

# High performance GPU computing based approaches for oil spill detection from multi-temporal remote sensing data



Ujwala Bhangale<sup>a</sup>, Surya S. Durbha<sup>a,\*</sup>, Roger L. King<sup>b</sup>, Nicolas H. Younan<sup>b</sup>, Rangaraju Vatsavai<sup>c</sup>

<sup>a</sup> Centre of Studies in Resources Engineering (CSRE), IIT Bombay, Mumbai 400076, India

<sup>b</sup> Department of Electrical and Computer Engineering, Mississippi State University, MS, USA

<sup>c</sup> Department of Computer Science, North Carolina State University, Raleigh, USA

## ARTICLE INFO

### Article history:

Received 17 July 2016

Received in revised form 5 January 2017

Accepted 25 March 2017

Available online 10 April 2017

### Keywords:

High performance computing (HPC)

Graphics processing unit (GPU)

Oil spill

Morphological attribute profiles

## ABSTRACT

Oil spills have adverse effects on the environment and economy. Near real time detection and response activities enable to better manage the required resources at the incident area for clean-up and control operations. Multi-temporal remote sensing (RS) technologies are widely used to detect and monitor oil spills on the Ocean surfaces. However, current techniques using RS data for oil spill detection are time consuming and expensive in terms of computational cost and related infrastructure. The main focus of this work is oil spill detection from voluminous multi-temporal LANDSAT-7 imagery using high performance computing technologies such as graphics processing units (GPUs) and Message Passing Interface (MPI) to speed up the detection process and provide rapid response. Kepler compute architecture based GPU (Tesla K40) with Compute Unified Device Architecture (CUDA), which is a parallel programming mechanism for GPU is used in the development of the detection algorithms. Oil spill detection techniques that were adapted to GPU based processing include band-ratio and Morphological attribute profile (MAP) based on six structural and shape description attributes namely, Gray mean, standard deviation, elongation, shape complexity, solidity and orientation. Experimental results show the significant gains in the computational speed of these techniques when implemented on a GPU and MPI. A GPU vs. CPU comparison shows that the proposed approach achieves a speedup of around 10× for MAP and 14× for band ratio approaches, which includes the data transfer cost. However, the MPI implementation using 64 cores outperforms the GPU, and executes the time intensive task of computing the above said attributes in only 18 min, whereas a GPU consumes around an hour.

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

Remotely sensed imagery provides a vast source of information for monitoring and management activities for natural and man-made disasters such as earthquakes, floods, hurricanes, forest fires, tsunamis, oil spills etc. Oil spilling is the release of liquid petroleum hydrocarbon in the ocean or coastal water and has adverse effects on the environment and economy. Marine and coastal habitats, wildlife species, recreational activities, local industry, and fisheries are some of the sectors that get affected by oil spills. Frequent occurrence of oil spills across several countries clearly indicates the need for rapid oil spill detection and monitoring approaches. This would greatly help in the implementation of preventive and control actions, and also facilitate real time emergency response (Brekke and Solberg, 2005).

Many countries use a combination of spaceborne and airborne sensors for oil spill surveillance. Spaceborne remote sensing (RS) covers large geographical area; hence the information obtained from it can be

used to generate the first warning, while further detailed analysis can be done using airborne data (Brekke and Solberg, 2005). Also, spaceborne sensors are more cost effective than the airborne sensors (Casciello et al., 2007).

In this work, satellite based remote sensing data is used for the detection of the oil spill in its initial stage as the oil spill thickness is more, and can be detected in the visible range of Electromagnetic radiation. However, after certain duration, the oil spill forms a thin layer over the sea surface, and hence it is challenging to detect the oil spill using the visible wavelength range of space borne sensors.

Some of the state of the art techniques used for oil spill detection include, Robust Satellite Techniques (RST) (Casciello et al., 2007; Grimaldi et al., 2010; Grimaldi et al., 2009; Casciello et al., 2011), thermal infrared based techniques such as Robust Estimator of TIR Anomalies (RETIRA) using thermal inertia and sea surface temperature (Grimaldi et al., 2009), collective techniques on satellite images such as region selection, feature extraction, and oil spill classification (Brekke and Solberg, 2005; Del Frate et al., 2000).

The trajectory of oil spill is affected greatly by wind flow and ocean current, slick gets elongated because of wind and starts to drift over

\* Corresponding author.

E-mail address: [sdurbha@iitb.ac.in](mailto:sdurbha@iitb.ac.in) (S.S. Durbha).

the sea surface (NOAA, 2014) and oil spill regions form unique shapes; considering this phenomenon, oil spill regions can be distinguished using its shape description characteristics. Various oil spill satellite images shown in Fig. 1 further substantiate this observation, since unique shapes of the oil spill regions are clearly noticeable.

Solberg et al. (2007) have used various shape features for classifying oil regions such as *slick complexity*, *slick area*, *width* etc., and contrast-based features such as *slick local contrast*, *border gradient* etc., and obtained 78% classification accuracy for oil spills and 99.4% classification accuracy for lookalikes. They have added around 37 rules to improve the accuracy of the classification; these rules resemble attribute filtering criteria of MAP technique. Alawadi (2012), lists various shape-based features that can be used for oil spill detection such as *slick area*, *slick perimeter*, *perimeter to area ratio*, *slick complexity*, *slick width* etc. Topouzelis (2008), discusses various shape characteristics of the oil spill which helps to discriminate it from its lookalikes such as roundness, elongation etc. Stathakis et al. (2006), have extracted around 25 geometrical, physical and texture features for oil spill detection and observed around 86% accuracy using neural network classification.

In this work, after applying the attributes selection process on various characteristics of oil spill objects, six most relevant attributes are considered for MAP based oil spill detection process.

It is worth to mention here that the shape description attributes i.e. *solidity* and *orientation* are proposed for the first time using convex hull characteristics, to identify the oil spill regions, which plays an important role to distinguish between lookalike and oil spill regions. A systematic and suitable MAP approach is used in this work for oil spill detection.

To capture the unique shape of the oil spill, and describe its attributes, a polygon, which encloses the connected components, seems to be an appropriate choice. Six different attributes are considered for profiling the images. These attributes are, Gray Mean (GM) that observes the spectral similarity of objects, Standard Deviation (SD) that observes the homogeneous regions, Slick Complexity (SC) that observes shape complexity, Elongation (EL) that observes the length of the object, Solidity (SL) that observes the solidity of the object and Orientation (OR) that observes the direction of the object.

These six attributes are discussed in more detail in Section 3.

Since, the current techniques for oil spill detection are time consuming and computationally expensive, hence in this work we develop high performance computing (HPC) based techniques using Kepler compute architecture based GPU (Tesla K40) to enable rapid oil spill detection.

However, it can be argued about the need for a GPU-based HPC for executing a task, which normally takes only a few minutes on the CPU. This argument has merit when the processing is a one-time activity. But detection of disaster-affected areas involves continuous multi-temporal collection, and evaluation of the data for monitoring and control action, which requires several images to be processed per day. In the case of the Sichuan earthquake, 1300 images from 22 different sensors

were used for monitoring and evaluating the affected area Lewis (2009), for the Nepal earthquake, which occurred in April 2015, around 18 sensors were used for analysis (Disasters Charter, 2015); several pre and post disaster images were used for evaluation. Considering the tremendous amount of data involved in disaster management related activities, even if a High performance computing (HPC) approach (e.g. GPU-based computing) could save around a minute per image, then for example, for 1000 images it would save around 16 h.

Earlier work on GPU based processing for satellite data illustrates its suitability in terms of infrastructure requirement, implementation complexity, cost, and speed (Plaza et al., 2011), also other different HPC technologies such as field-programmable gate array (FPGA), message passing interface (MPI) are explored for remotely sensed data. Recently, GPUs are used for hyperspectral unmixing (Bernabe et al., 2013a), hyperspectral anomaly detection (Bernabe et al., 2013b; Ma et al., 2014) have designed parallel programming templates to facilitate the programmer with simplified development of parallel remote sensing image processing algorithms on GPU.

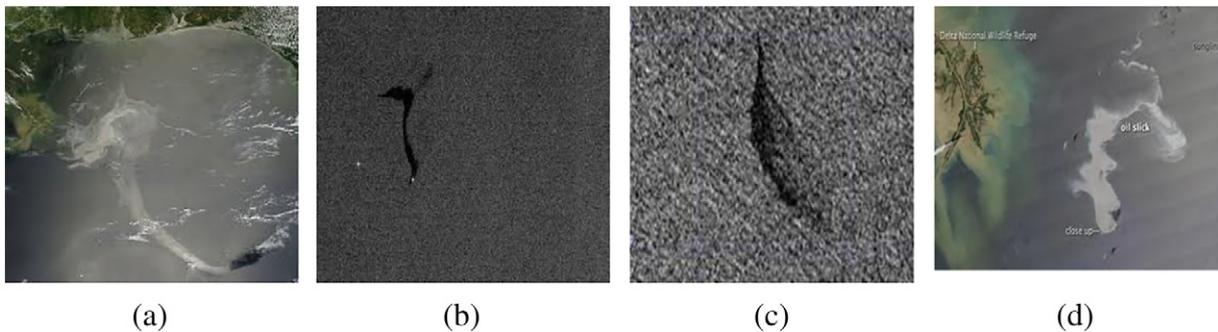
The core objectives of this work are:

- Develop rapid high performance computing (HPC) based oil spill detection approaches
- Exploit the inherent parallelism of GPU, and reformulate the algorithms to take full benefit of the high performance computing capability of GPU. Also, exploring MPI platform for time intensive attributes computation task, and its performance comparison with GPU.
- Evaluate the computational gains obtained by rapid detection approaches developed on a GPU with their sequential implementations on a CPU.
- Validation of the oil spill detection results obtained with relevant geometric and thematic accuracy metrics.

## 2. Need for high performance computing approaches for rapid oil spill detection

Currently, spaceborne, airborne, multispectral, hyperspectral with varying spectral range such as visible bands, short wave infrared bands, thermal bands etc., have been used for oil spill detection. There are a variety of algorithms for oil spill detection (Grimaldi et al., 2009; Casciello et al., 2011; Del Frate et al., 2000; Plaza et al., 2005; Cai et al., 2007)

- Algorithms that use different spectral properties (such as visible bands, thermal bands, radio bands etc.)
- Algorithms that take advantage of the spatial properties (such as shape, texture, and homogeneity etc.) with high spatial resolution, with varying temporal resolution



**Fig. 1.** Oil spill regions captured by satellite imagery, (a) MODIS Terra image of oil spill at gulf of Mexico acquired on 17th May 2010, (b) Possible oil slick captured by RADARSAT-2 SAR image on 25th June 2009, (c) 3–4 km long oil slick detected by RADARSAT-2 in deep water of Australian coast (d) MODIS Aqua image of oil spill at gulf of Mexico acquired on 25th April 2010.

- Algorithms that combine spectral characteristics, spatial characteristics, temporal characteristics of the data to get robust response and avoid false alarm

The implementation of this wide range of algorithms for oil spill detection in near real time is challenging because of the huge amounts of data that needs to be processed. Hence, high performance computing infrastructure is essential to overcome this problem. Further, the inherent characteristics of RS data, and the sequential algorithms in vogue are not beneficial for the rapid detection of the oil spills due to:

(i) *High spatial, spectral, and temporal resolution of the satellite data*

Nature of the satellite data is complicated; it includes spatial, spectral and temporal components, for example, Landsat ETM+ sensor provides 8 bands for spatial resolution 30 m (bands 1–5,7), 60 m (bands 6), 15 m (band 8-Panchromatic). A single scene of Landsat ETM+ used for detection of oil spills is around 231 MB. Hence, this grows quickly if the detection and monitoring is carried out on multi-temporal imagery.

(ii) *Computationally expensive algorithms*

Oil spill algorithms can be inherently computationally expensive, for example: techniques involving multiple steps such as region selection, feature extraction, oil spill classification etc. in which each of these components are computationally expensive (Brekke and Solberg, 2005; Del Frate et al., 2000). The approach proposed by Del Frate et al. (2000) have multiple steps in the detection process such as selection of ROI, computation of physical and geometrical features for extracting specific characteristics of the object and classification of the object into oil spill.

Some algorithms combine spectral, spatial, physical, temporal characteristics together to get more robust results. For example, Plaza et al. (2005) have used both spatial and spectral characteristics of hyperspectral data (CASI and AVIRIS) simultaneously to detect pure and mixed pixels of oil spill using extended morphological operations. They have also acknowledged the heavy computation burden while processing such high dimensional data.

Another approach for oil spill detection was proposed by Cai et al. (2007). They have calculated the sea surface temperature using four thermal infrared bands; surface albedo is obtained through weighted index of spectral reflectance of ASTER data using 6S model. Further, apparent thermal inertia is calculated as a ratio of surface albedo and diurnal temperature difference. They have found that the apparent thermal inertia difference between the oil spill and the surrounding seawater is higher. Hence, the apparent thermal inertia is more suitable to detect the oil spill. Also, the method used to derive sea surface temperature was developed by Xue et al. (2005) which is robust, but very time consuming as it involves multiple stages (it takes around 20 min for an image of size  $256 \times 256$ ). This high computing time is mentioned as a limitation for working with large image sizes.

Recently a comparative study of the several classification algorithms for oil spill detection was carried out using RADARSAT-1 data (Xu et al., 2014). Classification algorithms such as SVM, ANN, tree-based ensemble classifiers (bagging, bundling and boosting), Generalized Additive Model (GAM) and Penalized Linear Discriminant Analysis (PLDA) were studied for their predictive capability. However, the training phase in the development of these models is highly computationally expensive, for example, Support Vector Machines (SVM) includes computationally intensive tasks during the training phase i.e. it solves a quadratic programming problem to find separating hyperplanes which involves computationally expensive operations (Taşkın et al., 2011).

Further, Del Frate et al. (2000), have used a neural network (NN) to classify the oil spill features. The training time is very high for NN since it requires iterative adjustment of the weight coefficients to minimize the error function equal to mean square difference between the desired and actual output, in addition, several attempts are required to finalize the number of units in hidden layers. If back-propagation algorithm is used in NN, the training time increases significantly. In general, to reduce the false alarm rate using machine-learning approaches, a large

training set is required and this can lead to high computational overhead.

Solberg et al. (2007) have used different types of features such as shape based features, contrast based features, homogeneity features etc., for feature extraction of Radarsat and Envisat SAR imagery; further they have classified it using a combination of statistical classifiers based on probability principle and rule based approach. They have added 37 rules to overcome false alarm.

(iii) *Lack of near instantaneous feedback for end users and developers*

Generally, for optimum parameter selection, repetitive execution of the application with different parameter values is performed to build classification models, such as selecting cost parameter or penalty parameter in SVM, which is time intensive process (Gokaraju et al., 2011). A parameter set is applied on an algorithm and the response is evaluated. Currently, the time it takes for many of these algorithms to build the model prohibits their use on large data sets. Hence, instantaneous feedback using HPC will be really helpful for developers and end users to avoid a long waiting time to see the response of how well a particular parameter set has performed.

(iv) *Limited capabilities for immediate response to enable near real time detection and monitoring*

Many coastal countries have oil surveillance systems in place to manage oil spills (Brekke and Solberg, 2005). However, it is a challenge to provide near real time response for detection and monitoring due to large spectral ranges, voluminous data, and complex algorithms. Therefore, high performance computing is essential to provide near real time response for oil spill events.

Based on the above, it is evident that HPC can play an important role in disaster applications. Below is a brief description of the high performance GPU Kepler architecture, and CUDA platform used for parallel implementation - a technology used in this work for HPC.

### 3. Methodology for oil spill detection

In this work, two oil spill detection techniques i.e. Band ratio (Taravat and Del Frate, 2012; Srivastava and Singh, 2010), Morphological Attribute Profile (MAP) (Mura et al., 2010), which are shape description attributes are considered. In addition, new shape description attributes specifically for oil spill detection have been proposed in this work. Fig. 2 shows the proposed methodology.

The algorithms used for parallelization and subsequent implementations on HPC technologies to achieve high performance are described below.

#### 3.1. Oil spill index based on spectral characteristics

Imagery of Oil spills is classified using different spectral bands such as visible, near infrared, thermal infrared, ultraviolet and microwave bands. Oil has high surface reflectance as compared to water in the visible range (480–570 nm) of the electromagnetic spectrum. The thermal infrared (TIR) spectral band (7000–14,000 nm) can be used to identify contrasting patterns to discriminate oil from water. Information such as oil thickness and oil to water emulsion ratios can be derived from oil absorption features in the NIR (700–2500 nm) spectral band (Leifer et al., 2012). Also, ultraviolet (UV) (250–350 nm) passive sensors can be used to detect oil spills due to its strong reflectance than water in this region. Microwave (2–8 mm) radiometer, which is a passive sensor, can also be used for oil spill detection and oil thickness measurement (Fingas and Brown (2014), Jha et al. (2008).

Shorter wavelengths are more sensitive for optical signature of oil (Taravat and Del Frate, 2012). Oil and water reflectance increases in the range of 0.475–0.675  $\mu\text{m}$  and have different levels of absorption in the 0.675–0.800  $\mu\text{m}$  range (Taravat and Del Frate, 2012; Srivastava and Singh, 2010). Oil spill index can be obtained as a ratio of B4/B2 or B3/B2 for LANDSAT ETM+. LANDSAT ETM+ band 1 (0.45–0.515

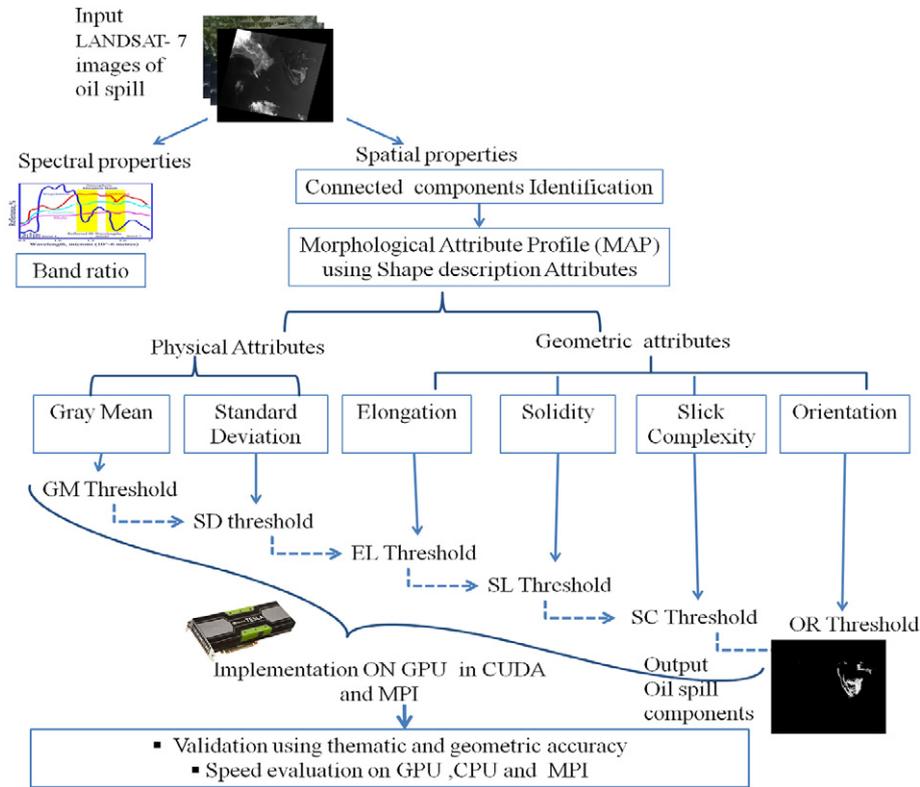


Fig. 2. High performance oil spill detection methodology using spectral and spatial properties.

$\mu\text{m}$ ), band 2 (0.525–0.605  $\mu\text{m}$ ) and band 3 (0.63–0.69  $\mu\text{m}$ ) is used in this work. The oil spill signature is calculated as given in Eq. (1).

$$\text{Oil spill index} = (B3/B2)/B1 \quad (1)$$

The above ratio is normalized using band1 as it also includes the signature of biogenic materials such as algae.

### 3.2. Morphological attribute profile (MAP)

Attribute profile characterizes an image by applying different attribute filters (area, standard deviation moment of Inertia, etc.) sequentially, which can model various kind of structural information to enable the detection of relevant geospatial objects. It provides the flexibility to model different kinds of objects by varying the attribute filtering criteria, or by selecting a subset of relevant attributes and then applying the filtering criteria.

In order to derive the appropriate profile for geospatial objects, the three major steps are:

1. Selection of appropriate attributes, which are responsible for modeling the structural information.
2. Deciding the sequence of attributes in which the filtering should be applied.
3. Selection of threshold criterion for each attribute to retrieve a specific kind of object.

Morphological attribute filters such as area, standard deviation, and moment of inertia are currently been used to obtained the structural information from satellite images such as buildings, roads etc. (Mura et al., 2010; Breen and Jones, 1996; Pesaresi and Benediktsson, 2001). However, they have not been fully explored for oil spill detection.

Binary attribute opening is extended in Breen and Jones (1996), for gray-scale image attribute opening using threshold sets.

For a gray-scale image  $f$ , Threshold set  $T_t(f)$  is defined as

$$T_t(f) = \{x \in E \mid f(x) \geq t\} \quad (2)$$

where  $E$  is a finite and a discrete set and  $f(x)$  is the gray-scale value at point  $x$  of image  $f$ .

Gray-scale attribute opening  $\gamma^T$  of image  $f$  at point  $x$  is defined as

$$\gamma^T(f)(x) = \max\{t \mid x \in \Gamma_x[T_t(f)]\} \quad (3)$$

where  $T_t(f)$  is an image obtained by thresholding  $f$  at gray level  $t$ ,  $\Gamma_x$  is the connected opening which preserves the regions from the image that contains point  $x$ , and all other regions are eliminated. Gray-scale attribute closing is analogous to Gray-scale attribute opening (Mura et al., 2010; Breen and Jones, 1996).

**Attribute opening profile** with increasing threshold criteria  $T = \{T_t \mid t = 0, \dots, n\}$  is defined as

$$\Pi_{\gamma^T}(f) = \{\Pi_{\gamma^T_t} \mid \Pi_{\gamma^T_t} = \gamma^T_t(f), \forall t \in [0, \dots, n]\} \quad (4)$$

where  $\Pi_{\gamma^T}(f)$  is an attribute opening profile as a set of all attribute opening images  $\gamma^T_t(f)$ , where  $\gamma^T_t(f)$  is a gray-scale attribute opening of image  $f$  at threshold  $t$ . If the criterion  $T$  is non-increasing in the above equation, then it is an attribute thinning operation. Filtering attributes such as standard deviation, moment of inertia can be used to perform attribute thinning. Attribute closing profile is analogous to opening profile. More details of this methodology are given in (Mura et al., 2010).

#### 3.2.1. Relevance of the attributes selected for oil spill detection

Attributes used for MAP process are based on the shape of the region such as *elongation*, and/or derived from geometric characteristics such as *Slick Complexity*, which is derived from area and perimeter of the polygon enclosing the region. The attributes listed below were selected as optimal subset after applying an attribute selection process. A Wrapper based genetic algorithm (Durbha et al., 2010) approach is used for

the feature subset selection. The attributes, Solidity and Orientation are proposed for the first time in this work for oil spill detection.

#### 1. Gray mean (GM)

Spectral similarity i.e. the regions having similar spectral reflectance, can be detected using Gray mean attribute; Mean value of each connected component C is calculated as,

$$GM = \frac{\sum_{i=1}^n p_i}{n} \quad (5)$$

where,  $p_i$  is intensity value of each pixel that belongs to C, and  $n$  is the total number of pixels that belong to C;  $n$  is also referred to as the *area* of C.

Regions belonging to one category will have similar mean gray value. It may include the components, which are lookalikes such as cloud and oil spill regions. GM is a non-increasing filtering attribute; hence, attribute opening is referred to as attribute thinning (Mura et al., 2010; Breen and Jones, 1996).

#### 2. Standard deviation (SD)

Structural similarity such as homogeneity can be observed by obtaining Standard Deviation (SD) of the connected components. SD calculation is benefited because of earlier calculated GM of each connected component,

$$SD = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - GM)^2} \quad (6)$$

SD is non-increasing filtering criteria.

#### 3. Slick complexity (SC)

Slick complexity (Solberg et al., 2007) is used to measure the complexity of the oil spill region. It is similar to shape index (Jiao and Liu, 2012). High slick complexity values indicate complex shapes, whereas low values indicate simpler shapes. Oil spill objects usually have complex geometrical shapes as they are altered continuously by wind or current, this property helps to distinguish them from their lookalikes. Slick complexity is calculated as

$$SC = \frac{P^2}{n} \quad (7)$$

where, P is the perimeter and n is the area (i.e. total number of pixels) of the connected component. Area for each connected component is calculated earlier for GM.

#### 4. Elongation (EL)

Geometric similarity, which is based on the length of the region, can be observed using elongation attribute. The use of Elongation attribute is appropriate since, it is one of the well-observed characteristics of the oil spill as seen in Fig. 1. It is an increasing filtering attribute. Elongation of the connected component is computed as:

$$Elongation = \frac{\text{Length of Major Axis of Convex hull enclosing C}}{\text{Length of Minor Axis of Convex hull enclosing C}} \quad (8)$$

#### 5. Solidity (SL)

Solidity is a ratio of the area occupied by the object to the area of its convex hull, basically it analyses the concavity or convexity of a region. It is calculated as,

$$Solidity = \frac{\text{Area of C}}{\text{Area of convex hull enclosing C}} \quad (9)$$

Lookalike regions can be filtered out further using solidity, for example, cumulus types of clouds which are cellular (individual) in nature and have rounded top always have maximum solidity as compared to oil spill regions.

#### 6. Orientation (OR)

Orientation defines the angle between the x-axis and the major axis of the convex hull drawn around the region. Orientation helps to

understand the direction of oil flow influenced by the water current and the wind.

To compute orientation and major, minor axis of convex hull covering the region, the technique discussed below (Chaudhuri and Samal, 2007) is applied. Consider the object A with  $n$  boundary points  $(x_i, y_i)$ ,  $i = 1, 2, 3, \dots, n$ .

Centroid of A  $(\bar{x}, \bar{y})$  is calculated as,

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (10)$$

The direction of a major axis is determined based on the principle that sum of perpendicular distances of all boundary points is minimum, and given by,

$$\tan 2\theta = \frac{2 \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n [(x_i - \bar{x})^2 - (y_i - \bar{y})^2]} \quad (11)$$

where  $\theta$  is the angle between the x axis and a major axis of the object. Next, Major axis points are computed using Eq. (12) such that it returns  $v = 0$ , if the point lies on a major axis.

$$v = (y_i - \bar{y}) - \tan\theta(x_i - \bar{x}) \quad (12)$$

Similarly, minor axis points are computed using,

$$v = (y_i - \bar{y}) - \cot\theta(x_i - \bar{x}) \quad (13)$$

Out of these 6 attributes, 5 attributes i.e. (SD, EL, SC, SL, OR) are generic attributes, and selecting threshold criterion for each of them enables to extract various types of similar objects irrespective of the nature of object (e.g. building, cultivated lands, rivers, roads etc.) (Jiao and Liu, 2012). Only Gray mean is the attribute, which is object specific, and used to detect a specific type of object.

#### 3.2.2. Connected components extraction

Connected components are identified using 8 connectivity rules. Further, the above-mentioned six attributes are computed for each connected component. In order to obtain *elongation* and *solidity* attributes, the convex hull for each connected component is computed. Given a set of points P in the plane, a convex hull of the set P is the smallest convex polygon that contains all the points of P, as illustrated in Fig. 3(a). The Jarvis's March method (Jarvis, 1973) is used for finding the convex hull of each connected component. It does not require explicit sorting since all points are obtained in the anticlockwise direction only. Fig. 3(b) illustrates the working of Jarvis's March algorithm. It starts with the left most point l of the given set of points, it selects the next point p such that all other points lies at right of the line (l, p), this process continues till point l reaches again.

Finally, a threshold criterion considering different attributes is applied on each connected component to generate various profiles. The components, which satisfy a certain criteria are merged together to obtain the desired oil spill regions.

The GPU based implementations of the above-mentioned approaches enables the rapid detection of the affected areas. Those details are presented in the subsequent sections.

## 4. Nvidia GPU architecture

Nvidia's Kepler architecture has dramatically increased the compute capability of the GPU. Tesla K40 GPU provides 2880 cores (Nvidia, 2012). They are organized into 15 streaming multiprocessor (SMX) responsible for scheduling and dispatching thread blocks and six 64-bit memory controllers as shown in Fig. 4, where each multiprocessor consists of 192 cores (in green). Each SMX has registers and L1 cache. Each processor has a fully pipelined integer arithmetic logic unit (ALU) and

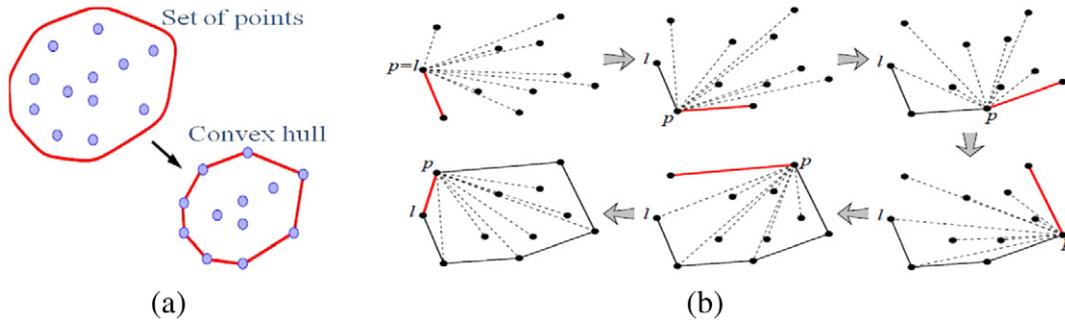


Fig. 3. (a) Convex hull from set of points (b) Execution steps of Jarvis's March convex hull algorithm on set of points.

floating point unit (FPU). Kepler's 15 SMX are positioned around a common L2 cache. Nvidia's Tesla K40 GPU with 2880 cores is used in this work.

Some of the features of the Kepler architecture over the earlier generation architectures such as Fermi, GT200 (for example, Tesla 2075 with 448 cores, Tesla C1060 GPU with 240 cores) are (Nvidia, 2012),

- Increased number of cores
- Dynamic parallelism
- Improved Double precision performance
- More global, shared memory
- Introduced minimum, maximum, logical AND, OR, XOR atomic operations

#### 4.1. Compute unified device architecture (CUDA)

CUDA is a programming model that enables general purpose programming on GPU. Basic building blocks of CUDA (Kandrot and Sanders, 2011) are kernel, block, and thread. Fig. 5 depicts the organization of these CUDA components.

Kernel is a user defined function, it runs N times in parallel by N different CUDA threads; threads are basic elements of parallel execution in CUDA. A Block consists of concurrently executing threads. Blocks are organized into one-dimensional, two-dimensional, or three-dimensional

grid of thread blocks. A grid is an array of thread blocks that execute the same kernel.

#### 4.2. Optimized data transfer using pinned memory on GPU

Christophe et al. (2011) performed NDVI calculations on a GPU and observed that most of the time the GPU capacity is wasted in data transfer as it involves multiple bands to transfer from CPU to GPU. Various solutions available for optimizing data transfer are:

- *Page locked or pinned memory:* `cudaMallocHost()` or `cudaHostAlloc()` function allocates memory on the CPU, it can be directly accessible from GPU, also data from this memory can be further copied to the device memory rapidly (Nvidia, 2015).
- *Zero copy memory:* As the name suggests, no data is copied on the GPU, instead the pointer is set and transferred to GPU to access the data directly from the host (Nvidia, 2015).
- *CUDA streaming by means of asynchronous transfer:* Different CUDA streams for data transfer are created for different tasks, and each stream works asynchronously, thus enabling small transfer overhead.
- Different small blocks of data can be grouped together and can be sent at once to the device memory; this also reduces the transfer time.



Kepler15 Streaming Multiprocessors are positioned around a common L2 cache

Kepler's Streaming Multiprocessor (SMX) with 192 cores

Fig. 4. Kepler architecture of GPU with 15 streaming multiprocessors (Enlarged Streaming Multiprocessor at right).

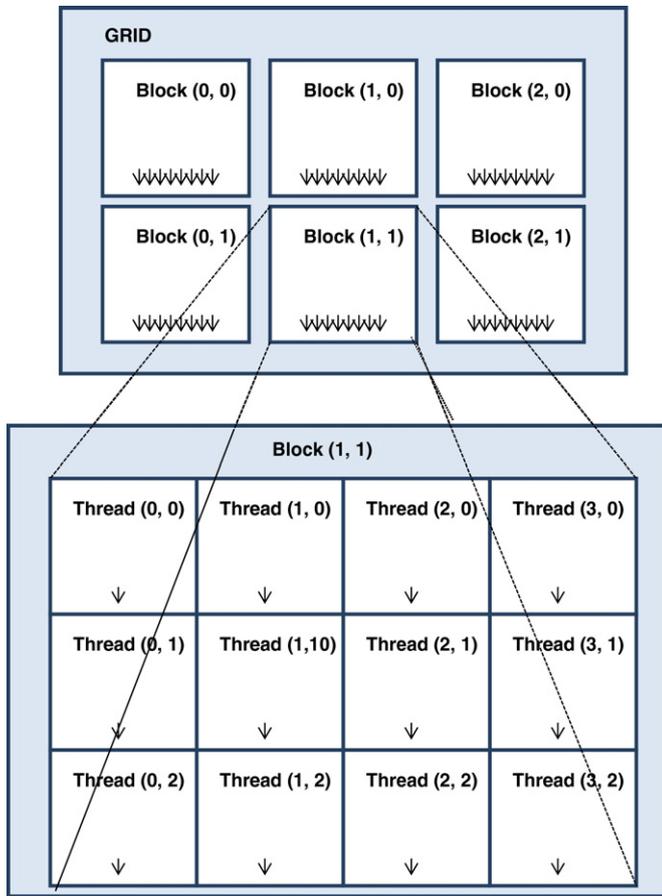


Fig. 5. CUDA building blocks: grid consist of multiple blocks, each block consist of multiple threads.

- **GPU Direct technology:** GPU direct technology uses special adapters for direct data transfer, which is generally used for GPU to GPU data transmission. Pinned/Page locked memory approach is used in this work to reduce the data transfer cost.

## 5. HPC implementation of oil spill detection approaches

Two state of the art HPC platforms (i.e. GPU and MPI) are used for the implementation of the oil spill detection approaches that are described in Section 3. MAP process is implemented using GPU; however one of the most time consuming task of MAP i.e. attribute computation is also implemented using MPI platform. Nvidia TESLA K40 based on Kepler architecture and Intel Xeon CPU (E5-2620) is used for processing the RS images for band ratio and morphological attribute profile operations. The technical specifications of K40 are given in Table 1.

CUDA with compute capability 3.5 is used for the software development. For MPI based implementation, a cluster of 4 nodes is used, where

Table 1  
Nvidia Tesla K40 GPU technical specification.

(Source: [http://www.nvidia.in/content/PDF/kepler/Tesla-K40-Active-Board-Spec-BD-06949-001\\_v03.pdf](http://www.nvidia.in/content/PDF/kepler/Tesla-K40-Active-Board-Spec-BD-06949-001_v03.pdf))

Number of CUDA cores	2880
Double precision processing performance	1.43 Tflops
Memory capacity	12 GB
Memory clock	3 GHz
Memory interface	384 bits
Memory bandwidth	288 GB/s
Max power consumption	235 W

each node has two Intel(R) Xeon(R) CPU E5-2640 processors, each processor has 8 cores, in total of 64 cores. CUDA aware OpenMPI v1.8.1 is used for the MPI implementation (Forum, 2015).

### 5.1. GPU implementation of oil spill index

GPU implementation of oil spill index involves transferring three different band images (each of size  $7191 \times 8331$ ) on to the device memory. However, such a transfer results in a huge overhead on the computational performance.

Christophe et al. (2011) have experienced such high data transfer cost while performing Normalized Difference Vegetation Index (NDVI) calculations as compared to the actual computation cost, when multiple bands are involved in the computation. Although, there is a huge data transfer cost involved in band ratio type of algorithms, the algorithms of this kind can take full advantage of the inherent parallelism on a GPU. Since this work focuses also on optimizing data transfer for effective utilization of GPU computing power, the approach of page locked/pinned memory is used for transferring three different bands on to the device memory. As a result, the transmission time reduces significantly using the pinned memory.

Pinned memory is allocated using the following statements in CUDA,  
`cudaMallocHost((void**)&d_band1, size*sizeof(double));`  
 or  
`cudaHostAlloc((void**)&d_band1, size*sizeof(double), cudaHostAllocPortable);`

Intentionally, the data is considered as of type of double to observe the data transfer performance shown in Table 2. For faster access the pinned data is further copied to the device memory. To locate each pixel in the image, sequential implementation requires two *for* loops that corresponds to the number of rows and columns of the image. Appendix I (A) shows a code snippet for oil spill index calculation on the GPU.

Whereas, in the GPU implementation, each pixel is located using thread identifiers, hence, there is no need of *for* loops, and reduces the computational complexity of the function from  $O(n^2)$  to  $O(c)$  where  $c$  is some constant time taken for multiple blocks as shown in Appendix I (A).

### 5.2. GPU implementation of morphological attribute profile

Six different attributes are considered for implementation on Landsat ETM + Panchromatic data. In order to compute these attributes, following variables are calculated for each connected component, and attributes are derived using the following listed variables,

(a) *count\_c (Area)*: Indicates total number of pixels in each component. It is inherently sequential operation (i.e. new value of a variable depends upon an earlier calculated value.)

(b) *sum\_c*: Indicates sum of the intensities of pixels of each connected component. It is also an inherent sequential operation.

(c) *Perimeter\_c*: Indicates number of points forming the border of the connected component.

*Count\_Convex (Area)*: Indicates total number of pixels in each convex hull. It is inherently a sequential operation.

Table 2  
Geometric errors of MAP results (oil-spill dataset).

MAP results using multilevel attribute filtering (GM + SD + EL + SL + SC + OR)	OS %	US %	ED %	SH %
Pan-Image 1	11.12	25.66	12.10	14.90
Pan-Image 2	24.04	28.12	38.40	27.34
Pan-Image 3	35.21	14.18	27.466	8.80

(d) *Convex\_major\_axis*: Indicates major axis length of convex hull. It involves computation of sum, averages ( $sum_x, sum_y, avg_x, avg_y$ ) of all x co-ordinates and y-coordinates of the convex hull perimeter and the angle ( $\theta$ ) with respect to x-axis.

(e) *Convex\_minor\_axis*: Indicates minor axis length of convex hull.

GM is derived from *count\_c* and *sum\_c*. Further, SD attributes are derived using GM. Slick complexity is computed using *perimeter\_c* and *count\_c*. *Elongation* is computed using *Convex\_major\_axis* and *Convex\_minor\_axis*. Computation of *Convex\_major\_axis* involves identifying  $\theta$ , which is actually the *orientation* of the convex hull. Solidity is derived from *count\_c* and *count\_convex*.

In parallel flow on a GPU, the calculation of sum of pixel intensities (*sum\_c*) and count of the number of pixels (*count\_c, count\_convex*) that belong to each component and a convex hull, are inherently sequential operations and cannot be fully parallelized, it is performed using *atomicAdd()* function on CUDA platform which executes the sequential sum of intensities and pixel count number in each of the connected component/convex hull. *CUDA reduction* (Harris, 2007) is another approach that can be used in place of *atomicAdd()*. Merging of all the connected components that satisfy the specific threshold criteria is performed in parallel.

### 5.2.1. GPU kernels for attribute profile computation

Attribute profiles are implemented using five different kernels, for identification, labeling, sum and count of connected components, calculation of squared difference, computation of convex hull, calculation of major and minor axis, and merging of the components based on threshold criteria. Implementing GM on GPU involves sequential sum and count of pixel values for each connected component. This is achieved using CUDA statements.

```
atomicAdd(&sum_c[i], *(band1 + x*cols + y));
```

```
atomicAdd(&count_c[i], 1);
```

Array *sum\_c [i]* stores the sum of all the pixels belonging to the connected component *i*, and *count\_c[i]* stores number of pixels in *ith* connected component; here each thread adds its data synchronously. The mean value for each connected component is stored in a constant memory where GPU has fastest access and is used further to calculate SD. Shape description attributes that are based on convex hull are implemented in CUDA based on Jarvis's convex hull algorithm is given below. The steps executed in parallel are also mentioned.

Let *l* be the left most point in *P*.

```
C_hull(P) ← {l}
```

```
p=l
```

```
repeat
```

```
for all i ∈ P do
```

```
Search for point q such that orientation (p, i, q)
```

```
is anticlockwise for any other point i and p ≠ q
```

```
if orientation (p-i-q) < 0 then q ← i
```

```
end if
```

```
p=q
```

```
end for
```

```
Append (q) to C_hull(P).
```

```
Until p = l
```

Parallel Implementation

For CPU, the computational complexity is  $O(P \cdot h)$  and for GPU it is  $O(P)$  where *P* is the total number of points in the data set for which convex hull is to be found, and *h* is the total number of iterations required to find points of the convex hull; code snippet for the same given in Appendix I (B). For each component, to identify the points of the first

half of the convex hull, the kernel is invoked by passing flag *direct* = 1 and the points of the second half of the convex hull are identified by passing flag *direct* = 0 to the same kernel.

All the points of connected components from global memory are stored in the shared memory so that each block should compute the rightmost point from the data stored in its shared memory, thread 1 of each block is responsible to compute the rightmost point and store it at thread 0, hence, thread 0 of each block holds the rightmost point of the respective block, block level thread synchronization is achieved using *\_\_syncthreads()*; At the end, using *atomicMax()* function, the rightmost point is identified from the points held by thread 0 of the respective blocks. Here, synchronization among multiple blocks is achieved using *\_\_threadfence()*. After obtaining convex hull points of connected components, major/minor axis is computed. It involves *avg\_x, avg\_y*, that requires sequential operations, *sum\_x* and *sum\_y*, which are computed using CUDA reduction mechanism.

### 5.3. MPI implementation for attribute computation

Six attributes from each connected component are computed in parallel using MPI technology. Root process is responsible for distributing the input data and uniform number of components at each core from each node. All cores work in parallel, and computes the attributes from the connected components assigned to it. After completing the calculation for all the components, each core sends back the computed values to the root process. It collects the attributes after finishing with all components, and stores it at an appropriate location of the feature matrix, which holds all features of all connected components together. Implementation is done by calling MPI routines in C code. MPI calls are invoked to initiate the MPI process after obtaining connected components from the image. The first process, i.e. rank 0 calculates the number of components that can be distributed to each core, such that all cores should get assigned equal amount of work, only the last process gets remaining components, may be lesser than number of components assigned to all other processes. The labeled components matrix and original image matrix (required for GM, SD computation) are broadcasted to all processes, the following statement demonstrates the broadcast of the labeled components matrix,

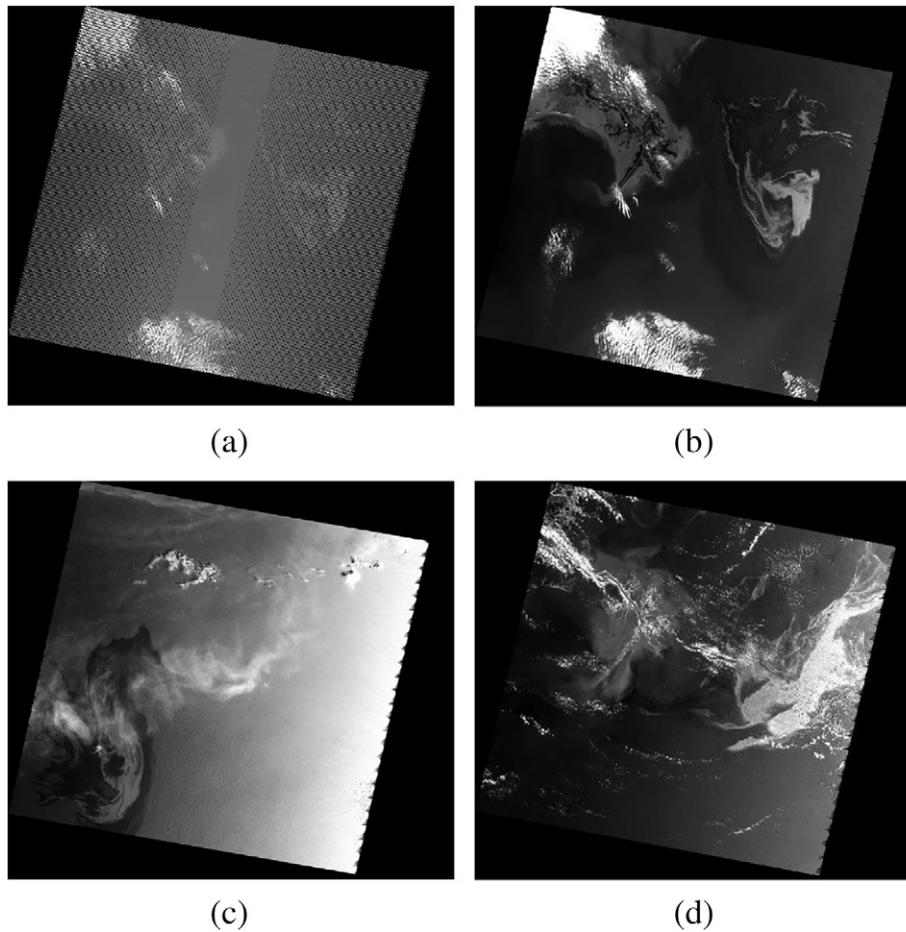
```
ierr = MPI_Bcast(band_1_conn, no_of_rows*no_of_cols, MPI_INT, 0, MPI_COMM_WORLD);
```

where *band\_1\_conn* is the labeled components matrix, *no\_of\_rows\*no\_of\_cols* is the size of matrix, *MPI\_INT* represents the data type of the matrix, 0 represents the process or rank broadcasting the data, *MPI\_COMM\_WORLD* is a MPI communicator. *MPI\_Bcast* is observed as more time efficient as compared to *MPI\_Send* when same data is to be transferred to all cores, also other processes need not have to explicitly collect the data using *MPI\_Recv* as required by *MPI\_Send*, data relevant to specific core is transferred using *MPI\_Send*. Appendix I (C) shows the code snippet where rank 0 distributing the components across all cores. Root process computes *start\_comp, end\_comp* i.e., start and end component identifiers for all other processes so that all processes should get uniform number of components; it then sends that data to all other processes. Each process then extracts the features only for the specific components assigned by the root process.

## 6. Experimental results

### 6.1. Data sets

On 20th April 2010, Gulf of Mexico experienced world's worst accidental oil spill. This event released net liquid oil of 4.1 million barrels in sea water at Gulf of Mexico. Petroleum hydrocarbons spewed from leaking well 1.5 km below sea surface (Reddy et al., 2012). Oil spill



**Fig. 6.** (a) LANDSAT-7 ETM+ original image with striping due to failure in scan line corrector. LANDSAT-7 ETM+ panchromatic (gap filled) images of oil spill at Gulf of Mexico captured on (b) 1st May 2010 (16801 × 14441) (Pan-Image1) (c) 10th May 2010 (16681 × 14221) (Pan-Image2) (d) 17th May 2010 (16801 × 14381) (Pan-Image3).

affected area about 960 km of beaches and wetlands spreading across five states of United States. Lasting for 87 days, this event had adverse impact on environment and economy. Marine species, migratory shore-birds have had a long term incalculable impact on their survival rate, breeding process.

Reddy et al. (2012) performed impact (chemical) analysis of Deepwater Horizon oil spill, regarding oil flow rate, total oil volume released and trajectories and fates of hydrocarbon components in marine environment. They observed that gas and oil experienced a significant residence time in water column causing volatile species not to vaporize in atmosphere, hence leading to much water soluble compound dissolved in to water column as compared to surface oil spill.

In response to Deepwater horizon oil spill, National Oceanic and Atmospheric Administration (NOAA) (NOAA, 2015) with other natural resources trustees released a 15 year comprehensive, integrated environmental ecosystem restoration plan for the Gulf of Mexico which will cost around \$8.8 billion.

Oil spill images of Gulf of Mexico captured in May 2010 by LANDSAT-7 ETM+ are obtained from USGS Hazards Data Distribution System (HDDS) (USGS, 2014). LANDSAT-7 ETM+ Panchromatic images with 15 m spatial resolution from the path 21/ row 40 captured on 1st (denoted as Pan-Image1), 10th (Pan-Image 2) and 17th May 2010 (denoted as Pan-Image 3) are used for the spatial analysis. Bands1, 2, 3 of LANDSAT-7 ETM+ are used for spectral analysis (band-ratio). Original image consists of striping due to failure in scan line corrector, (one of the image is shown in Fig. 6(a)), these images are subjected to pre-processing for gap filling (USGS, 2013), the resulted images are shown in Fig. 6(b) (c) and (d).

## 6.2. Results

### (i) Oil Spill Index

LANDSAT-7 ETM+ Bands1, 2, 3 images (7191 × 8331) are used for oil spill detection using band ratio, as it has spectral ranges required for oil spill index calculation.

Fig. 7 shows the oil spill detected by thresholding the image at positive index values, since the cloud and oil have similar spectral characteristics for the used range, the resulting image shows presence of cloud along with the oil spill.



**Fig. 7.** Oil spill detected using band ratio approach from LANDSAT ETM+ data (with positive index).

(ii) Morphological Attribute Profiles

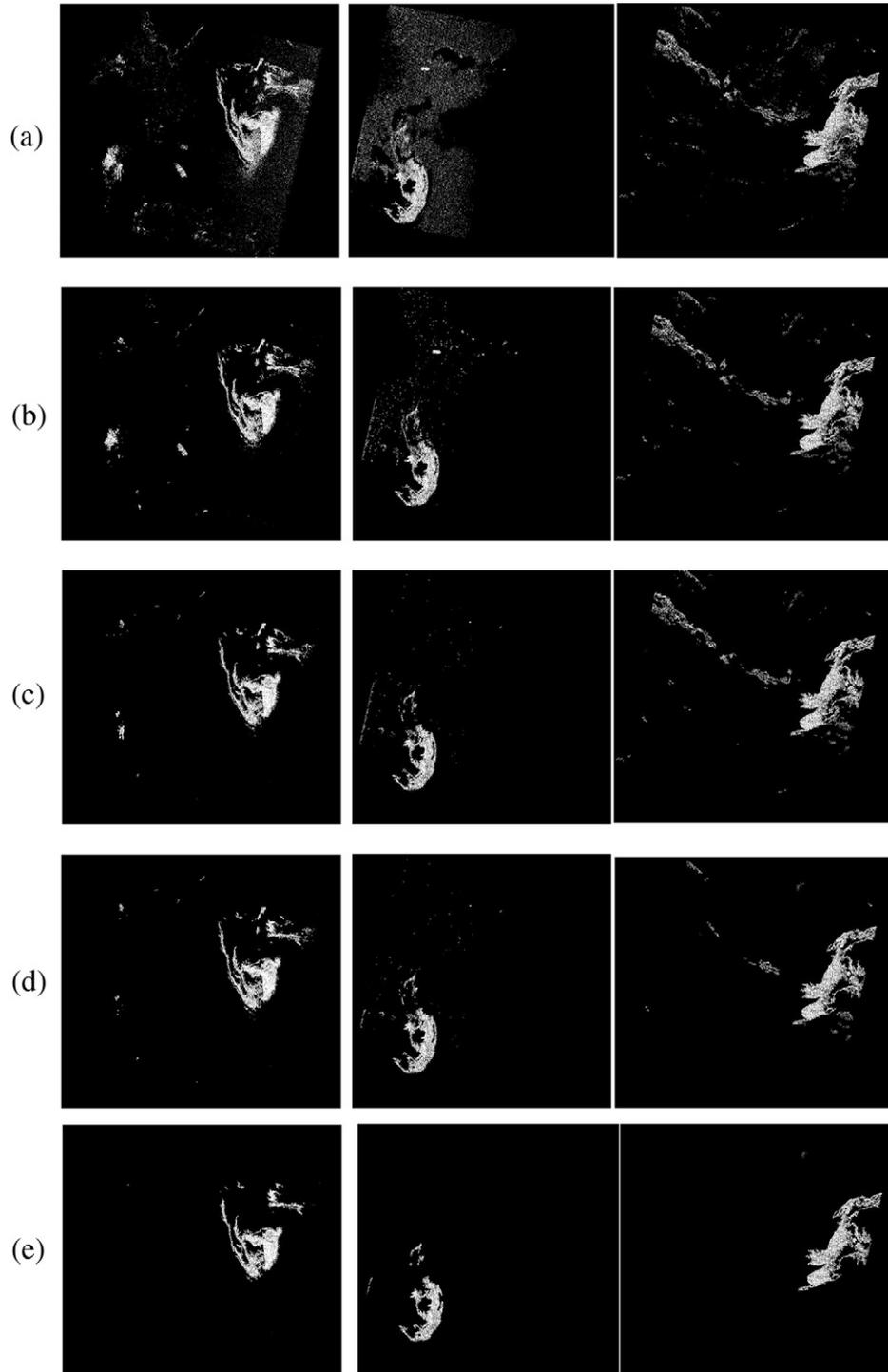
Three different LANDSAT ETM+ panchromatic images containing oil spill regions of size,  $16801 \times 14411$  captured on 1st May 2010,  $16681 \times 14,221$  (Pan-Image 1), captured on 10th May 2010 (Pan-Image 2), and  $16,801 \times 14,381$  captured on 17th May 2010 (Pan-Image 3) are used for experimentation. Profiles with 6 different attributes (GM, SD, EL, SL, SC, OR) are calculated for each connected component of each image using MAP. Attribute filtering is applied in hierarchy to eliminate the undesired or lookalike components from the data. Oil

spill signatures are obtained at below given thresholds of each attribute for each image,

Pan-Image 1:  $\{50 < X_{GM} < 100, 1.8 < X_{SD} < 6.0, X_{EL} > 1.5, X_{SL} < 0.5, X_{SC} > 2, -90 < X_{OR} < -10\}$

Pan-Image 2:  $\{50 < X_{GM} < 120, 1.8 < X_{SD} < 6.0, X_{EL} > 1.5, X_{SL} < 0.5, X_{SC} > 2, 50 < X_{OR} < 80\}$

Pan-Image 3:  $\{50 < X_{GM} < 130, 1.8 < X_{SD} < 6.0, X_{EL} > 1.5, X_{SL} < 0.5, X_{SC} > 2, 20 < X_{OR} < 60\}$  where X represents all the components, which satisfies the given threshold criteria with respect to a subscripted attribute.



**Fig. 8.** MAP of oil spill using multilevel attribute filtering on multi-temporal Panchromatic Images (each column represents: Pan-Image1, Pan-Image2, Pan-Image3) (a) GM filtering (b) GM + SD + EL (c) GM + SD + EL + SL (d) GM + SD + EL + SL + SC (e) GM + SD + EL + SL + SC + OR with the following threshold, Pan-Image 1:  $\{50 < X_{GM} < 100, 1.8 < X_{SD} < 6.0, X_{EL} > 1.5, X_{SL} < 0.5, X_{SC} > 2, -90 < X_{OR} < -10\}$  Pan-Image 2:  $\{50 < X_{GM} < 120, 1.8 < X_{SD} < 6.0, X_{EL} > 1.5, X_{SL} < 0.5, X_{SC} > 2, 50 < X_{OR} < 80\}$  Pan-Image 3:  $\{50 < X_{GM} < 130, 1.8 < X_{SD} < 6.0, X_{EL} > 1.5, X_{SL} < 0.5, X_{SC} > 2, 20 < X_{OR} < 60\}$ .

Filtering is applied in the same sequence as mentioned above. In Solberg et al. (2007), 37 different rules are defined for improving the accuracy of classification, these rules resemble attribute filtering criteria of MAP technique. Other than *GM* and *OR* attribute, all attributes are quite generic and have same threshold levels across all the images; whereas the changes in *GM* and *OR* attributes represent the temporal changes on oil spill components; change in *GM* indicates increase in intensity along with time and change in *OR* indicates the influence of wind on oil spill regions along with time. Agreement on threshold values of some of the attributes will actually be useful to derive more generic profiles of distinct categories of objects such as roads, rivers, buildings, ponds etc., whereas inclusion of intensity and direction related attributes gives more specific object profiles.

Fig. 8 represents different profiles obtained after applying various attribute filtering on oil spill images captured at different dates. Fig. 8(a) is a profile generated using only *GM* attribute with the given threshold criteria for each image. Fig. 8(b) till (e) represents the multilevel attribute filtering that eliminates not only distinct non oil-spill components, but also eliminates lookalikes from each of the image. *GM* eliminates all the objects, which are distinct in nature in terms of their brightness. Result of *GM* includes oil spill regions along with its lookalike; further filtering attempts to eliminate the look alike as well. Oil spill regions are observed to have *SD* above 1.8, after imposing this filter, many small (in terms of area) connected components gets eliminated automatically, as they usually have very less *SD* (below 1). *EL* attribute filtering further restricts the components, which are not elongated; Fig. 8(b) shows the components after *SD* and *EL* filtering. All the images under experimentation contain cumulus clouds, which usually have high solidity as compared to oil spill regions; Fig. 8(c) shows the components after *SL* filtering. Oil spill signatures are observed to be of complex shape;

hence *SC* filtering is applied to eliminate the lookalike components, which are simpler in shape, results of *SC* are presented in Fig. 8(d).

Since oil spill regions are greatly influenced by wind and current flow, its direction is one of the characteristics that will identify the oil spill regions uniquely. Fig. 8(e) shows oil spill regions based on its orientation property. After applying multilevel filtering at different thresholds, it is observed that, most of the oil spill components pass through all filtering criteria.

Fig. 9(a) highlights prominent cloud components, which are eliminated due to more solidity after applying *SL* threshold criteria. At the final level of filtering, almost all lookalikes are eliminated due to the orientation criterion; Fig. 9(b) shows such components from Pan-image 2.

Some of the oil-spill regions highlighted in Fig. 9(c) are also eliminated due to unfulfilled orientation constraint. However, eliminating many other non oil-spill components from the same scene increases the recall.

### 6.3. Result validation

MAP technique used in this work for oil spill detection is mainly based on geometric characteristics of the data. Thematic accuracy does not consider geometrical quality of the region on the scene; hence, the correctness of the geometry of the regions obtained after classification is not verified. For low or medium resolution images, it is difficult to describe the geometry of the region, but for high resolution images (e.g. Pan-Image1, Pan-Image2, Pan-Image3 used in this work) it is beneficial to validate geometric accuracy along with thematic accuracy. Hence, correctness of the methodology is measured using both, thematic and geometric accuracy measures.

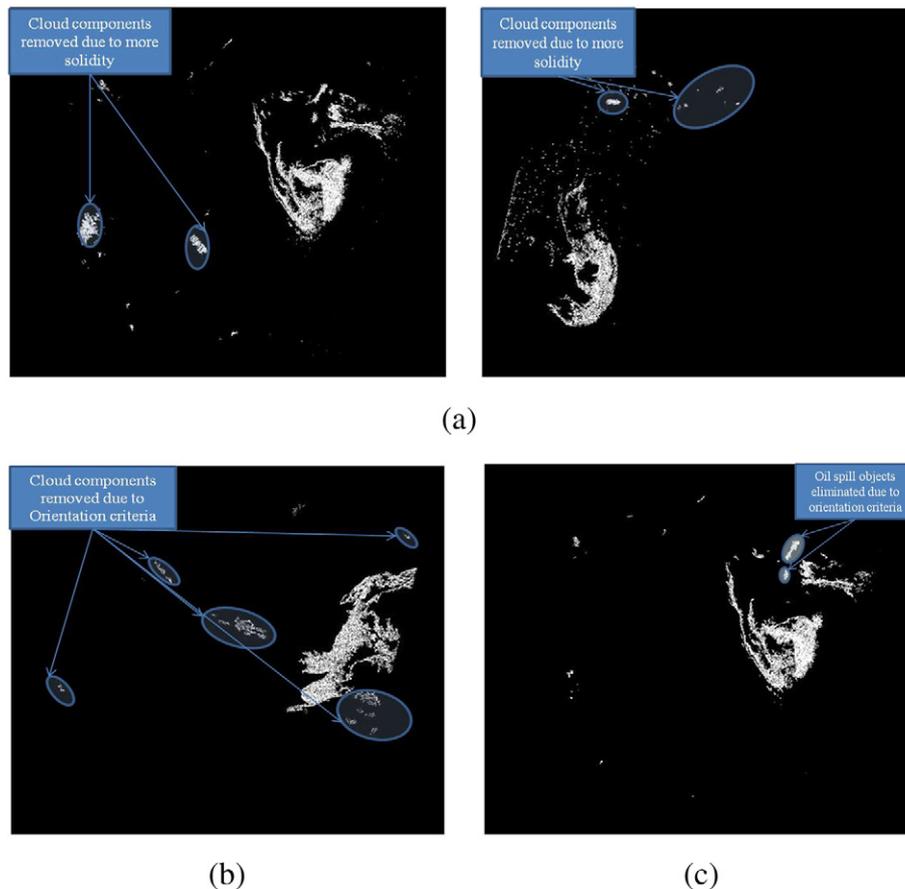


Fig. 9. MAP of oil spill using multilevel attribute filtering (a) highlighted cloud objects eliminated due to more solidity (b) highlighted cloud objects eliminated due to orientation criteria (c) highlighted Oil spill objects eliminated due to orientation criteria.

6.3.1. Geometric accuracy

Persello and Bruzzone (2010), have proposed a novel protocol for accuracy assessment as the thematic accuracy is not sufficient for geometrical properties of high resolution classification. They have proposed geometric error indices which computes the correctness while extracting the geometry, shape of the objects (buildings, roads, oil spills, etc.) considered for investigation. Computation of these error indices includes a set of reference objects,  $R = \{R1, R2, \dots, Rn\}$  of specific class of interest that defines the exact shape, and geometry. Another set of objects  $M = \{M1, M2, M3, \dots, Mm\}$  is obtained through experimental result using various other algorithms. For a pair of  $(Ri, Mi)$  where  $Mi$  corresponds to  $Ri$ , following error indices can be calculated:

- Over segmentation (OS)

It calculates a ratio of overlapping area of two objects and the area of reference object.

$$OS(Ri, Mi) = 1 - \frac{|Ri \cap Mi|}{Ri} \tag{14}$$

- Under segmentation (US)

It calculates a ratio of overlapping area of two objects and the area of object obtained through experiment, given in Eq. (15).

$$US(Ri, Mi) = 1 - \frac{|Ri \cap Mi|}{Mi} \tag{15}$$

- Edge location error (ED)

In edge location error, given in Eq. (16),  $e(Ri)$  denotes the pixels that form a border of reference objects, and  $e(Mi)$  denotes the pixels that form a border of classified object.

$$ED(Ri, Mi) = 1 - \frac{|e(Ri) \cap e(Mi)|}{|e(Ri)|} \tag{16}$$

- Shape error (SH)

Shape error is calculated as absolute value of the differences of the specific shape factor such as roundness, compactness etc.

$$SHi = |sf(Ri) - sf(Mi)| \tag{17}$$

OS, US, ED index will vary in the range of 0 to 1, whereas SH index range depends upon the range of values of the shape factor.

Mura D. et al. (2010) have used this validation approach for validating the results of MAP on VHR images for classification of road, buildings, vegetation etc.

In this work, four different geometric error indices are used for evaluation, namely, over segmentation (OS), under segmentation (US), Edge location (ED), and shape error (SH) (Persello and Bruzzone, 2010). Oil spill reference objects are obtained by performing segmentation using Sobel operation on the image data. Further, manual verification of segments (based on published data) is done to identify the exact oil spill segments. Fig. 10(a) presents the oil spill reference objects and Fig. 10(b) presents the oil spill objects identified using MAP multilevel attribute filtering.

All oil spill objects obtained as a result of MAP multilevel attribute filtering criteria are evaluated against corresponding reference objects by computing OS, US, ED, SH geometric error indices as described above. Table 2 shows the validation results obtained by OS, US, ED, SH

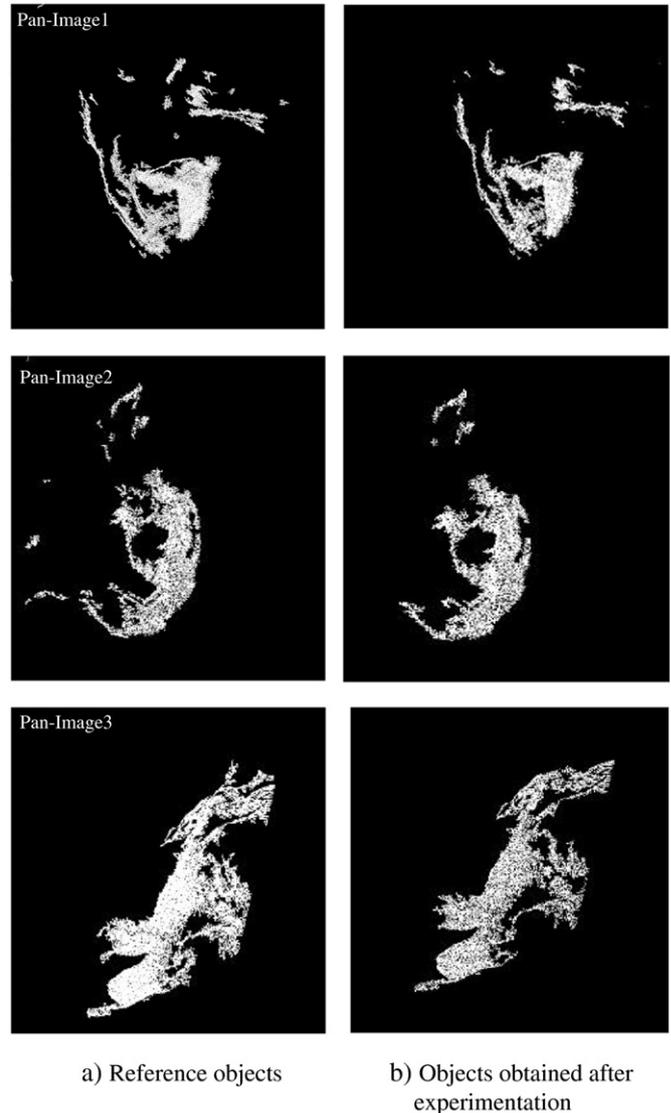


Fig. 10. (a) Reference objects obtained using segmentation technique (b) corresponding Oil spill objects obtained after applying MAP multilevel attribute filtering.

measures; figures in this table are average percentile errors calculated considering all resulted objects of MAP multilevel attribute filtering.

6.3.2. Thematic accuracy

Thematic accuracy is measured by subjecting the component's attribute data as features to SVM classification, LIBSVM (Chang and Lin, 2011) classifier is used. 195 features are considered for classification with almost equal number of features for oil spill and non-oil spill classes. RBF kernel is used for training with default gamma i.e. (1/number of features), and cost with 10 fold cross validation. SVM provides 97.95% accuracy for oil spill sample dataset.

Table 3 Optimized data transfer (pinned memory vs. unpinned memory).

Image size (double data type)	Unpinned memory transfer time (ms)	Pinned memory transfer time (ms)
2048 × 2048	18.61	3.18
4096 × 4096	72.77	12.71
8192 × 8192	281.96	50.87
12,288 × 12,288	532.47	114.47
18,432 × 18,432	1000	247.44

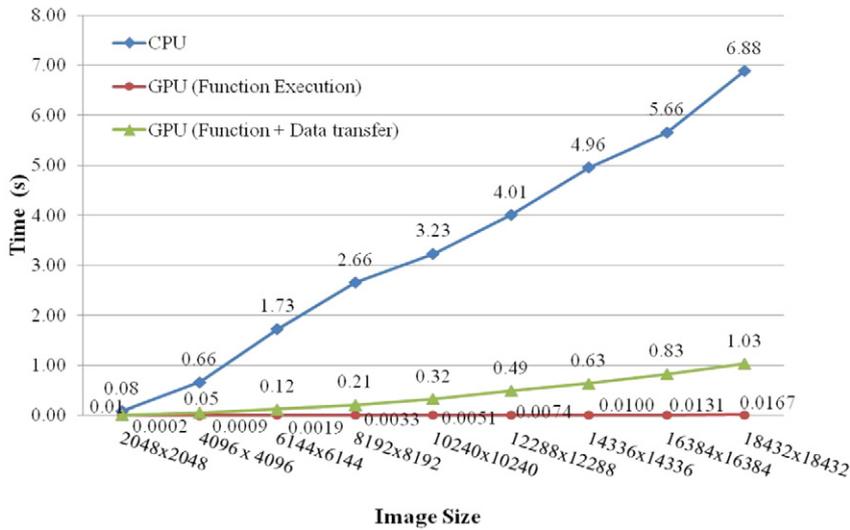


Fig. 11. Performance evaluation of oil spill index calculation on GPU and CPU.

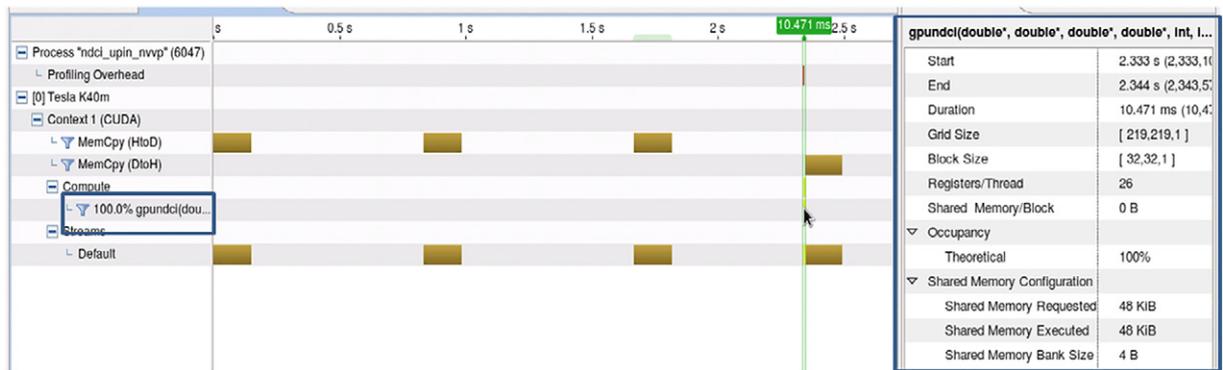


Fig. 12. Visual profiler data for kernel gpundci() (Right side block), highlighted kernel in time line (details of gpundci() can be seen in right side block).

6.4. GPU performance

Transfer cost of pinned memory to device is very less, approximately 5 times less than unpinned memory. Table 3 shows the performance for both pinned and unpinned memory data transfer on GPU.

For oil spill index, Fig. 11 shows the performance improvement of this computation on GPU as compared to CPU.

GPU implementation of oil spill index is around 14 times faster than CPU implementation with data transferring (i.e. 3 full scene images corresponding to 3 bands) and around 800 times faster over CPU if only the kernel execution is considered. It is an embarrassingly parallel task. Fig. 12 shows the time-line results obtained using Nvidia visual profiler for oil spill index (gpundci()) kernels.

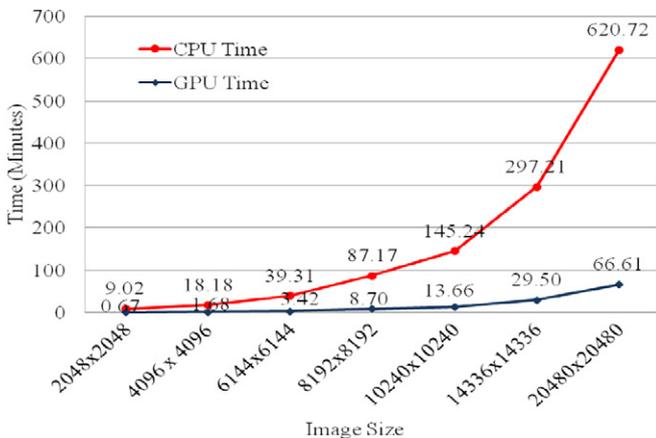


Fig. 13. Performance evaluation for MAP process on CPU and GPU.

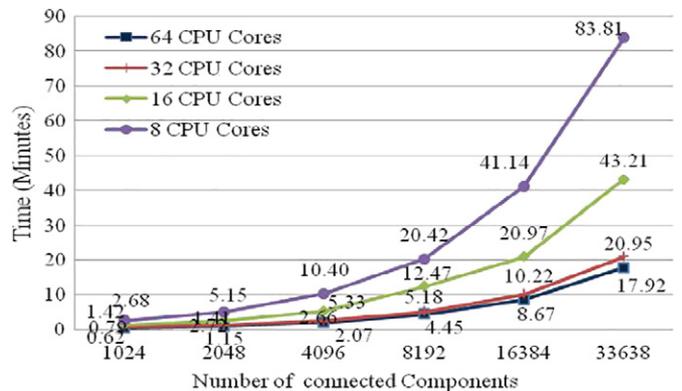


Fig. 14. MPI Performance by varying number of cores for attribute computation for different number of components.

In attribute profiles, speedup is also observed for the kernels calculating sequential sum, count and squared difference required for all the attributes on GPU. Fig. 13 shows the performance evaluation for entire MAP process ON GPU over CPU. In spite of involving lots of sequential tasks, for the entire MAP process, considering all attributes, GPU speed up is around 10× over CPU.

### 6.5. MPI performance

Attributes computation for each component is also implemented using the OpenMPI platform. Execution time is measured by varying the number of cores as 8, 16, 32, and 64. Fig. 14 presents the performance of features extraction process by varying number of cores against number of connected components. Execution using all 64 cores performs 5× faster as compared to execution on 8 cores.

Only attribute computation process on GPU takes almost an hour whereas MPI implementation on 64 cores takes only 18 min.

As compared to 64 cores of CPU, the GPU performance is lower because of lots of inherently sequential operations involved in the process such as computation of variance, convex hull based attributes, in this case, at block level, the computation of rightmost point is done sequentially only. Further, synchronization between all threads inside a block and across a block introduces a delay. Though these tasks are performed in parallel using atomic or reduction operations and can gain speed over a single core execution, but multiple cores (64 in this case) intended for sequential processing outperforms GPU in such a kind of operations.

## 7. Conclusions

Oil spill signatures are obtained using spectral and spatial techniques on LANDSAT ETM+ data. Spatial analysis approach, called MAP is applied on oil spill data using *GM, SD, EL, SL, SC, OR* attributes. For each attribute threshold criteria is identified after manual observation. Multilevel filtering is applied with these attribute threshold criteria to obtain the oil spill signature. Two different validation approaches, geometric validation and thematic validation are used to measure the accuracy of MAP results. Geometric accuracy is calculated using *OS, US, ED, SH* error measures which give 23.46, 22.65, 25.98, and 17.01% average errors respectively for all images. Thematic accuracy is measured by subjecting all attributes of 195 components as features for SVM classification with almost equal number of features for oil spill and *non-oil* spill classes; with 10 fold cross validation, 97.95% accuracy is observed.

Different oil spill detection techniques are implemented on CPU and GPU. It is observed that GPU execution time is considerably lower than CPU execution time for these techniques and is inversely proportional to the data size. Amount of speedup achieved is around 14 X for band-ratio approach and 10× for entire MAP process over CPU. Data transfer is optimized using pinned memory as it is observed that optimized data transfer benefits percentage utilization of GPU. Pinned memory transfer is around 5× faster than unpinned memory. Most time intensive task of MAP i.e. attribute computation is also implemented on MPI using 64 cores and observed significant speedup.

Further optimization can also be done using concurrent execution of CUDA kernels using CUDA streaming; further scope of this work includes implementation of the MAP using hybrid parallelism with CUDA aware MPI that uses heterogenous platform i.e. GPU and MPI to exploit multilevel parallelism.

## Appendix I (A)

Oil spill index calculation snippet in CUDA C.

```

__global__ void gpundci(double *band1, double *band2, double *band3, double *band_ratio, int
rows, int cols)
{
    double b1,b2,b3;
    int x= blockIdx.y*blockDim.y + threadIdx.y;
    int y= blockIdx.x*blockDim.x + threadIdx.x;
    if(x< rows && y < cols)
    {
        b1= *(band1+x*cols+y);
        b2=*(band2+x*cols+y);
        b3=*(band3+x*cols+y);
        *(band_ratio + x*cols+y) =((b3/b2)/b1);
    }
}

```

**Appendix I (B)**

CUDA Code snippet for convex hull implementation.

```

:
__shared__ int shared_x[1024],shared_y[1024];
temp_q=*q; final_q=0;
int gid = (blockDim.x * blockDim.x) + tid; // convertig to one dimentional
shared_x[0]=0;
// move data from global memory to shared memory of each block
while (gid < (*n) && tid!=0)
{
shared_x[tid] = *(co1+(gid*4)+0); //fetched x coordinates
shared_y[tid] = *(co1+(gid*4)+1); //fetched y coordinates
gid += blockDim.x*blockDim.x;
}
__syncthreads();
gid = (blockDim.x * blockDim.x) + tid;
if(tid==1) // only thread 1 of each block will do this task , no other thread need to do it
{ for (unsigned int i=1; (i<1024 && (blockDim.x * blockDim.x)+i < *n) ; i++)
{ if (((blockDim.x * blockDim.x) + i) != *p) && (((blockDim.x * blockDim.x) + i)!=temp_q) &&
*(co_flag+(blockDim.x * blockDim.x)+i)!=1)
{
x1= *(co1+(*p)*4+0); y1=*(co1+(*p)*4+1);
x2= *(co1+(*q)*4+0); y2 =*(co1+(*q)*4+1);
x3=shared_x[i]; y3 = shared_y[i];
if(*direct==1)
val1=(y2- y1) * (x3-x2)- (y3- y2) * (x2- x1);
else
val1=(y3-y1)*(x2-x3)-(x3-x1)*(y2-y3);
if ((val1 < 0) && (val1 != 0))
*(q) = (blockDim.x * blockDim.x)+i;
shared_x[0]=*q; // store that row number value to 0th thread , will overwrite
__syncthreads();
}
} __threadfence(); } // tid =1 task finishes here
if(tid==0) { atomicMax(&final_q,shared_x[0]);
__threadfence(); }

```

## Appendix I (C)

C MPI code snippet for feature extraction process from each connected components.

```

if(rank == 0)
{
    avg_comps_per_process = (comp / (numtasks-1)) +1; // calculates no .of components to transfer to
    each processor
    /* distribute equal no of components to each processor */
    for(id = 1; id < numtasks ; id++)
    {
        start_comp = (id-1)*avg_comps_per_process + 1;
        end_comp = start_comp +avg_comps_per_process-1;
        if((comp - start_comp) < avg_comps_per_process) //for
        remaining last components
        end_comp = comp - 1;
        num_comps_to_send = end_comp - start_comp + 1;
        ierr = MPI_Send( &num_comps_to_send, 1 , MPI_INT, id, send_data_tag,
MPI_COMM_WORLD);
        ierr = MPI_Send(&start_comp ,1 , MPI_INT, id, send_data_tag, MPI_COMM_WORLD);
    }
else //for all other process except root
{ ierr = MPI_Recv( &num_comps_to_send, 1 , MPI_INT, 0, send_data_tag, MPI_COMM_WORLD,
&status);
    ierr = MPI_Recv( &start_comp, 1 , MPI_INT, 0, send_data_tag, MPI_COMM_WORLD, &status);
    :
}
for (k=start_comp;k<=start_comp+num_comps_to_send-1;k++)
{ // all the processors extracts the features only from components assigned to it.
}

```

## References

- Alawadi, F.A.M., 2012. Identifying the impact of oil spills. Environmental Remote Sensing and Systems Analysis. CRC Press.
- Bernabe, S., Sanchez, S., Plaza, A., Lopez, S., Benediktsson, J.A., Sarmiento, R., 2013a. Hyperspectral unmixing on GPUs and multi-core processors: A comparison. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 6 (3), 1386–1398.
- Bernabe, S., Lopez, S., Plaza, A., Sarmiento, R., 2013b. GPU implementation of an automatic target detection and classification algorithm for hyperspectral image analysis. IEEE Geosci. Remote Sens. Lett. 10 (2), 221–225.
- Brekke, C., Solberg, A.H.S., 2005. Oil spill detection by satellite remote sensing. Remote Sens. Environ. 95, 1–13.
- Breen, E.J., Jones, R., 1996. Attribute openings, thinnings, and granulometries. Comput. Vis. Image Underst. 64 (3), 377–389.
- Casciello, D., Lacava, T., Pergola, N., Tramutoli, V., 2007. Robust Satellite Techniques (RST) for oil spill detection and monitoring. Proc. MultiTemp 2007–2007 Int. Work. Anal. Multi-Temporal Remote Sens. Images, pp. 1–6.
- Casciello, D., Lacava, T., Pergola, N., Tramutoli, V., 2011. Robust satellite techniques for oil spill detection and monitoring using AVHRR thermal infrared bands. Int. J. Remote Sens. 32 (14), 4107–4129.
- Cai, G., Wu, J., Xue, Y., Wan, W., Huang, X., 2007. Oil spill detection from thermal anomaly using ASTER data in Yinggehai of Hainan, China. Int. Geosci. Remote Sens. Symp. 1, 898–900.
- Chang, C.-C., Lin, C.-J., 2011. Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3), 1–27.
- Christophe, E., Michel, J., Inglada, J., 2011. Remote sensing processing: from multicore to GPU. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 4 (3), 643–652.
- Chaudhuri, D., Samal, A., 2007. A simple method for fitting of bounding rectangle to closed regions. Pattern Recogn. 40 (7), 1981–1989.
- Del Frate, F., Petrocchi, A., Lichtenegger, J., Calabresi, G., 2000. Neural networks for oil spill detection using ERS-SAR data. IEEE Trans. Geosci. Remote Sens. 38 (5), 2282–2287.
- Disasters Charter, 2015. [Online]. Available: <https://www.disasterscharter.org/web/guest/-/landslide-in-nep-2> (Accessed: 01-Jul-2015).
- Durbha, S.S., King, R.L., Younan, N.H., 2010. Wrapper-based feature subset selection for rapid image information mining. IEEE Geosci. Remote Sens. Lett. 7 (1), 43–47.

- Forum, M.P.I., 2015. MPI: A Message-Passing Interface Standard. University of Tennessee Knoxville, TN, USA, Technical Report.
- Fingas, M., Brown, C., 2014. Review of oil spill remote sensing. *Mar. Pollut. Bull.* 83 (1), 9–23.
- Grimaldi, C.S.L., Casciello, D., Coviello, I., Lacava, T., Pergola, N., Tramutoli, V., 2010. Satellite oil spill detection and monitoring in the optical range. *Int. Geosci. Remote Sens. Symp.* 4487–4490.
- Grimaldi, C.S.L., Coviello, I., Lacava, T., Pergola, N., Tramutoli, V., 2009. Near real time oil spill detection and monitoring using satellite optical data. 2009 IEEE Int. Geosci. Remote Sens. Symp. IV–709–IV–712.
- Gokaraju, B., Durbha, S.S., King, R.L., Younan, N.H., 2011. A Machine Learning Based Spatio-Temporal Data Mining Approach for Detection of Harmful Algal Blooms in the Gulf of Mexico. 4(3) pp. 710–720.
- Harris, M., 2007. Optimizing Parallel Reduction in CUDA. NVIDIA Dev. Tech. [Online]. [http://developer.download.nvidia.com/compute/cuda/1.1-Beta/x86\\_website/projects/reduction/doc/reduction.pdf](http://developer.download.nvidia.com/compute/cuda/1.1-Beta/x86_website/projects/reduction/doc/reduction.pdf) ([Accessed:05-Apr-2017]).
- Jarvis, R.A., 1973. On the identification of the convex hull of a finite set of points in the plane. *Inf. Process. Lett.* 2 (1), 18–21.
- Jha, M.N., Levy, J., Gao, Y., 2008. Advances in remote sensing for oil spill disaster management: state-of-the-art sensors technology for oil spill surveillance. *Sensors* 8 (1), 236–255.
- Jiao, L., Liu, Y., 2012. Analyzing the shape characteristics of land use classes in remote sensing imagery. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 1–7, 135–140.
- Kandrot, E., Sanders, J., 2011. CUDA by Example. Addison-Wesley Professional (ISBN: 0131387685 9780131387683).
- Leifer, I., Lehr, J.W., Simecek-Beatty, D., Bradley, E., Clark, R., Dennison, P., ... Wozencraft, J., 2012. State of the art satellite and airborne marine oil spill remote sensing: application to the BP deepwater horizon oil spill. *Remote Sens. Environ.* 124, 185–209.
- Lewis, S., 2009. Remote Sensing for Natural Disasters Facts and Figures - SciDev, 2009. [Online]. Available <http://www.scidev.net/global/earth-science/feature/remote-sensing-for-natural-disasters-facts-and-figures.html> ([Accessed: 01-Jul-2015]).
- Ma, Y., Chen, L., Liu, P., Lu, K., 2014. Parallel programming templates for remote sensing image processing on GPU architectures: design and implementation. *Computing* 98 (1–2), 7–33.
- Mura, M.D., Benediktsson, J.A., Waske, B., Bruzzone, L., 2010. Morphological Attribute Profiles for the Analysis of Very High Resolution Images. *IEEE Trans. Geosci. Remote Sens.* 48 (10), 3747–3762.
- NOAA, 2014. Wind Effects on Oil Spills in Rivers Response.[Online]. Available: <http://response.restoration.noaa.gov/oil-and-chemical-spills/oil-spills/resources/wind-effects-oil-spills-rivers.html> ([Accessed: 01-May-2014]).
- NOAA, 2015. Deepwater horizon restoration plans for Gulf of Mexico. [Online]. Available: <http://response.restoration.noaa.gov/oil-and-chemical-spills/significant-incidents/deepwater-horizon-oil-spill>.
- Nvidia, 2012. NVIDIA's Netx Generation CUDA Compute Architecture: Kepler GK110. pp. 1–24.
- NVIDIA, 2015. Programming Guide: CUDA Toolkit Documentation. [Online]. Available: <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>.
- Plaza, A., Du, Q., Chang, Y.-L., King, R.L., 2011. High performance computing for hyperspectral remote sensing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 4 (3), 528–544.
- Pesaresi, M., Benediktsson, J.A., 2001. A new approach for the morphological segmentation of high resolution satellite imagery. *IEEE Trans. Geosci. Remote Sens.* 39 (2), 309–320.
- Persello, C., Bruzzone, L., 2010. A novel protocol for accuracy assessment in classification of very high resolution images. *IEEE Trans. Geosci. Remote Sens.* 48 (3), 265–268.
- Plaza, J., Pérez, R., Plaza, A., Martínez, P., Valencia, D., 2005. Mapping oil spills on sea water using spectral mixture analysis of hyperspectral image data. *Proc. SPIE* 5995, 1–8.
- Reddy, C.M., Arey, J.S., Seewald, J.S., Sylva, S.P., Lemkau, K.L., Nelson, R.K., ... Camilli, R., 2012. Composition and fate of gas and oil released to the water column during the Deepwater horizon oil spill. *Proc. Natl. Acad. Sci. U. S. A.* 109 (50), 20229–20234.
- Stathakis, D., Topouzelis, K., Karathanassi, V., 2006. Large-scale feature selection using evolved neural networks. 13th SPIE Symposium on Remote Sensing: The International Society for Optical Engineering. 6365 (636513-1).
- Solberg, A.H.S., Brekke, C., Husøy, P.O., C., 2007. Oil spill detection in Radarsat and Envisat SAR images. *IEEE Trans. Geosci. Remote Sens.* 45 (3), 746–754.
- Srivastava, H., Singh, T.P., 2010. Assessment and development of algorithms to detection of oil spills using MODIS data. *J. Indian Soc. Remote Sens.* 38 (1), 161–167.
- Taravat, A., Del Frate, F., 2012. Development of band ratioing algorithms and neural networks to detection of oil spills using Landsat ETM+ data. *EURASIP J. Adv. Signal Process.* 1, 107.
- Topouzelis, K.N., 2008. Oil spill detection by SAR images: dark formation detection, feature extraction and classification algorithms. *Sensors* 8 (10), 6642–6659.
- Taşkın, G.K., Ersoy, O.K., Kamaşak, M.E., 2011. Support vector selection and adaptation for remote sensing classification. *IEEE Trans. Geosci. Remote Sens.* 49 (6), 2071–2079.
- USGS, 2013. [http://landsat.usgs.gov/sci\\_an.php](http://landsat.usgs.gov/sci_an.php).
- USGS, 2014. Hazards Data Distribution System (HDDS) Explorer.[Online]. Available: <http://hddsexplorer.usgs.gov> ([Accessed: 01-Aug-2014]).
- Xue, Y., Cai, G., Guan, Y.N., Cracknell, A.P., Tang, J., 2005. Iterative self-consistent approach for Earth surface temperature determination. *Int. J. Remote Sens.* 26 (1), 185–192.
- Xu, L., Li, J., Brenning, A., 2014. A comparative study of different classification techniques for marine oil spill identification using RADARSAT-1 imagery. *Remote Sens. Environ.* 141, 14–23.