

# DCT 係数上での動画像解像度変換の高画質化に関する研究 A Study on Improving Image Quality of Moving Picture in DCT Coefficients

後藤 富朗<sup>†</sup>  
Tomio Goto

杉本 和彦<sup>‡</sup>  
Kazuhiko Sugimoto

花村 敏明<sup>†</sup>  
Toshiaki Hanamura

北村 正<sup>†</sup>  
Tadashi Kitamura

## 1. はじめに

画像の解像度には受信機の表示機能や各種アプリケーションに応じてさまざまな種類が存在する。これに伴い多くの解像度変換法が研究されている。画像の解像度変換は処理の容易さから一般に画像空間上で行われている。

また、静止画像や動画像の圧縮方式として、国際標準規格である JPEG (Joint Photographic Expert Group) [1] や MPEG (Motion Picture Experts Group) [2] が用いられている。これらの符号化方式には DCT 符号化が採用されており、DCT で圧縮されたデータの解像度変換法として DCT 係数上での変換が注目されている。JPEG や MPEG では、基底の長さが固定された復号化器を使用しているが、一般的な DCT 係数上での解像度変換では、基底の長さが変化してしまい、JPEG や MPEG の復号化器をそのまま使用することができない。また、固定して処理できても処理時間が増大したり、ブロックひずみが発生するなどの問題がある。

そこで本研究ではこれらの問題を解決する手法として、隣接するブロックの DCT 係数を用いた DCT 係数上での解像度変換法を提案する。隣接ブロックの DCT 係数を用いることで、ブロックひずみの発生しない解像度変換画像を得ることができ、提案手法を動画像に適用することで動画解像度変換の高画質化を図る。また、PSNR, GBIM による客観評価および復写画像を視覚的に比較することによる主観評価を行い、提案法の有効性について検討する。

## 2. DCT 係数上での解像度変換

### 2.1 従来法

$N \times N$  点の基底サイズを変更することなく解像度変換を行う方法が提案されている [3]。この方法を用いて縦横それぞれ  $Q$  倍に拡大する場合、 $N \times N$  点の DCT 係数 1 ブロック  $F_0$  より図 1 のように縦横  $Q$  倍に拡大し、それぞれ  $Q \times Q$  ブロックの DCT 係数  $F_{11} \sim F_{QQ}$  を式 (1) より行列演算で求める。ここで、 $U_1 \sim U_Q$  は DCT 基底より求めることができる  $N \times N$  点の行列である。

$$F_{ij} = U_j F_0 U_i^t \quad (i, j = 1, \dots, Q) \quad (1)$$

また、画像を  $R$  分の 1 に縮小する場合、同様に  $R \times R$  ブロックから 1 ブロックを求める。

この方法は解像度変換後のブロックサイズが変更されないため、JPEG 等の  $8 \times 8$  点の逆 DCT のみで復号化でき、行列演算を用いることにより処理時間も短縮できる。しかし、ブロック間の相関を考慮していないためブロックの境界部分でブロックひずみが発生し、拡大率が上がるにつれてブロックひずみも増加してしまう。

<sup>†</sup>名古屋工業大学 情報工学専攻

<sup>‡</sup>名古屋工業大学 電気情報工学科

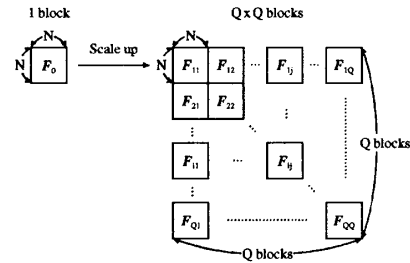


図 1: DCT 係数上の解像度変換 (従来法)

### 2.2 提案法

解像度変換時においてブロックひずみが発生するという問題を解決するため本研究では、単一ブロックだけでなく隣接ブロックを加えた複数ブロックを用い、ブロック間の相関を考慮した解像度変換法を提案する。提案法の拡大手順を縦横 2 倍の拡大を例にとり以下に示す。

- (a)  $N \times N$  点の DCT 係数  $F_1$  を式 (2) を用いて  $4 \times 4$  ブロックの  $N/4 \times N/4$  点 DCT 係数  $F'_1$  に変更する (図 2 (a) 参照)。

$$F'_1 = A F_1 A^t \quad (2)$$

ここで、 $A$  は基底変換行列であり、 $N=8$  の場合、式 (3) で表され、 $2 \times 2$  点の DCT 基底行列と  $8 \times 8$  点の DCT 基底行列の積より導出できる。

$$A = \begin{bmatrix} 0.50 & 0.64 & 0.46 & 0.22 & 0 & -0.15 & -0.19 & -0.13 \\ 0 & 0.05 & 0.19 & 0.36 & 0.50 & 0.54 & 0.46 & 0.27 \\ 0.50 & 0.27 & -0.46 & -0.54 & 0 & 0.36 & 0.19 & -0.05 \\ 0 & 0.13 & 0.19 & -0.15 & -0.50 & -0.22 & 0.46 & 0.64 \\ 0.50 & -0.27 & -0.46 & 0.54 & 0 & -0.36 & 0.19 & 0.05 \\ 0 & 0.13 & -0.19 & -0.15 & 0.50 & -0.22 & -0.46 & 0.64 \\ 0.50 & -0.64 & 0.46 & -0.22 & 0 & 0.15 & -0.19 & 0.13 \\ 0 & 0.05 & -0.19 & 0.36 & -0.50 & 0.54 & -0.46 & 0.27 \end{bmatrix} \quad (3)$$

- (b)  $N/4 \times N/4$  点の DCT 係数を拡大したいブロックに加え隣接ブロックを加えた計  $3 \times 3$  ブロック (計  $3N/4 \times 3N/4$  点) を用い拡大する (図 2 (b) 参照)。変換式は

$$G' = B G B^t \quad (4)$$

となる。ここで  $G$ ,  $G'$  はそれぞれ拡大前、拡大後の DCT 係数である。また拡大行列  $B$  は  $N=8$  の場合、式 (5) で表され、 $2 \times 2$  点の DCT 基底行列と 3 次畳み込み補間法の拡大行列と  $2 \times 2$  点の DCT 基底行列の積より導出できる。

$$B = \begin{bmatrix} -0.01 & 0.01 & 1.03 & 0.50 & -0.01 & -0.01 & 0 & 0 \\ 0.01 & -0.01 & -0.03 & 0.50 & 0.01 & 0.01 & 0 & 0 \\ 0 & 0 & 0.75 & -0.78 & 0.25 & 0.28 & 0 & 0 \\ 0 & 0 & 0.25 & -0.23 & -0.25 & -0.28 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

(c) 4x4 ブロックの  $N/4 \times N/4$  点の DCT 係数  $G_m$  を式 (6) を用いて  $N \times N$  点の DCT 係数  $G'_m$  に変換する (図 2 (c) 参照).

$$G'_m = A^t G_m A \quad (6)$$

このように行列演算を行うことで、縦横それぞれ整数倍の拡大を実現することができる。また、拡大行列  $B$  を用いて拡大操作を行うことで得られる拡大画像は、画像空間上で 3 次畳み込み補間により得られる画像に相当し、この拡大行列  $B$  を変更することで他の補間法による拡大画像も得ることができる。

次に、提案法の縮小手順を縦横 2 分の 1 の縮小を例にとり示す。  $N \times N$  点の DCT 係数  $2 \times 2$  ブロックを式 (7) を用い  $N \times N$  点の DCT 係数 1 ブロックに変換する。

$$F' = C F C^t \quad (7)$$

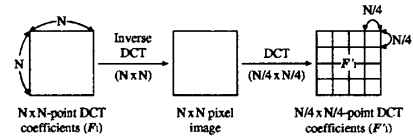
ここで  $F, F'$  はそれぞれ縮小前、縮小後の DCT 係数である。また、 $C$  は縮小行列であり、 $N=8$  の場合、式 (8) で表される。

このように複数ブロックから 1 つのブロックを求めることで、整数分の 1 の縮小も行うことができる。また、任意の倍率の解像度変換は拡大と縮小を組み合わせることで実現できる。

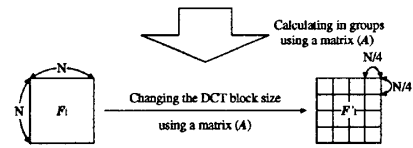
### 3. 動画像の高画質化

DCT 係数上での解像度変換を動画像の各フレームに対して行うことで動画像の解像度変換を実現できる。しかし、各フレームを独立に解像度変換すると、解像度変換後の画像にちらつきが生じ、画質が劣化してしまう。これは、各フレームを独立に処理することにより、解像度変換によって新たに作り出された画素の時間軸方向の相関がなくなることに起因する。また、一般に動画像のあるフレームとその前後のフレームは相関が非常に高いことが知られている。そこで本研究では、動画像に対して時間軸方向にフィルタ処理を施すことで、ノイズの除去を行い、動画像の高画質化を図る。しかし、この処理は動画像内の物体の動きを考慮しないと動く物体のエッジ部分がぼやけると考えられる。そこで、動きベクトルを用いたフィルタ処理によりエッジのぼやけを解消するとともにノイズ除去を行う。この処理は、動画像の各フレームを  $8 \times 8$  点のブロックに分割し、ブロックマッチングにより動きベクトルを求め、動きベクトルを考慮してブロック毎にフィルタ処理を施す。ただし、ブロックマッチングによる差分データがある基準値より大きい場合には、誤った動きベクトルが検出されたと判断し、処理を施さずフレーム内の画素をそのまま用いる。

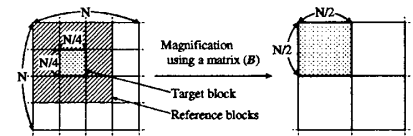
$$C = \begin{bmatrix} 0.50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.50 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.45 & 0.20 & -0.03 & 0.01 & 0 & -0.01 & 0.01 & -0.04 & -0.45 & 0.20 & 0.03 & 0.01 & 0 & -0.01 & -0.01 & -0.04 \\ 0 & 0.49 & 0 & 0 & 0 & 0 & 0 & -0.10 & 0 & -0.49 & 0 & 0 & 0 & 0 & 0 & 0.10 \\ -0.16 & 0.39 & 0.24 & -0.04 & 0 & 0.03 & -0.10 & -0.08 & 0.16 & 0.39 & -0.24 & -0.04 & 0 & 0.03 & 0.10 & -0.08 \\ 0 & 0 & 0.46 & 0 & 0 & 0 & -0.19 & 0 & 0 & 0 & 0.46 & 0 & 0 & 0 & -0.19 & 0 \\ 0.11 & -0.17 & 0.35 & 0.20 & 0 & -0.14 & -0.15 & 0.03 & -0.11 & -0.17 & -0.35 & 0.20 & 0 & -0.14 & 0.15 & 0.03 \\ 0 & 0 & 0 & 0.42 & 0 & -0.28 & 0 & 0 & 0 & 0 & 0 & -0.42 & 0 & 0.28 & 0 & 0 \\ -0.09 & 0.14 & -0.17 & 0.36 & 0 & -0.24 & 0.07 & -0.03 & 0.09 & 0.14 & 0.17 & 0.36 & 0 & -0.24 & -0.07 & -0.03 \end{bmatrix} \quad (8)$$



(a) DCT 係数の基底サイズ変更 ( $N \times N \rightarrow N/4 \times N/4$ )



(b) 縦横 2 倍に拡大



(c) DCT 係数の基底サイズ変更 ( $N/4 \times N/4 \rightarrow N \times N$ )

図 2: 提案法の縦横 2 倍の拡大操作の流れ

## 4. 実験結果

動画像への解像度変換および高画質化の有効性を確認するため、それぞれの手法を個別に実験する。

### 4.1 解像度変換

実験画像として  $352 \times 240$  ピクセル、140 フレームの Mobile & Calendar 画像を用い、全フレームを縦横 2 分の 1 にダウンサンプリングした後、縦横 2 倍に拡大した画像のフレーム毎の PSNR 及び GBIM [4] を図 3 に、実験画像を縦横 2 倍に拡大した画像の一部を図 4 に示す。GBIM は画像中のブロックひずみ成分を定量化する指標であり、値が小さいほど画像中に含まれるブロックひずみが少ないことを表す。図 3 において PSNR が従来法で約 16 dB、提案法で約 18 dB となっており、比較的低い値となっている。これは、ダウンサンプリングによる情報の欠落と考えられる。しかし、本実験での目的は、より高い PSNR の画像を得ることではなく、解像度変換手法におけるブロックひずみ混入度合いの調査であるため、全体的な PSNR の低さは問題とはならない。また、PSNR においては提案法の方が約 2dB 高くなっており、GBIM においては提案法の方が低くなっており、確認できる。故に PSNR, GBIM ともに改善できていることが確認できる。また、図 4 より従来法ではブロックひずみが現れているのに対し、提案法では現れていないことが確認できる。

表 1: 演算回数の比較 (拡大率: 2 倍)

Method	加算	乗算	合計
提案法	7680	7680	15360
従来法	2368	2368	4736
画像空間	6112	10816	16928

## 4.2 演算回数の比較

8×8 点の DCT 係数 1 ブロックを縦横 2 倍に拡大して、8×8 点の DCT 係数 4 ブロックを求めるのに必要な演算回数を比較した。実験結果を表 1 に示す。表 1 より演算回数を比較した結果、提案法は従来法に比べ演算回数が多い。これは、隣接ブロックを参照したため演算量が増えたためである。また、提案法と画像空間上での拡大の演算量を比較すると、提案法の方が少ない処理量で演算を行うことができています。これは、画像空間上での拡大では、DCT、逆 DCT の操作が必要となるが、DCT 係数上での解像度変換はその操作が必要ないためである。さらに、拡大率を上げるにつれ、この差は大きくなるため、提案手法が画像空間上の拡大に比べ高速に処理できることが確認できる。

## 4.3 高画質化

動きベクトルを用いた高画質化の有効性を確認するため、動きを考慮しない場合と考慮した場合の比較を行った。実験画像として 352×288 ピクセル、300 フレームの Hall monitor 画像を用い、高画質化を行った画像を図 5 に示す。図 5 より、動きを考慮しない場合には画像のエッジ部分が不鮮明になっているが、動きを考慮した場合には鮮明で良好な画像となっていることが確認できる。

## 5. むすび

本研究では、拡大時に発生するブロックひずみの解消を目的として隣接ブロックを用いた DCT 係数上での解像度変換を動画に適用した。また、拡大時に認知しやすくなるノイズを軽減するため、動画のフレーム間情報を用いた高画質化について検討した。その結果、ブロックひずみを発生させることなく画像を拡大することができた。また、動画の高画質化において、動きを考慮することにより画像をぼやけさせることなく、処理が施せることが確認できた。これらにより、ちらつきを軽減させつつブロックひずみを発生させることのない解像度変換が可能となり、提案法が動画の拡大において有効であることが確認できた。

しかし、DCT 係数上での解像度変換は周辺ブロックを参照し、また、動画の高画質化は前後のフレームを参照するため使用メモリ量や処理時間が増加してしまう。そこで今後の課題としてアルゴリズムの簡略化による使用メモリの削減および処理の高速化が挙げられる。

## 謝辞

本研究の一部は、文部科学省科学研究費補助金 (若手研究 (B) 課題番号 15760261) および電気通信普及財団の助成によるものである。ここに記して謝意を表す。

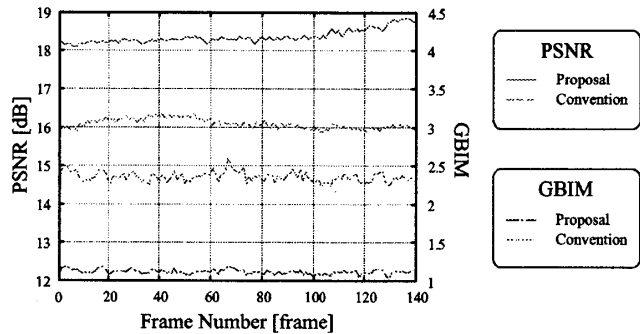


図 3: 各フレームにおける PSNR 及び GBIM の関係

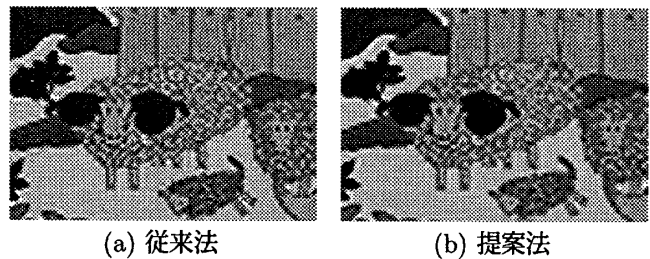


図 4: 拡大画像の一部 (100 フレーム)

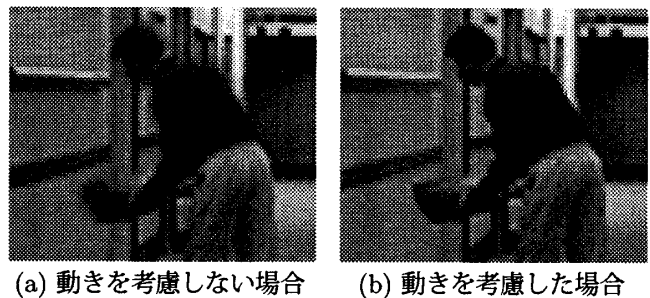


図 5: 高画質化画像の一部 (100 フレーム)

## 参考文献

- [1] Gregory K. Wallace, "The JPEG Still Picture Compression Standard", IEEE Transactions on Consumer Electronics, Dec. 1991.
- [2] ISO/ICE 11172-2, "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbit/s - Part 2: Video", 1993.
- [3] 宮本 義弘, 赤岩 芳彦, "DCT 係数上での符号化画像のサイズ変換", 電子情報通信学会論文誌 D-II Vol.J-82-D-II No.1 pp.53-60, Jan. 1999.
- [4] H.R.Wu, M.Yuen, "A generalized block-edge impairment metric for video coding", IEEE Signal Processing Letters, vol.4, no.11, pp.317-320, Nov. 1997.