Fourier Transform Based GPU Acceleration

Jingshen Li

Chengdu Aeronautic Vocational and Technical College
Chengdu, China
lijingsen2012@yeah.net

Abstract—**In digital image processing, Fourier transform is an important algorithm of image transformation. In order to improve the speed of Fourier transform, the paper proposes to deal with the image with GPU parallel computing through the method of GPU accelerating MATLB. The relationship of data scale and calculation speed is analyzed through the traditional CPU serial operation and GPU parallel computing. Computer simulations verify that the calculation speed can be improved by GPU about large scale data.**

Keywords-GPU; Fourier Transform; Parallel Computing

I. INTRODUCTION

In digital image processing, the Fu Liye transform is mainly the image from the spatial domain into the frequency domain, the inverse transform is mainly the image from the frequency domain is transformed into the space domain, time domain and frequency domain by the switching of image, image features extraction and analysis, simplified calculation workload, it is widely used in image change, image coding and compression, image segmentation, image denoising, image reconstruction etc.. The discrete Fu Liye transform the large amount of computation, the computation time is longer, and the fast Fu Liye transform will Fu Liye change the complexity of the algorithm is reduced, although to a certain extent, can reduce the amount of computation, but the calculation of large-scale data still need to consume a lot of time, especially the use of MATLB in the treatment of massive data of low speed, low efficiency, because CPU a wide variety of tasks, in the face of increasingly complex data serial calculation requires a lot of time, GPU graphics processor core, along with the development of technology, GPU in floating point calculation has made rapid development, GPU computing model in heterogeneous collaborative processing calculation model, CPU and GPU combined utilization. The application of serial sections of running in the CPU, and the computing task segments from GPU to accelerate. Because the application using the GPU high performance to enhance the computing performance. GPU with strong abilities of floating point arithmetic, in molecular simulation, image processing, fluid mechanics, finance, mathematics, astronomy and communication and has important application prospects. This paper mainly introduced the GPU based on the CUDA framework programming model, focusing on CPU and GPU computation ability, detailed introduction of digital image processing in the Fu Liye transform principle and significance, through computer simulation, compared to CPU's computing ability can come to the conclusion that, CUDA in parallel data processing ability is very strong, in image processing can be used to accelerate the calculation method of MATLB based on CUDA to improve the image processing speed, through the Fu Liye transform analysis under the two methods of accelerated MATLB program.

II. FAST FOURIER TRANSFORMATION

A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse. There are many different FFT algorithms involving a wide range of mathematics, from simple complex-number arithmetic to group theory and number theory.

The DFT is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields (see discrete Fourier transform for properties and applications of the transform) but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly: computing the DFT of N points in the naive way, using the definition, takes $O(N2)$ arithmetical operations, while an FFT can compute the same DFT in only $O(N \log N)$ operations. The difference in speed can be substantial, especially for long data sets where N may be in the thousands or millions. In practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to N / log(N). This huge improvement made the calculation of the DFT practical; FFTs are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers.

The best-known FFT algorithms depend upon the factorization of N, but there are FFTs with $O(N\log(N))$ complexity for all N, even for prime N. Many FFT algorithms only depend on the fact that $e^{-\frac{2\pi i}{N}}$ is an N-th primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms. Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a 1/N factor, any FFT algorithm can easily be adapted for it.

The FFT has been described as "the most important numerical algorithm of our lifetime". There are lots of researches based on FFT.

III. BASED ON THE GPU PROGRAMMING AND THE FAST FOURIER TRANSFORM

A. The CUDA programming model

The CUDA programming model is CPU as the host, GPU as the equipment end, this programming model by CPU and GPU work together to accomplish complex transaction processing, CPU is mainly responsible for the small amount of data of the serial arithmetic, GPU is mainly responsible for performing high speed thread parallel processing tasks, CPU has an independent memory address space host memory, GPU has an independent memory address space the device memory, in memory of the operation process needs to call the CUDA memory management functions, including open, release and initializing the memory space, as well as in the host side and the terminal equipment for data transmission. For the large-scale data and parallel operation part can consider this part of the work to the GPU to complete, run on the GPU CUDA parallel computing by the kernel function to achieve the kernel, the kernel function is not a complete program but the program can be executed in parallel steps. Therefore, a complete CUDA program is composed of a series of equipment of kernel functions parallel steps and the host side of serial processing steps to complete. CPU serial code complete work is mainly the preparation of data and the initialization of the device, to the greatest extent possible to improve the calculation speed, should try to avoid the data in CPU and GPU pass between the frequency, but because the GPU function is very limited, most of the work needs to be carried out in CPU. In a CUDA program, the host has mainly completed the following functions.

(1) CUDA starts, in the side of the device for data memory space allocation, input data.

(2) the host memory copy of the data to the memory of.

(3) in the GPU on the distribution of memory, used for storing the output data.

(4) calls the equipment side of the kernel function for calculation, the results into the memory of the corresponding region.

(5) CPU memory allocation, used to store GPU came from the output data.

(6) the memory of the results are read into memory, using CPU for data and other processing.

(7) the release of memory and memory space.

The CUDA compiler is the main tool of nvcc, this article through the MATLB and CUDA mixed programming to realize the CPU and GPU assistance operations, first using CUDA and MATLB to write applications, mainly on the core function of the writing, and then through a plugin for MATLB, nvmex will use CUDA source program is compiled into a Mex file, and then in the MATLB directly call the corresponding parallel program mex file. For GPU accelerated MATLB fast Fu Liye transform, one is through the CUDA and MATLB mixed programming, a MATLB2012 via parallel computing function to achieve. This article mainly through digital image processing in the fast Fu Liye change algorithm for GPU accelerated MATLB

tested, the original image data is in the CPU, in the fast Fu Liye transform to data from the CPU copied to the GPU to execute in parallel operation, and then to shorten the running time.

B. Fast Fu Liye transform algorithm

Size of $M \times N$ images $f(x, y)$, the two-dimensional Fu Liye transform, the expression is：

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + uy/N)}$$

For spectrum $F(u, v)$ symmetric，It contains the image frequency domain signals. In image processing, many times through the frequency domain image processing to analyze and improve image. Especially in image low-pass filter, high pass filter, can make use of Fu Liye transform in the airspace problem into the frequency domain for processing, fast Fu Liye transform to discrete Fu Liye transform of the computing speed is greatly improved, so that the fast Fu Liye transform in digital image processing in a more extensive application. Size of $M \times N$ images If the above formula directly of the Fu Liye transform, the complex multiplications and additions were M2N2 and MN (M-1) (N-1). For the current 512*512 digital image, the complex multiplication and addition of the number will reach about 70000000000 times. And the use of fast Fu Liye transform only needs time 2 millions of times. Cooley and Tukey in 1965 put forward the fast FFT algorithm, using the algorithm in the implementation of F (x) (x = 0,1, ... , N-1) of the discrete Fu Liye transform, needs to complete complex multiplication and addition of the number were 1/2Nlog2 (N) and Nlog2 (N). While the direct use of the discrete Fu Liye transform formula to calculate, need for complex multiplications and additions were N2 and N (N-1). In the MATLB image fast Fu Liye transform the two-dimensional image data to a matrix, then directly call the system fft2 function directly in the CPU two dimensional images of the fast Fu Liye transform. Based on the CUDA MATLB accelerated first in the CPU distribution of two-dimensional image data, and then use the CUDA built-in function in data transfer function cudaMemcpy of data from the CPU copy to GPU, GPU use CUDA own CUFFT library functions for image fast two-dimensional Fu Liye transform, and then call the cudaMemcpy function to the result copy to CPU, free storage space which is completed by the use of CUDA fast Fu Liye transform. Another version of matlab2012 GPU integration natively support, you can directly use the built-in MATLB parallel operation function library in GPU of data copy and fast Fu Liye transform, and GPU accelerated MATLB.

IV. THE RESULTS OF COMPUTER SIMULATION AND ANALYSIS

For fast Fu Liye transform test, mainly through the processing of dimension 200200400400600 60080080010001000 2D image. The system is windows7, support CUDA plugin for nvmex MATLAB2009A, the card

is in support of the CUDA parallel computing GTX580. Then the processing of single and double precision data, mainly through the MATLB platform has been compiled based on GPU parallel executable mexw files, and then processing of single and double precision data, corresponding to the processing time and acceleration such as shown in figures 1, 2, 3 and 4 .

From the results it can be seen, whether single or double precision data processing accuracy, with the increase of data quantity CPU arithmetic of time consumed by the gradient increased, but GPU operation time increases slowly. In addition from CPU and GPU acceleration ratio curve can be seen, only in the processing of large amounts of data and the complexity of algorithm can reflect better the accelerating effect.
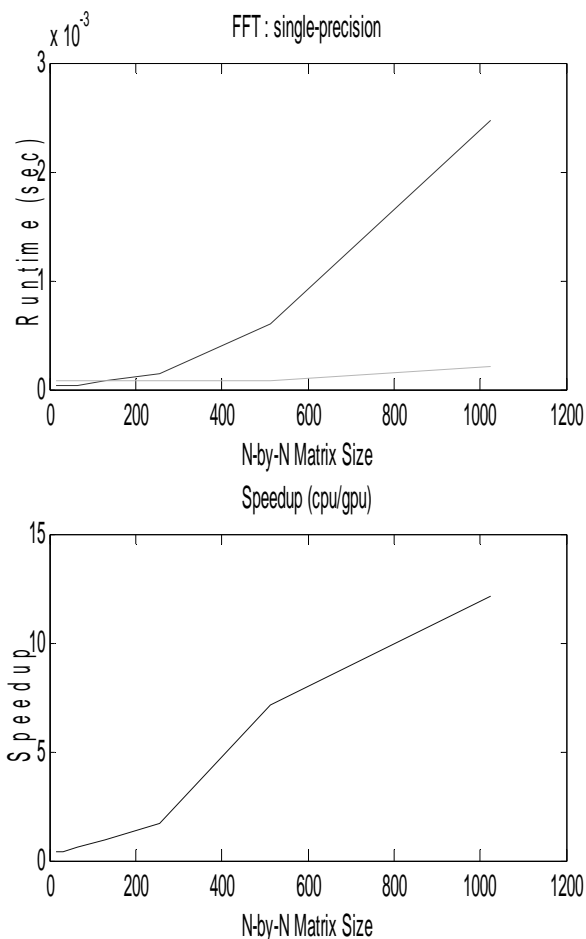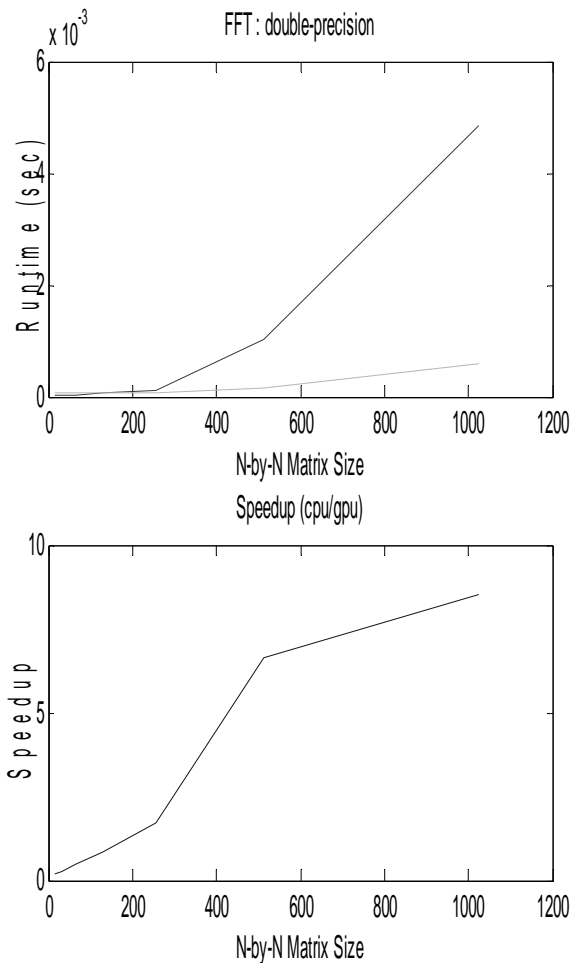
Figure 1. Single precision FFT.

Figure 2. Double precision FFT.

V. CONCLUSION

With large capacity digital image development, how to improve the processing speed as soon as possible to obtain the image feature and information, is a term in calculation tasks, especially in the timely recognition plays a very important role. With the development of high performance computing technology development, GPU in floating point arithmetic and parallel computing, computing capacity to rapidly improve. This paper mainly introduces the utilization based on GPU accelerated MATLB approach to the analysis of GPU processing fast Fu Liye change speed. Through the computer simulation can be seen, for large-scale data, through CPU and GPU help, give full play to the use of GPU powerful computing capacity and parallel processing ability, improve operation speed. Use GPU to high speed, high efficiency, low cost of processing large capacity image or multiple images, based on the GPU acceleration method, will be in the digital image has important application value.

REFERENCES

[1] Jason Sanders, Edward kandrot, CUDA BY Example an introduction to General-Purpose GPU Programming[M].China Machine Press.

[2] Nvidia Corporation, http://www.nvdia.com/cuda.

[3] NVIDIA Corporation, Accelerating MATLAB with CUDA UsingMEX files. https://developer.nvidia.com/object/matlab_cuda.html.

[4] Ruan Qiuqi . Digital image processing [M]. Beijing: Publishing House of electronics industry, 2003:118-146.

[5] Zhou Jinping. Image processing and graphics applications [M]. Beijing: Science Press, 2003:114-147.

[6] Wu Enhua, based on the graphics processor (GPU) is used to calculate the [J]. Journal of computer-aided design & computer graphics, 2004

[7] Rob Farber. CUDA- for large amounts of data of the supercomputer [J]. China Academic Journal Electronic Publishing House, 2008,11:166-168.

[8] Deng Yangdong VDIA CUDA large scale parallel program design and training course [Z]. Tsinghua University, 2008:10-200.

[9] Zhang Shu. GPU high performance calculation CUDA[M]. Beijing: China Water Power Press.

[10] Claire, Chen Qingkui CUDA based fast image compression [J]. computer engineering and design, 2010,14,: 3302-3308.