

# GPUによる合成開口レーダ画像再生処理の評価

## Evaluation of SAR Image Processing using GPUs

佐藤 裕幸<sup>†</sup>  
Hiroyuki Sato

高橋 勝己<sup>†</sup>  
Katsumi Takahashi

若山 俊夫<sup>†</sup>  
Toshio Wakayama

尾崎 敦夫<sup>†</sup>  
Atsuo Ozaki

### 1. はじめに

合成開口レーダ (SAR : Synthetic Aperture Radar) の画像再生処理は非常に時間を要する。そのため、クラスターサーバを用いた並列処理による高速化が行われてきたが、装置規模が大きくなるという課題がある。近年、グラフィック用プロセッサ (GPU : Graphics Processing Unit) を汎用の処理へ活用する GPGPU (General Purpose GPU) [1] が、安価に高速化を図れる手段として注目を浴びている。そこで我々は、SAR 画像再生処理に、この安価で高速な GPU を適用することで、再生処理装置の小型化を図る研究開発を行った[2]。

GPU ではホストマシンとのデータ転送が演算速度に比べて相対的に低速なため、これがボトルネックとなってしまう。そこで、データ転送の遅さを隠蔽するために、GPU 演算処理と CPU-GPU 間のデータ転送を同時に行った。また、データ転送回数を削減するために、画像再生方式としてデータを分割して処理可能なサブアパーチャ法を採用した。今回これらの効果を評価したので報告する。

### 2. 合成開口レーダ画像再生処理

図1の左側は SAR センサによる観測状況を示し、右側は対応する SAR 画像を示している。また、図2は SAR 画像再生の処理手順を示している。ここで、図2の②ではアジマス方向 (センサの移動方向) の一次元 FFT (高速フーリエ変換) を全レンジに対して行っている。従って、アジマス方向にデータを並べた方が効率が良い。一方④はレンジ方向の FFT であり、レンジ方向にデータを並べた方が効率が良い。⑥、⑧の IFFT についても同様である。なお、参照関数乗算には処理の方向は無いため、どちらの方向に並んでも構わない。

このように FFT/IFFT では処理の方向があるため、①~③はアジマス方向にデータを並べ、④~⑥はレンジ方向に、

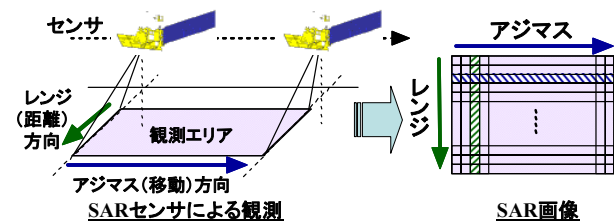


図1 SARによる観測と画像再生処理

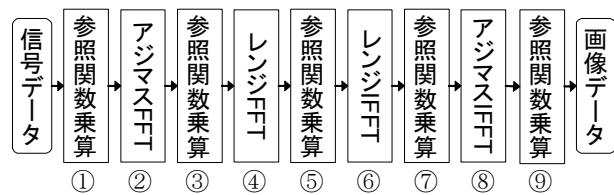


図2 SAR 画像再生処理手順

⑦~⑨は再びアジマス方向にデータを並べて処理する。そのため、③と④及び⑥と⑦の間に二次元データの縦横の並びを変える転置 (コーナータン) を行うのが一般的である。

### 3. GPU 適用

#### 3.1 課題と解決策

SAR 画像のデータ量は、評価に用いたデータで約 7GB であり、今後も観測領域の広域化、高分解能化により、増大していく。GPU のメモリは、この SAR 画像データ全てを搭載可能なほど大きくないため、データの一部のみを GPU 側に転送して処理を行うことしか出来ない。前述のように SAR 画像再生処理においては、アジマス方向とレンジ方向の処理があるため、処理方向が変わる度にデータを転送し直さなければならない。しかし、GPU へのデータ転送性能は、演算性能に比べて相対的に低速なため、これがボトルネックとなる。この課題を解決するために、我々は以下の2つの方策を適用した。

#### (1) サブアパーチャ法の採用

SAR 画像再生アルゴリズムの1つであるサブアパーチャ法 (図3) は、観測データをアジマス方向に重複を持たせて複数の小領域 (サブアパーチャ) に分割して、低分解能画像を複数作成し、それらを結合して高分解能画像を取得する手法であり、商用衛星 TerraSAR-X での画像再生に採用されている[3]。各サブアパーチャのデータであれば、GPU に充分搭載可能なデータ量であるため、1つのサブアパーチャ内の処理の過程ではデータ転送が不要となる。従

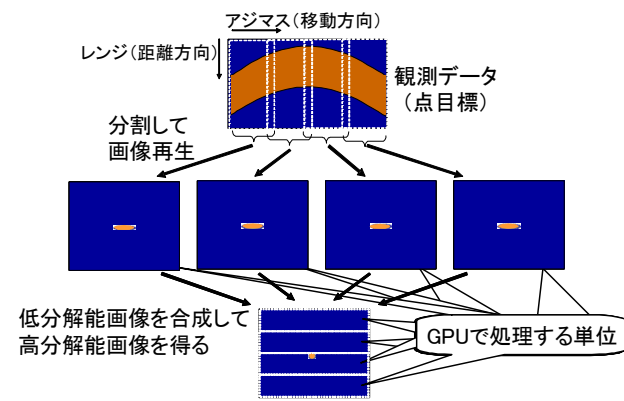


図3 サブアパーチャ法

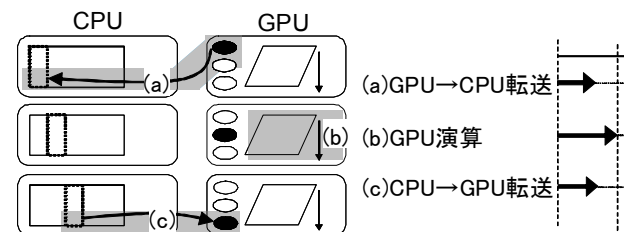


図4 データ転送と GPU 演算の同時実行

<sup>†</sup> 三菱電機株式会社 情報技術総合研究所, MELCO.

って、最後の高分解能処理を含め、4回であったデータ転送回数を2回に削減できる。

### (2) データ転送と GPU 演算の同時実行

今回使用した NVIDIA 社の GPU では、データ転送と GPU 演算を同時に行うことができる。そこで、GPU 内に3つのデータ領域 (バッファ) を設けることで、一連のサブアパーチャの処理において、1つの前のサブアパーチャの (a)GPU→CPU のデータ転送、(b)GPU 演算、1つ後のサブアパーチャの(c)CPU→GPU のデータ転送を同時に行うようにする (図 4)。これにより、データ転送の遅さを隠蔽することが出来る。

### 3.2 計測

以上の方策を適用して計測を行った。使用した GPU は、NVIDIA 社の TeslaC2070 (448CUDA コア, 1.15GHz, 6GB メモリ) ×2, ソフトウェアは、CUDA, CUFFT 3.2 である。またホスト CPU は、Intel 社 Xeon X5680 (6 コア×2, 3.33GHz, 48GB メモリ) である。[2]での評価に対して、GPU, ホスト CPU とも最新に更新されている。

表 1 に全処理時間の計測結果 (汎用 CPU 及び GPU) を示す。ここで、「逐次実行 1GPU」とは GPU 演算処理と CPU-GPU 間のデータ転送を逐次に行う方式、「同時実行 1GPU」とはそれらを同時実行する方式、「2GPU」とは更に2つの GPU を使用する方式の結果である。

GPU による実行は、汎用 CPU 1 コアに対して約 21 倍、2 チップ同士 (2GPU 対 CPU12 コア) では 5.5 倍高速である。なお、GPU は 1 筐体に 4 つまで搭載可能なので、更に高速化可能である。また、GPU 演算処理と CPU-GPU 間のデータ転送を同時に行うことにより、1.6 倍の高速化が図れており、この効果は高いと言って良い。

### 3.3 サブアパーチャ法採用による効果

今回、GPU を使用するに当たり、CPU-GPU 間のデータ転送回数を抑えるために、SAR 画像再生アルゴリズムとしてサブアパーチャ法 (SA 法) を採用した。この効果を評価するために、SA 法を採用しない実装も行い実行時間を計測した。この場合、処理のデータアクセス方向が変わる度に、CPU と GPU 間のデータ転送が必要となるため、4 回のデータ転送となる。

表 2 に SA 法を採用した場合と採用しない場合について、GPU 演算処理と CPU-GPU 間のデータ転送を逐次に行う方式 (逐次実行) と同時に行う方式 (同時実行) の各処理毎の実行時間総和を示す。SA 法を採用する場合、サブアパーチャ分割と結合が必要であるが、途中のデータ転送が不要となるという違いがある。また、アジマス方向にデータを分割するので、(4), (16)のアジマス方向の FFT/IFFT の実行時間は短くなる。なお、高分解能処理に関しては、相違点はない。

逐次実行では、データ転送回数が少ない分、SA 法採用が 1.5 倍高速である。一方、同時実行ではデータ転送の遅さが隠れてしまったので高速化は 1.2 倍まで低減している。SA 法採用・非採用の実行時間差 (2,126 ミリ秒) はアジマス FFT/IFFT の実行時間差とほぼ等しく、これはデータ転送回数の削減効果ではなく、演算量の削減効果である。SA 法採用は、データ転送不要な連続処理の粒度を増大させるためであったが、今回 GPU 演算だけでなくデータ転送の双

表 1 全処理の実行時間

(単位:秒)

	汎用CPU		GPU			
	1コア	12コア	逐次実行 1GPU	同時実行 1GPU	2GPU	
全サブアパーチャ処理	179.595	29.273	11.023	7.446	4.283	
高分解能処理	37.539	5.168	5.821	3.039	1.953	
合計	217.134	34.441	16.844	10.485	6.236	
対1コア比率	1.0	6.3	12.9	20.7	34.8	
対12コア比率	0.2	1.0	2.0	3.3	5.5	
対逐次実行比率				1.6		
					対1GPU比率	1.7

表 2 SA 法と非 SA 法の実行時間比較

(単位:ミリ秒)

処理内容	逐次実行		同時実行	
	SA法	非SA法	SA法	非SA法
(1) サブアパーチャ分割	926			
(2) CPU→GPUデータ転送	1,405	1,370		
(3) 参照関数乗算	232	183		
(4) アジマスFFT	324	1,356		2,374
(5) 参照関数乗算	425	571		
(6) GPU→CPUデータ転送		1,560		
(7) CPU→GPUデータ転送		2,167		
(8) コーナーターン	514	654		
(9) レンジFFT	773	755		
(10) 参照関数乗算	505	973	7,446	4,351
(11) レンジIFFT	1,075	881		
(12) コーナーターン	359	495		
(13) GPU→CPUデータ転送		2,306		
(14) CPU→GPUデータ転送		1,370		
(15) 参照関数乗算	1,068	836		
(16) アジマスIFFT	559	1,461		2,848
(17) 参照関数乗算	201	174		
(18) GPU→CPUデータ転送	1,597	1,560		
(19) サブアパーチャ結合	1,040			
(20) CPU→GPUデータ転送		2,291		
(21) 高分解能処理		1,087		3,039
(22) GPU→CPUデータ転送		2,443		
合計	16,824	24,492	10,485	12,611
対SA法比率	1.0	1.5	1.0	1.2

方向通信も同時実行することにより転送ネックは全く無くなったため、その効果が低くなった。

### 4. おわりに

以上、SAR 画像再生処理の GPU への適用に関する評価を報告した。この中で、CPU-GPU 間のデータ転送と GPU 演算を同時に行う効果が高いことが分った。また、SAR 画像再生アルゴリズムとしてサブアパーチャ法を採用しなくても、ある程度の高速実行が可能であることが分った。これは、GPU を適用する際に、画像再生アルゴリズムの選択肢を拡げられることになる。

#### 参考文献

- [1]Owens,J.D 他,“GPU Computing,” Proc. IEEE, vol.996, no.5, pp.879-899, 2008.
- [2]佐藤裕幸他,“グラフィックス処理用プロセッサ GPU による SAR 画像再生処理の高速化,”信学技報, Vol.110, No.8, SANE2010-2, pp.7-12, 2010.
- [3]J. Mittermayer 他,“Sliding spotlight SAR processing for TerraSAR-X using a new formulation of the extended chirp scaling algorithm,” vol.3, pp.1462-1464, IGARSS03, 2003.