

SAR 画像再生処理における並列処理方式の比較検討

水野 政治, 中島 克人

三菱電機(株) 情報技術総合研究所

合成開口レーダ (Synthetic Aperture Radar, SAR) は, 雲霧等の天候に左右されずに, 高い分解能で地表を撮像することができるセンサであるが, 人間が理解できる画像を生成するための再生処理が必要である. この SAR 画像再生処理は演算量が大きく, しばしば並列処理による高速化が必要となる. しかし, ハードウェア構成によって並列処理による効果は異なるため, 適用するシステムに適した並列処理方式を選択する必要がある, そのための指標が必要である. そこで, 本稿では, 2 種類の SAR 画像再生処理の並列処理方式について, 演算量とプロセッサ間のデータ転送量に着目し比較することにより, 2 つの方式の特性を明らかにした.

A Comparative Study of Parallel Processing Techniques for Synthetic Aperture Radar Processing

Masaji Mizuno, Katsuto Nakajima

Information Technology R&D Center,
Mitsubishi Electric Corporation

Synthetic Aperture Radar (SAR) is a remote sensing instrument capable of obtaining high resolution images of the Earth surface. One of the traditional problems with SAR as a remote sensing tool is the huge amount of signal processing that is required to form an image from the raw data. SAR processing often makes use of parallel processing for its speed-up. In this paper, we compared two parallel processing techniques for SAR processing in terms of computational cost and data transfer cost.

1 まえがき

合成開口レーダ (Synthetic Aperture Radar, SAR) は, リモートセンシングの分野において, 雲霧等の天候に左右されず, 高い分解能で地表を撮像することができるセンサとして知られている¹⁾²⁾. SAR では, レーダで受信した元データから人間が理解できる画像を再生する処理 (これを SAR 画像再生と呼ぶ) が必要となるが, データ量, 演算量が非常に大きい. そのため, SAR 画像再生処理を高速に行うためには, 従来は専用のハードウェアによって実現されていた³⁾⁴⁾⁵⁾. このような専用のハードウェアによるアプローチでは, 装置自体が複雑で高価となり, また, 柔軟性, 拡張性に欠ける等の問題を抱える. 一方, 近年の汎用の計算機システムの進歩は著しく, また安価である. そのため, このような汎用品/市販品 (Commercial off-the-shelf, COTS) を用いた, より安価で, 柔軟性, 拡張性の高いシステムでの高速な

SAR 画像再生処理の実現に移行しつつある.

さて, COTS によるアプローチにおいては, 適用する SAR システムの運用, 要求性能, 規模等の種々の条件に応じ, 使用する COTS 及びその上での処理方式を選択する上で, 何らかの指標が必要であると考える. 特に, 現状では, 単一のプロセッサによる逐次処理では要求性能に満たない場合が多く, 複数のプロセッサを用いた並列処理が必要となるため, 使用する COTS に適した並列処理方式の選択も必要であろう. SAR 画像再生処理の並列処理方式については, MPP (Massively Parallel Processor), ワークステーションやパソコンによるクラスタシステム等, 個々のシステムに対する実現方式については検討されている⁶⁾⁷⁾ のものの, どのようなハードウェア構成に対しても最適な結果を得られる, ハードウェア構成に依存しない並列処理方式は確立されていない.

そこで, COTS ベースの SAR 画像再生処理シス

テムを構築する上で一指標を得るため、今回、2種類の SAR 画像再生処理の並列処理方式について、演算性能及びプロセッサ間のデータ転送性能が与える影響を比較検討する。

2 SAR 画像再生処理

SAR 画像再生のアルゴリズムは種々提案されているが、ここでは、最も広く一般的に使用されている、レンジドップラ アルゴリズムを取り上げる。

SAR で受信した元のデータは、レーダパルスの照射方向（これをレンジ方向と呼ぶ）とこれに直交する方向（これをアジマス方向と呼ぶ）との二次元のデータである。この受信データは、レンジ方向とアジマス方向の二次元に広がりをもつホログラムである。これをレンジ方向、アジマス方向各々について、参照関数との畳込み演算（マッチドフィルタ）を行うことによって、画像を再生するものである（ここで、レンジ方向の相関処理をレンジ圧縮、アジマス方向の相関処理をアジマス圧縮と呼ぶ）。畳込み演算は、高速フーリエ変換（Fast Fourier Transform, FFT）を用いて周波数領域で行うことによって、演算量を小さくすることができる⁸⁾ので、一般的にはレンジ圧縮及びアジマス圧縮においても FFT が適用される。

図 1は、SAR 画像再生処理の一連の流れをまとめたものである。

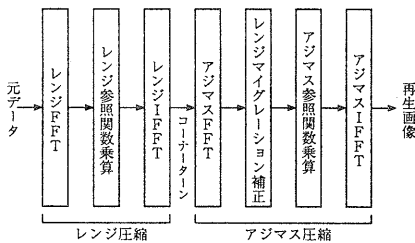


図 1: SAR 画像再生処理の流れ

1. レンジ FFT: 各レンジライン毎に、FFT を行う。各レンジラインの処理を並列処理することが可能である。また、1 レンジラインの FFT についても並列処理可能である。
2. レンジ参照関数乗算: 各レンジライン毎に、レンジ参照関数との項別積を行う。各データ単位で並列処理することが可能である。
3. レンジ IFFT: 各レンジライン毎に、IFFT(Inverse FFT, 高速

フーリエ逆変換) を行う。レンジ FFT と同様に、各レンジラインの処理毎に並列処理可能であり、1 レンジラインの IFFT も並列処理可能である。

4. アジマス FFT: 各アジマスライン毎に、FFT を行う。並列化に関しては、レンジ FFT と同様であり、各アジマスライン毎に並列処理することができる。
5. レンジマイグレーション補正: アジマス方向の相関処理をより効果的に行うため、観測点（地表）との距離の変化を補正する処理である。本稿では、近傍 5 点のデータから sinc 関数を用いて内挿する方式を想定する。各データ単位で並列処理することが可能である。
6. アジマス参照関数乗算: 各アジマスライン毎に、アジマス参照関数との項別積を行う。各データ単位で並列処理可能である。
7. アジマス IFFT: 各アジマスライン毎に、IFFT を行う。アジマス FFT と同様に並列処理可能である。

以上のように、SAR 画像再生処理の中心はレンジ方向／アジマス方向の FFT 及び IFFT であり、全体の処理時間の大部分を占めることとなる（3.3節参照）。

3 SAR 画像再生処理の並列処理

3.1 想定するシステム構成

並列処理方式を検討する上で、図 2に示すような並列処理システムの構成を想定する。

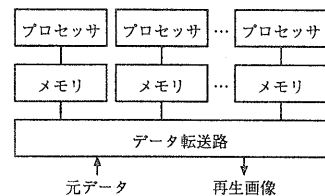


図 2: 想定するシステム構成

- 元データは、一箇所からデータ転送路を介して各プロセッサに与えられる。磁気テープ／磁気ディスク等の外部記憶装置、いずれのプロセッサからもアクセスできる共有メモリ、受信アンテナとのインターフェース等が考えられる。
- 処理結果（再生画像）は、各プロセッサからデータ転送路を介して一箇所に集められる。

3.2 対象とする並列処理方式

次の2つの並列処理方式を取り上げ、比較する。

1. レンジ-アジマス分割方式
2. 部分画像分割方式

3.2.1 レンジ-アジマス分割方式

レンジ圧縮は各レンジライン毎に独立した処理であり、アジマス圧縮は基本的には各アジマスライン毎に独立した処理である。

本方式は、まず、受信した元データをレンジ方向で分割し、各プロセッサでレンジ圧縮を並列処理した後、データを再分配し、次に、レンジ圧縮後のデータをアジマス方向で分割して各プロセッサでアジマス圧縮を並列処理するものである(図3)。なお、アジマス圧縮におけるレンジマイグレーション補正では、隣接するレンジラインのデータが必要となるため、アジマス方向で分割する際にデータをオーバーラップさせることとした。

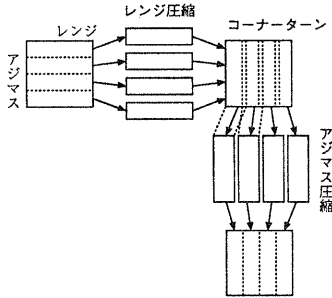


図3: レンジ-アジマス分割方式

本方式においては、レンジ圧縮とアジマス圧縮との間に、コーナーターンに伴うプロセッサ間の大規模なデータ転送が発生することとなり、このデータ転送が性能に及ぼす影響が危惧される。

3.2.2 部分画像分割方式

SAR 画像再生処理は、本来、レンジ方向、アジマス方向各々について参照関数との畳込み演算を行うものであるから、再生画像の一部分を、それに対応する元データの一部分に参照関数との畳込み演算を施すことにより再生することが可能である(図4)。この時の処理内容は、元データ全体について再生処理を行う場合と等価である。

本方式は、元データを複数の部分画像に分割し、各プロセッサで個々の部分画像の再生処理を並列に処理するものである(図5)。この場合、元データを部

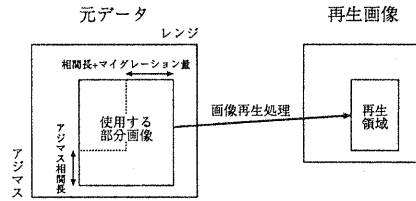


図4: 部分画像の再生

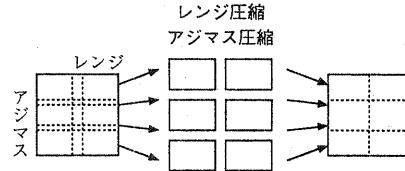


図5: 部分画像分割方式

分画像に分割する際には、レンジ方向ではレンジ参照関数の大きさ(レンジ相関長)とレンジマイグレーション量の分だけ、アジマス方向ではアジマス相関長の分だけ、それぞれデータをオーバーラップさせる必要がある。

本方式では、コーナーターンに伴うプロセッサ間のデータ転送は発生しないものの、部分画像に分割することによる全体の演算量の増加が心配される。

3.3 演算量による比較

まず、上記の二方式について演算量の観点から比較する。

表1は、各々の方式における1プロセッサの演算量をまとめたものである。ここで、

- R_{in} : 元データのレンジ数
- R_{ref} : レンジ相関長(レンジ参照関数の大きさ)
- R_{mig} : レンジマイグレーション量
- R_{out} : 再生画像のレンジ数

$$(R_{out} + R_{ref} + R_{mig} = R_{in})$$

- A_{in} : 元データのアジマス数

- A_{ref} : アジマス相関長

- A_{out} : 再生画像のアジマス数

$$(A_{out} + A_{ref} = A_{in})$$

- P : プロセッサ数(部分画像分割方式の場合: $P = P_r P_a$)

- P_r : レンジ方向の分割数(部分画像分割方式の場合)

- P_a : アジマス方向の分割数(部分画像分割方式の場合)

- $N_{fft}(x)$: データ長 x に適用する FFT 点数

(x 以上の最も小さい2の冪乗の数)

- $C_{fft}(x)$: x 点の複素数 FFT の演算量

表 1: 各方式の1プロセッサ当たりの演算量

	レンジ-アジマス分割方式	部分画像分割方式
レンジFFT	$\frac{A_{in}}{P} \cdot C_{fft}(N_{fft}(R_{in}))$	$(\frac{A_{out}}{P_a} + A_{ref}) \cdot C_{fft}(N_{fft}(\frac{R_{out}}{P_r} + R_{ref} + R_{mig}))$
レンジ参照関数乗算	$\frac{A_{in}}{P} \cdot C_{mul}(N_{fft}(R_{in}))$	$(\frac{A_{out}}{P_a} + A_{ref}) \cdot C_{mul}(N_{fft}(\frac{R_{out}}{P_r} + R_{ref} + R_{mig}))$
レンジIFFT	$\frac{A_{in}}{P} \cdot C_{fft}(N_{fft}(R_{in}))$	$(\frac{A_{out}}{P_a} + A_{ref}) \cdot C_{fft}(N_{fft}(\frac{R_{out}}{P_r} + R_{ref} + R_{mig}))$
アジマスFFT	$(\frac{R_{out}}{P} + R_{mig}) \cdot C_{fft}(N_{fft}(A_{in}))$	$(\frac{R_{out}}{P_r} + R_{mig}) \cdot C_{fft}(N_{fft}(\frac{A_{out}}{P_a} + A_{ref}))$
レンジマイグレーション	$(\frac{R_{out}}{P}) \cdot C_{mig}(N_{fft}(A_{in}))$	$(\frac{R_{out}}{P_r}) \cdot C_{mig}(N_{fft}(\frac{A_{out}}{P_a} + A_{ref}))$
アジマス参照関数乗算	$(\frac{R_{out}}{P}) \cdot C_{mul}(N_{fft}(A_{in}))$	$(\frac{R_{out}}{P_r}) \cdot C_{mul}(N_{fft}(\frac{A_{out}}{P_a} + A_{ref}))$
アジマスIFFT	$(\frac{R_{out}}{P}) \cdot C_{fft}(N_{fft}(A_{in}))$	$(\frac{R_{out}}{P_r}) \cdot C_{fft}(N_{fft}(\frac{A_{out}}{P_a} + A_{ref}))$

$C_{mul}(x)$: x 点の項別積の演算量

$C_{mig}(x)$: x 点分のレンジマイグレーション補正の演算量

ここで、より具体的な傾向をみるために、幾つかの条件を与えることとする。まず、各処理の演算量($C_{fft}()$, $C_{mul}()$, $C_{mig}()$)は、その処理で行われる実数加算及び実数乗算の数に比例するものと仮定し、次式の関係が成り立つものとする。

$$C_{fft}(x) : C_{mul}(x) : C_{mig}(x) = 5x \log_2 x : 6x : 22x \quad (1)$$

次に、RADARSAT/SAR⁹⁾を参考に、処理のパラメータ (R_{in} , R_{ref} , R_{mig} , R_{out} , A_{in} , A_{ref} , A_{out})を表2のように想定する。

表 2: 処理のパラメータ

元データのレンジ数 R_{in}	6000
レンジ相関長 R_{ref}	600
レンジマイグレーション量 R_{mig}	30
再生画像のレンジ数 R_{out}	5370
元データのアジマス数 A_{in}	8192
アジマス相関長 A_{ref}	600
再生画像のアジマス数 A_{out}	7592

以上の条件をもとに算出した結果を図6, 図7に示す。図6は、逐次処理、すなわちプロセッサ数 $P=1$ における各処理の比率を示すものである。図7は使用するプロセッサ数 P と各プロセッサの演算量との関係を示すものであり、横軸はプロセッサ数、縦軸は逐次処理 ($P=1$) の時を1とした場合の1プロセッサ当たりの演算量である。なお、部分画像分割方式においては、レンジ方向の分割数 P_r とアジマス方向の分割数 P_a と組み合わせによって演算量が異なる¹⁰⁾ため、演算量が最も小さくなる組み合わせを選択した。

- レンジ-アジマス分割方式では、プロセッサ数の増加に伴う演算量の減少は比較的理想的に近い

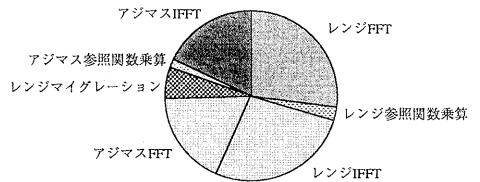


図 6: 逐次処理における演算量の比率

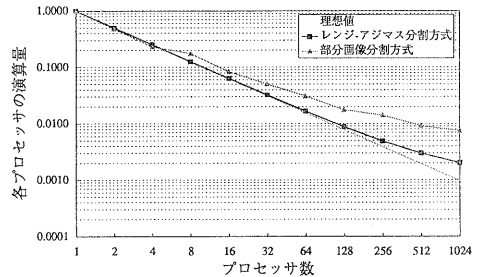


図 7: 1プロセッサ当たりの演算量の変化

い。プロセッサ数が大きくなると、アジマス圧縮でデータをオーバーラップしている影響が表面化するため、理想値を下回ることとなる。

- 一方、部分画像分割方式では、部分画像に分割した結果、総演算量が大きくなってしまったため、全体的にはレンジ-アジマス分割方式よりも各プロセッサの演算量は大きくなってしまふ。

すなわち、演算量の観点からは、レンジ-アジマス分割方式の方が効果的に並列処理を行える可能性が高いことを示唆する。

なお、部分画像分割方式において、プロセッサ数が小さい場合 ($P=2$ 及び $P=4$) に、理想値を上回るという結果が得られた。これは、SAR 画像再生処理では FFT 及び IFFT の占める割合が大きい (図6)

表 3: 各方式のデータ転送量

	レンジ-アジマス分割方式	部分画像分割方式
データの入力	$B_{in} \cdot R_{in} \cdot A_{in}$	$B_{in} \cdot P \left(\frac{R_{out}}{P_r} + R_{ref} + R_{mig} \right) \left(\frac{A_{out}}{P_a} + A_{ref} \right)$
コーナーターン	$B_{ct}(P-1) \left(\frac{R_{out}}{P} + R_{mig} \right) A_{in}$	0
データの出力	$B_{out} \cdot R_{out} \cdot A_{out}$	$B_{out} \cdot R_{out} \cdot A_{out}$

ため、部分画像に分割した結果、FFT 及び IFFT のポイント数が小さくなり、分割数以上に演算量が小さくなる場合があるためである。このことは、逐次処理及びレンジ-アジマス分割方式による並列処理においても、適切に部分画像に分割して処理することによって高速化できる可能性があることを意味する。

3.4 データ転送量による比較

次に、プロセッサ間で生じるデータ転送量の観点から比較する。

表 3は、各々の方式で発生するデータ転送量をまとめたものである。ここで、

B_{in} : 元データの 1 データあたりの大きさ

B_{ct} : コーナーターン時の 1 データあたりの大きさ

B_{out} : 再生画像の 1 データあたりの大きさ

演算量の場合と同様に、表 2 のパラメータを適用し、また $B_{in} : B_{ct} : B_{out} = 1 : 4 : 2$ であると仮定した場合の、データ転送量を図 8 に示す。図 8 の横軸はプロセッサ数、縦軸は逐次処理、すなわちプロセッサ数 $P=1$ の時を 1 とした場合のデータ転送量である。なお、部分画像分割方式においては、演算量が最も小さくなる、レンジ方向の分割数 P_r とアジマス方向の分割数 P_a との組合わせを選択した場合のデータ転送量である。

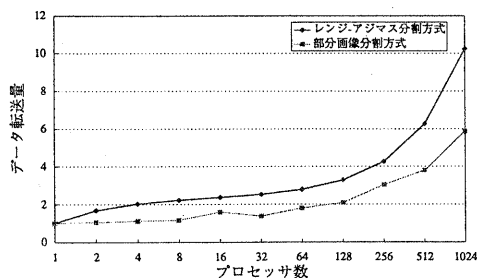


図 8: データ転送量の変化

いずれにおいても、プロセッサ数の増加に伴いデータ転送量が急激に増加する傾向にあり、並列処理による高速化を阻害する可能性が高い。しかし、部

分画像分割方式は、レンジ-アジマス分割方式に比較し、データ転送の影響を受けにくいことがわかる。

3.5 全体性能による比較

以上のように、演算量の観点からはレンジ-アジマス分割方式が、データ転送量の観点からは部分画像分割方式が適していることがわかった。システム全体の性能は、演算量とデータ転送量のいずれにも影響を受けることとなるので、これらを併せて評価する必要がある。そこで、対象とするシステムにおける、演算性能とデータ転送性能との性能比を想定し、この性能比別の並列処理による効果、すなわち高速化率について検討する。

いま、演算量 x の演算に要する時間を $T_c(x)$ 、データ転送量 y のプロセッサ間データ転送に要する時間を $T_t(y)$ とし、それぞれ演算量 x 及びデータ転送量 y に単純に比例するものと仮定する。

$$T_c(x) = a \cdot x \quad (2)$$

$$T_t(y) = b \cdot y \quad (3)$$

ここで、演算量 1 の演算に要する時間 $T_c(1)$ とデータ転送量 1 のデータ転送に要する時間を $T_t(1)$ との比 τ を次式で定義する。

$$\tau = \frac{T_t(1)}{T_c(1)} = \frac{b}{a} \quad (4)$$

式 2~4 を使って、演算量 x とデータ転送量 y を有する処理の実行時間 $T(x, y)$ は、次式によって成り立つものと仮定する。

$$T(x, y) = T_c(x) + T_t(y) = a(x + \tau \cdot y) \quad (5)$$

逐次処理 ($P = 1$) の演算量及びデータ転送量を各々 x_1, y_1 、プロセッサ数が p であるときの演算量及びデータ転送量を x_p, y_p とした場合、高速化率は次式によって算出する。

$$\text{高速化率} = \frac{T(x_1, y_1)}{T(x_p, y_p)} = \frac{x_1 + \tau \cdot y_1}{x_p + \tau \cdot y_p} \quad (6)$$

図 7 で示した演算量と、図 8 で示したデータ転送量とを使い、演算性能とデータ転送性能との性能比 τ

が変化した場合の高速化率を、式6によって計算した。その結果を図9に示す。図9の横軸はプロセッサ数、縦軸は逐次処理、すなわちプロセッサ数 $P = 1$ の時を1とした場合の高速化率である。

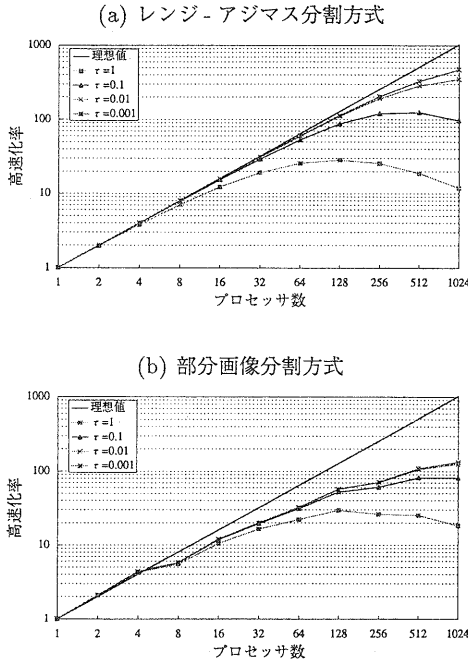


図9: 高速化率

この結果、次のような特徴が確認できる。

- τ が小さい、すなわち演算性能に比較しデータ転送性能が高い場合、データ転送の影響が小さくなるため、演算量の観点から優れたレンジ-アジマス分割方式で高い高速化率が得られる。
- 逆に、 τ が大きい、すなわち演算性能に比較しデータ転送性能が低い場合、データ転送の影響が大きくなるため、データ転送量の観点から優れた部分画像分割方式の方が良好な結果が得られる。
- 部分画像分割方式では、レンジ-アジマス分割方式に比較し、データ転送性能の影響を受けにくい。
- いずれの方式においても、 τ が大きくなるとデータ転送に要する時間がネックとなり、高速化率は小さい。すなわち、 τ が大きいシステム構成においては並列処理による利点は小さく、 τ が小さくなるような何らかの工夫が必要である。

4 むすび

本稿では、SAR 画像再生処理の並列処理方式として、(1) レンジ-アジマス分割方式、(2) 部分画像分割方式の二方式を取り上げ、比較検討した。

レンジ-アジマス分割方式は、データ転送性能に大きく依存し、データ転送性能が高い場合には高い高速化率が得られるものの、データ転送性能が低い場合の性能劣化が著しいことがわかった。一方、部分画像分割方式は、データ転送性能の影響を受けにくい、プロセッサ数が小さい場合には高い高速化率が得られるものの、プロセッサ数が大きくなると効果が得られにくいことが確認された。

今後、他の並列処理方式の検討を行うと共に、実際に幾つかのプラットフォーム上に実装し、今回机上で検討した結果を検証していく予定である。

参考文献

- 1) Curlander, J. C. and McDonough, R. N.: "SYNTHETIC APERTURE RADAR Systems and Signal Processing", John Wiley & Sons, Inc. (1991).
- 2) 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).
- 3) 山口明俊, 菊池剛, 九十歩修, 村田稔: 「リアルタイム SAR 処理システム」, 電子情報通信学会技術研究報告, Vol.93, No.540(SANE93 101-110), pp.71-76 (1994).
- 4) Premkumar, A. and Purviance, J.: "An Architecture for Real Time Processing of SAR Signals", In *IEEE Int. Symp. Circuits and Systems*, Vol.3, pp.1483-1486 (1992).
- 5) Chen, M., Cheng, T., and Jin, M.: "Magellan SAR processing algorithm and H/W design", In *IGARSS*, Vol.3, pp.1707-1712 (1989).
- 6) Fabbretti, G., Farina, A., Laforenze, D., and Vinelli, F.: "Mapping the Synthetic aperture radar signal processor on a distributed-memory MIMD architecture", In *Parallel Computing*, Vol.22, No.5, pp.761-784 (1996).
- 7) Meisl, P. G., Ito, M. R., and Cumming, I. G.: "Parallel Synthetic Aperture Radar Processing on Workstation Networks", In *10th International Parallel Processing Symposium (IPPS '96)*, pp. 716-723 (1996).
- 8) 電子情報通信学会 (編): 「デジタル信号処理」, コロナ社 (1987).
- 9) <http://radarsat.space.gc.ca/>
- 10) 水野政治, 中島克人: 「画像分割方式による SAR 画像再生の並列処理方式の基礎検討」, 情報処理学会第 58 回 (平成 11 年前期) 全国大会, Vol.2, pp.13-14 (1999).