HIGH-SPEED IMPLEMENTATIONS OF FRACTAL IMAGE COMPRESSION FOR LOW AND HIGH RESOLUTION IMAGES

by

ABDUL-MALIK HAIDER YUSEF SAAD

Thesis submitted in fulfilment of the
requirements for the degree of
Doctor of Philosophy

February 2018

ACKNOWLEDGEMENT

First and above all, I thank the Almighty Allah for his infinite blessings.

Next, I would like to express my sincere gratitude to my Supervisor **Prof. Mohd Zaid Abdullah** for the continuous support during my PhD study and research. I thank him also for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me to do my research and write this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

Last but not the least, I would like to thank my parents, Haider and Huda, my wife Ilham, and my brothers and sisters for their sincere prayers, unconditional love, and never-ending support over past years.

TABLE OF CONTENTS

LIST OF PUBLICATIONS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AGU	Address Generation Unit
ASIC	Application-Specific-Integrated-Circuit
CLB	Configurable Logic Blocks
CPU	Central Processing Unit
CR	Compression Rate
DSP	Digital-Signal-Processor
EA	Error Adder
FC	Fractal Code
FCIC	Fractal Colour Image Compression
FCM	Fast Comparison Module
FF	Flip Flop
FHCBC	Fractal Hierarchical Colour Block Coding
FIC	Fractal Image Compression
FPGA	Field-Programmable-Gate-Array
Fps	Frames Per Second
FSM	Finite State Machine
GPP	General Purpose Processor
GPU	Graphic Processor Unit
HV	Horizontal/Vertical
I/O	Input And Output
IFS	Iterated Function Systems
IP	Intellectual Property
LE	Logic Element

LSB	Least Significant Bits
LUT	Look-Up-Table
MCDCU	Mean And Contracted Domain Computation Unit
MCDCU-Ctrl	Mean And Contracted Domain Computation Control Unit
MCU	Memory Control Unit
MSE	Mean Square Error
OCU	Offset Computation Unit
PC	Personal Computer
PE	Processing Element
PG	Primary Group
PIFS	Partitioned Iterated Function Systems
PSNR	Peak-Signal-To-Noise-Ratio
RAM	Random Access Memory
RTL	Register Transfer Level
SAD	Sum Of Absolute Differences
SADCU	Sum Of Absolute Differences Computation Unit
SADCU-Ctrl	Sum Of Absolute Differences Computation Control Unit
SCtrl	Storing Control Unit
SCU	Sum Of Absolute Differences Computation Unit
SFC	Separated Fractal Coding
SG	Secondary Group
SOC/SoC	System-On-Chip
SRAM	Static Random Access Memory
VHDL	Very High Speed Integrated Circuit Hardware Description Language

PELAKSANAAN PEMAMPATAN IMEJ FRAKTAL BERKELAJUAN TINGGI UNTUK IMEJ BERESOLUSI RENDAH DAN TINGGI

ABSTRAK

Pemampatan Imej Fraktal (PIF) adalah teknik pengekodan yang sangat terkenal digunakan dalam aplikasi berkaitan imej/video disebabkan keringkasannya dan prestasinya yang tinggi. Walau bagaimanapun, kelemahan besar PIF adalah dari segi algoritmanya yang mengambil masa panjang, terutamanya apabila pencarian-penuh dibuat. Oleh itu, pencapaian operasi masa nyata adalah sangat mencabar terutamanya apabila algoritma ini dioperasikan pada pemproses biasa mahupun grafik. Oleh itu, penyelidikan ini mencadangkan perlaksanaan perkakasan baharu bagi mempercepatkan proses nyahkod menggunakan keselarian dan penalian paip. Pelbagai pendekatan telah dikaji untuk mencapai prestasi berkelajuan tinggi. Sebagai permulaan, pengiraan kompleks operasi fraktal dikaji bagi memilih saiz bit yang minimum dan efisien yang memberi hasil kualiti mengekod sama atau hampir sama. Ini menghasilkan perkakasan PIF yang agak baharu dan dikenali dalam tesis ini sebagai Rekaan I (Design I). Dalam rekaan ini, pendekatan pencarian-penuh diguna pakai untuk membolehkan penjanaan semula pada kadar kualiti tertinggi. Rekaan ini sesuai untuk mengekod imej beresolusi rendah memandangkan masa pengekodan meningkat secara eksponen apabila memproses imej yang beresolusi tinggi. Masalah ini telah diselesaikan melalui Rekaan II (Design II) yang menggunakan dasar skema pencarian-separa untuk mencapai operasi masa nyata. Kaedah ini mengeksplotasikan kewujudan hubung kait yang tinggi antara piksel di sekitar kawasan berdekatan dalam imej digital, seterusnya merangkumkan ruang pencarian pada kawasan tersebut sahaja. Dengan menetapkan ruangan-ruangan ini untuk setiap kumpulan lingkungan blok dan membahagikan imej di mana setiap domain blok mengandungi empat lingkungan blok, membolehkan dua operasi pemadanan dilakukan serentak. Ini mengurangkan capaian memori sehingga separuh dan

seterusnya menyebabkan kelajuan meningkat dua kali ganda. Rekaan ini telah diperluaskan untuk mengekod imej RGB, menghasilkan satu lagi rekaan baharu dikenali; sebagai Rekaan III (Design III). Dalam rekaan ini, hubung kait silang yang kuat antara komponen imej dieksploitasikan menyebabkan hanya komponen *G* dikod menggunakan pendekatan sama seperti dalam rekaan II, sementara komponen *R* dan *B* dikod menggunakan skema dasar tanpa pencarian dengan pemetaan terus antara blok yang bertindih. Ketiga-tiga rekaan telah diuji dari segi masa jalan, nisbah puncak isyarat ke hingar (PSNR) dan kadar pemampatan. Keputusan eksperimen Rekaan I apabila dilaksanakan dalam Altera Cyclone II FPGA, menunjukkan kenaikan halaju dengan purata 3 kali ganda, sementara PSNR tidak menunjukkan perubahan besar. Keputusan empirik menunjukkan bahawa perisian tegar ini adalah setara apabila dibandingkan dengan pencarian-penuh perkakasan lain dengan PSNR purata 30 dB, kadar pemampatan 5.82% dan masa jalan 9.8 ms. Manakala itu, Rekaan II telah disintesiskan pada Altera Stratix IV FPGA dan menunjukkan kebolehan untuk mengekod resolusi imej 1024×1024 pada 395 MHz dalam 10.8 ms dengan PSNR purata 27 dB dan kadar pemampatan 34. Keputusan ini menunjukkan bahawa pendekatan yang dicadangkan membolehkan imej berwarna dikod pada halaju yang hampir sama seperti imej skala kelabu. Selain itu, senibina yang dicadangkan boleh mencapai prestasi lebih baik jika dibandingkan dengan rekaan canggih, dengan purata halaju sebanyak 100, 92 dan 83 fps untuk Rekaan I, II dan III masing-masingnya.

HIGH-SPEED IMPLEMENTATIONS OF FRACTAL IMAGE COMPRESSION FOR LOW AND HIGH RESOLUTION IMAGES

ABSTRACT

Fractal Image Compression (FIC) is a very popular coding technique that is used in image/video applications due to its simplicity and superior performance. The major drawback of FIC is that it is a time consuming algorithm, especially when a full search is attempted. Hence, it is very challenging to achieve a real-time operation especially when this algorithm is run on a general or graphic processor unit. Therefore, in this research new hardware implementations of FIC are proposed for accelerating the encoding process by means of parallelism and pipelining. Various approaches have been investigated for achieving high speed performance. The computational complexity of fractal operations are first investigated in order to select the minimum and efficient bit sizes that can provide similar or nearly similar encoding quality. This has resulted in a relatively new FIC hardware which is referred in this thesis as Design I. In this design, a full-search approach was adopted in order to enable reconstruction at highest possible quality. However, full-search scheme is not suitable for encoding larger images since the encoding time is increased dramatically when processing high-resolution images. This problem is solved in Design II which used a partial-search based scheme in order to achieve high-speed operation. This method exploits the inherently high degree of correlation between pixels in the neighbourhood areas in digital image to restrict the search space to those areas. By fixing these areas for each group of range blocks and partitioning an image in which each domain block contains four range blocks, enabled two matching operations be performed simultaneously. This reduced the

memory access by half, thereby, doubling the speed by a factor of 2. This design was extended to encode RGB image, resulting in another new design referred to as Design III. In this design, the strong cross-correlation between the image components was exploited so that only the *G* component was encoded using the same approach as in Design II, while the *R* and *B* components were encoded by searchless-based scheme with direct mapping between overlapped blocks. All three designs were examined in terms of runtime, peak-signal-to-noise-ratio (PSNR), and compression rate. The experimental results of Design I when implemented in Altera Cyclone II FPGA, showed speedup of 3 times, on average, while the PSNR was not significantly affected. Empirical results demonstrated that this firmware is competitive when compared to other existing full-search hardware with PSNR averaging at 30 dB, 5.82 % compression rate and a runtime of 9.8 ms. On the other hand, Design II was synthesised on Altera Stratix IV FPGA and showed an ability to encode a 1024×1024 image at 395 MHz in 10.8 ms with PSNR averaging at 27 dB and compression rate of 34. These results suggest that the proposed approach enables colour images be encoded at approximately same speed as grayscale images. Also the proposed architectures have achieved better performance compared to the state-of-the-art designs, with speed averaging at 100, 92 and 83 fps for Design I, II and III respectively.

CHAPTER ONE

INTRODUCTION

1.1 Introduction

In recent years, images are found everywhere and become an essential element in our daily lives. Images are produced intensively every single day in various forms such as medical images, personal images, social media images, surveillance images, and graphics images. Storing these images or transferring them through the network creates an unbearable burden for memory devices and the network bandwidth too. Taking this into account, compression is an indispensable tool for archiving images in fewer spaces of storage and also for transferring images in less time of transmission. Furthermore, compression can save the network bandwidth efficiently.

Theoretically, image compression is the process of encoding the image information using bits that are fewer than the original representation (Mahdi et al., 2012). In general, the compression process can be either lossless or lossy. The lossless compression attaches the utmost importance to the entire image data. Thus, the image data must be compressed in a way that it can be fully recovered without any loss (i.e. the decompressed image must match the original image) (Pujar and Kadlaskar, 2010). On the other hand, the lossy compression attaches more importance to the compression rate and consequentially accepts some loss in the data. Unlike other types of data (such as text data), image data, under a certain degree of compression, can be compressed with lossy technique without any visual effects (i.e., visually lossless appear). Yet, the image still can maintain its main

features even for higher rate of compression. Thus, the most well-known image compression techniques are lossy-based.

Compared to the existing lossy image compression techniques, Fractal Image Compression (FIC) is one of the most popular technique due to its distinctive way in compressing the images. This compression technique uses the self-similarity features as a means to compress an image (Fisher, 1995). It also offers a high compression ratio (CR), especially when applied to a digital image with a high degree of self-similarity like aerial photography or satellite imagery (Jacquin, 1992). Owing to its popularity in digital archiving, FIC has been found in numerous applications such as character recognition (Mozaffari et al., 2005), watermarking (Kiani and Moghaddam, 2011; Wu and Chang, 2003), and digital signature embedding (Puate and Jordan, 1997). FIC also possesses other attractive features like good peak signal-to-noise ratio (PSNR) performance and simple decoding method (Wohlberg and De Jager, 1999; Polvere and Nappi, 2000; Martin and Curtis, 2013). However, FIC suffers one major drawback arising from the computational complexity of the algorithm. Typically, FIC requires a very large number of searches in order to find excellent or good-enough maps between image blocks. Theoretically, the time complexity of the fractal compression algorithm approximately equals to $O(n^4)$ for $n \times n$ size image (Salarian et al., 2015; Liu et al., 2017). Therefore, a 256×256 image requires a $256^4 = 2^{32} \cong 4$ billion number of comparisons in order to be fully encoded; each comparison requires hundreds of mathematical operations. Owing to this, the encoding time required for FIC is generally measured in terms of minutes and sometimes in hours. For example, the runtime reported in the work by Dhawan (2011) for encoding a single image of 256×256 pixels size is 5.6 hours. Moreover, this complexity increases dramatically when targeting larger image sizes.

1.2 Problem Statement

To-date various methods have been proposed for accelerating the fractal algorithm. These methods have mainly focused on reducing the size of the range and domain pool, adoption of classification approach (Kovács, 2008), restriction of search space (Truong et al., 2004), combination of different code schemes (Curtis et al., 2002), and application of feature vector approach (Saupe, 1995). For instance, Bani-Eqbal (1995) arranged the domain blocks in tree structure to reduce the number of candidate blocks for matching search. In an earlier study, Saupe and Hamzaoui (1994) discarded the domain blocks that have low variance. Tong and Pi (2001) presented an adaptive search for excluding the domain blocks that do not satisfy the necessary condition of better matching. Meanwhile, Tong and Wong (2002) converted the matching search problem to a nearest neighbour search problem. Although these schemes are successful in speeding-up the FIC, the attained encoding times are still too slow and not adequate to deliver real-time implementation. This is partly due to the fact that these algorithms are developed on general purpose processors (GPPs) which execute the operations sequentially.

Compared to those implemented in GPPs, the number of other types of implementations that have targeted another image processing platforms, is relatively less. For instance, few researchers have targeted DSP processors to gain some speed (Vidya et al., 2000; Thai et al., 2011) through utilising the hardware built-in units which are used specifically for executing repetitive and numerically intensive operations. The major outcome of these studies is the improvement of the runtime performance. However, the speed-up gains are still small because of DSP hardware architecture lacks the flexibility.

Another attempt is the use of graphic processor units (GPUs). This processor type is targeted because it comprises hundreds or thousands of small processing units that can be used for parallel-data processing. As FIC algorithm is inherently parallel, GPU processors can attain performance that is higher than the performance of GPP and DSP. Taking this into account, Erra (2005) and Wakatani (2012) implemented fractal image compression on GPU for fast pairing search and achieved a speedup of more than 70 times. Despite their attempts, however, the encoding time is still more than one second, which does not suit most of real-time applications. Similar to DSP, the speed-up limitation of GPU can be attributed to the lack of flexibility in the architecture. Furthermore, the other drawback of GPUs is the high power consumption which makes them unsuitable choice for power-sensitive applications.

From the above arguments, it can be concluded that GPP, DSP and GPU processors cannot meet the real-time requirement for FIC implementation. This can be attributed specifically to the architecture inflexibility. Thus, a custom hardware design of FIC is proposed to overcome the real-time problem.

Some recent works targeting FIC implementation in hardware have been published (Acken et al., 1998; Panigrahy et al., 2015; Vidya et al., 2000; Samavi et al., 2010; Jackson et al., 2007; Son et al., 2012). Some of these designs are based on a full-search scheme that searches the entire image for the best match for a given range block (Vidya et al., 2000; Panigrahy et al., 2015; Acken et al., 1998; Liang and Wang, 2005). However, the encoding time of 256×256 grayscale image using this search scheme is still large even with the use of massive parallelism in the architecture (Acken et al., 1998). The best encoding time reported by these authors is approximately 82 ms which is equivalent to 12 fps.

Therefore, this current research aims to develop a full-search based FIC architecture that can provide real-time performance of more than 50/60 fps of 256×256 pixels size each. To design such a real-time design of FIC, the complexity involved in the fractal computations needs to be reduced. In order to do that, the most complex operations need to be determined first and their bit-width size should be reduced while maintaining the image quality.

Since there is a high demand for coding high resolution image, another high-speed design is proposed for coding 1024×1024 grayscale images. To the best of the researcher's knowledge, the largest image size that has been coded by hardware is 512×512 pixels size. This image size was targeted in the designs of Acken et al. (1998) and Jackson et al. (2007). In fact, the design of Acken et al. (1998) is based on full search scheme, while the design proposed by Jackson et al. (2007) adopted a searchless strategy. Among the search types, the full-search scheme is the most time-consuming scheme. However, it yields higher image quality. On the other hand, the searchless scheme is considered to have the lowest time consumption, but it generally yields lower compression rate and image quality. As a compromise, partial search technique is adopted in the proposed architecture to overcome the time complexity resulting from coding high resolution image and to maintain the image fidelity at acceptable level. Thus, the adopted search method is chosen to search the similarity in the neighbourhood area instead of the entire imaging space.

With the wide spread of colour images nowadays, FIC technique should not be limited in compressing grayscale images only. To the best of the researcher's knowledge, so far there has been no such hardware design proposed for compressing colour image through the employment of the fractal method. Principally this is due to

the fact that colour images comprise of at least red, green and blue components compared to single component as in achromatic or grayscale images. Hence, designing a hardware to process colour images is much more complicated compared to designing a hardware for processing grayscale images. One simplest approach is to treat each colour component individually and perform compression component-by-component basis. However, for three colour planes in an image, the encoding time becomes threefold. Consequently, the encoding time problem becomes more critical. Furthermore, this technique does not improve the compression efficiency and the image quality. This is because the strong correlation between colour image planes are not exploited. Taking this into consideration, this current study aims to develop a new hardware design of FIC targeting colour images. As a result of utilising the strong correlation between the colour planes in the compression, it is possible to deliver high-speed compression system, larger compression rate and better image quality.

1.3 Research Objectives

In general, this study aims to develop high-speed architectures of fractal image compression, targeting grayscale and colour images. Both low and high resolution images are considered. Thus, the research objectives of this study are:

1. To reduce the computational complexity of fractal operations via bit-width optimisation method.
2. To design a high-speed architecture for implementing full-search FIC.
3. To develop a high-speed architecture for encoding high-resolution images.

4. To develop an efficient-time architecture for encoding colour image by utilising the inherently high correlation between colour components.

1.4 Contributions of the Research

This study aims to develop high-speed implementations of fractal image compression, in which the time performance is more than 60 frames-per-second. Generally, the objectives of this study are to explore the existing implementations and to find efficient implementations for low and high resolution images and also for grayscale and colour images. Within this context, this study has achieved four new contributions which are:

1. **Bit-width optimisation:** The computational complexity of fractal operations has been investigated for the purpose of decreasing it. Here, the bit-width sizes of the selective fractal operations are reduced into lower sizes at the same time ensuring the PSNR is maintained at an acceptable level. The operations required for computing scale parameter are selected and then implemented at lower precision. Thus, the subtraction and division operations are reduced from 24-bit precision to 10-bit and 5-bit respectively. As a result, the runtime and LEs utilization for *s* computation are significantly improved while maintaining the PSNR value.
2. **High-speed full-search compression:** In this study, a new parallel hardware architecture for full-search FIC was developed. This design is referred to in this study as Design I. This architecture targets low resolution grayscale images of size 256×256 pixels. In order to reduce the encoding time, three different methods have been investigated. The first one is the optimisation between arithmetical precision and image quality. Reducing memory access constitutes the second method, while

the third (final) one is the parallel computation. The proposed architecture outperformed the state-of-the-art methods of full-search FIC in terms of runtime.

3. **High-speed compression for high resolution image:** In this study, a new high-speed FIC design was proposed for coding a high-resolution image of 1024×1024 pixels size. This high-speed FIC design is referred to as Design II. To overcome the time complexity resulting from coding a big image size, the proposed architecture was designed for searching the similarity in the neighbourhood area instead of the entire imaging space. In this way, a significant reduction is achieved in both memory access and runtime. Also, an image is partitioned in such a way that the size of each range block is exactly one quarter of a domain block. In other words, each domain block contains four range blocks. Thus, fetching a domain block also means acquiring four range blocks simultaneously, or vice versa. As a result, two matching operations can be performed in parallel. This leads to a 50% reduction in memory access, and hence increases the speed-up by the same factor. Further improvement is achieved by parallel implementation and deep data pipelining. Finally, the design is optimised at circuit level to achieve high-frequency speed and low utilisation of logic elements.
4. **High-speed colour image compression:** A new high-speed design was proposed in this study for coding a high-resolution colour image of 1024×1024 pixels size. This new design is referred to as Design III. To overcome the threefold time complexity resulting from coding colour image components individually, the proposed architecture was designed to exploit the cross-correlation between colour components. Both search and searchless based schemes have been investigated. As a result of encoding two of three components using the proposed searchless scheme, the runtime is nearly one third of that using grayscale-based designs. Furthermore, it