Indian Journal of Science and Technology, *Vol 9(6), DOI: 10.17485/ijst/2016/v9i6/72651, February 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

1D Chaos-based Image Encryption Acceleration by using GPU

Leila Habibpour*, Shamim Yousefi, Mina Zolfy Lighvan and Hadi S. Aghdasi

Faculty of Electrical and Computer Engineering, University of Tabriz, 29th Bahman Boulevard, Tabriz, Iran;
Habibpour.leyla@gmail.com, Shamimyousefi75@gmail.com, Mzolfy@tabrizu.ac.ir, Aghdasi@tabrizu.ac.ir

Abstract

Chaos-based image encryption algorithm is one of the most important methods that are considered as the main part of many structuring encryption systems. In this paper, a new implementation of One-Dimension (1D) chaos-based image encryption algorithm is presented using the parallelism features of GPU and CPU. In order to use the parallelism power of CPU, the parallel computing toolbox of MATLAB, provides efficient methods for Parallel Task Processing (PARFOR) and Parallel Data Processing (SPMD). For further improving the execution time of the algorithm, sequential partitions are performed on CPU and the parallel pieces are executed on the GPU. The results of serial and parallel implementation on the color images with different resolutions, using MATLAB parallelism methods show when the size of the pictures increase, the performance of the 1D chaos-based image encryption algorithm in parallel implementation by the both parallel task "PARFOR" and data processing "SPMD" methods, becomes better. Also, the results of the implementation illustrate that the execution time when PARFOR scheme is used becomes better when the image size is higher than a threshold. Furthermore, the results of Cuda and Visual C++ implementation on the color images with different resolutions show that the simulation time using Cuda C++ is almost three times better than visual C++. Total results of the comparison show that when a combination of CPU and GPU is used, the execution speed reached its best state. Because of accelerating the image encryption process using the power of CPU and GPU, the proposed implementation is suitable for the multimedia application systems.

Keywords: Accelerating Image Encryption Process, Graphic Processing Unit (GPU), One-Dimension (1D) Chaos-based Image Encryption Algorithm, Parallel Data Processing (SPMD), Parallel Task Processing (PARFOR)

1. Introduction

The increasing requirement for private data in current multimedia applications has made the information security as one of the most frequently stated problems in image communication. Therefore, several techniques of cryptography and steganography are proposed[1,2].

Encryption plays a key role in information security of multimedia application systems[3,4]. In the history of multimedia applications development, the security considerations and the encryption/decryption execution speed are key factors in designing image encryption algorithms, especially over fast communication channels[1].

In current communication applications, the traditional cryptographic algorithms such as DES[5], IDEA[6,7] and RSA[8] that require high computing load are not suitable candidates. An appropriate image encryption algorithm in present application systems must have a high throughput for being used in real time systems[9]. For such cases, chaos-based algorithms[10] have been considered as suitable ones.

Recently, many algorithms developed related to encryption by considering a characteristic of image data and its improvements such as circular random grids[11–14], vector quantization[15], SCAN[16,17], fractional wavelet transforms[18,19] and chaos[20,21].

H. Cheng and et al. in[22] proposed a tree structure-based method, that the quad tree structure was selected to be encrypted. The image encryption algorithm based on self-adaptive wave transmission that proposed in[23]

**Author for correspondence*

can encrypt image in parallel and be also applied to color image encryption. So far, for increasing the efficiency of image encryption methods, a variety of chaos-based image encryption algorithms were proposed[10,24–28]. With respect to developing chaos-based image encryption algorithms, few methods have been implemented in parallel by using GPGPU[29].

The main aim of this paper is to increase the execution speed of 1D chaos-based image encryption algorithm significantly by taking the advantages of algorithm parallelism and using the processing power of the GPU. In our advancement, the importance of security information, image encryption and encryption/decryption speed in modern applications is considered. Implementation results of the comparison show when a combination of CPU and GPU is used, the execution speed reached its best state.

The paper is organized as follows. In Section 2, improvement of the 1D chaos-based image encryption algorithm implementation is discussed and its methods are explained. Section 3, includes the simulation setup and implementation of the 1D chaos-based image encryption algorithm in MATLAB and CUDA C++ and the results of the test and the analyses have been investigated. Finally, we conclude in Section 4.

2. Encryption Algorithm Parallelization

Studies on the image encryption show the importance of parallelism in this area. Today's microprocessor development efforts, concentrate on using several cores rather than increasing single-thread performance. The highly parallel Graphic Processing Unit (GPU) is a powerful engine for computationally demanding image processing applications. Due to considerable differences in GPU architecture and programming model with the majority of other single-chip processors, GPU performance and potential offer a great deal in computing systems. The GPU is designed for a particular class of applications with large computational requirements, substantial parallelism and especially the applications that the throughput is more important than latency[30].

2.1 1D Chaos-based Image Encryption Algorithm

In this paper, we improve the implementation of the 1D chaos-based image encryption algorithm[10] that has a 4-round image encryption structure. As shown in Figure 1, each image encryption round includes five steps: inserting a random pixel in the beginning of each row, separating each row into a 1D data matrix, applying a substitution process to change data values, combining all 1D matrices back into a 2D data matrix and rotating the 2D matrix 90 degrees counterclockwise.

The more detailed routine of the algorithm is illustrated in the flowchart shown in Figure 2.

2.2 Parallel Implementation

The most important limitation in the implementation of the cryptography algorithms is their high execution time. In this work, the execution time of the 1D chaos-based image encryption algorithm is reduced and its performance is improved using both GPU and CPU. In order to use the parallelism power of CPU, the parallel computing toolbox of MATLAB provides very efficient methods for both Parallel Task Processing (PARFOR) and Parallel Data Processing (SPMD). As well, when CPU processing is combined with GPU processing, the execution time of the algorithm becomes much better.

2.2.1 Parallelism using Main Processor Cores

MATLAB is known as a simple instrument for scientific computing, engineering designs and simulations. The main feature of MATLAB is allowing the use of separate

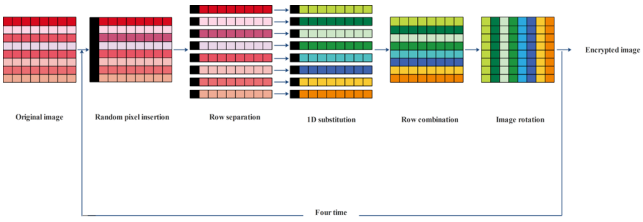

Figure 1. An example of the 1D chaos-based image encryption algorithm.

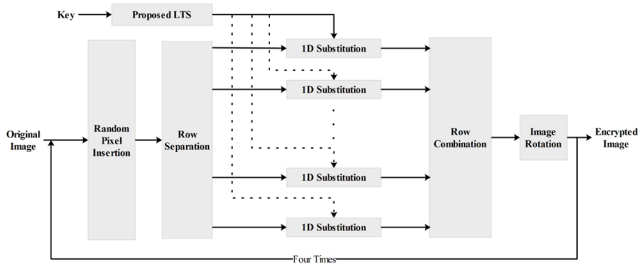

Figure 2. 1D chaos-based image encryption algorithm.

modules that can be ordered as a Toolbox. One of these packages is Parallel Computing Toolbox. The same toolbox can be used to run multiple threads independently on multiple cores of a computer. The simplest and most useful construct for parallel task processing in the toolbox is the PARFOR statement[29].

In the first step of 1D chaos-based image encryption algorithm, original image is recalled by the CPU and being converted to three separate matrices. After creating a Red (R), a Green (G) and a Blue (B) matrices, PARFOR replaces the regular "for" loops of the encryption algorithm and can be used when loop iterations are completely independent of each other. As shown in Figure 3, behind the scenes, MATLAB creates duplicates of the workspace for threads, runs a part of iterations per core until all iterations are done then gathers all results and returns them to the initial workspace.

The PARFOR method is easy to use, but it only lets us do parallelism in terms of loops. The just choice we make is whether a loop is to run in parallel. So, we can't determine how the loop iterations are divided up and which of the cores run any iteration. Furthermore, we can't examine the work of any individual core.

SPMD is another parallelism capability of MATLAB. In the SPMD (Single Program, Multiple Data) method, in order to obtain better results, multiple instances of the single program are performed on several different input data on multiple cores[31]. Each instance of the single program has a unique identifier, which is used to determine the partition of the program that will operate on. In high-performance computing, instances of an SPMD program have separate workspaces, but they can communicate directly with each other using Message Passing Interface (MPI) functionality via messages. MPI provides not only point-to-point communication but also collective operations. The implementation of one round of 1D chaos-based image encryption algorithm by using this method is shown in Figure 4.

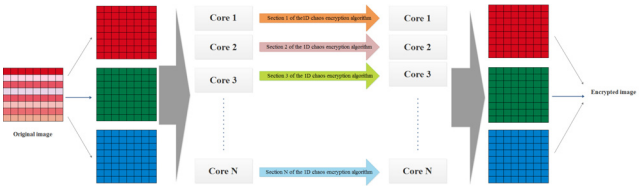

Figure 3. Implementation of one round of the 1D chaos-based image encryption algorithm using PARFOR.

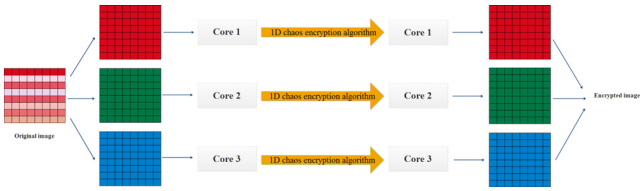

Figure 4. Implementation of one round of the 1D chaos-based image encryption algorithm using SPMD.

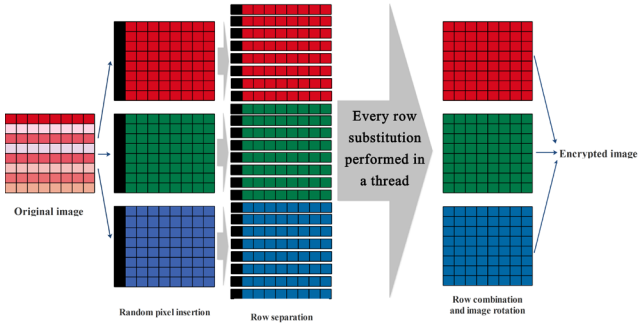

Figure 5. Implementation of one round of the 1D chaos-based image encryption algorithm using CUDA C++.

2.2.2 *Parallelism using GPU*

In this section, the parallel implementation of the 1D chaos-based image encryption algorithm on GPU is described. For this purpose, sequential partitions have been performed on CPU and parallel pieces have been executed on the GPU[32,33]. As shown in Figure 5, parallel portions of the algorithm are executed on the kernel devices where every kernel is run on each thread. As shown in Figure 1 the 1D chaos-based image encryption algorithm does substitution process row-by-row. That is, substitution process in each row is a serial process, but the substitution process of each row is independent of other rows. So, we assign each row to one thread. Every thread executes an instance of substitution algorithm and applies it to the row. For an image of size 128 × 128, for instance, we need 128 threads to execute the substitution algorithm on each row of image as mentioned.

3. Discussion

In this section, 1D chaos-based image encryption algorithm is tested and obtained experimental results are analyzed. Our experimental results have been conducted under MATLAB, in a computer with the Windows 8 operating system, Intel(R) Core (TM) i3-4030U CPU @ 1.90

GHz and 4 GB RAM with NVIDIA GeForce 820M graphic card. The computation capability of this graphic card is 2.1. Furthermore, in order to stimulate this research, the platforms of CUDA C++ 6.5 and Visual Studio have been used.

In order to compare the execution time of the 1D chaos-based image encryption algorithm, we implemented this algorithm by serial MATLAB, SPMD and PARFOR methods. The implementation results are shown in Table 1.

The results of Cuda C++ and Visual C++ implementation on the color images with different resolutions are shown in Figure 6 and Figure 7. The experimental results show that because of parallel implementation using Cuda C++, the simulation time for different pictures by Cuda C++ is almost three times faster than visual C++. Results of the Figure 6 and Figure 7 indicate that when the sizes of the images are increasing, the required time for encryption is grown up.

The results of serial and parallel implementation on the color images with different resolutions using MATLAB and its parallelism methods are shown in Figure 8 and Figure 9. Implementation results show when the size of the pictures increase, the performance of the 1D chaos-based image encryption algorithm in parallel implementation by the both parallel task "PARFOR" and data processing "SPMD" methods becomes better. Also Figure 8 and Figure 9 illustrate that the execution time when PARFOR scheme is used become better when the image size is higher than a threshold.

Furthermore, total results of the comparison figures show when a combination of CPU and GPU is used, the execution speed reached its best state.

Table 1. The implementation results using GPU and CPU

File name	Image size	MATLAB			Visual C++	
		Serial	SPMD	PARFOR	Serial	CUDA
Ball.jpg	64×64	0.084	0.033	0.772	0.001	0.111×10^{-3}
Flower.jpg	128×128	0.231	0.065	0.403	0.005	0.247×10^{-3}
Pens. jpg	256×256	0.876	0.204	0.660	0.028	0.839×10^{-3}
Stat. jpg	512×512	3.498	0.560	1.271	0.051	3.226×10^{-3}
Fruit. jpg	1024×1024	13.911	3.188	3.558	0.216	10.478×10^{-3}
Lena.jpg	2048×2048	56.367	25.354	15.924	0.997	42.488×10^{-3}

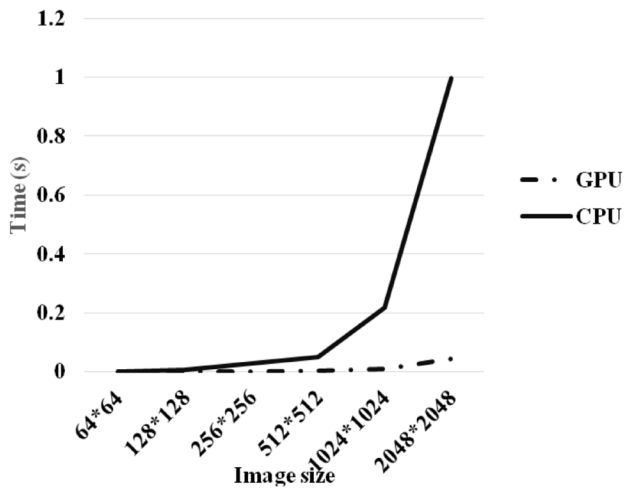

Figure 6. The simulation results for images with different resolutions by Cuda C++ C++ and Visual C++.

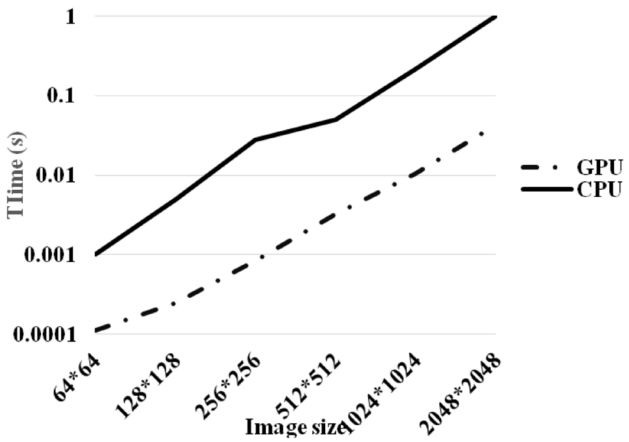

Figure 7. The simulation results for images with different resolutions by Cuda C++ C++ and Visual C++.

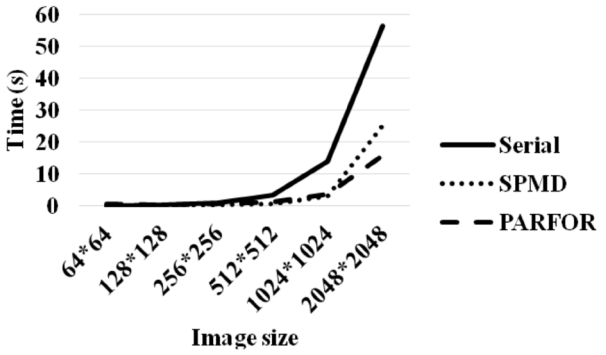

Figure 8. The simulation results for images with different resolutions by MATLAB and its parallelism methods.

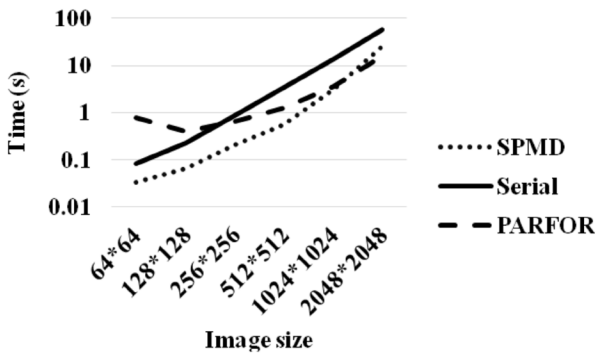

Figure 9. The simulation results for images with different resolutions by MATLAB and its parallelism methods.

4. Conclusion

This study set out to reduce the execution time of 1D chaos-based image encryption algorithm over color images. For this purpose, an algorithm has been implemented with different parallelization styles. Then execution time of all implementations on images with different sizes has been investigated. The parallelization styles included MATLAB based implementation using parallel task "PARFOR" and data processing "SPMD" and also C++ based implementation on CPU and GPU using Visual Studio and CUDA. This study has been found that the execution time of the 1D chaos-based image encryption algorithm in parallel implementations had considerable improvement for large colored images. Furthermore, the execution time of the algorithm has been improved 93.28% when a combination of CPU and GPU is used. That means, based on the parallel architecture of the graphic processing units, they are good choices to run image encryption algorithms.

5. References

1. El-Latif AAA, Li L, Wang N, Han Q, Niu X. A new approach to chaotic image encryption based on quantum chaotic system exploiting color spaces. Signal Processing. 2013 Nov; 93(11):2986–3000.
2. Hou YC, Quan ZY, Tsai CF, Tseng AY. Block-based progressive visual secret sharing. Information Sciences. 2013 Jun; 233:290–304.
3. Liao X, Lai S, Zhou Q. A novel image encryption algorithm based on self-adaptive wave transmission. Signal Processing. 2010 Sep; 90(9):2714–22.
4. Patel KD, Belani S. Image encryption using different techniques: A review. International Journal of Emerging Technology and Advanced Engineering. 2011 Nov; 1(1):30–4.
5. Coppersmith D. The Data Encryption Standard (DES) and its strength against attacks. IBM Journal of Research and Development. 1994 May; 38(3):243–50.
6. Lai X, Massey JL. A proposal for a new block encryption standard. In Advances in Cryptology - EUROCRYPT'90; 1991; 473:389–404.
7. Ranjan R, Poonguzhali I. VLSI Implementation of IDEA Encryption Algorithm. Mobile and Pervasive Computing (CoMPC–2008); 2008. p. 97–101.
8. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM. 1983 Jan; 26(1):96–9.
9. Kwok HS, Tang WK. A fast image encryption system based on chaotic maps with finite precision representation. Chaos, Solitons and Fractals. 2007 May; 32(4):1518–29.
10. Zhou Y, Bao L, Chen CP. A new 1D chaotic system for image encryption. Signal Processing. 2014 Apr; 97:172–82.
11. Shyu SJ. Image encryption by multiple random grids. Pattern Recognition. 2009 Jul; 42(7):1582–96.
12. Chen TH, Li KC. Multi-image encryption by circular random grids. Information Sciences. 2012 Apr; 189:255–65.
13. Roy S, Sadhukhan S, Sadhu S, Bandyopadhyay SK. A novel approach towards development of hybrid image Steganography using DNA sequences. Indian Journal of Science and Technology. 2015 Sep; 8(22):1–7.
14. Tamilselvi R, Ravindran G. Image encryption using pseudo random bit generator based on logistic maps with radon transform. Indian Journal of Science and Technology. 2015; 8(11):1–7.
15. Chen TH, Wu CS. Compression-unimpaired batch-image encryption combining vector quantization and index compression. Information Sciences. 2010 May; 180(9):1690–701.
16. Chen RJ, Horng SJ. Novel SCAN-CA-based image security system using SCAN and 2-D von Neumann cellular automata. Signal Processing: Image Communication. 2010 Jul; 25(6):413–26.
17. Maniccam SS, Bourbakis NG. Image and video encryption using SCAN patterns. Pattern Recognition. 2004 Apr; 37(4):725–37.
18. Li X, Chen T, Xinjun Z, Biyuan L, Linlin W, Xiusheng Y. Image/video encryption using single shot digital holography. Optics Communications. 2015 May; 342:218–23.
19. Bhatnagar G, Wu QJ, Raman B. Discrete fractional wavelet transform and its application to multiple encryption. Information Sciences. 2013 Feb; 223:297–316.
20. Chen CK, Lin CL, Chiang CT, Lin SL. Personalized information encryption using ECG signals with chaotic functions. Information Sciences. 2012 Jun; 193:125–40.

21. Bao L, Zhou Y, Chen CP, Liu H. A new chaotic system for image encryption. 2012 International Conference on System Science and Engineering (ICSSE); Dalian, Liaoning. 2012 Jun 30-Jul 2. p. 69–73.
22. Cheng H, Li X. Partial encryption of compressed images and videos. IEEE Transactions on Signal Processing. 2000 Aug; 48(8):2439–51.
23. Liao X, Lai S, Zhou Q. A novel image encryption algorithm based on self-adaptive wave transmission. Signal Processing. 2010 Sep; 90(9):2714–22.
24. El-Latif AAA, Li L, Wang N, Niu X. Image encryption scheme of pixel bit based on combination of chaotic systems. 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP); Dalian. 2011 Oct 14-16. p. 369–73.
25. Liao X, Lai S, Zhou Q. A novel image encryption algorithm based on self-adaptive wave transmission. Signal Processing. 2010 Sep; 90(9):2714–22.
26. Yeo JC, Guo JI. Efficient hierarchical chaotic image encryption algorithm and its VLSI realization. IEE Proceedings Vision, Image and Signal Processing. 2000 Apr; 147(2):167–75.
27. Tong X, Cui M. Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator. Signal Processing. 2009 Apr; 89(4):480–91.
28. Sathishkumar GA, Sriraam DN. Image encryption based on diffusion and multiple chaotic maps. Computer Science. 2011 Mar; 3(2):181–94.
29. Suh JW, Kim Y. Accelerating MATLAB with GPU computing: A primer with examples. Computer Science; 2013 Dec. p. 258.
30. Owens JD, Houston M, Luebke D, Green S, Stone JE, Phillips JC. GPU computing. Proceedings of the IEEE. 2008; 96(5):879–99.
31. Darema F. The SPMD model: Past, present and future. In Recent Advances in Parallel Virtual Machine and Message Passing Interface, Berlin Heidelberg: Springer; 2001. p. 1.
32. Zoric DP, Olcan D, Kolundzija BM. Solving electrically large electrodynamic problems using graphics processing units. Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP); Rome. 2011 Apr 11-15. p. 2263–7.
33. CUDA C++ homepage on NVIDIA. (Online). 2011. Available from: http://www.nvidia.com/object/Cuda C++_home_new.html