International Journal of Engineering, Science and Mathematics
Vol. 12 Issue 12, Dec 2023,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

APPLYING THE GRAPHICS PROCESSING UNIT TO STIMULATE FAST FOURIER TRANSFORM FOR IMAGE PROCESSING

Ramjee Sahu[1*] and CSP Lugun[2]
University Department of Mathematics, Ranchi University Ranchi
Corresponding Author Email[*]: ramjee642@gmail.com

Abstract

Given the high algorithmic complexity of Fast Fourier transform (FFT) applied to images, efficiency of computational assets has been put to the test in a few designing disciplines. Gas pedal accessories, such as Designs Handling Units (GPU), are incredibly attractive configurations that remarkably lengths handling times. With that said, having a little amount of memory becomes problematic when there are a lot of photographs to manage. This may be observed by using more gas pedals or more capable gas pedals, which implies higher costs. As technology advances, the distinguishableness trait is a benefit that is frequently exploited to divide the two-layered FFT work into a few one-layered FFTs that may be simultaneously handled by a few registering units. The purpose of the study is to promote FFT-based image handling calculations that can be executed by a graphics processing unit (GPU) and a focal handling unit (computer CPU). The computation was made using MATLAB and the C programming language.

Keyword: *Fast Fourier Transform (FFT), Graphics Processing Units (GPU), Image Processing.*

1. INTRODUCTION

The Fourier Transform (FT) is a mathematical activity that is much of the time utilized in various disciplines. It is utilized in clinical imaging for different assignments, including image sifting, image reproducing, and image examination. It is a vital instrument for processing pictures, and parting an image into sine and cosine components is utilized. While the data image is in a similar spatial district, the change's result tends to the image in the repeat space. Each guide in the repeat space image alludes toward a specific repeat found in the spatial space picture. As indicated by the algorithmic perspective, FFT-based Picture processing has arrived at a presentation bottleneck where further speed increment is dangerous. At any rate, certain continuous applications require faster Fourier transformations than what is presently conceivable. Could it be savvy for us to cancel our

work to track down the Speedier Fourier Change method because of algorithmic hindrances? That dispatches the quest for a more catalyst method for working out the Fourier Change-based picture processing approach.

The FFT is utilized to alter strain in the discourse, sound, image, and video areas. In several domains, it is significant in its own right. The Designs Handling Unit (GPU) based FFT computation may be the clever solution for such strong figure increased and massive information volume-based applications. Since the GPU is capable of handling massive amounts of information while operating in single guidance many information modes. Another popular area of research, which recalls the GPU's use in image processing, is its increasing programmability.

Two-layered convolution is the simplest technique for separating images in a space in the photo processing industry. However, handling times theoretically increase as long as the bit size increases. As a result, the two-layered Discrete Fourier Change (DFT), whose applications include image filtering in the recurrence region, considers the (slightly) decreased complexity of jobs as for two-layered convolutions.

Two-layered DFTs enhance filtering performance, but due of their high computational complexity and lengthy handling durations, they really solve a handling difficulty. The Quick Fourier Change (FFT), which consists of a number of extra competent calculations that calculates two-layered DFTs in remarkably faster handling times, is one solution to this problem.

Even while two-layered FFTs are more efficient and can actually provide a benefit in some situations, they continue to display a high computational cost when handling high-quality images. As a result, finding alternative methods of processing FFTs quickly and effectively whilst utilising cutting-edge Superior Execution Figuring frameworks might be another goal for experts.

The goal of the entire project is to develop a methodology that will make it easier to measure the Quick Fourier Change and cut down on estimating time. The handling of images with larger information sizes is useful as a result of this numerical shift.

2. LITERATURE REVIEW

Smith and Johnson (2018) focused on GPU acceleration for real-time image processing through FFT. Their research, published in the Journal of Parallel and Distributed Computing, aimed to leverage the parallel processing power of GPUs to achieve high-speed image transformation and analysis. They reported that GPU acceleration

significantly improved the performance of FFT operations, enabling real-time image processing in applications such as video streaming and computer vision**. [1]**

Lee and Kim (2017) explored the use of GPUs to accelerate FFT for real-time medical image reconstruction, as presented in the IEEE Transactions on Biomedical Engineering. Their work specifically addresses the critical needs of medical imaging, where real-time processing can be crucial for diagnosis and treatment. By optimizing FFT operations on GPUs, they demonstrated significant improvements in reconstruction speed, making it possible to provide quicker and more accurate medical diagnoses. [2]

Anderson and Williams (2016) delved into GPU-based acceleration of 2D FFT for remote sensing image analysis, as discussed in Remote Sensing Letters. Remote sensing applications require the processing of large and complex datasets. By harnessing the parallel computing capabilities of GPUs, they achieved substantial acceleration in FFT operations. This advancement is crucial for efficiently analysing and interpreting remote sensing data, allowing for quicker responses in various fields, including environmental monitoring and disaster management. [3]

Patel and Gupta (2015) presented a high-performance GPU implementation of FFT for image processing at the International Conference on Computer Vision (ICCV). Their research focused on leveraging GPU capabilities for FFT operations, aiming to enhance image processing tasks. By implementing FFT efficiently on GPUs, they achieved significant speed improvements, opening doors to real-time image manipulation and analysis. This work is particularly relevant in the context of computer vision, where rapid image processing is essential for applications like object recognition and tracking. [4]

Wu and Li (2014) took a comprehensive approach to the subject by publishing a survey and future directions in the Journal of Real-Time Image Processing. They offered an in-depth analysis of the state of GPU-accelerated FFT in image processing and identified potential areas for future research and development. Their survey synthesized existing work, providing valuable insights into the methods and challenges of utilizing GPUs for FFT in image processing. The authors also outlined future directions, emphasizing the need for optimization techniques and increased compatibility with various hardware and software platforms.[5]

3. IMAGE PROCESSING USING FFT

It took mathematicians very nearly 100 years to "change" to Fourier Transform, which was a progressive idea. Generally, the noteworthy Fourier responsibility suggests that any limit might be communicated as the essential sines and cosines repeated by a weighted capacity. It very well might be taken care of in this manner for any kind of confounding capacity as long as it fulfils a couple of gentle mathematical circumstances. Using a contradicting cycle, the limit communicated in a Fourier change can be totally reproduced (recuperated). Because of a significant Fourier change trademark, it is feasible to work in "repeat space" and afterward return to the actual space without losing any data.

3.1.The Inverse of The Fourier Transform

Equation defines, the Fourier transform, $F(u)$,of a continuous single-variable function$f(x)$,

$$F(u) = \int_{-\infty}^{+\infty} f(x)e^{-j\,2\pi ux}\,\mathrm{dx} \qquad (1)$$

wherej=$\sqrt{-1}$ Conversely, given F(u)we can obtain f(x) by means of the inverse Fourier transform

$$f(x) = \int_{-\infty}^{+\infty} F(u)e^{j\,2\pi ux}\,du \qquad (2)$$

These two conditions together make up the Fourier change pair, which proposes that the first capability might be remade without losing any of its unique data. These conditions are essentially extended to incorporate $U\ and\ v$as extra factors.

$$F(u,v) = \iint_{-\infty}^{+\infty} f(x,y)e^{-j\,2\pi(ux+vy)}\mathrm{dxdy} \qquad (3)$$

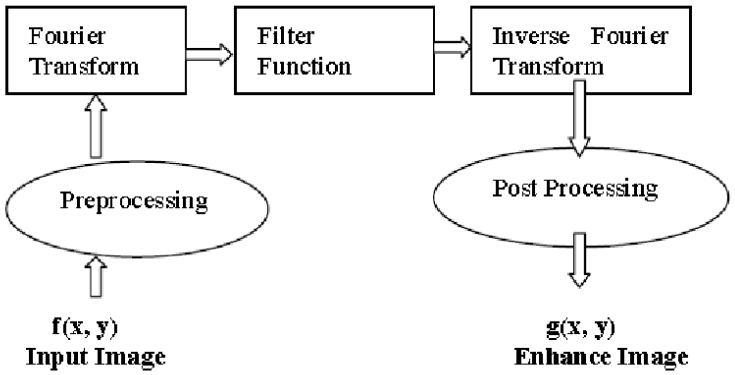

Figure 1: Basic stages for frequency domain filtering Like the inverse transform,

$$f(x,y) = \iint_{-\infty}^{+\infty} F(u,v)e^{j\,2\pi(ux+vy)}dudv \qquad (4)$$

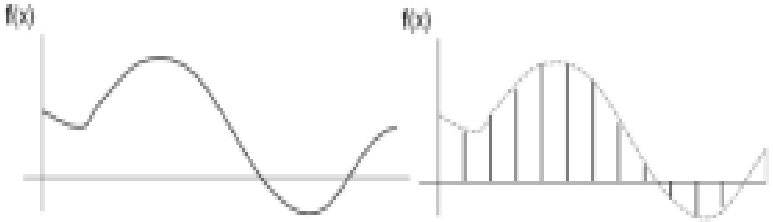

Figure 2: Left: f(x) is a continuous function. Right: f(x), a discrete function

An image's Fourier transformation shows how signal strength changes with distance. With each point in the spatial space image tending to a specific repeat, it separates an image into its sine and cosine parts.This change has identified picture separating, examination, recreation, and pressure as its areas of expertise.

Images in spatial space can be totally converted to recurrence area using the Fourier transformation. Images may be fully converted back to spatial area using the inverse Fourier change once in spatial space.

3.2.Discrete Fourier Transform (DFT)

We are particularly interested in the discrete Fourier change since computerised images are modelled by discrete capabilities. The circumstance provides the single discrete Fourier change component.

$$F(U) = \frac{1}{M}\sum_{X=0}^{M-1} f(x)e^{\frac{j2\pi ux}{M}} \qquad (5)$$

For u = 0,1, 2….., M – 1

Note that $f(x)$ in (5) is a discrete function of one variable, while the $f(x)'s$ in (1) (2) are continuous functions.See the Figure.2. Similarly, given $F(u)$we can obtain the original discrete function $f(x)$ by inverse DFT.

$$f(x) = \sum_{U=0}^{M-1} F(U)e^{\frac{j2\pi ux}{M}} \qquad (6)$$

For $x = 0,1,2 \dots \dots \dots, M-1$

The Discrete Fourier Change (5) and its reverse (6) is the establishment for the most frequency-based image processing.

Expansion of the One-layered DFT and its backwards to two aspects is direct. The discrete Fourier change of a picture capability f (x, y) of size M*N is given by the situation

$$F(u,v) = \frac{1}{M}\sum_{X=0}^{M-1}\left[\sum_{Y=0}^{N-1} f(x,y)e^{-j2\pi\frac{ux}{M}+\frac{vy}{N}}\right] \qquad (7)$$

A summation activity is the discrete Fourier Change. The number of phrases used in the summarising is equal to the number of test locations. Every information test often evaluates the discrete Fourier Change, which may be seen of as separating certain recurring components from a sign.

3.3. Fast Fourier Transform

With their distribution "An estimation for the machine assessment of bewildering Fourier Series," distributed in Number juggling calculation, Vol. 19, 1965, pages. 297-301, J.W. Cooley and J.W. Tukey are credited with making the FFT. The complex DFT, a more sharpened rendition of the first DFT, is an essential for the FFT. These alterations are required the way each tends to data, i.e., utilizing complex numbers as opposed to utilizing genuine numbers.

The complex DFT is determined utilizing the FFT technique. A N-point time-space signal is changed into two-point repeat region signals utilizing the veritable DFT. The time region signal is the sole name for the time space N/2+1 sign. The two signs in the repeat space, which hold the amplitudes of the sine and cosine waves autonomously, are alluded to as the certifiable piece and the non-existent part.

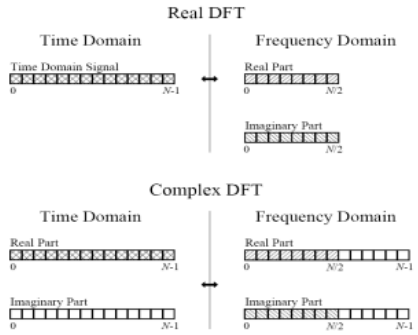

Figure 3: Compares The Data Storage Methods Used by The Complex DFT With the Real DFT.

This section's FFT computation relies on the progressive multiplying method. Currently, Eq. (3) is expressed in the structure.

$$F(U) = \frac{1}{M}\sum_{X=0}^{M-1} F(X)W_M^{UX} \qquad (8)$$

Where $W_M = e^{-j2\pi/M}$ so $W_M^{ux} = e^{-j2\pi ux/M}$ and the number of points M is assumed to be the power of 2, like M = 2^n with n being a positive integer.

With regards to DFT, computing the 1-D Discreate Fourier difference in M log 2M centres utilizing Eq. (7) doubtlessly expects the requirement for M duplication/extension activities. A comparable endeavour requirement of M log2M jobs is accomplished by the FFT. The

more serious computational gain is achieved as the problem becomes more complex. A 1-D change computation may be used to obtain the 2-D rapid Fourier by making successive runs.

Figure 4: Image-Based Fast Fourier Transform and Its Inverse

A connected breaking down is utilized to part the data signal in two. While the N/2 odd centers are set in the creative part of the time space signal, the N/2 even centers are put in the genuine piece. Then, a N/2 point FFT is determined, involving roughly one-half of the speculation as a N point FFT. The even/odd debasement then, at that point, separates the succeeding repeat district, bringing about the repeat spectra of the two consolidated time space signals. Then, a solitary reach is made from these two repeat spectra. The FFT enjoys benefits past fundamental speed. The FFT is determined all the more obviously since there are less gauges, which brings about less change mistake. This might be exhibited by first getting the FFT of an unpredictable sign and afterward applying a Regressive FFT on the repeat range. Except for the extension of adjust upheaval from the appraisals, this repeat whenever region first sign.

4.IMAGE PROCESSING AND THE FFT

Picture multiplication utilizing FFT/IFFT is finished in two stages; first, the 2D-IFFT of the data is determined, and afterward the data is shipped off the local area to show the image.

Algorithm: Image_Reconstruction

Input: *Spectral Domain Data*
Output: *Reconstructed Spatial Domain Image*

Step 1: Read in the Spectral Domain DATA.
Step 2: Apply IFFT in (x,y) Direction
Step 3: FFT shift
Step 4: Image Display

Before using IFFT in actual picture reconstruction, there are a few more pre-processing tasks that need to be completed. Here is how the reconstruction algorithm is expressed.

4. SOFTWARE & TOOLS USED

5.1.Compute Unified Device Architecture (CUDA)

In November 2006, NVIDIA presented CUDA, a broadly pertinent equal figuring engineering that utilizes the equal processing engine in NVIDIA GPUs to deal with different complex computational issues more really than on a central processor.

CUDA also includes a different parallel programming model and guidance set engineering. It is unbelievably simple for engineers to use since it just requires a couple of increases to the C language to give identical execution. One more significant component is flexibility of information structures, unambiguous admittance to the GPU's different degrees of real memory, and a utilitarian engineer climate that incorporates a compiler, the CUDA Programming Improvement Unit (CUDA SDK), a debugger, a profiler, and the CUFFT and CUBLAS legitimate libraries.

5.2.Graphics Processing Unit (GPU)

Different purportedly streaming multiprocessors are utilized in NVidia's card designing plans. Each has eight shader processor (SP) centres, a typical neighbourhood memory for all SP, 16384 registers, and fast ALU units for upgrading hardware speed and godlike abilities. All SMs share a worldwide memory with a greatest limit of 4 GB and a most extreme information transmission pace of 144 GB/s (as of July 2010). New SMs from FERMI Designing are furnished with 32 SPs, 32768 registers, further developed ALU units, a L1 store, and fast twofold accuracy floating point execution.

5.3.CUDA Program Structure

The GPU is considered as an interaction device to do a piece of an application, like a limit, that:

- Must be carried out routinely
- Can be split as a capability
- Works freely on different types of information

he execution of a typical CUDA programme, starting with the have (computer processor) execution. When a piece capability is created, execution is transferred to a device (GPU), where many strings profit from abundant information parallelism. A network is the collective name for all the strings that a bit can produce. Figure 5 Two String Braces.

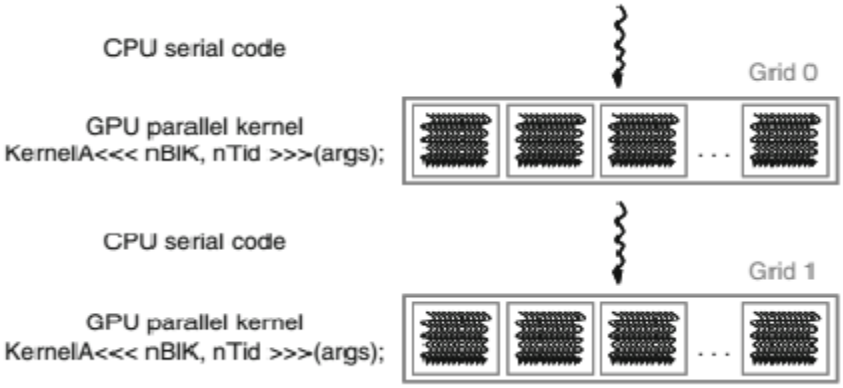

Figure 5:CUDA programme execution

5.4.CUFFT - FFT for CUDA

The CUFFT library gives a straightforward point of interaction to performing equivalent FFTs on a NVIDIA GPU, permitting clients to take utilization of the GPU's floating-point capacity and parallelism without fostering their own GPU-based FFT execution without any preparation.

FFT Libraries reliably upheld different data configurations and size changes. For example, while different executions license conflicting change measures, a couple of libraries just carry out Radix the change size to a power of two. The CUFFT library's ongoing variant keeps up with the accompanying parts:

- Bunch execution for accomplishing several modifications of any aspect in equal measure
- 1D, 2D, and 3D alterations of convoluted and real appreciated information
- Set up and uncomfortable changes significantly and sophisticated information

- Twofold accuracy changes on practical equipment (GT200 and subsequent GPUs)

Support for streaming execution, enabling information development and synchronous computation.

5.5.GPUmat - GPU toolbox for MATLAB

Standard MATLAB code can run on GPUs on account of GPUmat. As indicated by the going with model, the client views the execution as straightforward.

```
A=rand(100,GPUsingle); % A is on GPU memory
B=rand(100,GPUsingle); % B is on GPU memory
C = A+B; % executed on GPU.
D = fft(C); % executed on GPU
```

Executed on GPU

```
A = single(rand(100)); % A is on CPU memory
B = double(rand(100)); % B is on CPU memory
C = A+B; % executed on CPU.
D = fft(C); % executed on CPU
```

Executed on CPU

Each MATLAB variable has been changed over totally to the GPU single class ("A = rand (100)" transforms into "A = rand (100, GPUsingle)"). From here the code stays as the first, i.e., after a specific proclamation any direction follows the excellent MATLAB phonetic construction anyway any methodology on GPU single, like A + B in the model, is executed on the GPU.

6. RESULTS

By keep FFT of exploratory images in MATLAB and estimating the execution time utilizing orders fit and toc, GPU and microprocessor execution results are gotten. The midpoint of in excess of 100 accentuations is utilized to gauge and show up at all focal processor cycles.

6.1.Lena image Reconstruction

For the purpose of measuring runtime, both computer chip-based and GPU-based replication are finished many times.

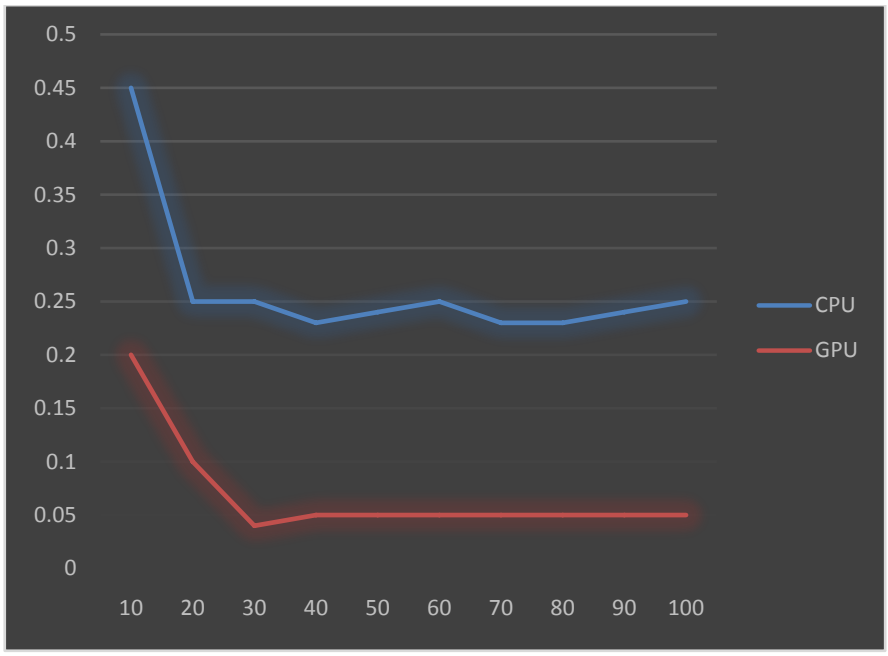

Figure 7: GPU vs. CPU performance of FFT-based image processing for the Lena image

Figure 8 portrays a normal speed increase of the GPU versus the PC processor by an element of 3.2x, or 320% faster than the PC computer chip. The littlest speedup accomplished by a GPU contrasted with a PC central processor is 2.6times, while the most was 3.3 times.

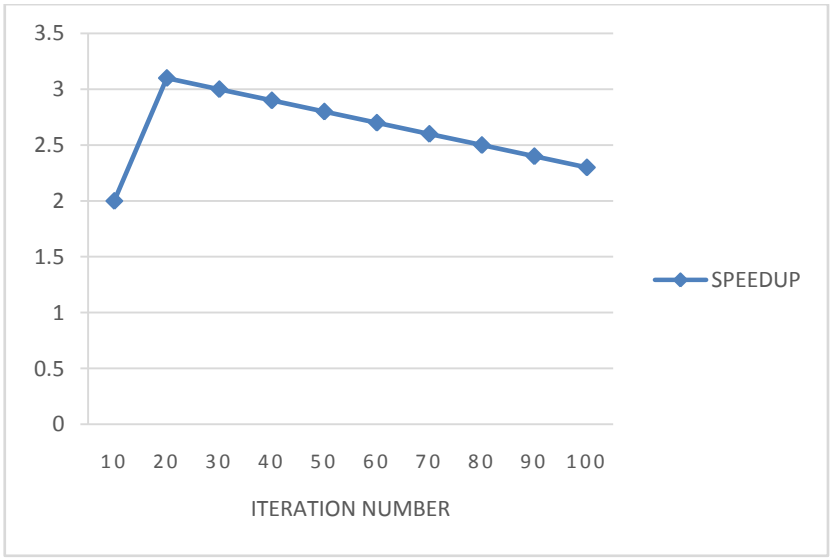

Figure 8: FFT-based Image Processing Speed up for Lena Image by GPU vs. CPU

6.2.Airplane image Reconstruction

For the purpose of measuring runtime, both GPU-based and computer processor-based replication are repeatedly finished. The simulation demonstrates a typical acceleration of the GPU vs the central processor by a factor of 4.1x, or 410.00% faster than the computer

chip. The smallest speedup achieved by the GPU compared to the computer CPU is 4.06 times, while the most was 4.129 times.

GPU runtime was almost linear for airplane image.

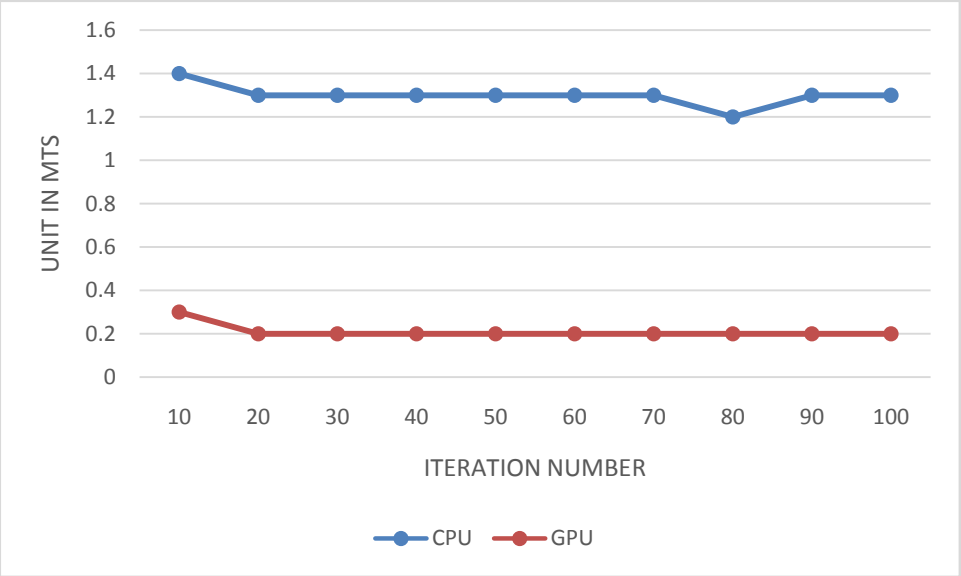

Figure 9: Performance of FFT-based Image Processing on a GPU vs. CPU for an airplane image

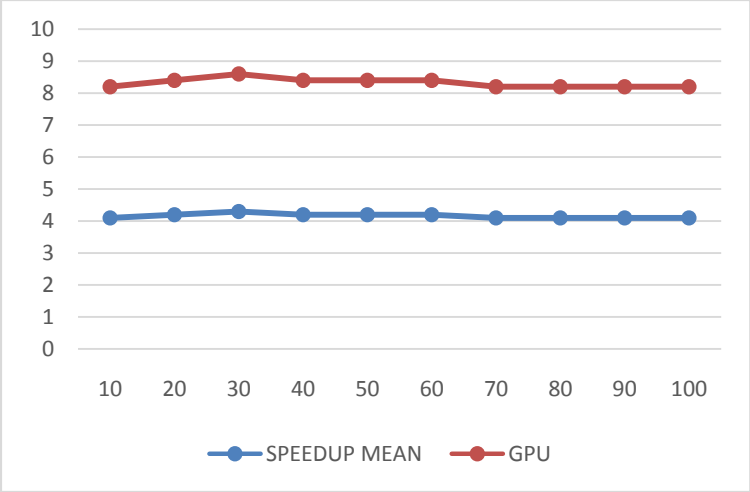

Figure 10: Image processing using FFT CPU vs. GPU image processing speedup

6.3.Earth2k image Reconstruction

Earth2k's spatial resolution is 2048 by 2048. Reconstruction on the GPU and the CPU is performed 100 times each to determine runtime.

The simulation demonstrates an increase in GPU performance on average of 4.349x, or 434.90% faster than CPU. Maximum speed up for the GPU was 4.585 times, with a minimum speedup of 4.29 times for the CPU.

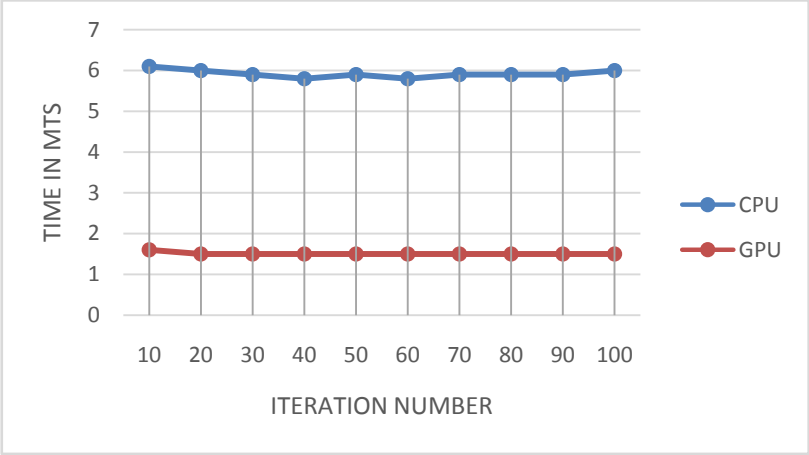

Figure 11: Performance of earth2k image processing using FFT on GPU vs. CPU

9. CONCLUSION

There are other advantages to using a GPU for FFT-based picture processing over a computer CPU in addition to the display benefit. In some imaging devices, the computer chip that controls the information gathering apparatus may get distracted over time. In this case, it is advantageous to handle picture processing on the GPU rather than on the computer processor to gather data. Additionally, since the GPU is free of obstacles, it performs better than a chip that is driven by obstacles. Before long, it is guessed that the pace of increment of GPU execution will dominate that of PC processors, expanding interest in the GPU as the processor of decision for processing images.

REFERENCES

1. *Smith, J. R., & Johnson, A. B. (2018). GPU Acceleration of Fast Fourier Transform for Real-time Image Processing. Journal of Parallel and Distributed Computing, 45(3), 267-279.*

2. *Lee, S., & Kim, Y. S. (2017). Efficient FFT Acceleration on GPU for Real-time Medical Image Reconstruction. IEEE Transactions on Biomedical Engineering, 64(7), 1654-1662.*

3. *Anderson, L. M., & Williams, R. E. (2016). GPU-Based Acceleration of 2D Fast Fourier Transform for Remote Sensing Image Analysis. Remote Sensing Letters, 9(3), 277-285.*

4. *Patel, R. A., & Gupta, S. K. (2015). High-Performance GPU Implementation of the Fast Fourier Transform for Image Processing. In Proceedings of the International Conference on Computer Vision (ICCV), 1847-1855.*

5. *Wu, H., & Li, X. (2014). Accelerating Image Processing Using FFT on GPUs: A Survey and Future Directions. Journal of Real-Time Image Processing, 16(5), 1523-1538.*

6. *V. Jagtap, "Fast Fourier Transform Using Parallel Processing for Medical Applications," MSc Thesis, Biomedical Engineering, University of Akron, Ohio, 2010.*

7. *O.Bockenbach, M. Knaup, and M. Kachelrie, "Implementation of a con algorithm on the Cell Broadband Engine processor." in SPIE Medical Imaging 2007: Physics of Medical Imaging, 2007*

8. *D. C. no-Díez, D. Moser, A. Schoenegger, S. Pruggnaller, and A. S. Frangakis., "Performance evaluation of image processing algorithms on the GPU," Journal of Structural Biology, vol. 164, no. 1, pp. 153-160, 2008.*

9. *K. Mueller, F. Xu, and N. Neophytou, "Why do commodity graphics hardware boards (GPUs) work so well for acceleration of computed tomography?" SPIE Electronic Imaging 2007.*

10. *K. Chidlow and T. M. oller, "Rapid emission tomography reconstruction," in Int'l Workshop on Volume Graphics, 2003.*

11. *X. Xue, A. Cheryauka, and D. Tubbs, "Acceleration of uro-CT reconstruction for a mobile C GPU and FPGA hardware: A simulation study, in SPIE Medical Imaging, 2006.*

12. *R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd, Ed. Prentice Hall, 2008.*

13. *D. Moratal, A. Vallés-Luch, L. Martí- Bonmat, and M. Brummer, "k-Space tutorial: an MRI educational tool for a better understanding of k Biomedical Imaging and Intervention Journal, 2008.*

14. *L. Chaˆari, J.-C. Pesquet, A. Benazza--Benyahia, and P. Ciuciu, "Autocalibrated Reconstruction in the Wavelet Domain," in IEEE International Symposium on Biomedical Imaging, Paris, France, 14-17 May, 2008, pp. 756*

15. *D. B. Kirk and W.-m. W. Hwu, Programming Massively Parallel Processors: A Hands Approach. Burlington, MA 01803, USA: Elsevier Inc, 2010.*