

# 重複保持法に基づくFFT全探索ブロックマッチングの並列実装

Parallel Implementation of FFT Based Full-Search Block Matching Algorithm Using an Overlap-Save Method

土橋 俊之 李 禎 貴家 仁志  
Toshiyuki Dobashi Zhen Li Hitoshi Kiya

首都大学東京大学院システムデザイン研究家情報通信システム学域  
Dept. of Information and Communication Systems, Tokyo Metropolitan University

## 1 はじめに

ブロックマッチングは、画像圧縮や雑音除去、画像復元など、多くの分野で応用されている。ブロックマッチング法の一つに、FFTを用いて全探索SSDブロックマッチングを高速に実行するアルゴリズムがある[1]。また、このFFT全探索ブロックマッチングにおいて、探索領域を分割して実行する方法も提案されている[2]。しかし、探索領域の分割はメモリ使用量の削減や計算の効率化を目的としており、並列処理の有効性は確認されていなかった。

本稿では、文献[2]の手法をマルチコアCPU上で実装し、その有効性を確認した。

## 2 FFTによる高速ブロックマッチング法

ブロック信号  $b(x, y)$  のサイズを  $A \times B$ 、探索領域信号  $f(x, y)$  のサイズを  $M \times N$ 、シフト量を  $u, v$  とすると、SSDは以下の式で定義される。

$$SSD_{b,f}(u, v) = \sum_{x=0}^{A-1} \sum_{y=0}^{B-1} \{f(x+u, y+v) - b(x, y)\}^2,$$

$$u \in [0, M-A+1], v \in [0, N-B+1],$$

右辺を展開すると、以下のように変形できる。

$$SSD_{b,f}(u, v) = C_b - 2cor_{g_b, f} + S_{f^2}(u, v),$$

ただし、

$$C_b = \sum_{x=0}^{A-1} \sum_{y=0}^{B-1} \{b(x, y)\}^2,$$

$$S_{f^2}(u, v) = \sum_{x=0}^{A-1} \sum_{y=0}^{B-1} f^2(x+u, y+v).$$

$cor_{g_b, f}(u, v)$  は信号  $f(x, y)$  と信号  $g_b(x, y)$  との相互相関であり、 $g_b(x, y)$  はブロック信号  $b(x, y)$  をゼロ詰めして信号  $f(x, y)$  と同じサイズに拡張した信号である。この相互相関はFFTを用いて高速に計算することが可能である[1]。また、 $S_{f^2}(u, v)$  は再帰的FIRフィルタの計算により効率的に求めることができる。

### 2.1 重複保持法に基づく相互相関の計算

重複保持法とは、あるデータに対する直線たみ込みを、データを分割して実行する方法の一つである。文献[2]では、探索領域信号を重複保持法に基づいて分割することで、相互相関を効率的に計算している。さらに、FFTの計算では複素数を計算することができるため、二つの分割信号の相互相関計算を一回で実行している。

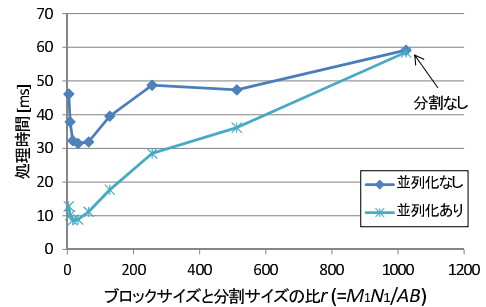


図1 ブロックサイズと分割サイズの比  $r$  と処理時間の関係

本稿では、相互相関の計算に加え、 $S_{f^2}(u, v)$  の計算も分割信号ごとに実行することで、計算の更なる並列化を行っている。

## 3 評価

前節のアルゴリズムを Visual C++ 2010 と OpenCV 2.2 とを用いてマルチスレッドで実装し、 $16 \times 16$  画素のブロック画像と  $512 \times 512$  画素の探索領域画像を用いて、評価した。分割信号のサイズ  $M_1 \times N_1$  を変化させたときの処理時間を図1に示す。横軸の  $r$  は、ブロックサイズ  $A \times B$  と分割サイズの比である。また、使用したCPUは intel Core i5 2500K (4コア 3.3GHz) である。

図1から、最適な分割数が存在することがわかる。今回の評価では、 $r = 8$  のときが最適な分割数であった。 $r = 8$  のときの処理時間は、並列化しない場合は 37.9 ms、並列化した場合は 10.2 ms であり、並列化により 3.72 倍高速化された。以上の結果から、並列処理が効果的であることを確認した。

## 4 おわりに

本稿では、重複保持法に基づくFFT全探索ブロックマッチングを並列実装し、その有効性を確認した。4コアCPUにおいて、並列化によって最大で3.72倍高速化されたことを確認した。

### 参考文献

- [1] Z. Li and H. Kiya, "An FFT-based full-search block-matching algorithm with sum of squared difference criterion," IEICE Trans. Fundamentals, vol.E93-A, pp.1748-1754, Oct. 2010.
- [2] H. Sasaki, Z. Li, and H. Kiya, "FFT-Based Full-Search Block Matching Using Overlap-Add Method," Proc. Picture Coding Symposium, no.P4-28, pp.586-589, Nagoya, Japan, 10th Dec. 2010.