# Synthetic Aperture Radar Image Processing by Range Migration Algorithm using Multi-GPUs

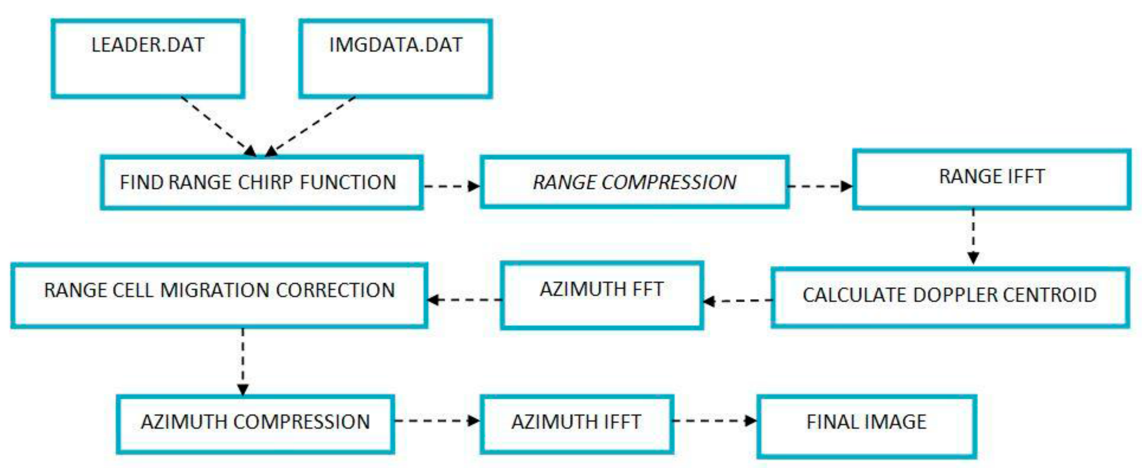
Barath Sastha S, Sachin B Patkar and Y. S. Rao
Indian Institute of Technology Bombay, Mumbai 400076

## Abstract

Range Doppler algorithm is a radar image processing algorithm used to produce high resolution images of strip map satellite data. Two dimensional images can be produced from raw data by finding the intensity and location of scatterers in the scene. Range Migration is an effect which significantly affects the resolution of the image. Range Doppler algorithm eliminates this effect and produces high quality image. Here we have implemented Range Doppler algorithm on a multi-GPU and accelerated the computation process.

## Introduction

The Range Doppler algorithm implemented using CUDA program is described below. *Note that Range Migration Algorithm is a general term for algorithms which eliminates Range Migration effects.*




The raw SAR data used for this experiment contains 27200 lines * 5616 complex elements. The entire data is subdivided into patches of size 2048 * 5616. Each patch is computed separately and the output data is merged together to produce the final image.

**Range Chirp Function** The range chirp function is found using the centre frequency and slope of the transmitted chirp which are specific to ERS satellites. The frequency domain version of range chirp function is obtained by applying an FFT and conjugate on it. Manually negating the imaginary component of complex numbers was 4 times faster than using cuConjf function to find the conjugate.

**Range Compression** The signals are brought into range spatial frequency domain by applying FFT [2]. cuFFT library has been used throughout the CUDA program to find FFTs and IFFTs. Each line of the patch is multiplied by the matched filter which is a complex multiplication. Manual complex number multiplication was around 4 times faster than using cuCmulf on a matrix of size 2048 * 8192.

**Range IFFT** After the range compression is done an IFFT operation is done on each line of the patch. As the compressed data is shorter than uncompressed data, a part of each line is thrown away as junk [3].

**Calculate Doppler Centroid** Doppler centroid frequency is the frequency of the return from the target when it is located in the centre of the beam [3]. It locates the signal energy in the azimuth frequency domain. Manual complex multiplication and addition was done instead of using cuCmulf and cuCaddf function as they were around 7 times faster for a matrix data of size 2047 * 4912.

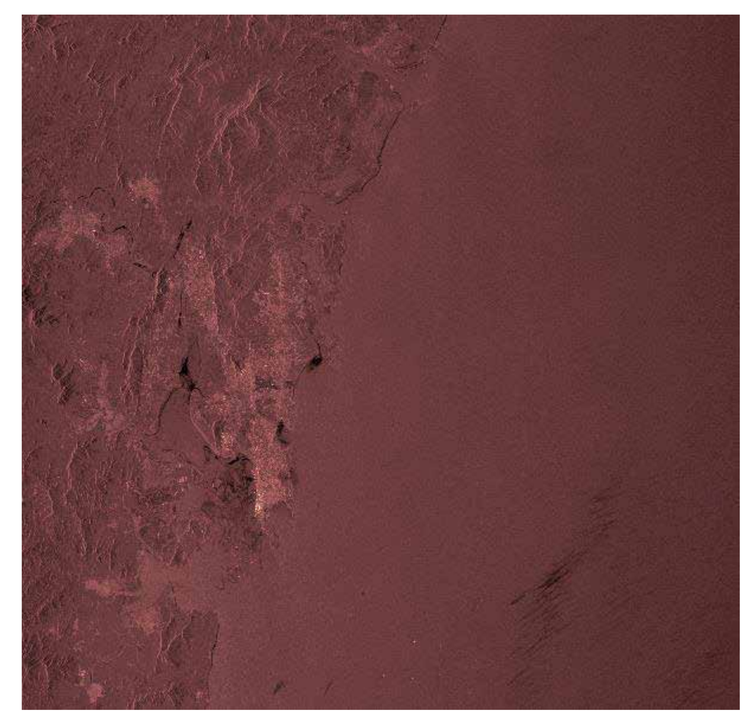
**Azimuth FFT** An azimuth reference function is found for each range bin and an FFT is applied on it.

**Range Cell Migration Correction** A scatterer is located by its range and range rate. [2]Processing a raw SAR data becomes difficult as the differential range and differential range rate to each scatterer vary over the synthetic aperture length. This effect is called range cell migration. Range cell migration correction aligns each scatterer or target in its range bin before azimuth compression is done.

**Azimuth Compression** This process is for multiplying azimuth reference function with range migration corrected data.

**Azimuth IFFT** The complex look image is obtained by taking IFFT on the azimuth compressed data. Instead of dividing each element by vector size, multiplying them by its inverse was around 5 times faster for normalization.

## SAR Image

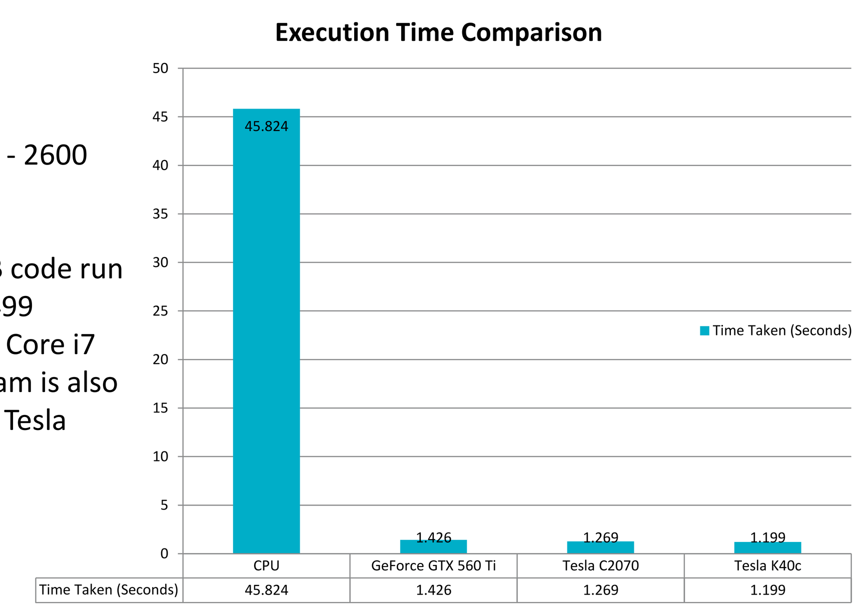


*The above image is that of Mumbai taken by ERS2 satellite.*

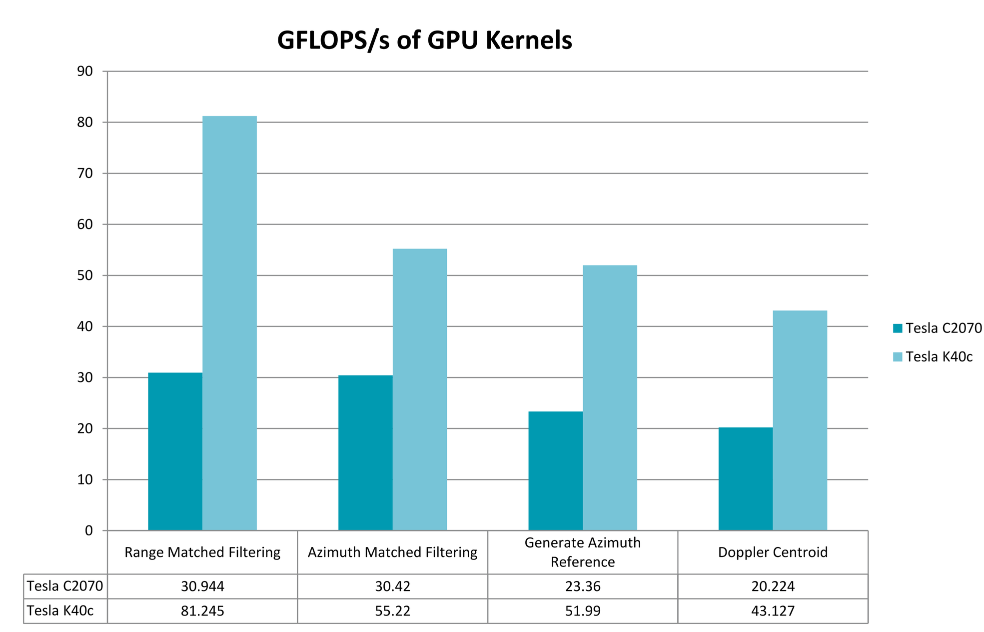
## Results

Time taken by different GPUs to produce the complete image matrix is as follows:

**Hardware Setup**
**CPU**: Intel (R) Core (TM) i7 - 2600 CPU @ 3.40 GHz
**GPU**: GeForce GTX 560 Ti
The CPU code is a MATLAB code run on 64 bit MATLAB 7.10.0.499 R2010a on the above Intel Core i7 machine. The CUDA program is also tested on Tesla C2070 and Tesla K40c GPUs.



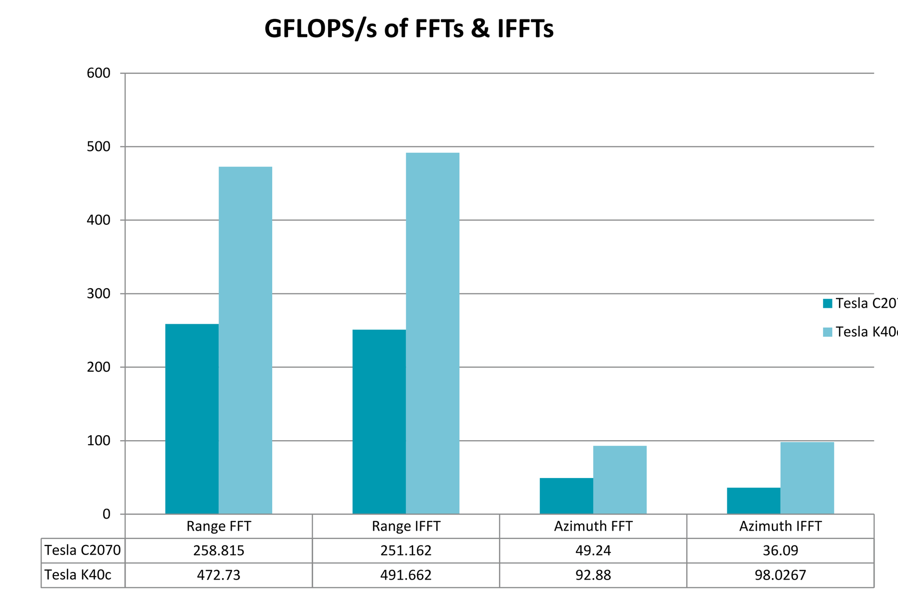

| | CPU | GeForce GTX 560 Ti | Tesla C2070 | Tesla K40c |
|---|---|---|---|---|
| Time Taken (Seconds) | 45.824 | 1.426 | 1.269 | 1.199 |




| | Range Matched Filtering | Azimuth Matched Filtering | Generate Azimuth Reference | Doppler Centroid |
|---|---|---|---|---|
| Tesla C2070 | 30.944 | 30.42 | 23.36 | 20.224 |
| Tesla K40c | 81.245 | 55.22 | 51.99 | 43.127 |

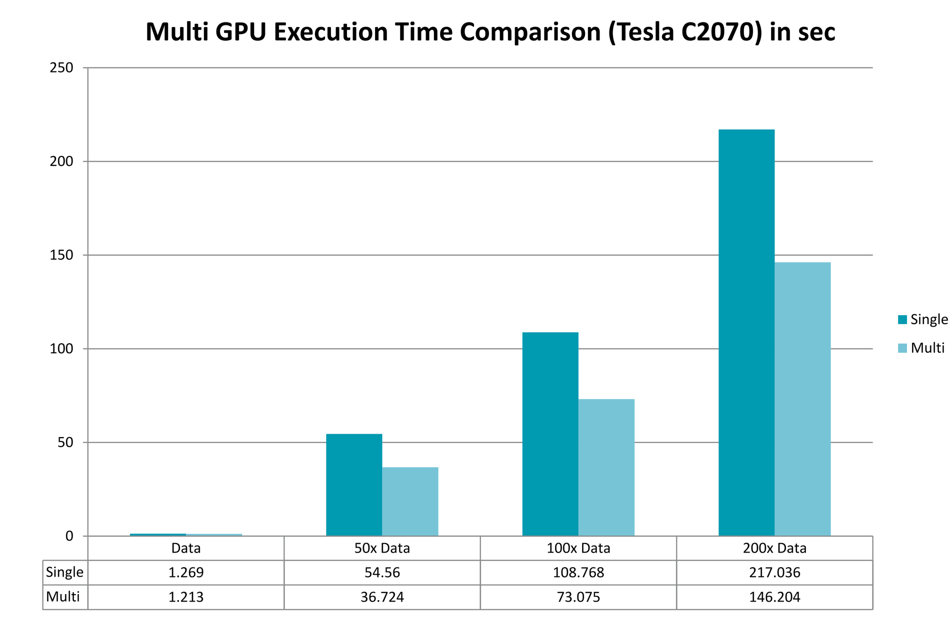
**Multi-GPU System**
Multi-GPU system used for this experiment consists of two Tesla C2070 GPUs. CUDA code is modified to invoke two pthreads and send half of the data to the first GPU and the remaining data to the second GPU.

## Results and Conclusion




| | Range FFT | Range IFFT | Azimuth FFT | Azimuth IFFT |
|---|---|---|---|---|
| Tesla C2070 | 258.815 | 251.162 | 49.24 | 36.09 |
| Tesla K40c | 472.73 | 491.662 | 92.88 | 98.0267 |

Due to overhead occurred due to cudaMalloc and cudaFree statements, there is not much advantage in using two GPUs over one for small data. As the data grows larger you can clearly see the improvement in speed of computation.




| | Data | 50x Data | 100x Data | 200x Data |
|---|---|---|---|---|
| Single | 1.269 | 54.56 | 108.768 | 217.036 |
| Multi | 1.213 | 36.724 | 73.075 | 146.204 |

Range Doppler algorithm is implemented on both single GPU as well as multi GPU system and acceleration is obtained. Future work of this project involves implementing Range Doppler algorithm for
•Raw data obtained in scansar mode which is complex and computationally intensive .
•Interferometric SAR data and data containing more range migration effects.
Our work differs from that of [4] in the following sense. In addition to optimization techniques, we have studied GFLOPS/s performance of each kernel and modified them for better performance, better asynchronous memory copy, compile time memory allocation using static global memory. Our CUDA code performance in terms of GPU runtime seems to be marginally better than that obtained in their work.

## Acknowledgements

Authors thank CUDA Centre of Excellence, IIT Bombay for access to hardware resources and Mandar Gurav for discussions.

## References

1) I.G. Cumming, F.H. Wong, Digital Processing of Synthetic Aperture Radar Data, Artech House, 2004
2) W.G. Carrara, R.S. Goodman, R.M. Majewski, Spotlight Synthetic Aperture Radar Signal Processing Algorithms, Artech House (1995)
3) https://earth.esa.int/handbooks/asar/CNTR2-6-1-2-3.html
4) Prof. Nagendre Gj jar , Vishal Mehta ,Nilesh M. Desai, PP04 Parallel Implementation of Range Doppler Algorithm for Synthetic Aperture Radar on GPU, GTC 2013.

The author can be reached at this mail id: *bsastha@gmail.com*