FPGA Implementation for Fringe Pattern Demodulation Using the One-Dimensional Modified Morlet Wavelet Transform

Abdulbasit Abid
Umm Al-Qura University, College of Computer and Information Systems, Computer Engineering Department, P.O. Box: 715 Makkah, Saudi Arabia

Abstract—The novel FPGA implementation for fringe pattern demodulation using the one-dimensional modified Morlet wavelet transform algorithm is presented. Firstly, the modified Morlet wavelet transform is programmed using the C programming language and compiled into VHDL using the ImpulseC tool. Then this VHDL code is implemented on the Altera Cyclone IV GX EP4CGX150DF31C7 FPGA. A fringe pattern image with the size of 512 × 512 pixels is presented to the FPGA, which processed the image using the modified Morlet algorithm. The FPAG requires about 100 milliseconds to process the image and produce a wrapped phase map. The modified Morlet wavelet is more suitable to phase-demodulate fringe patterns than the frequently used Morlet wavelet transform; since the modified Morlet has better spatial localization than the Morlet wavelet transform. For performance comparison purposes, the modified Morlet wavelet algorithm is programmed using the C language. The C code is then compiled using the Intel compiler version 13.0. The compiled code is run on a Dell Precision state-of-the-art workstation. The time required to process the fringe pattern image is one second approximately. In order to further reduce the execution time, the modified Morlet wavelet is reprogrammed using the Intel Integrated Primitive Performance (IPP) Library Version 7.1. The execution time was reduced to 600 milliseconds approximately. This confirms that at least six folds, approximately, speed up was gained using the FPGA implementation over a state-of-the-art workstation that executes heavily optimized implementation of the modified Morlet wavelet algorithm.

Index Terms—Fringe analysis, Phase demodulation, Wavelet, FPGA.

I. INTRODUCTION

Fringe pattern demodulation using digital computers has recently seen significant interest due to its widespread application in science, medicine and industry [1, 2]. Many methods for retrieving phase information from fringe patterns have been researched in considerable depth and some have now reached a mature state. For example, using phase stepping or phase shifting (PS) algorithms [3], Fourier transform profilometry (FTP) [4], More recently windowed Fourier transform (WFT) profilometry [5] and wavelet transform profilometry (WTP) [6 - 10], have all been the topics of active research.

There are many mother wavelets that can be used to extract the phase of a fringe pattern, for example, Morlet, Gaussian, b-spline, Shannon and Paul [8]. The most frequently used mother wavelet is the Morlet since it has the best combined frequency and spatial localization properties. The traditional complex Morlet wavelet is a sine wave modulated by a Gaussian function [10], and is defined as

\[ \psi(x) = \pi^{1/4} \exp(i \nu x) \exp(-m x^2) \]  

(1)

Where \( c \) is a fixed spatial frequency and chosen to be about 5 to 6 to satisfy the admissibility condition and \( x \) is the index to pixels in the x direction. The value of the parameter \( m \) is set to 1. Fig. 1 shows the real part (dashed line) and the imaginary part (solid line) of the Morlet wavelet.

Fig. 1. The traditional complex Morlet wavelet

The spatial localization property can be improved at the expense of the frequency localisation by setting the \( m \) parameter to be less than one. The resultant mother wavelet is designated as modified Morlet wavelet transforms [10]. Experimental results have shown that setting the value of the parameter \( m \) to 0.5 allow the WTP algorithm to demodulate fringe patterns with high bandwidth and rapid phase changes in comparison with the Morlet wavelet transform [10]. Fig. 2 shows the real part (dashed line) and the imaginary part (solid line) of the modified Morlet wavelet with \( m \) set to 0.5.

Fig. 2. The Modified Complex Morlet Wavelet

Extracting the phase of a fringe pattern using the one-dimensional modified Morlet wavelet has many potential advantages in extracting the phase of a fringe pattern over the Fourier transform and the windowed Fourier transform.
methods [8]. But the execution time of the wavelet method is greater than the other two methods. This long execution time is considered to be as an obstacle that prevents the wide use of the wavelet transform profilometry and it will be an advantageous if this execution time is reduced. In an effort to reduce the execution time of the one-dimensional modified Morlet wavelet algorithm, it is programmed using the C programming language. We have published the C source code of this algorithm on the Internet for the benefit of researchers in this area [11]. The C code is compiled using the Intel compiler version 13. The measured execution time to process a fringe pattern image with the size of 512 x 512 pixels is one second approximately. The C code is executed on a Dell Precision T7600 workstation with Dual Eight Core XEON with clock speed of 2.7 GHz. The workstation has 4 GByte RAM. In an effort to further reduce the execution time of the one-dimensional modified Morlet wavelet algorithm, it is reprogrammed using the C programming language and Intel integrated primitive performance library (IPP) version 7.1. The algorithm is then run on the Dell workstation. The execution time is reduced to 650 milliseconds approximately. This execution time can be considered long for many practical applications and it is advantageous if it can be reduced further. The execution time of the one-dimensional modified Morlet wavelet algorithm can be reduced further by using FPGA and this novel implementation will be discussed in detail in the next section.

II. FPGA IMPLEMENTATION

In this paper, we propose the use of FPGA to improve the computational performance of the modified Morlet wavelet transform algorithm and the use of this algorithm to extract the phase of a fringe pattern. In order to implement the modified Morlet wavelet transform algorithm on an FPGA, this wavelet technique should be programmed using VHDL and this is not an easy task to do. In order to avoid this obstacle, the author has used the ImpulseC compiler [12]. This tool imposes restrictions on the C language such as the use of pointers is prohibited. The code in [11] is reprogrammed according to these limitations and then compiled into VHDL. There are two approaches to calculate the 1D-CWT algorithms: time domain and frequency domain. The frequency domain method is considerably faster and it uses the discrete Fourier transform (DFT) or fast Fourier transforms (FFT). We have used the DFT/FFT to calculate the 1D-CWT algorithms. The VHDL program is then implemented on the Altera Cyclone IV GX EP4CGX150DF31C7 FPGA [13]. This FPGA has 149,760 logic elements, 6,635,520 memory bits and 720 embedded multiplier 9-bit elements. The rich combination of logic, memory, and digital signal processing (DSP) make this FPGA suitable for the intense DSP applications like the design addressed in this paper. Altera Cyclone IV GX FPGA development board was used, shown in Fig. 1, as the hardware platform for the wavelet transform design. This development board has on-Board memory: of 4-MB (×16) synchronous static random access memory (SSRAM), two 32-MB (×32) DDR2 SDRAM, 64-MB flash and on-board clocking circuitry: 50.000-MHz oscillator, 125.000-MHz oscillator, programmable oscillator (default: 100.000-MHz). The SSSRAM memory was used due its fast speed transfer suitable for moving image contents for processing using the wavelet design. All the calculations have been carried out using 32 bits float data type in both the FPGA system and the Dell workstation. This is to ensure that no reduction in precision occurs when using the FPGA system in comparison with the C code executed using the Dell workstation.

The FPGA was connected to a PC using Joint Test Action Group (JTAG) interconnectivity. The whole fringe pattern image, shown in Fig. 2, is downloaded to the FPGA’s external SSSRAM, shown in Fig. 3. The wavelet transform design was linked to a NIOS II processor with two DMA controllers to write and read both to and from the input and output buffers of the wavelet transform core and the external SSSRAM. The overall design is shown in the diagram in Fig. 3. The NIOS processor triggers an image DMA transfer (32-bit float 512 x 512 pixels) from the external SSSRAM to the internal 1MByte input buffer, which is then read and processed by the wavelet core. Next the result is written into the 1MByte output buffer, which as a result triggers another DMA transfer from the output buffer to the SSSRAM. Once the entire fringe pattern image is processed, the resultant wrapped phase map is transferred back to the PC. The output wrapped phase map that is produced by the FPGA for the input fringe pattern that was shown in Fig. 2 is presented in Fig. 4.

![Fig. 1. Altera Cyclone IV GX FPGA development board.](image-url)
The design was compiled using the Altera Quartus II Compiler in order to achieve both an optimal design footprint on the FPGA and also fast processing speeds. The design used almost 92% of the FPGA logic elements, 65% of the FPGA on-chip memory and 80% of the FPGA 9-bit embedded multiplier elements. The FPGA design footprint on the FPGA die is shown in Fig. 5. The darker blue areas represent used logic gates and the lighter blue areas represent unused logic gates of the FPGA.

Fig. 2. A Real Fringe Pattern Image.

Fig. 3. FPGA System Design.

Fig. 4. FPGA Generated Wrapped Phase Image.

Fig. 5. The FPGA Design Footprint upon the FPGA Die

III. CONCLUSION

A significant effort has been made here to reduce the execution time of the one-dimensional continuous modified Morlet wavelet transform profilometry algorithm. Initially, the algorithm was programmed using the C language and implemented upon a state-of-the-art Dell Precision workstation using the very powerful Intel compiler. This was enhanced still further using the Intel integrated primitives performance library (IPP). Both of these tools produce very optimized object code. The time required to execute the modified Morlet algorithm for a typical 512 x 512 pixel source image when using this arrangement was approximately 60 milliseconds. In order to further reduce the execution time, the modified Morlet algorithm was reprogrammed using the C language and following the requirements necessitated by Impulse C. Impulse C was then used to compile this C program and produce output code in VHDL format. The VHDL code may subsequently be implemented upon a specific FPGA hardware device, namely an Altera Cyclone IV GX FPGA. Finally, a typical real 512×512 pixel fringe pattern image was processed using the FPGA approach and here the total measured execution time was reduced to approximately 100 milliseconds. This time includes the communications overhead between the PC and the FPGA. A speed up of approximately six folds has therefore been achieved by using this hardware implementation when compared to heavily optimized conventional C code. This type of hardware acceleration makes the use of Wavelet Transform Profilometry algorithms viable for many practical applications that require higher measurement quality than is possible using established methods such as Fourier Transform Profilometry.

IV. ACKNOWLEDGMENT

The author would like to acknowledge King Abdulaiz City for Science and Technology (KACST) for the funding of this project.

REFERENCES


AUTHOR’S PROFILE

The Author is an assistant Professor in the Computer Engineering Department at Umm Al-Qura University, Makkah, Saudi Arabia. He has been awarded a Doctor of Philosophy in Image Processing from Liverpool John Moores University, UK in 2008. He has studied Master of Science in Computer and Information Networks at University of Essex, UK in 2004. He has also studied Master of Science in Electrical Engineering at Colorado State University, USA in 1995. He has gained Bachelor in Electrical and Computer Engineering from Umm Al-Qura University, Makkah, KSA. Currently he is the vice dean of Computer Engineering Department at Umm Al-Qura University. His research interests are fringe pattern analysis, wavelet transform and FPGA.