In association with
32nd Spring Symposium of KSRS
25th Annual Workshop of EMSEA

Organized by:
The Korean Society of Remote Sensing (KSRS)
The Remote Sensing Society of Japan (RSSJ)
The Chinese Society of Photogrammetry and Remote Sensing (CSPRS)
The Environmental Monitoring from Space for East Asia (EMSEA)

Supported by:
The Korean Federation of Science and Technology Societies (KOFST)
Jeju Special Self-Governing Province, Jeju Convention & Visitors Bureau (JCVB)

Final Program of

ISRS 2016
International Symposium on Remote Sensing 2016
In association with 25th Anniversary Celebration of EMSEA

20 – 22 April 2016
International Convention Center (ICC) JEJU
Jeju, Korea

Image Acquired from Kompsat-3, copyright ©2016 by KARI
21 April 2016

Room Samda Hall A

**KIOST Special Session**

**Geostationary Ocean Color Mission I**

*Chair:* Young-Je Park  
(Korea Institute of Ocean Science and Technology)

0900  **VERSION 1.4 UPDATE IN GOCI DATA PROCESSING SYSTEM (GDPS) AND INTRODUCTION OF GOCI DATA REPROCESSING SYSTEM (GDPS)**  
Hyun Yang, Nam-Su Kim, Jeung-Mi Ryu, Hee-Jeong Han, Young-Je Park and Yu-Hwan Ahn  
Korea Institute of Ocean Science and Technology

0920  **SPATIO-TEMPORAL MODELING OF AEROSOL OPTICAL THICKNESS OVER THE GOCI LAND AREA FOR THE ATMOSPHERIC CORRECTION OF LAND PRODUCTS**  
Hwa-Seon Lee and Kyu-Sung Lee  
Inha University

0940  **A NEW MULTIPLE-SCATTERING ALGORITHM FOR THE GOCI ATMOSPHERIC CORRECTION**  
Jae-Hyun Ahn  
Korea Ocean Satellite Center, KIOST/Ocean Science & Technology School

Young-Je Park and Wondeok Kim  
Korea Ocean Satellite Center, KIOST

Boram Lee  
Korea Ocean Satellite Center, KIOST/Sejong University

1000  **SCENE-BASED TURBID WATER ATMOSPHERIC CORRECTION FOR GOCI**  
Boram Lee  
Korea Ocean Satellite Center, KIOST/Sejong University

Young-Je Park and Jae-Hyun Ahn  
Korea Ocean Satellite Center, KIOST

Sang-Wan Kim  
Sejong University

Room Samda Hall A

**KIOST Special Session**

**Geostationary Ocean Color Mission II**

*Chair:* Young-Heon Jo  
(Pusan National University)

1040  **SPATIAL VARIABILITY OF FISHING GROUNDS IN THE EAST SEA IN RESPONSE TO OCEANIC FRONT CHANGES**  
Yejin Oh and Young-Heon Jo  
Pusan National University

1100  **CLASSIFICATION AND CHANGE DETECTION OF INTERTIDAL SEDIMENT GRAIN SIZE IN YELLOW SEA USING A TWO-STEP PCA MODEL APPLIED TO GOCI IMAGE**  
Dong-Jae Kwon, Wook Park, Hongtaek Lee and Joong-Sun Won  
Yonsei University

1120  **EVALUATION OF CHLOROPHYLL ALGORITHMS FOR GOCI-II USING IN SITU RADIOMETRIC DATA**  
Wondeok Kim, Jeong-Eun Moon, Naehun Kim and Young-Je Park  
Korea Ocean Satellite Center, Korea Institute of Ocean Science & Technology

1140  **CURRENT DEVELOPMENT STATUS OF GOCI-II**  
Sang-Soon Yong, Haegong-Pal Heo, Gm-Sil Kang and Sang-Youl Shin  
Korea Aerospace Research Institute

**KIOST Special Session**

**Geostationary Ocean Color Mission III**

*Chair:* Seongik Cho  
(Korea Institute of Ocean Science and Technology)

1300  **GOCI-II ON-ORBIT CALIBRATION PLAN: LESSONS FROM GOCI**  
Seongik Cho, Ki-Beom Ahn and Einseong Oh  
Korea Ocean Satellite Center, Korea Institute of Ocean Science & Technology

Young-Je Park  
Korea Ocean Satellite Center, Korea Institute of Ocean Science & Technology

1320  **GOCI-II MISSION OPERATION PLAN**  
Ki-Beom Ahn, Seongik Cho and Einseong Oh  
Korea Ocean Satellite Center, Korea Institute of Ocean Science & Technology

Young-Je Park  
Korea Ocean Satellite Center, Korea Institute of Ocean Science & Technology

1340  **SYSTEM DESIGN OF GOCI-II GROUND SEGMENT**  
Hee-Jeong Han, Hyun Yang, Jae-Moo Heo and Young-Je Park  
Korea Ocean Satellite Center (KOSC), Korea Institute of Ocean Science and Technology

Sunghee Kim  
Saticec Initiative Co., Ltd.

1400  **TECHNICAL ANALYSIS OF PARALLEL ALGORITHM IN GOCI-II DATA PROCESSING**  
Jae-Moo Heo, Hyun Yang, Hee-Jeong Han and Young-Je Park  
Korea Ocean Satellite Center, Korea Institute of Ocean Science & Technology
1. INTRODUCTION

Geostationary Ocean Color Imager II (GOCI-II), follow-up satellite of GOCI that was launched from French Guiana in June 2010 and is operating currently (Ryu, 2012), is expected to be launched in 2016. GOCI-II will succeed the GOCI’s mission and undertake a various missions additionally. It has a spatial resolution of 250m, four times greater than GOCI, and an observation area of $2500 \times 2500$ km. The observation number is also increased from eight to ten times a day and full disk observation is added once a day. In addition, four ocean observation bands and one wide band for star observation are added to existing eight bands of GOCI. The kind of products is increased from 13 to 26 as well. Thus, GOCI-II is expected to perform more data processing 24 times more than GOCI. Especially, GOCI-II Ground System (G2GS) should perform all courses that include data reception, processing, storage, and distribution of satellite data obtained from GOCI-II within 37 minutes. It is necessary to develop efficient mass data processing system in order to satisfy above requirements.

There are many papers related to parallel processing techniques to take full advantage of limited hardware resources (Chen, 2011; Ehsan, 2102; Rexer, 2015). GOCI-II data will provide marine environmental information (green tide, red tide, turbidity, sea ice, etc.) around the Korean Peninsula ten times per day and full disk observation data once a day (Ryu, 2012). It will also be used as important public information to improve the quality of people’s lives, as well as study materials for researchers. We found the most efficient parallel processing methodology for GOCI-II algorithms. Through the application of various parallelism techniques, we will be able to develop the G2GS reliably and contribute to the successful operation of GOCI-II.

Parallelization of GOCI-II data processing should be able to run on both Windows and Linux systems and achieve effective performance improvements in the limited hardware resources. Furthermore, it should be able to apply various parallelization techniques in consideration of the possibility of adding new algorithms in the future. For this, we used Chundoong Cluster (http://chundoong.smu.ac.kr), a heterogeneous cluster which refers to a cluster of nodes connected by a high-speed network and used with accelerators of CPU, GPU, and Xeon Phi Coprocessor, etc., at Seoul National University. We tested parallelization techniques for algorithm as below. Open specification for Multi-Processing (OpenMP) is a multi-threaded application programming interface for parallel programming (http://openmp.org). It provides a simple and easy-to-use API and is the de facto standard of shared memory parallel programming. Open Computing Language (OpenCL) is a standard parallel programming model for various heterogeneous platforms and has highly portable because it can be programmed for devices of the various architecture such as CPU, GPU and FPGA (https://www.kchronos.org/opencl/). Message Passing Interface (MPI) is specific to the message passing model of the distributed memory system (http://www.openmpi.org). MPI can be used in most clusters as a de facto standard for message passing parallel programming. In short, we applied three parallel processing techniques to multiple ocean color algorithms for GOCI, found the most appropriate techniques and evaluated the degree of parallelism level.

This paper is organized as follows. Section 2 describes the status of the GOCI data processing and GOCI algorithms for parallel processing. In section 3, we describe parallelization methods of GOCI algorithms. Section 4 describes the experimental results of proposed parallelization methods and Section 5 describes a speedup of each method and a parallelism level of each algorithm. Finally, we conclude the paper in Section 6.
2. GOCI DATA PROCESSING

2.1 Status of GOCI Data Processing

Raw data obtained from GOCI is digitized through 12-bit quantization. Digital data is converted into two-dimensional radience data consisting of 5667 × 5685 pixels through the geometric correction. This is a two-dimensional radience data and the data for each of the eight bands constitutes one set. This is called level-1 data (Ahn, 2012). The remote-sensing reflectance (Rrs) data is produced from the radience information in level-1 data and Rrs data is used as input data for estimation algorithms of Total Suspended Solid concentration (TSS), the Chlorophyll concentration (chl) and absorption coefficient of Colored Dissolved Organic Matter (CDOM), etc. after the atmospheric correction (Yang, 2012). These algorithms produce a data called level-2.

2.2 GOCI Level-2 Algorithm

In order to analyze the difference in the algorithm-specific parallelism level, we considered the algorithms having differences in computational complexity and amount of data size. We tested the algorithms of TSS using Yellow Sea Large Marine Ecosystem (YSLME) schemes (equation 1), chl using the ocean chlorophyll 2-band (OC2) (equation 2) and CDOM using YSLME schemes (equation 3) (Siswanto, 2011). Each algorithm has one, two and three Rrs data, respectively. And each Rrs data has a capacity of 130MB.

\[
TSS = c_s \times R_{\lambda} (555) \tag{1}
\]

\[
CHL_{OC2} = e_c + 10^{0.33 R_{\lambda} (490) + 0.4 R_{\lambda} (544) + 0.9 R_{\lambda} (443)} \tag{2}
\]

\[
CDOM_{YSLME} = 10^{0.5 \log_{10} (C_{\lambda})} \tag{3}
\]

\[R = \frac{R_{\lambda} (490)}{R_{\lambda} (555)} \times \frac{R_{\lambda} (443)}{R_{\lambda} (443)} \]

where \( c_s, c \) coefficient values
\( R_{\lambda} (443), R_{\lambda} (490), R_{\lambda} (555) \) = Rrs data at 443 nm, 490 nm and 555 nm wavelengths, respectively

3. PARALLELLIZATION METHODS OF GOCI LEVEL-2 ALGORITHMS

3.1 Analysis of Algorithm

Table 1 shows the specifications of Chundoong Cluster and Figure 1 shows the sequential data processing time of each algorithm. Assuming the rest is a part that cannot be resolved through these parallel programming languages in the existing system, the parallelism is available at the part of algorithm. Therefore, the possible part for parallelism of TSS, chl and CDOM algorithms are 68%, 81%, 84%, respectively. We can quantify the speedup from parallelism on any system through the Amdahl’s law as below (equation 4). And it is also used in the prediction of the maximum speedup that can be obtained when we parallelize any program and it can be expressed in the format as below (equation 5) (Mark, 2008),

\[
Speedup_{parallel} (P, S) = \frac{1}{(1 - P) + \frac{P}{S}} \tag{4}
\]

\[
Speedup_{parallel} (P, N) = \frac{1}{(1 - P) + \frac{P}{N}} \tag{5}
\]

where \( P \) = possible part for parallelization of algorithm
\( S \) = speedup from parallelism
\( N \) = the number of processors or nodes

If N is assumed as the 16 that is the number of core in the single compute node and P is the possible part for parallelism of TSS, chl and CDOM algorithm, maximum speedup showed 2.74, 4.19 and 4.68, respectively.

3.2 Parallelization of Algorithm

We tested OpenMP, OpenCL and MPI programming for GOCI level-2 algorithms. Each parallelization algorithm was processed in the cores of processor, GPU and nodes, respectively. In the performance of the OpenMP, there is
an interrelation between cores and threads because threads can be executed as much as the number of cores. One computing node of Chunodoog Cluster is equipped with two 8-core processors, for a total of 16 physical cores and 32 logical cores. Thus, it was divided into the appropriate number of threads when we tested parallel programming in the OpenMP (Shen, 2013). In the OpenCL programming, the amount of data being processed in the single local work group, involved in the running units at a time, of the GPU. Because the single GPU of the experiment can process the 64 works in parallel, we performed the experiment where the size of the local work group was increased gradually (Pennycook, 2013). In the MPI programming, data was divided into the number of nodes and were transferred to each node. Output data of processing was collected again after data was processed at each node. At this time, we performed the experiment of the MPI programming where the number of nodes was increased gradually, because the network bottlenecks was aggravated with an increase in the number of nodes and the size of data (Shen, 2013).

4. EXPERIMENTAL RESULTS

Figure 2 (a) shows the speedup of parallel processing by the OpenMP. It shows an improved speedup when the number of threads increased gradually. But speedup was small after the number of threads was 16 which is equal to the number of physical cores in the single compute node and was the largest when the number of threads was 32 which is equal to the number of logical cores in the single compute node. Figure 2 (b) shows the speedup of parallel processing by the OpenCL. It shows a little improved speedup when the size of local work group increased to 256 that is the maximum size restricted from hardware specification of GPU. Figure 2 (c) shows the speedup of parallel processing by the MPI. It shows improved speedup when the number of nodes increased gradually and speedup of TSS, chl and CDOM was largest when the number of nodes is 10, 16 and 18, respectively. We confirmed that there is a difference in data transmission times because a distributed memory environment requires relatively long time for the data transmission.

![Figure 2. Speedup of each parallel algorithm by the (a)OpenMP, (b) OpenCL and (c) MPI](image)

Table 2 shows an analysis result of run-time when the best speedup draw by each parallel processing technique. The actual algorithm calculation took the least time by the OpenCL programming, but the network bottlenecks caused in file I/O and data transmission. As a result, total run-time took longer than the OpenMP and it was the same result in the MPI.

5. DISCUSSION

Table 3 shows the actual performance effect due to the parallelization experiments. The values of Karp-Flatt metric mean the ratio of the sequential run-time in parallelization algorithm. The values of Amdahl's law mentioned above and these values are closely related. The possible part for parallelization of TSS, chl and CDOM algorithms which obtained by Amdahl’s law is 68%, 81%

<table>
<thead>
<tr>
<th>Programming Model</th>
<th>Sequential</th>
<th>OpenMP (32 threads)</th>
<th>OpenCL</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>File I/O &amp; Data Transmission</td>
<td>TSS</td>
<td>0.86</td>
<td>0.93</td>
<td>1.82</td>
</tr>
<tr>
<td></td>
<td>chl</td>
<td>0.94</td>
<td>1.01</td>
<td>1.86</td>
</tr>
<tr>
<td></td>
<td>CDOM</td>
<td>0.97</td>
<td>1.03</td>
<td>2.15</td>
</tr>
<tr>
<td>Algorithm Calculation</td>
<td>TSS</td>
<td>1.81</td>
<td>0.11</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>chl</td>
<td>4.05</td>
<td>0.26</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>CDOM</td>
<td>5.04</td>
<td>0.36</td>
<td>0.14</td>
</tr>
<tr>
<td>Total (Sec)</td>
<td>TSS</td>
<td>2.67</td>
<td>1.04</td>
<td>1.90</td>
</tr>
<tr>
<td></td>
<td>chl</td>
<td>4.99</td>
<td>1.27</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>CDOM</td>
<td>6.01</td>
<td>1.39</td>
<td>2.29</td>
</tr>
</tbody>
</table>
Table 3. Indicators of parallel programming

<table>
<thead>
<tr>
<th>Programming Model</th>
<th>OpenMP (32 threads)</th>
<th>OpenCL</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karp-Flatt Metric</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSS</td>
<td>37%</td>
<td>71%</td>
<td>53% (10 nodes)</td>
</tr>
<tr>
<td>chl</td>
<td>23%</td>
<td>38%</td>
<td>33% (16 nodes)</td>
</tr>
<tr>
<td>CDOM</td>
<td>21%</td>
<td>37%</td>
<td>30% (18 nodes)</td>
</tr>
<tr>
<td>Speedup</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSS</td>
<td>2.56</td>
<td>1.40</td>
<td>1.73</td>
</tr>
<tr>
<td>chl</td>
<td>3.92</td>
<td>2.56</td>
<td>2.67</td>
</tr>
<tr>
<td>CDOM</td>
<td>4.45</td>
<td>2.62</td>
<td>2.96</td>
</tr>
<tr>
<td>Efficiency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSS</td>
<td>1.28</td>
<td>0.70</td>
<td>0.17</td>
</tr>
<tr>
<td>chl</td>
<td>1.96</td>
<td>1.28</td>
<td>0.17</td>
</tr>
<tr>
<td>CDOM</td>
<td>2.23</td>
<td>1.31</td>
<td>0.16</td>
</tr>
</tbody>
</table>

and 84%, respectively. And Karp-Flatt metric in the OpenMP is 37%, 23% and 21%, respectively.

Compared the two indicators, the parallelization by the OpenMP is relatively more effective than the others. And we can confirm how much speedup of each algorithm. Especially, speedup of TSS algorithm was very poor in the OpenCL and MPI. And the higher computational complexity of the algorithm, the higher speedup came out. And indicator of how efficiently parallelized is the efficiency. The efficiency is the value that is divided by the number of added hardware devices from speedup.

Because two processors was used in the OpenMP, one processor and one GPU were used in the OpenCL and each of 10, 16, and 18 nodes were used in the MPI. The number of hardware devices has been selected from these values. Although efficiency is generally more efficient as close to 1, the values of the above could come since the existing algorithms did not take full advantage of the hardware.

These results show that the performance is bad considering the hardware construction cost for MPI programming. Consequently, it showed the best speedup and efficiency in case of the OpenMP. If the computational complexity of algorithm is high and the amount of data is small, we can consider the OpenCL programming. Also, MPI programming is inefficient in parallelization of the GOCI level-2 algorithm.

6. CONCLUSION

In this paper, we come up with the most efficient parallel processing methodology for Geostationary Ocean Color Imager-II (GOCI-II) algorithms. Through the application of various parallel programming languages, we tried to develop the GOCI-II Ground System (G2GS) reliably and contribute to the successful operation of GOCI-II. For this, we tested the OpenMP, OpenCL and MPI programming to TSS, chl and CDOM algorithms which is current GOCI algorithm. In experimental results, Speedup is 2.56, 1.40, 1.73 in TSS algorithm, 3.92, 2.56, 2.67 in chl algorithm and 4.45, 2.62, 2.96 in CDOM algorithm, respectively, in experimental results from Chundoong cluster at Seoul National University. Through Comparison of the Amdahl’s law and Karp-Flatt metric, it came close to the ideal speedup and the higher efficiency in the OpenMP.

Because only three kinds of algorithms were tested in this experiment, it requires experiments on other algorithms. And other parallel processing techniques should also be tested such as Straming SIMD Extensions (SSE) or Multi Media eXtension (MMX). In the future work, we will test the parallelism of atmospheric correction which takes a lot of time in the G2GS.

7. REFERENCES


8. ACKNOWLEDGEMENTS

This research was supported by the “Research for applications of Geostationary Ocean Color Imager (GOCI)” and the “Development of the integrated data processing system for GOCI-II” funded by the Ministry of Ocean and Fisheries, Korea. This research was also supported by the “Operations in Korea Ocean Satellite Center” funded by the Korea Institute of Ocean Science and Technology (KIOST).