EPSC Abstracts
Vol. 9, EPSC2014-476, 2014
European Planetary Science Congress 2014

Calculation of illumination conditions at the lunar south pole - parallel programming approach

R. Marco Figuera (1), P. Gläser (1), J. Oberst (1,2) and D. De Rosa (3)
(1) Technical University Berlin, Department of Geodesy and Geoinformation Science, Str. des 17. Juni 135, 10623 Berlin, Germany, (rmafi@campus.tu-berlin.de)
(2) German Aerospace Center, Institute of Planetary Research, Rutherfordstrasse 2, 12489 Berlin, Germany
(3) European Space Agency, ESTEC, Keplerlaan 1, 2200 AG Noordwijk ZH, The Netherlands

Abstract

In this paper we present a parallel programming approach to evaluate illumination conditions at the lunar south pole. Due to the small inclination (1.54°) of the lunar rotational axis with respect to the ecliptic plane and the topography of the lunar south pole, which allows long illumination periods, the study of illumination conditions is of great importance. Several tests were conducted in order to check the viability of the study and to optimize the tool used to calculate such illumination. First results using a simulated case study showed a reduction of the computation time in the order of 8-12 times using parallel programming in the Graphic Processing Unit (GPU) in comparison with sequential programming in the Central Processing Unit (CPU).

1. Introduction

The calculation of illumination conditions for a certain Region of Interest (RoI) can be a highly time consuming process and mainly depends on the size of the RoI. The illumination calculation is achieved by comparing the elevation of the Sun w.r.t. an observer on the surface of the moon to the maximum elevation of the terrain in the same direction. The calculation of the visible fraction of the solar disk is then derived which represents the degree of illumination.[2]. While sequential programming executes processes one after another, parallel programming executes multiple processes at the same time [3]. If we consider each pixel of our RoI as an individual process we can apply parallel programming techniques to evaluate the illumination for multiple pixels at the same time, and thus reducing the computation time.

To achieve such an improvement we will have to consider the size of the input data as well as the size of the process to be executed concurrently. The parallel programming approach has been programmed using Open Computing Language (OpenCL) and the sequential approach using C++. The test case was run on a quad-core Intel Xeon CPU and a Nvidia Quadro FX 1800 GPU.

2. Method

We have conducted a simulation by using a variable region of interest (1000 to 10000000 pixels) and different platforms (CPU and GPU) to check whether and to which degree the use of OpenCL decreases the computation time. The case study has been set up as a simple vector addition problem to be executed for each pixel and all the process repeated for 360 times. The reason to do such a repetition is to simulate the derivation of a complete horizon as seen in Fig. 1 from an observer on the lunar surface.

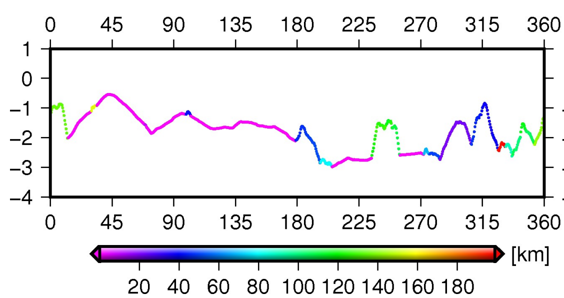

Figure 1: Preliminary results of an horizon as seen from an observer near the lunar south pole. The distance to the horizon is color coded.

Five different approaches have been studied in this paper, one sequential programming approach using C++ on the CPU, one parallel approach using OpenCL on the CPU and 3 parallel approaches using OpenCL on the GPU.

For the OpenCL approaches different configurations have been considered. The OpenCL CPU approach uses the maximum number of cores in our card. For the OpenCL GPU approaches 3 different scenarios have been considered varying the number of processes (threads) that can be launched concurrently: 1 thread, 2 threads and an optimal number of threads. In this case the variation of the number of threads will change the number of pixels that can be computed at the same time.

3. Results

For a small number of pixels, no difference between the approaches can be noticed. As the number of pixels of the RoI increase the approaches start to show different behaviors. From Fig. 2 we can conclude that the usage of the GPU with 1 thread is twice as slow as the CPU approach. This is due to the time necessary for the GPU to access the data stored in the memory. When the number of threads is increased the time will be reduced as there will be more than one process running at the same time. The use of an optimal number of threads will increase the performance of our tool by a factor of 8-12 with respect to the CPU C++ approach and by an order of 3 with respect to the CPU OpenCL approach. As can be noted in Fig. 2 the use of OpenCL running on the GPU with an optimal number of threads is unrivaled for larger areas.

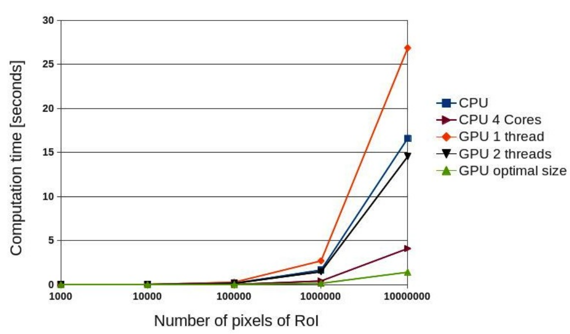

Figure 2: Comparison of computation time for each approach depending on the size of the input data.

4. Summary and Conclusions

Five different tests were conducted in order to determine the performance of parallel programming in general and especially for the calculation of illumination conditions. The results show that the use of parallel programming on the GPU in combination with larger RoIs can improve the performance significantly in comparison with sequential programming approaches.

Since the analyzed RoIs are in the order of hundred thousands to millions of pixels [1], the GPU approach is evidently favorable.

Considering the results we can assume that an OpenCL GPU approach using its optimal number of threads will decrease the computation time in the order of 8-12 compared to an approach using C++ in the CPU and in the order of 3 compared to an approach using OpenCL in the CPU.

5. Future Work

The final goal of such a study is to develop an OpenCL tool to calculate the horizon and illumination conditions using the CPU and the GPU respectively. Currently a first version of the OpenCL CPU tool for the calculation of the horizon is cross-validated. Once the tool is fully operational we will be able to quickly derive illumination conditions for various sites at the lunar south pole.

Acknowledgements

This project was funded by a Grant of the German Space Agency (FKZ 50OW1202).

References

[1] De Rosa, D., Bussey, B., Cahill, J.T., Lutz, T., Crawford, I.A., Hackwill, T., van Gasselt, S., Neukum, G., Witte, L., McGovern, A., Grindford, P.M., Carpenter, J.D., : Characterisation of potential landing sites for the European Space Agency's Lunar Lander project. Planetary and Space Science 74, 224-246. 1208.5587., 2012.

[2] Gläser, P., Scholten, F., De Rosa, D., Marco Figuera, R., Oberst, J., Mazarico, E., Neumann, G.A., Robinson, M.S.: A highly illuminated landing site at the lunar south pole - Connecting Ridge. EPSC 2014.

[3] Munshi, A., Gaster,B., Mattson, T. G., Ginsburg, D.: OpenCL Programming Guide, Pearson Education, 2011.