B-004

ウェーブレット変換に基づく学習型超解像の GPU による高速化手法

A Technique for GPU-Based Acceleration of Learning-Based Super-Resolution Using Wavelet Transform

坂本 博之† 佐々木 仁† 雫 譲† 黒木 修隆† 廣瀬 哲也† 沼 昌宏†
Hiroyuki Sakamoto Jin Sasaki Yuzuru Shizuku Nobutaka Kuroki Tetsuya Hirose Masahiro Numa

1. はじめに

辞書と呼ばれる事前に作成したデータベースを用いて，劣化の少ない高解像度画像を推定・復元する学習型超解像 [1] という技術がある。その 1 つであるウェーブレット変換に基づく学習型超解像 [2] では，まず学習処理において，離散ウェーブレット変換（DWT：Discrete Wavelet Transform）により抽出したエッジに対する主成分分析によってノイズ因子を除去した後，二分木構造化して辞書とする。一方，作成された辞書に基づく超解像処理においては，エッジに最も近い辞書のデータから高解像度画像を推定し，超解像画像を求める。この学習型超解像は，演算量が大きいため，CPU（Central Processing Unit）のみで処理を実行すると，処理に時間を要する点に問題がある。

本稿では，ウェーブレット変換に基づく学習型超解像における超解像処理について，一般の CPU よりも高い並列処理能力をもつ GPU（Graphics Processing Unit）を処理の一部に用いることで，学習型超解像の超解像処理の高速化を実現する手法について提案する。

2. ウェーブレット変換に基づく学習型超解像

ウェーブレット変換に基づく学習型超解像における超解像処理では，入力画像の輝度成分に DWT を適用し，抽出されたエッジ成分に辞書を用いて高解像度なエッジ成分を推定する。その離散ウェーブレット逆変換（IDWT：Inverse Discrete Wavelet Transform）を適用し，輝度成分の超解像画像を求める。

DWT とは，入力画像の垂直・水平方向にそれぞれに 2 種類のフィルタのいずれかを適用し，生成した画像の信号を 1 画素おきに間引くことにより，原画像の 1/4 倍の大きさの低域画像と，それぞれ垂直・水平・斜め方向のエッジ成分をもつ高域画像 3 枚の計 4 枚の画像を段階的に生成する多重解像度解析である。また，DWT により生成した画像を用いて IDWT を適用することで，もとの画像を復元できる。学習型超解像では，高解像度化の処理を行うため信号の間引きは行わず，DWT を適用して生成する画像は垂直・水平方向のエッジ成分をもつ高域画像 2 枚である。IDWT の処理では，入力画像と 2 枚の高域画像の各画像の信号間に 0 を挿入するアップサンプリング処理を行い，IDWT 用のフィルタを適用する。

辞書探索では，DWT で求めたエッジ成分から，4×4 pixel のブロックを切り出し，二分木辞書のデータ群を根から葉まで探索し，最も類似するデータを利用することで高解像度なエッジ成分を推定する。

3. GPU を用いた高速化手法

本章では，ウェーブレット変換に基づく学習型超解像の主な処理である，DWT，IDWT，辞書探索に GPU を用いた高速化手法について述べる。

3.1 離散ウェーブレット変換

学習型超解像において，DWT の主な処理はフィルタ処理である。よって，出力画像の画素数分のスレッドを生成して並列にフィルタ処理を行うことで，DWT の高速化を実現する。

3.2 離散ウェーブレット逆変換

IDWT の主な処理は，アップサンプリング処理とフィルタ処理である。しかし GPU では，メモリへのアクセスに相対的に時間を要する。そこで，フィルタ処理に用いるフィルタを偶数番号の要素と奇数番号の要素で分割し，出力画像の画素座標の偶奇により用いるフィルタを変更することにより，アップサンプリング処理を取り除き，メモリアクセス回数を削減する。このとき，ある出力画像の画素とその近くの画素において，適用するフィルタが異なるだけで，その画素値を求めるときに用いる入力画素の集合は同じ集合を用いることがある。そこで，入力画像の画素数分のスレッドを生成し，並列にフィルタ処理を行うことで，さらにメモリアクセルの回数を削減し，IDWT の高速化を実現する。

3.3 辞書探索

辞書探索では，以下の手順により，低解像度なエッジ成分から高解像度なエッジ成分を求める。

1) ブロックの切り出し
2) ノルム計算
3) 二分木探索
4) 逆主成分分析
5) ブロックの貼り合わせ

1) の処理では，低解像度なエッジ成分から 4×4 pixel のブロックを切り出し，主成分分析を行う。2) の処理では，切り出したブロックの要素の二乗和をとり，その平方根をノルムとする。3) の処理では，事前に作成した二分木構造の辞書のデータと，ノルムにより正規化されたブロックとの差の二乗和である SSD（Sum of Squared Difference）を計算し，SSD が最小となる辞書中のブロックデータを求める。ただし，ブロックにおけるノルムの大きさにより，そのブロックに二分木探索を実行するか否かの条件判断を行う。実行しないと判断したブロックには，以降の処理を行わない。4) の処理では，二分木探索で求めたデータにノルムをかけて逆主成分分析を行い，ブロックの復元を行う。5) の処理では，復元したブロックを高解像度な高域画像の対応する座標に貼り合せる。

これらすべての処理に対して，GPU を用いて並列処理を実行することで高速化を実現する。1) の処理では，（ブロ

†神戸大学大学院工学研究科，
Graduate School of Engineering, Kobe University

ックの数）×（ブロックの要素数）分のスレッドを生成して，並列処理を行う。2)，3) の処理では，ブロックの数だけスレッドを生成して，並列処理を行う。また，3) の処理において，二分木探索を行わないブロックの要素の値をすべて 0 にする。4) の処理では（ブロックの数）×（ブロックの要素数）分のスレッドを生成する。5) の処理では，高解像度な高域画像の画素数分のスレッドを生成して，並列処理を行う。

4. 実験による評価と考察

4.1 評価項目と実験環境

提案手法を評価するために，図 1 に示す 256×256 pixel, 512×512 pixel，768×512 pixel の 3 種類の大きさの異なる画像各 3 枚，計 9 枚の画像で学習型超解像を適用し，縦横 2 倍の 4 倍拡大画像と縦横 4 倍の 16 倍拡大画像を生成する際の，従来手法と提案手法での処理時間について評価する。輝度成分 Y では超解像を実行し，色差成分 Cb，Cr では Bicubic 法により拡大を実行する。また 16 倍拡大は，4 倍拡大した画像に対して再度 4 倍拡大を実行することで求める。実験には，GPU として GeForce GT 240 を用い，CPU として Intel Core i5-650 3.20 GHz，RAM 3.24 GB を用いた。

辞書には，自然画像から 4×4 pixel で作成したブロックデータ 20 万組から作成し，従来手法，提案手法ともに同じ辞書を適用する。

4.2 評価結果

表 1 に 4 倍拡大を実行した時の処理時間を示し，表 2 に 16 倍拡大を実行した時の処理時間を示す。表 1 および表 2 より，すべての画像で高速化を実現しており，4 倍拡大では 52.10 倍から 73.83 倍の高速化，16 倍拡大では 61.77 倍から 82.21 倍の高速化を実現した。また，提案手法による 256×256 pixel 画像の 4 倍拡大については，30 fps の動画に適用可能な時間で処理を終えた。処理時間は同じサイズの画像でも異なり，処理時間が長い画像ほど高速化倍率は小さい。これは，画像の種類により辞書探索の処理内容が変わるためであり，二分木探索を行うブロックが多いと演算量や条件分岐が増加し，処理時間が長くなる。GPU では条件分岐が発生すると効率が低下するため，条件分岐が増加する二分木探索を行うブロックが多い画像では高速化倍率が低下する。また，拡大率が大きくなるほど高速化倍率も向上している。この要因としては，提案手法では，CPU と GPU 間のデータ転送を必要としており，演算結果を CPU 上のメモリに転送するまでの演算量が，4 倍拡大のときより 16 倍拡大のときでは増加したため，データ転送の時間が全体の処理時間に占める割合が小さくなったことや，4 倍拡大を実行したことにより，もう一度 4 倍拡大を実行する時の全ブロック中で二分木探索を行わないブロックの割合が増加したためと考えられる。

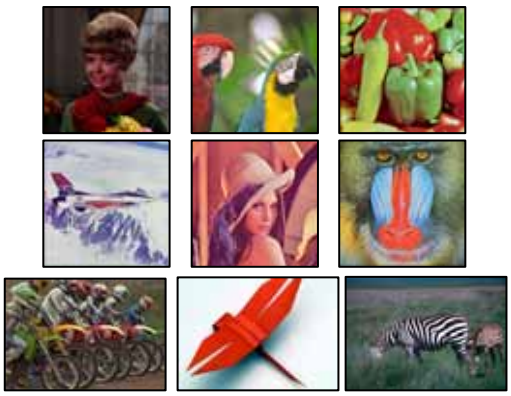

図 1　実験に用いた画像

表 1　4 倍拡大の処理時間 (ms)

評価画像	従来手法	提案手法	高速化倍率 (倍)
girl	741.10	12.78	57.99
parrots	757.86	13.65	55.52
pepper	764.29	14.67	52.10
airplane	3,066.57	45.39	67.11
lena	3,046.24	45.39	67.11
mandrill	3,582.97	67.16	53.35
bikes	5,056.88	96.01	52.67
origami	4,443.55	60.19	73.83
zebra	4,860.32	80.99	60.01

表 2　16 倍拡大の処理時間 (ms)

評価画像	従来手法	提案手法	高速化倍率 (倍)
girl	3,332.57	48.60	68.57
parrots	3,372.16	51.22	65.84
pepper	3,359.36	52.64	63.82
airplane	13,703.55	180.78	75.80
lena	13,951.03	175.84	79.34
mandrill	15,176.08	245.70	61.77
bikes	21,307.55	326.75	65.21
origami	20,943.08	254.76	82.21
zebra	21,156.68	298.38	70.91

5. まとめ

本稿では，ウェーブレット変換に基づく学習型超解像の超解像処理について，高い並列処理性能をもつ GPU を処理の一部に用いることで高速化を実現する手法を提案した。提案手法により，4 倍拡大を実行したとき最大 73.83.倍，16 倍拡大を実行したとき最大 82.21 倍の高速化を実現した。

今後の課題として，辞書として有効な学習用画像の検討や処理時間の削減が挙げられる。

参考文献

[1] W. Freeman, E. Pasztor, and O. Carmichael, “Leaning low-level vision”, International Journal of Computer Vision, vol. 40, no. 1, pp. 25-47, 2000.

[2] 中矢知宏, 近松慎伍, 黒木修隆, 廣瀬哲也，沼 昌宏，“ウェーブレット係数の主成分分析を用いた学習型超解像”，第 9 回情報科学技術フォーラム (FIT2010)，pp. 299-300，2010 年 9 月.