

機動的SARプラットフォームを対象とした GPU並列処理による時間領域逆投影処理

Othmar Frey
Gamma Remote Sensing Switzerland
Earth Observation & Remote Sensing,
ETH Zurich

Charles L. Werner, Urs Wegmuller
Gamma Remote Sensing
Switzerland

概要:自動車に搭載するような機動性のあるSARプラットフォームでは、合成開口をする間センサーは非直線的な軌道を辿ります。その場合、柔軟なSAR処理の仕組みが必要です。本稿では、時間領域逆投影法によるSARの結像アルゴリズムについてNVIDIA社製CUDAのGPU計算フレームワークを用いた並列実装について解説します。車載SARのデータセットを用いて検討し、様々なハードウェアを用いた処理性能の評価を行いました。

さらに、フル3Dの動き補正を可能とする前処理の仕組みについて説明します。これは、好都合なことに、シングルルック複素SAR画像の従来のスラントレンジ/アジマス方向のジオメトリで扱えます。

キーワード: 合成開口レーダ(SAR)、地上設置型SARシステム、SARイメージング、SAR干渉法、車載SAR、CARSAR、GPU、CUDA、並列処理、アジマスフォーカシング、非線形センサー軌道

1. はじめに

自動車のような機動性のある合成開口レーダ(SAR)のプラットフォームでは、合成開口処理の間の非直線的なセンサー軌道を考慮した柔軟なSAR処理の仕組みが必要です。最近、我々はそのような車載SARおよびInSAR実験での最初の測定結果を報告しました(参考文献[1])。

本稿では、NVIDIA社製のCompute Unified Device Architecture(CUDA)と呼ばれるアプリケーションプログラミングインターフェースに基づき、時間領域逆投影アルゴリズム(TDBP)(参考文献[2, 3])を画像処理装置(GPU)に新しく並列実装する方法について解説、評価しています。2パターン

表1
合成開口レーダモードで使用した
GPRI-II地上設置型レーダシステムの仕様

搬送周波数	17.2 GHz
チャープバンド幅	200 MHz
方式	FM-CW
チャープ長	0.001 s
レンジ方向の3dビーム幅	18度
アジマス方向の3dビーム幅	16.9度
対地速度	21 m/s
干渉基線	0.25 m
オフナディア角	110度

でのTDBP結像アルゴリズムの実装の処理性能を比較しました:(1)CPU上でのANSI-C実装、(2)3種類のNVIDIA社製GPUデバイスをを用いたCUDAによる実装。

さらに、TDBP処理中のフル3D動き補正(3Dセンサー座標と3D再構築グリッド)を可能とする前処理の仕組みについて説明します。これは、好都合なことに、シングルルック複素(SLC)SAR画像の従来のスラントレンジ/アジマス方向のジオメトリで扱うことが可能です。

時間領域逆投影アルゴリズムのCUDA並列実装に関する検証および評価のために今回使用したサンプルデータセットは、車両上部に搭載したGPRI-II地上設置型レーダ(参考文献[4, 5])の構成を変更して、わずかにカーブした道路から取得しました(詳細は参考文献[1]を参照)。関連するシステムパラメータの概要を表1に示します。

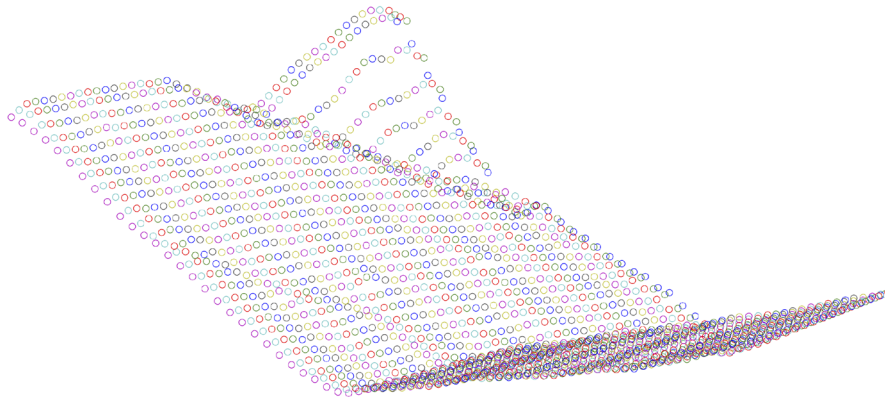


図1.SLC(レンジ/アジマス)ジオメトリの例、DEMリサンプリング手順後の3D再構築グリッド

II. 方法

A. レンジ圧縮

本実験に用いられた車載SARシステム(参考文献[1]を参照)では、線形FM-CW方式のGPRI-IIレーダ(参考文献[4, 5])を使用しています。レーダは受信時デチャープモードで動作します。つまり、受信信号 $s(t)$ は基準信号とミックスされます。この処理では、データを次の形式のデランプ信号 s_d に変換します(参考文献[6])。

$$s_d(t) = s^*(t) \exp(j2\pi f_s t + j\pi \gamma t^2) \quad (1)$$

ここで、 f_s はチャープの開始周波数、 γ はチャープ率です。その結果生じるデランプ信号の位相は次の通りです。

$$\varphi_d(t) = (2\pi f_s t_n - \pi \gamma t_n^2) + 2\pi \gamma t_n t \quad (2)$$

これは、レンジFFTを介してレンジ距離に直接関連付けることができます。 t_n は、ターゲット n に対する双方向の時間遅延です。データのレンジフォーカシングでは、残差ビデオ位相を補正する必要があります(参考文献[7])、開始から停止における近似の無効性も考慮する必要があります(参考文献[8])。

B. レンジ/アジマス(SLC)ジオメトリにおけるDEMベースの画像再構築グリッド作成手順

次のような前処理方法により、TDBPアジマスフォーカシングでフル3D動き補正を可能にしながら、レンジおよびアジマス方向に等間隔にサンプリングを行なった3D再構築グリッドの作成が行われました(標準的なSLCデータの成果物)。基本的には、地形座標を中心に考えた場合、不均一にサンプリ

ングされたグリッドにDEMをリサンプリングしますが、レンジ・アジマス領域への投影を考慮する場合は均一にサンプリングされます。前処理は次の手順で行われます。

- 1) 直交回帰によりセンサー軌道をパラメーター化する。
- 2) レンジ圧縮データからドップラー中心周波数を推定する。
- 3) ルックアップテーブルに基づいたSLC再構築グリッドをより効率的に計算できるようマルチルックジオメトリを作成する。
- 4) 数値標高モデル(DEM)とパラメーター化したセンサー軌跡に基づいて、ジオコード用ルックアップテーブルを作成する。ルックアップテーブルには、DEMの各所に対して浮動小数点でレンジ/アジマス座標が含まれている。
- 5) ルックアップテーブルを反転する。反転後のルックアップテーブルには、各スラントレンジ/アジマス座標におけるDEMグリッドのピクセル座標が浮動小数点で含まれる。
- 6) 反転したルックアップテーブルを3D DEM座標に適用して、マルチルックジオメトリの近似的な地形情報を取得する。
- 7) MLIからSLCジオメトリにX、Y、Z座標のアップサンプリングを行う。
- 8) レンジ距離が公称のレンジ距離に(倍精度で)一致するように3Dグリッドの位置を改善する。

レンジおよびアジマス方向に等間隔サンプリングによってリサンプリングした3D再構築グリッドの例については図1をご覧ください。表示の都合上、解像度を低くしています。

C. TDBPアジマスフォーカシングのCUDAでの実装

新規開発された、GPUへのTDBPアルゴリズムのCUDA

表2

CPUによる実装と3種類のNVIDIA社製GPU使用時の処理時間の比較

処理装置	(V)RAM	コア数	処理時間	処理速度の比 (対CPU)	処理速度の比 (対GT 650M)
CPU: 2.6 GHz Intel Core i7	8 GB	4 (使用数1)	1539 s	1	-
GPU: NVIDIA GeForce GT 650M	1 GB	384	315 s	4.9	1
GPU: NVIDIA GeForce GTX 660 Ti	2 GB	1344	79 s	19.5	4.0
GPU: NVIDIA Tesla K20c	4 GB	2496	51 s	30.2	6.2

並列実装では、時間領域逆投影アルゴリズムの単一命令-複数データ(SIMD)の性質が利用されています。NVIDIA社製CUDA高速フーリエ変換(FFT)ライブラリ(cuFFT)(参考文献[9])を使って、各エコーのFFT補間によるアップサンプリング(参考文献[10])が実装されました。実際の逆投影中のSIMDの操作には、NVIDIA社製CUDA Basic Linear Algebra Subroutines(cuBLAS)ライブラリ(参考文献[11])で利用可能な、最適化されたCUBLAS複素ベクトルの加算を活用して行われています。これは、完全標準BLASライブラリのGPUアクセラレータによる実装です。

現在の実装では、GPUにより並列化されたTDBPアジマスフォーカシングのコア処理シーケンスは、次の手順で行われます。

- 1) センサー位置および速度データを読み込む。
- 2) 画像の再構築グリッドの座標を読み込む。
- 3) SLC画像ブロック用にGPUデバイスにメモリを割り当てる。
- 4) レーダエコーデータ、ドップラー中心値、エコー毎の位相補正值用に、GPUデバイスにメモリを割り当てる。
- 5) レーダエコーのループを開始して、逆投影を行う。
 - a) ループ中の現在のレーダエコーに合致するように、レンジおよびアジマス方向に変動するドップラー中心値を更新する。
 - b) レンジ変動する位相補正值を更新する。値はアジマスエコー毎に更新される。
 - c) レーダエコーをデバイスメモリにコピーする。
 - d) cuFFTライブラリを使用し、GPUデバイス上のレーダエコーのFFTによる補間を行う。
 - e) アップサンプリングされたエコーをベースバンドからバンドパスへ変換する(時間遅延の導入)。

- f) 現在置と再構築グリッド座標間のレンジ距離を計算する。
 - g) 補間されたエコーを中間SLC画像グリッドに逆投影する。
 - h) 中間画像グリッドに保存されている現在の逆投影データを用いてSLC画像を更新する。
- 6) 逆投影ループを終了する。
 - 7) SLC画像のコヒーレントベースバンド変換を行う。
 - 8) 結像したSLC画像をGPUデバイスメモリからホストメモリにコピーする。

III. 結果

表2には、本試験に用いたハードウェアやそれぞれの処理時間、時間比率について概要が示されています。

ノートパソコン用のGeForce GT 650Mおよびデスクトップパソコン用のGeForce GTX 660 Tiの2種類のコンシューマ向けNVIDIA社製カードを比較すると、処理時間の比は315 s/79 s = 4倍です。この場合のCUDAコア数の比はちょうど3.5倍であり、測定された処理時間の比率とほぼ比例しています。同様に、Tesla K20cとGeForce GT 650Mとの間の処理時間の比は、315 s/ 51 s = 6.2倍であり、コア数の比(2498個/384個=6.5倍)とほぼ比例しています。

ANSI-Cを用いて時間領域逆投影アルゴリズムをCPUに実装する場合と比較して、Tesla K20c GPUを用いると**最大約30倍の高速化が達成されます**。

図2は、CARSAR実験で取得した処理済みSARデータを可視化したものです(参考文献[1])。黄色で示されている経路の左下部分がわずかにカーブしたセンサー経路です。

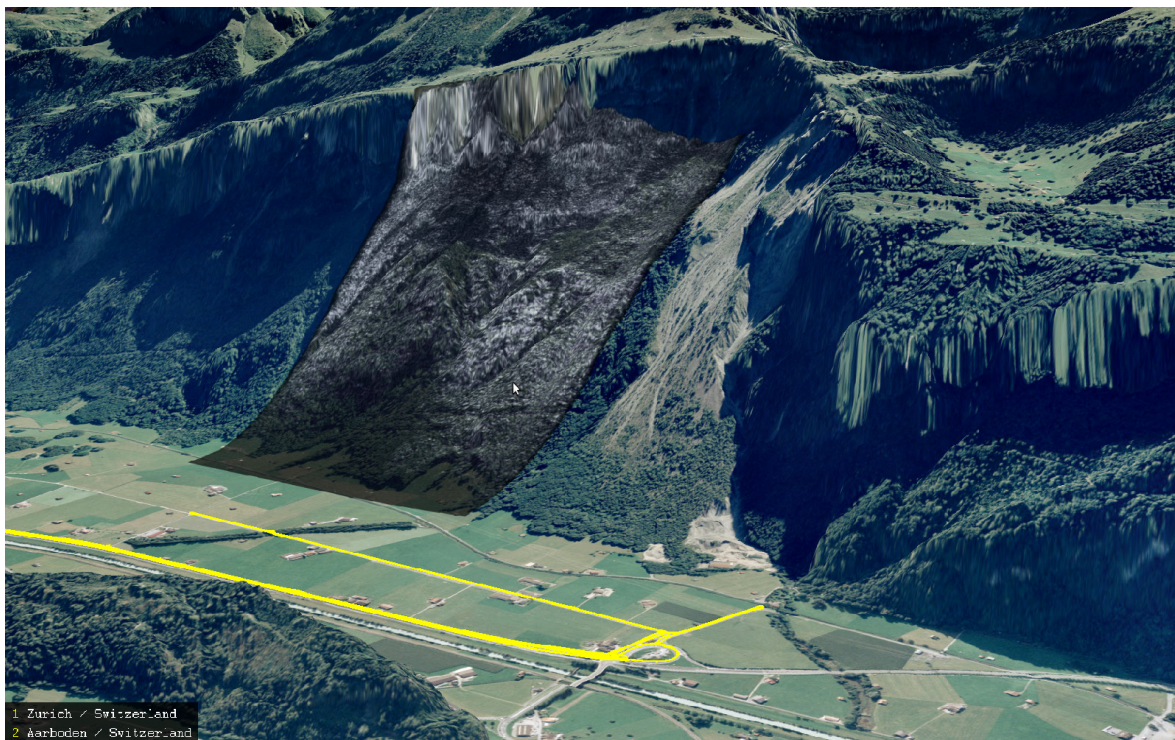


図2.車載干渉SARシステムから取得した谷斜面のSAR画像の例。SARデータは、当該領域の正射画像と高解像度DEM(© swisstopo)で3D表示できるようにジオコードされています。

IV. 考察および結論

高度に並列化されたSARアジマスフォーカシングの時間領域逆投影(TDBP)アルゴリズムについて、CUDAを用いたGPUへの実装を処理効率の観点から評価しました。

ノートパソコンおよびデスクトップパソコン用のコンシューマ向けGPUを使用することで、TDBPアジマスフォーカシングアルゴリズムをCUDAにより実装した場合、CPUが1つの場合の処理時間と比較して約5~20倍の顕著な高速化が得られました。高性能のTesla K20c GPUカードを使用すると、最大約30倍の高速化に達しました。この場合、SARフォーカシングに必要な処理時間は、車両が21 m/s(時速約76km)の平均対地速度で走行する場合の車載SARデータの取得時間と同程度です。

GPUでのTDBPフォーカシングは、曲線状のセンサー軌跡を含むSARデータ取得の魅力的な処理方式です。例えば、非常に非線形なセンサー軌道から取得した航空機搭載SARデータ(参考文献[2])や、InSAR用に曲線道路に沿って取得した車載SARの結像(参考文献[1])などが挙げられます。今回提示した実装のそれ以外の応用事例には、地球同期SARまたは飛行船搭載SARのような非定型的なSARモードのアジマスフォーカシングに関する試験や参照処理などがあります。

参考文献

- [1] O. Frey, C. L. Werner, U. Wegmuller, A. Wiesmann, D. Henke, and C. Magnard, "A car-borne SAR and InSAR experiment," in Proc. IEEE Int. Geosci. Remote Sens. Symp., 2013, pp. 93–96.
- [2] O. Frey, C. Magnard, M. Rüegg, and E. Meier, "Focusing of airborne synthetic aperture radar data from highly nonlinear flight tracks," IEEE Trans. Geosci. Remote Sens., vol. 47, no. 6, pp. 1844–1858, June 2009.
- [3] O. Frey, E. Meier, and D. Nüesch, "Processing SAR data of rugged terrain by time-domain back-projection," in SPIE Vol. 5980: SAR Image Analysis, Modeling, and Techniques X, 2005.
- [4] C. Werner, T. Strozzi, A. Wiesmann, and U. Wegmuller, "A real-aperture radar for ground-based differential interferometry," in Proc. IEEE Int. Geosci. Remote Sens. Symp., vol. 3, July 2008, pp. 210–213.
- [5] T. Strozzi, C. Werner, A. Wiesmann, and U. Wegmuller, "Topography mapping with a portable real-aperture radar interferometer," IEEE Geosci. Remote Sens. Lett., vol. 9, no. 2, pp. 277–281, Mar. 2012.
- [6] M. Soumekh, Synthetic Aperture Radar Signal Processing: with MATLAB Algorithms. John Wiley & Sons, 1999.
- [7] W. G. Carrara, R. S. Goodman, and R. M. Majewski, Spotlight Synthetic Aperture Radar: Signal Processing Algorithms. Artech House Inc., 1995.
- [8] A. Ribalta, "Time-domain reconstruction algorithms for FMCW-SAR," IEEE Geoscience and Remote Sensing Letters, vol. 8, no. 3, pp. 396–400, May 2011.
- [9] "CUFFT library user's guide," NVIDIA Corporation, Tech. Rep., 2013, <http://docs.nvidia.com/cuda/cufft>.
- [10] D. Fraser, "Interpolation by the FFT Revisited - an Experimental Investigation," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 5, pp. 665–675, May 1989.
- [11] "CUBLAS library user guide," NVIDIA Corporation, Tech. Rep., 2013, <http://docs.nvidia.com/cuda/cublas>.

(翻訳) 株式会社オープンGIS