International Journal of Applied Engineering and Technology ISSN: 2277-212X (Online)
An Open Access, Online International Journal Available at http://www.cibtech.org/jet.htm
2014 Vol. 4 (4) October-December, pp.61-67/Sivakumar and Goyal
Research Article

GEOSPATIAL INFORMATION EXTRACTION FROM MULTISPECTRAL SATELLITE IMAGERY THROUGH GPU'S BASED PARALLEL COMPUTATION APPROACH

***Sivakumar V. [1] and *Ankit Goyal [2]**
[1]SS & DM Group, C-DAC, Pune, India, 411007
[2]Symantec Corporation, Pune, India
**Author for Correspondence*

ABSTRACT
Earth observational satellite images provide massive quantities of data that in principle could be processed and provide very valuable information in several areas in real time. The Graphics Processing Unit (GPU) based parallel computation approach play an important role in processing and analyzing a large volume of satellite imagery and speeding up the computations. In the present paper, we implemented the satellite imagery based indices computation algorithm (MNDWI) with NVIDIA CUDA-enabled GPU's. We performed both parallel and serial approach and compare the execution times and quality of performances using NVIDIA CUDA-enabled GPU's. Performance results were compared with the conventional computing approach and achieved a speed up to ~13X on NVIDIA M2090 GPU. This experimental results show that outputs can be achieved in real time with the best utilization of GPU resources and an efficient parallelization approach. Surface water was successfully delineated from multispectral LANDSAT – 8 satellite images.

Keywords: *Indices Computation, Satellite Imagery, MNDWI, NVIDIA CUDA, GPU, Parallel Programming, High Performance Computing (HPC)*

INTRODUCTION
Advancement of satellite sensor technology, the spatial, spectral and radiometric resolution of acquired remotely sensed satellite images becomes much higher and the amount of data becomes huge (Bilotta *et al.,* 2013). However, the data processing technologies have been following far behind, especially in computing speed and efficiency in handling large volume of satellite data. It is agreed that conventional approaches are grossly inadequate for these new generation sensors. High Performance Computing (HPC) has evolved largely due to increasing demands for processing speed and handling large amount of data (Bhojne *et al.,* 2013). Traditional computer may not handle large volume of satellite imageries and lack in processing speed, hence parallel processing is required for processing and analyzing such a volumes of data in time (Plaza *et al.,* 2009). Graphical Processing Unit (GPU's) provides remarkable improvement in handling large volume of data and performance. GPU programming method is growing in the real time computing research field. GPU architectures are massively used for resource-intensive computation (Christophe *et al.,* 2011; Jeong *et al.,* 2012). Among many applications, the aim of this work is to demonstrate advantages of GPU's for computing Modified Normalized Difference Water Index (MNDWI) (extracting surface water feature) from large-scale satellite imagery in real time. This involves developing scalable and portable parallel programs for a range of satellite image processing and INDICES computing.

Algorithm for MNDWI Computing
Satellite remote sensing techniques have ability to map surface water features and monitor the dynamics of surface water. Several techniques are used for extraction of water information from satellite imagery. One of the important techniques is spectral water index method. MNDWI was used, which is one of the spectral water index methods for computing water index from satellite imagery. MNDWI is modified NDWI which will enhance open water features while efficiently suppressing and even removing built-up land noise as well as vegetation and soil noise. The MNDWI is expressed as follows (Xu, 2006):

$$MNDWI = Green - MIR / Green + MIR \quad (1)$$

Where, MIR is a middle infrared band such as Operational Land Imager (OLI)'s band 6 of Landsat 8 satellite.

NVIDIA CUDA

CUDA™ (Compute Unified Device Architecture) is a parallel computing platform and programming model created by NVIDIA and implemented by only NVIDIA's graphics processing units (GPUs) (CUDA, 2013; NVIDIA®, 2012). CUDA gives developers direct access to the virtual instruction set and memory of the parallel computational elements in CUDA GPUs. Using CUDA, the GPUs can be used for general purpose processing (i.e. not exclusively graphics), this approach is known as GPGPU. GPUs have a parallel throughput architecture that emphasizes executing many concurrent threads slowly, rather than executing a single thread very quickly. It enables dramatic increases in computing performance by harnessing the power of the Graphics Processing Unit (GPU). Currently, for this application we have used one GPU. The configuration of GPU (Tesla M2090 GPU) is as given in the Table 1).

Table 1: Configuration of GPU, Tesla M2090

Properties	**Value/Details**
Compute capability	2.0
Threads per block	1024
Block dimensions	1024, 1024, 64
Grid dimensions	65535, 65535, 65535
Blocks per multiprocessor	8
No. of multiprocessors(SM)	16
Clock rate	1.301 MHz
Global memory bandwidth	169.19 GB/s
Global memory	5.249 GB
Constant memory size	64 KB

Design Approach

The test satellite imagery was downloaded from USGS data archive and mosaicked using ERDAS image processing software based on single workstation environment (Figure 1).

Table 2 shows details of satellite image used for calculating MNDWI.

A pixel wise parallel approach was used for processing and obtaining result. Efficient utilization of GPU resources can be achieved by designing the flow of execution in such a way that the logic will be implemented on all the pixels of the image at a time, in parallel.

The total number of threads was chosen to be equal to the total number of pixels in the image (number of rows multiplied by number of columns in the input image), with each thread handling each pixel. Sequential approach also employed for computing MNDWI to understand the computational efficiency and comparison.

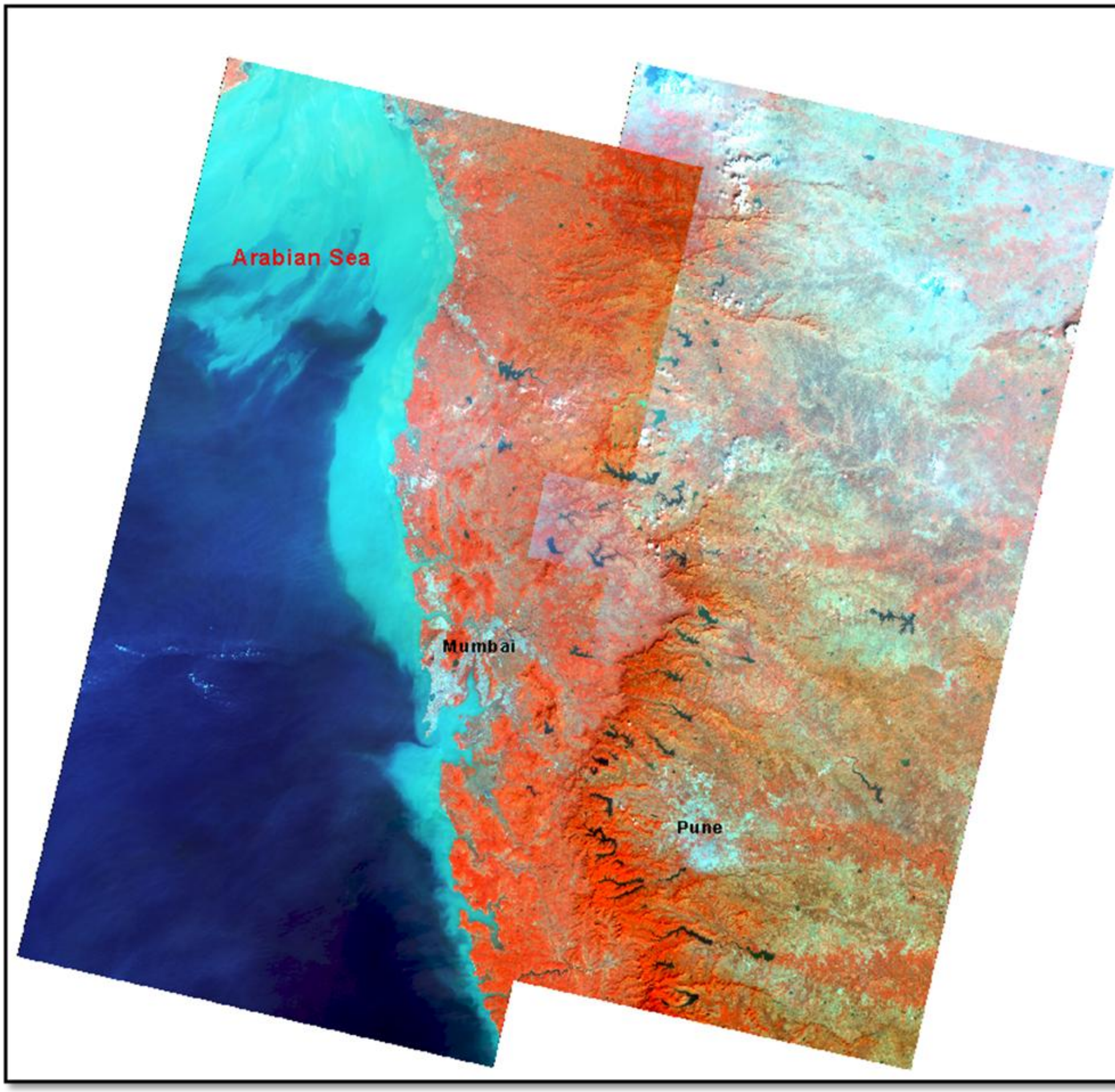

Figure 1: FCC (RGB: b5, b3, b2) of Operational Land Imager (OLI)'s of Landsat 8 satellite (input image for MNDWI computing). Image covers part of Maharashtra

Table 2: Specification of satellite image

Attribute	**Details**
Name of the satellite data	Landsat 8
No. of spectral bands	11
Spatial Resolution	30 meter
Radiometric Resolution	12 bit
Format	Geo-TIFF
Width(in pixels)	14241
Height(in pixels)	13191
File Size	2.5 GB

International Journal of Applied Engineering and Technology ISSN: 2277-212X (Online)
An Open Access, Online International Journal Available at http://www.cibtech.org/jet.htm
2014 Vol. 4 (4) October-December, pp.61-67/Sivakumar and Goyal
Research Article

RESULTS

Experimental Output

MNDWI algorithm was successfully implemented on GPU in parallel and serial approach. MNDWI was effectively calculated from multi-spectral Landsat 8 (Operational Land Imager) satellite imagery and indices range from 0.1 to 0.45 (Figure 3). MNDWI values shows higher for clear and deep water and low value for muddy and shallow water. Performance test was carried out using NVIDIA Visual Profiler (Table 3, 4, 5, 6 & 7). Figure 2 representing occupancy for various block sizes.

Table 3: Details of Computational Timing for MNDWI Computing

Parameter	Value
Total no. of threads	187853031 (14241*13191)
Grid size	[19, 13191, 1]
Block size	[768, 1, 1]
Threads that run concurrently	24576 (48 warps each of 32 threads * 16 SM)

Table 4: Statistics of achieved Efficiency on GPU

Attribute	Details
Performance	563.559093 GFlops
Achieved occupancy	81.4 %
Warp Execution Efficiency	100 %
Global memory Store Efficiency	100 %
Global memory Load Efficiency	100 %
Branch Efficiency	100 %
Multiprocessor activity	99.9 %
GPU Execution Time	22.675 ms

Table 5: GPU DRAM Usage Statistics

Operation	Time	Avg. throughput	Data size
Memory transfer Host DRAM to GPU DRAM	246.75 ms	5.67 GB/s	1.4 GB
Memory transfer GPU DRAM to Host DRAM	114.343 ms	6.12 GB/s	716.602 MB

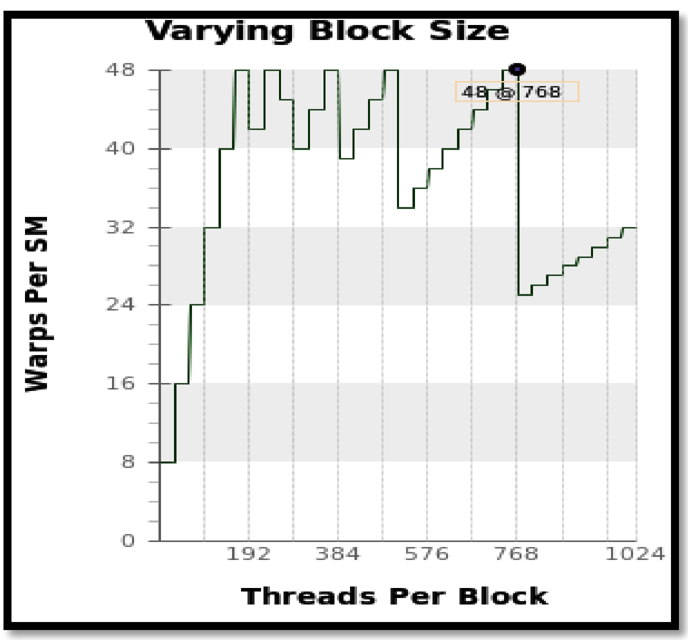

Figure 2: Representing Occupancy for Various Block Sizes

Table 6: Memory Bandwidth used by the Kernel

L1/Shared Memory			
Local Loads	0	0 B/s	
Local Stores	0	0 B/s	
Shared Loads	0	0 B/s	
Shared Stores	0	0 B/s	
Global Loads	11747712	75 GB/s	
Global Stores	5873760	37.5 GB/s	
L1/Shared Total	17621472	112.5 GB/s	Idle Low Medium High Max
Texture Cache			
Reads	0	0 B/s	Idle Low Medium High Max
L2 Cache			
Reads	46967456	75.01 GB/s	
Writes	23481629	37.5 GB/s	
Total	70449085	112.51 GB/s	Idle Low Medium High Max
Device Memory			
Reads	62616273	100.01 GB/s	
Writes	25806046	41.2 GB/s	
Total	88422319	141.21 GB/s	Idle Low Medium High Max

Table 7: Kernel Execution Statistics

Variable	Achieved	Theoretical	Device Limit	Grid Size: [19,13191,1] (250629 blocks) Block Size: [768,1
Occupancy Per SM				
Active Blocks		2	8	0 1 2 3 4 5 6 7 8
Active Warps	39.05	48	48	0 5 10 15 20 25 30 35 40 45 48
Active Threads		1536	1536	0 256 512 768 1024 1280 1536
Occupancy	81.4%	100%	100%	0% 25% 50% 75% 100%
Warps				
Threads/Block		768	1024	0 128 256 384 512 640 768 896 1024
Warps/Block		24	32	0 3 6 9 12 15 18 21 24 27 30 32
Block Limit		2	8	0 1 2 3 4 5 6 7 8
Registers				
Registers/Thread		9	63	0 8 16 24 32 40 48 56 63
Registers/Block		7680	32768	0 8k 16k 24k 32k
Block Limit		4	8	0 1 2 3 4 5 6 7 8
Shared Memory				
Shared Memory/Block		0	49152	0 16k 32k 48k
Block Limit		8	8	0 1 2 3 4 5 6 7 8

Research Article

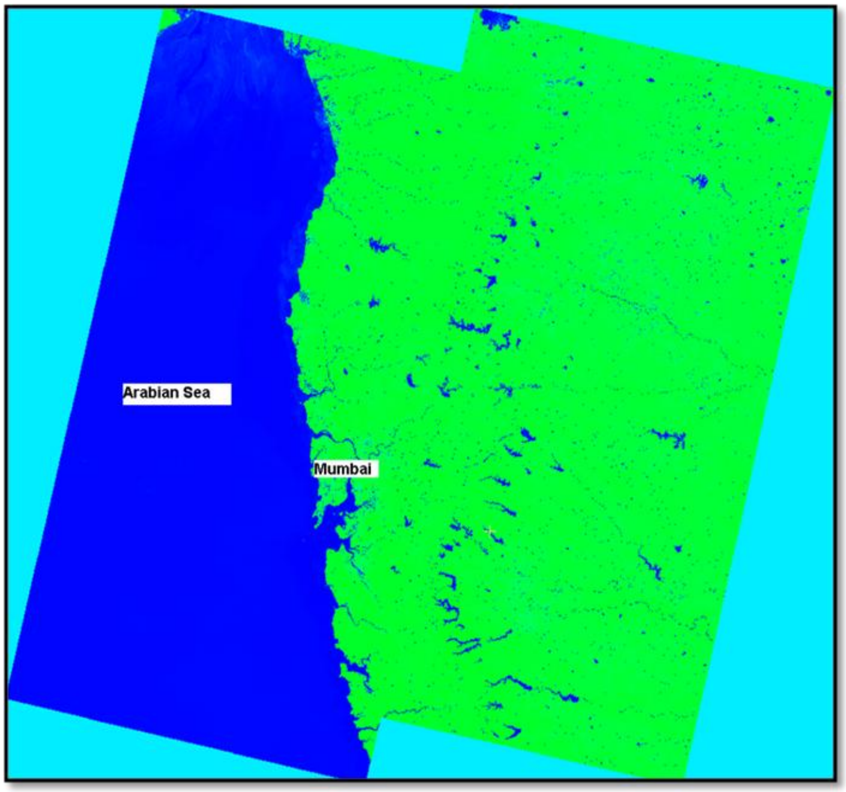

Figure 3: MNDWI output derived using NVIDIA-CUDA enabled GPU's. Blue color indicates water surface (MNDWI indices range from ~ 0.1 to 0.45)

Conclusion
MNDWI was computed on GPU in parallel and serial approach. MNDWI indices range from ~0.1 to 0.45. MNDWI values shows higher for clear and deep water; low value for muddy and shallow water. The achieved efficiency of parallel processing approach on GPU was 383.768 ms (memory operation (HàD and DàH) + GPU's SM Execution Time = 246.75+114.343+22.675 = 383.768 ms). Serial processing approach was carried out on Intel processor of clock speed 2.6 GHz and total execution time achieved on CPU was 4990 ms. Parallel and serial processing was compared and overall speedup was calculated. This experiment shows that parallel processing is ~13X faster than serial processing approach. The experimental results indicates that NVIDIA-CUDA enabled GPU's and parallel programming techniques provides the suitable solution for real time information extraction and handling large volume of satellite data. Future work is to implement different algorithms and methods to further speedup and validating results.

ACKNOWLEDGEMENT
Authors wish to thankfully acknowledged C-DAC for providing resources and support to carry out the study.

REFERENCES
Bhojne Mamta, Abhishek Chakravarti, Anshu Pallav and Sivakumar V (2013). High Performance Computing for Satellite Image Processing and Analyzing–A Review. *International Journal of Computer Applications Technology and Research* **2** 424-430.
Bilotta G, Sanchez RZ and Ganci G (2013). Optimizing Satellite Monitoring of Volcanic Areas Through GPUs and Multi-Core CPUs Image Processing: An OpenCL Case Study, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **6**(6) 2445,2452.

Christophe E, Michel J and Inglada J (2011). Remote Sensing Processing: From Multicore to GPU, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **4**(3) 643,652.

CUDA (2013). Toolkit documentation [Online]. Available: https://developer.nvidia.com.

Jeong In-Kyu, Min-Gee Hong, Kwang-Soo Hahn, Joonsoo Choi and Choen Kim (2012). Performance Study of Satellite Image Processing on Graphics Processors Unit Using CUDA. *Korean Journal of Remote Sensing* **28**(6).

Liu Y and Hu J (2011). GPU-based parallelization for fast circuit optimization, *ACM Transactions on Design Automation of Electronic Systems* **16**(3) 24.

NVIDIA® Nsight™ (2012). Development Platform, Visual Studio Edition 2.2 User Guide 2012, Available: http://http.developer.nvidia.com.

Plaza A, Benediktsson JA, Boardman J, Brazile J, Bruzzone L, Camps-Valls G, Chanussot J, Fauvel M, Gamba P, Gualtieri J, Marconcini M, Tilton JC and Trianni G (2009). Recent advances in techniques for hyperspectral image processing, *Remote Sensing of Environment* **113** 110–122.

Xu Hanqiu (2006). Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery. *International Journal of Remote Sensing* **27**(14) 3025-3033.