www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 11 November 2017, Page No. 22858-22860
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i11.06

Parallelization of Least Significant Bit Algorithm for Image Steganography in GPU using CUDA

Sabareesh K, Divya M

Department of Computer Science , Amrita Vishwa Vidyapeetham,
Ettimadai, Coimbatore, Tamil Nadu, India.

Abstract

This paper presents a method for accelerating the performance of LSB (Least Significant Bit) Algorithm by the concept of parallelization through GPU (Graphics Processing Unit) of the system using CUDA(Compute Unified Device Architecture).The paper also illustrates Image Steganography as an application of LSB algorithm for the encryption of audio file format within a higher resolution digital image file. The principal idea behind of the system is to perform optimal programing of the LSB algorithm for GPU compatibility to parallelize the computation by using M*N threads of the GPU for an image of M*N resolution in which the audio bit encryption is applied. The results of CPU and GPU aided image steganography are calculated separately and compared to estimate performance in gain achieved in parallelization.

Keywords— CUDA, GPU, LSB Algorithm, Image Steganography, Parallel Processing, Acceleration of Algorithm , Parallel Computing.

Introduction

Least Significant Bit Embedding (LSB) is a universal and most common steganography technique that is employed to embed data into a wide range of digital media. But one of the most popular applications of LSB embedding to hide an image within an another. The technique is most suitable for cover files that are larger than the image file , considered in bitmap standard. The algorithm primarily works on the idea of message file's bit insertion in the LSB of successive bytes of the cover pixel ,there by embedding 3 bits of message file in a pixel with 3 bytes of data in the cover file. This method is proved to be highly efficient and undetectable, but the performance of the algorithm on larger message file tends be slow down the running time of the system dramatically. With this paper ,the concept of parallelizing the LSB algorithm for image steganography is proposed to greatly reduce the running time delay of the LSB algorithm for larger message files.

Steganography is a technique which deals with encrypting message/file within an another message/file without creating any significant visual alteration to the cover file. The aim of image steganography is to hide a file within a cover image, such that the encoded file's existence is concealed , there by retaining the identity of the cover file. With such reliable data processing , image steganography finds a place in numerous real world application like, Image piracy ,Intelligence agency , Smart card identity ,secured data transfer and multi format file transfer .Having discussed about image steganography this paper deals with encryption of audio file within a image , unlike the conventional image within image methodology.

A GPU, or graphics processing unit, is a single-chip processor primarily used to manage and boost the performance of video and graphics. The computations are mathematically-intensive tasks, which otherwise, would put quite a strain on the CPU performance. But unlike the CPU the GPU is optimised to handle specific programs like graphics and high intensity computation in order to enhance the functionality of CPU. Moreover the GPU is capable of being programmed manually to handle specific user programs ,which widens the computational capability of GPU besides its conventional graphic rendering process.

CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units (GPU). With CUDA, developers are able to dramatically speed up computing applications by harnessing the power of GPUs. In GPU-accelerated applications, the sequential part of the workload runs on the CPU which is optimized for single-threaded performance ,while the computation intensive portion of the application runs on thousands of GPU cores in parallel. When using CUDA, developers program in popular languages such as C, C++, Fortran, Python and MATLAB and express parallelism. It includes the CUDA Instruction Set Architecture (ISA) and the parallel compute engine in the GPU. This accessibility makes it easier for specialists in parallel programming to use GPU resources, in contrast to prior APIs like Direct3D and OpenGL, which required advanced skills in graphics programming. This system talks about the parallelisation of LSB algorithm using CUDA parallel computing platform written in C# on Visual Studio environment.

Tools And Requirements

Following tool and specifications are expected for the computation

A. Software Requirements

- NVIDIA CUDA 5.5 Toolkit
- Microsoft Visual Studio 2012 (for Windows 7, 8, 8.1) or Eclipse (for Linux)

B. Hardware Requirements

- A CUDA capable NVIDIA Graphics Card
- Preferably 2 GB DDR3 RAM or Greater
- Preferably Intel Core i3-2330M CPU @ 2.20 GHz processor or Greater

Implementation

Implementation of the LSB Algorithm is done with inputs of an image of size M*N, where M represents number of pixels on the X-Axis and N represents number of pixels in the Y-Axis and an audio file which is chosen as .message file. The input cover image needs to be converted into a bitmap format, where each pixel is represent by the values taken by the RGB components. The audio file is read in binary format and stored in the LSB of the RGB components for the considered bitmap cover image in a cyclic order. Each pixel can store 3 bits of data from the considered audio file, where all the 3 bits are equally shared by RGB components in their LSB. Therefore 1 byte of the information is stored in 3 pixels of the cover image. The algorithm is implemented using CUDA (CUDA uses nvcc compiler which has C compiler with additional functions and keywords provided by NVIDIA)

Parallelization

A. Parallelization requirements and CUDA Architecture Support

To parallelize the proposed algorithm we require M*N threads corresponding to the resolution the image file. Since these threads can execute independent of each other, we can parallelize LSB to maximum with each thread performing bit insertion in single pixel of the image file. This requirements is satisfied by the utilisation nVIDIA Graphics Processing Unit, which provides innumerable kernels capable of independent execution .The GPU comprises of several Blocks with 1024 grids in each and with each grid carrying 1024 threads . The availability of multifarious system of block and threads within a single core of the GPU provides an ideal architecture for the parallelization of LSB Algorithm for high intensity image steganography process.

B. Data Structures

Every byte is stored as an unsigned character, with the values ranging from 0 to 255. Collection of all bytes are stored in unsigned character array. The byte values are converted to Boolean when the LSB of the image needs to be altered to store the audio bits and they are again converted back to bytes and these bytes ate written to the file. In order to make GPU operational we need to allocate memory in the device (device represents GPU) and pass the unsigned char array pointer to the __global__ function (__global__ is a keyword which is comprehended by CUDA).

C. Workflow in CUDA

The memory allocation for the device is followed coping the data in by means of the __global__ function call with M number of grids and N number of kernels. When the __global__ function is invoked the control is shifted from CPU to GPU. The LSB Algorithm is defined in the __global__ function with appropriate grid id and thread id. Once all the kernel finishes its execution, the control is transferred back to CPU.

Performance Analysis

The performance measure used here is the Speed-Up factor. Speed-Up is defined as a ratio of Time Taken for the Execution of a set of instructions in Single Core to Time Taken for the Execution of the same set of instructions in Multicore. For the calculation, five sample cover images are chosen. The information to be hidden are the five sample audio files corresponding to the respective cover images. LSB Algorithm is executed separately on both CPU and GPU and the response time for their execution is measured and analysed respectively. The information regarding the sample inputs are mentioned in TABLE I. The result obtained are shown in TABLE II. The comparison of their execution time between CPU and GPU can be understood from Fig. 1

TABLE I

SAMLE INPUTS TO LSB ALGORITHM

SAMPLE	COVER IMAGE NAME (.BMP)	MESSAGE FILE NAME (.MP3)	SIZE OF THE COVER IMAGE (IN KB)	SIZE OF THE MESSAGE FILE (IN KB)
1	Elephant	Elephant	481	10
2	Penguin	Penguin	480	8
3	Bike	Bike	482	36
4	River	River	481	32
5	Tiger	Tiger	480	23

TABLE III

Results Of Lsb Algorithm Running Time In Cpu And Gpu For The Sample Inputs

SAMPLE	TIME TAKEN IN CPU (IN MS)	TIME TAKEN IN GPU (IN MS)	SPEED-UP
1	344	47	9.58
2	234	31	9.13
3	1197	73	41.96
4	1140	62	30.87
5	828	47	21.41

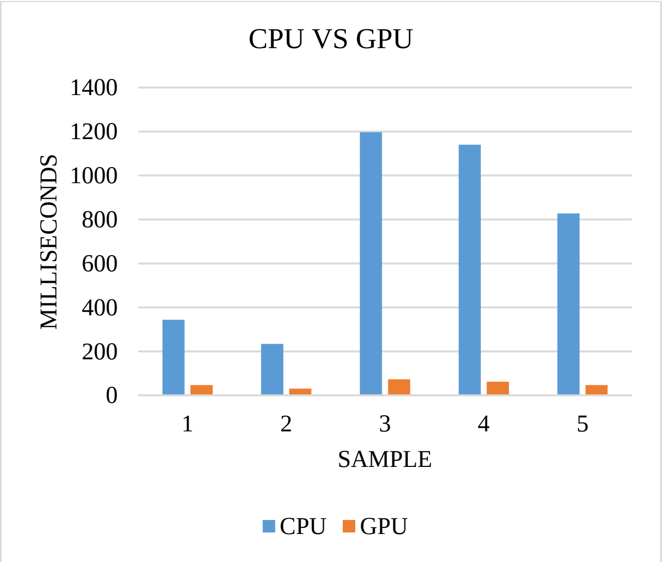

Fig. 1 Comparison of run time for execution in CPU and GPU

Conclusions

From the results of multiple computational trials, it is inferred that the execution time of LSB algorithm for image steganography is greatly accelerated by the utilizations of GPU capability, thereby accelerating the

LSB algorithm on an average 20 times faster than the corresponding CPU powered computation. It is also predicted that the performance gain can reach up 100 for relatively larger message file, yet the true power of CUDA can only be realized only high intensive real-time application where manifold computations are expected to be accomplished within specified time limit. Hence this parallelization approach to LSB algorithm that is established is found to be immensely productive and can be considered as a vital component of steganography methodologies.

Acknowledgment

I take this opportunity to express profound gratitude and regards to my project guide and Assistant Professor Miss Divya .M for her exemplary guidance, and constant encouragement throughout the course of this project. I also take this opportunity to express a deep sense of gratitude to Assistant Professor Miss Sikha O. K , for her cordial support, and suggestions in the conceptualisation of the project. I feel obliged by staff members of Security and Networking project panel, for their valuable contribution and suggestions with every phase of the project development. Finally, I thank the all my department staffs and friends without whom this project would not be successful.

References

1. Patel, B., Samir, Pradhan, N., Shrikant, Ambegaokar, U., Saumitra, ‘A Novel Approach for Implementing Steganography with Computing Power obtained by combining CUDA and MATLAB’, International Journal of Computer Science and Information Security (IJCSIS), Vol. 6, No. 2, pp.133-137, 2009.
2. Adnan Abdul-Aziz Gutub, ‘Pixel Indicator Technique for RGB Image Steganography’, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia.
3. Bloisi, Domenico, Ioochi, Luca, ‘Image based Steganography and Cryptography’, Sapienza University of Rome, Italy.
4. Gupta, Shilpa, Gujral, Geeta, Aggarwal, Neha, ‘Enhanced Least Significant Bit algorithm for Image Steganography’, IJCEM International Journal of Computational Engineering & Management, Vol. 15, Issue 4, pp. 40-42, July 2012.
5. Sorel, Y., ‘Massively Parallel Computing Systems with Real Time Constraints, The Algorithm Architecture Ad equation Methodology’, INRIA, Domaine de Voluceau, Rocquencourt B.P.105 78153 LE CHESNAY CEDEX FRANCE.
6. Mohammad Tanvir Parvez, Adnan Abdul-Aziz Gutub, ‘RGB Intensity Based Variable-Bits Image Steganography’, IEEE Asia-Pacific Services Computing Conference, pp. 1322-1327, 2008.
7. Yu-Chee Tseng, Member, IEEE, Yu-Yuan Chen, Hsiang-Kuang Pan, ‘A Secure Data Hiding Scheme for Binary Images’, IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 50, NO. 8, pp. 1227-1231, AUGUST 2002.
8. Channalli, Shashikala, Jadhav, Ajay, ‘Steganography, An Art of Hiding Data’, International Journal on Computer Science and Engineering Vol.1 (3), 2009, pp. 137-14