

SAR 画像再生処理の並列化 ～改善したコーナーターン法の適用評価～

5H-7

和泉 秀幸[†] 佐々木 和司[‡] 佐藤 裕幸[†]

[†] 三菱電機 (株) 情報技術総合研究所 [‡] 三菱電機 (株) 鎌倉製作所

1 はじめに

SAR(Synthetic Aperture Radar:合成開口レーダ)は、天候を問わずに高い分解能で地表を撮像できるセンサである [1]。SAR では、人間が理解可能な画像を生成する画像再生処理が必要になる。この処理は、画素あたりの演算量が多く、かつ画像サイズが大きいため、我々は、SMP(Symmetric Multi-processor)でマルチスレッドプログラムによる並列化(高速化)を進めている。

SAR 画像再生の 1 部分処理であるコーナーターンは、単純に並列化するとキャッシュミスを起こしやすい問題があった。我々は、キャッシュのヒットを考慮したコーナーターンの高速化法を考察し、部分処理単体での処理速度向上や台数効果改善を確認した [2]。本稿では、この改善したコーナーターン法を SAR 画像再生処理に組込む方法を検討し、実機上で評価した結果を示す。

2 SAR 画像再生処理とコーナーターン

種々の SAR 画像再生アルゴリズムのうち、レンジ・ドップラ・アルゴリズムを対象に、処理の高速化を検討している。この画像再生処理では、FFT、や IFFT といった一連の処理でデータを次々に加工していく(図 1 参照)ため、部分処理やサンプラーチン単位での並列化戦略を取りづらい(効果が小さい)。一方、データサイズが大きく、データ間の依存関係も小さいため、各部分処理単位(FFT や IFFT)でデータ並列による並列化を行えば良いことがわかっていてる。

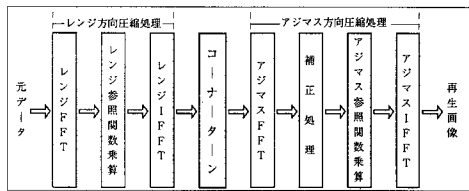


図 1 SAR 画像再生処理の流れ

ただし、各部分処理単位で、画像領域でのアクセス方向(レンジ方向とアジマス方向)が異なる(図 2 参照)。そこで処理を効率良く行うために、2次元画像データのメモリ上の配置を転置するコーナーターンを行う。一般に、処理負荷が高いレンジとアジマス方向の FFT や IFFT を

効率良く実行するため、レンジ圧縮処理とアジマス圧縮処理の間で行う。

なお、コーナーターンを実施しない場合には、単純な画素単位のコーナーターン(従来法)で実行した場合と比較しても、8プロセッサ使用時で約5倍ほど実行性能が低下することを確認している。

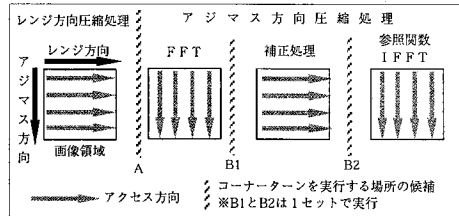


図 2 メモリアクセス方向

2.1 改善したコーナーターン法

コーナーターンは、単純に並列化するとキャッシュミスを起こしやすい。我々は、このキャッシュミスを軽減するために、キャッシュのラインにちょうど納まる幅の画素数を 1 辺とする正方形を“画像ブロック”として定義し、この単位で処理をプロセッサに分割してコーナーターンを並列実行する方法を考えた(図 3 参照)。

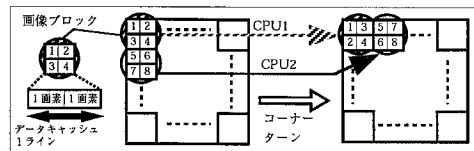


図 3 画像ブロック単位でのコーナーターン

このコーナーターン法では、階層型のキャッシュの場合、まず、プロセッサに最も近い最上位のキャッシュで画像ブロックを決定する。その後、下位のキャッシュで、ラインにちょうど納まる幅の“1つ上位のキャッシュの画像ブロック”数を 1 辺とする正方形を画像ブロックとして順次決定する。下位の画像ブロックでは、上位の画像ブロックのコーナーターンを繰り返す形式で実行する。画像ブロック内とブロック間のアクセス順序は、キャッシュの総容量(ライン数)から決めていく。

また、処理画像のサイズとキャッシュへのデータ読み込み方式(set associative方式など)により、画像ブロック内で同じキャッシュラインを使用するケースがある。この場合は、キャッシュラインの衝突を回避するためのオフセット領域を、処理対象画像に加える改善を施す [2]。

Parallelization of SAR Image Reconstruction on SMP: Applying The Optimized Corner-turn which Improving Cache Access Information Technology R&D Center, Mitsubishi Electric Corporation, 5-1-1 Ofuna, Kamakura, Kanagawa, 247-8501, Japan

今回は、MIPS R10000(250MHz) × 8 プロセッサ構成のCC-NUMA (cache coherent non-uniform memory access) アーキテクチャである SGI ORIGIN 2000 を対象とした。MIPS R10000 は 1 次キャッシュと 2 次キャッシュで構成され、ラインサイズはそれぞれ 32byte と 128byte である。SAR 画像の 1 画素は 8byte のため、4 画素を 1 辺とする正方形を「1 次画像ブロック」、4 個の 1 次画像ブロック (16 画素) を 1 辺とする正方形を「2 次画像ブロック」とする。

3 計測条件：コーナーターンの組み込み方法

コーナーターンを行う場所の候補として、レンジ圧縮処理とアジマス圧縮処理の間 (A) と、補整処理の前後 (B1,B2) の計 3 箇所がある (図 2 参照)。コーナーターンは、メモリの配置を並べ変えるだけであり、オーバーヘッドである。実行性能を向上させる場合は、コーナーターンにより各部分処理で実行時間が改善する量とコーナーターンにかかる時間がトレードオフとなる。これを検証するため、A で 1 回だけコーナーターンを行うケース (CT1) と A,B1,B2 で 3 回行うケース (CT3) の時間を比較する。

また、改善したコーナーターン法を画像再生処理へ組み込む方法として、コーナーターンを次の処理と結合して実行する方法 (図 4 参照) を考えた。この方法は、コーナーターン実行中にキャッシュに読み込んだデータを次処理で利用でき、かつコーナーターン直後のバリア同期を減らす利点がある。効果を証するため、次処理と結合する方法 (M1) と、単純にコーナーターンを実行後 (バリア同期後) に次処理を実行する方法 (M2) の実行時間を比較する。

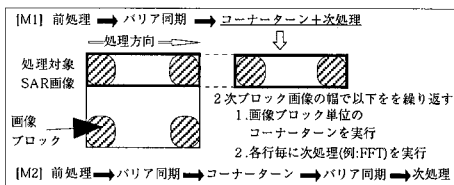


図 4 次の処理に結合して実行する方法

4 性能計測

単純な画素単位のコーナーターン (従来法) と改善したコーナーターン法 (改善法) を SAR 画像再生処理に組み込んで、実行時間を計測した。処理対象画像のサイズは 2048 × 16384 画素 (キャッシュラインの衝突発生ケースのため、改善法では、8 画素のオフセットを加えた 2056 × 16392 画素)。各処理の並列化は SPL (Simple Parallel Library) [3] で行った。

まず、コーナーターンの数回を、性能差が現れるアジマス方向圧縮処理の時間で比較する (図 5 参照)。従来法では、補整処理での性能改善量よりもコーナーターンの時

間が大きくなり、従来法 CT3 の性能が従来法 CT1 よりも

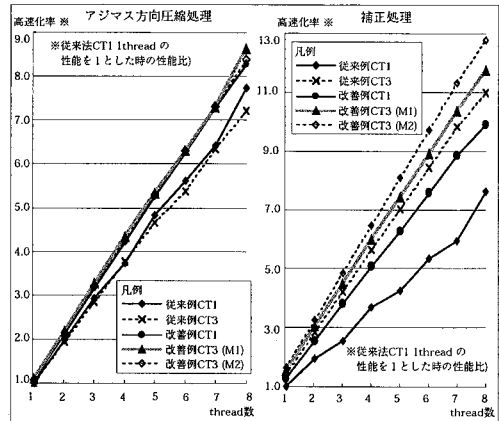


図 5 アジマス圧縮処理と補整処理の性能

低下した。一方で、改善したコーナーターンを使った場合、コーナーターンが高速化され、改善法 CT3 の性能が改善法 CT1 より向上する。

次に、コーナーターンの組み込み方法を評価する。同じ改善法 CT3 で組み込み方法だけを変えた、M1 と M2 のアジマス方向圧縮処理時間を比較すると、M1 の“次処理を結合した方法”で約 2% ほど性能が良く、今回考えた組み込み方法が効果的であることが確認できた。この改善量は、平均的なバリア同期 1 回分の時間に比べて十分大きいので、次処理でのキャッシュのヒット率改善の効果が主要因だと考える。

なお、前後にコーナーターンを実施しない、改善法 CT1 と従来法 CT1 の補整処理時間で差が生じている。これは、前後でコーナーターンを行わない補整処理では、キャッシュライン衝突が起きるようなメモリのアクセスを行うため、改善法でキャッシュライン衝突回避用に加えたオフセットの波及効果を受けたと考えている。

5 まとめ

SAR 画像再生処理に対して、改善したコーナーターン方法を組み込む方法を検討し、実機上で評価した。この結果、次処理に結合させ、レンジ圧縮とアジマス圧縮の間と、補整処理前後の合計 3 回コーナーターンを行う方法で、性能が良いことを確認した。また、詳細は省略したが、上記の方法を適用することで、画像再生処理全体の実行時間を約 12% 改善できることを確認している。

参考文献

[1] 春野 信義: 合成開口レーダ, 日本リモートセンシング学会誌, vol.1, no.1, pp.49-107, (1981).
 [2] 和泉, 佐々木, 水野, 中島: キャッシュラインを考慮したコーナーターン法の改善, 情報研究会 HPC 85-16, Mar. 2001.
 [3] 福地, 石塚, 和泉: マルチプロセッサ対応 UNIX 上での並列プログラム開発支援環境の開発, 第 48 回情報全国大会, 2G-9, (1994).