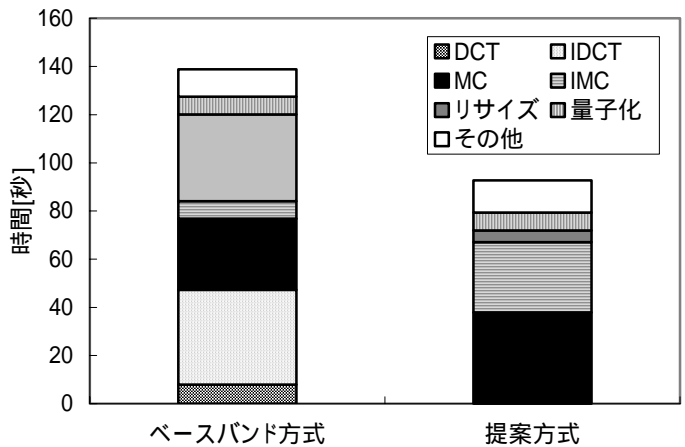


図5 各方式の処理時間およびその内訳

5. まとめ

DCT 領域で逆動き補償、解像度変換、動き補償の処理を行って多地点映像の合成を行う方法について検討した。高周波情報を削減することにより逆動き補償の高速化を図り、また動き予測情報を再利用することにより動き補償の高速化を図った。提案方式ではベースバンド方式の約66.8%に処理時間を削減できた。

今後は、より効果的な動きベクトルの選択方法による画質の向上、高速化についてさらに検討していく予定である。

参考文献

- [1] M. Baldi and Y. Ofek, "End-to-end delay of videoconferencing over packet switched networks," *IEEE Infocom '98*, Apr. 1998.
- [2] 瀧口, 加藤, 中島, "TV 会議用マルチ画面変換における DCT 領域でのサイズ変更の検討", 2003 年信学総大, 2003.
- [3] R. Dugad, "A Fast Scheme for Image Size Change in the Compressed Domain," *IEEE Trans. on circuits and Systems for Video Technology*, vol.11, No.4, Apr. 2001.
- [4] J. Song and B.L. Yeo, "A Fast Algorithm for DCT-Domain Inverse Motion Compensation Based on Shared Information in a Macroblock", *IEEE Trans. Circuits Syst. Video Tech.*, vol.10, pp767-775, Aug. 2000.