An Image Depth Processing Method Based On Parallel Computing and Multi-GPU

Chuting Yu[1,2], Minglun Cai[2]*
1.Engineering, Architecture & Info Tech, The University of Queensland, Australia
2.NingboTech University, Ningbo, China

***Abstract*—An image depth processing method based on CPU+GPU hybrid heterogeneous programming and Multi-GPU parallel computing is proposed in this paper. With the gradual increase of the image data, image processing algorithms have higher and higher requirements for GPU and CPU. Firstly, this paper introduces the heterogeneous programming system of the combination of CPU and GPU clusters and the technical points of CUDA. Secondly, the technical points based on Multi-GPU parallel computing image depth processing algorithm is introduced. Finally, the effectiveness of this algorithm is verified through experimental simulation.**

Keywords— Image Depth Processing, Parallel Computing, Multi-GPU, CUDA

I. INTRODUCTION

Image processing is the process of converting an image signal into a digital signal and processing it with a computer. The subject of digital image processing is originated in the 1950s. Thanks to the invention of computers, mankind has new tools on the road of exploring science. People began to learn to use computers, an efficient computing tool, to process graphics and image information. When the research on digital image processing was just emerging, the main purpose of this technology was to improve image quality and human visual effects. People often use the following digital image processing methods: image acquisition, digitization, encoding, enhancement, restoration, transformation, compression, storage, transmission, analysis, recognition, segmentation, etc[1-3].

Traditional digital image processing algorithms are mostly serial programming based on CPU. However, as the amplitude of the image to be processed increases and the number of pixels becomes denser, the requirements for the time complexity required by the digital image processing algorithm become higher and higher. This poses a great challenge to the real-time research of digital image processing.

In the process of computer development, people's requirements for image processing are not so complicated. The operation of images and related calculation methods are relatively simple. Therefore, there is no need to use corresponding hardware processing equipment to edit images. The powerful computing power of GPU can process graphics. However, with the continuous advancement of society and the continuous development of computer technology, people need faster computing speeds to perform higher-quality image processing, which has prompted the emergence and continuous development of GPU computing technology. From the current point of view, CPU generally refers to the central processing unit, which is a very large-scale integrated circuit. Its main function is to interpret computer instructions and process corresponding computer software data. Mainly rely on GPU to carry out, through the instruction to generate the corresponding operation control signal, in order to carry out the corresponding image processing. The GPU mainly refers to the graphics processor, which can also be called the visual processor. Its main function is to convert and drive the actual information required by the computer system, and to provide the corresponding scanning signal to the display to perform the display of the display. Correct control, in addition, the graphics processor is also the processor of the graphics card, and it is a more important part of the graphics card[4].

Parallel computing is the future development direction of the computer field. With the development of GPU (graphics processing unit) programmability, more and more researches have focused on GPU-based general-purpose computing technology. Combining the characteristics of digital image processing algorithms with the characteristics of GPU general-purpose computing, the acceleration of digital image processing using GPU is helpful for real-time computer image processing research. In recent years, with the continuous upgrade of GPU architecture and the continuous improvement of GPU programming technology, CPU+GPU hybrid heterogeneous programming has become a new research hot spot in the field of high-performance computing. The CPU+GPU heterogeneous collaborative computing architecture on a high-performance cluster is shown in Figure 1. On this architecture, the cluster can be divided into three parallel levels: inter-node parallelism, heterogeneous parallelism of CPU and GPU within nodes, and intra-device (CPU Or GPU) Parallel[5-7].

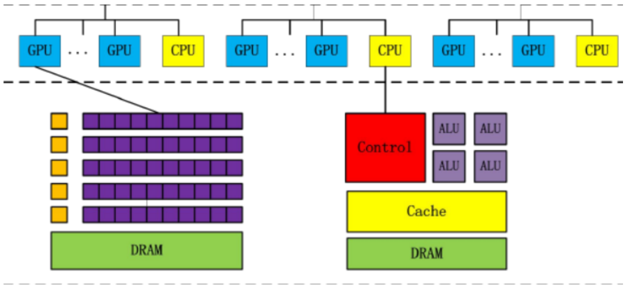


Fig. 1. CPU+GPU heterogeneous collaborative computing architecture

The birth of the above-mentioned new technologies has promoted the current Multi-GPU to become more efficient and flexible in terms of parallelism and programmability, and greatly improved the computing power and parallel processing capabilities of Multi-GPU. With the dramatic increase in Multi-GPU of computing performance, Multi-GPU-based general-purpose computing has gradually evolved into a new research field. This field is mainly aimed at how to use the powerful computing and parallel processing capabilities of GPUs in other scientific fields other than graphics processing, so that more general and extensive scientific calculations can be carried out[6-8].

II. The Proposed Methodology

A. Multi-GPU

There are two physical structures of the Multi-GPUs system, namely a single-node system and a multi-node system. Multi-node systems are GPU clusters, which connect single-node systems together through the network. Before 2012, the world's largest GPU system was Tianhe 1A, which contained 14,000 CPUs and more than 7,000 Tesla Fermi architecture GPUs, with 112 cabinets, and each cabinet contained 64 computing nodes. This article studies and uses a single computing node[9].

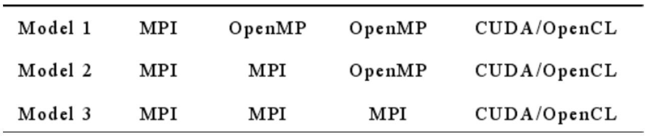
On a high-performance CPU+GPU cluster, each node is distributed and connected together through a high-speed network. In the actual programming process, MPI can be used to achieve communication between nodes. On each node, there is a multi-core CPU and several GPUs. Master-slave

The programming model is that the multi-core CPU is used as the host and the GPU is used as the slave. The execution of the kernel functions in the GPU needs to be called and managed by the CPU. On a high-performance CPU+GPU cluster, the CPU usually also has strong computing power, so the CPU may also perform part of the computing work. At this time, in the cluster, in addition to one CPU process/thread performing complex logic and transaction processing, other CPU processes/threads can also perform part of the parallel computing tasks, and the GPU is responsible for the main parallel computing tasks. In this case, all CPU cores can be unified as one device, and MPI processes or OpenMP threads can be used to control the communication and data division between devices in the node. For the CPU on a node, a CPU may contain multiple cores, and all the cores on the CPU are unified as a device, and a shared storage model is adopted between these computing cores. The parallelism between CPU cores can be accomplished by using MPI processes, OpenMP threads or p Thread threads. For the GPU on the node, CUDA or OpenCL programming can realize the parallel computing of the GPU many cores. It is worth noting that CUDA only supports GPU devices designed and produced by NVIDIA, while OpenCL supports both NVIDIA GPU and AMD GPU[10-12].

$$\phi = M_{\phi} \frac{\partial S}{\partial \phi} \tag{1}$$

$$\frac{de}{dt} = -\nabla \cdot M_e \nabla \frac{\partial S}{\partial e} \tag{2}$$

The following shows the CPU+GPU heterogeneous collaborative computing architecture



Model 1	MPI	OpenMP	OpenMP	CUDA/OpenCL
Model 2	MPI	MPI	OpenMP	CUDA/OpenCL
Model 3	MPI	MPI	MPI	CUDA/OpenCL

Fig. 2. CPU+GPU heterogeneous collaborative computing architecture

NVIDIA has developed the Scalable Link Interface (SLI) technology. SLI is an innovation on the platform that allows users to allow multiple GPUs to work in parallel in a single system, thereby intelligently expanding graphics performance. For example, in a system with two GPUs, when SLI is running, the tasks of rendering the screen will be divided equally, and each GPU only needs to complete half of the rendering tasks. This is how the user can nearly double the increase by installing two GPUs on the motherboard. Graphics performance. The implementation of SLI technology lays a technical foundation for parallel work of multiple GPUs under a single-node system. A single-node system is nothing more than installing two or more GPUs in a personal computer or workstation for parallel computing. Generally, personal computers expand the number of GPUs through multi-slot motherboard motherboards or multi-GPU boards. In theory, personal computers have the most.

B. Parallel Computing Based on Multi-GPU

There are two main types of parallel processing using Multi-GPUs: one is task-based parallel processing, and the other is data-based parallel processing.

Task-based parallel processing means that multiple GPUs process different tasks at the same time. For example, on a multi-GPU system, it generates N tasks, where N is equal to the number of GPUs in the system. Each GPU obtains a separate data packet from the memory to complete its task. If N tasks are not related to each other in parallel, there is no need to transfer data between GPUs. In this way, N GPUs can efficiently complete a huge computing task together. If N tasks depend on each other, then task-based parallelism will have great limitations. Data-based parallel processing is to divide a whole data into several blocks and hand them over to different GPUs for processing. For example, in a multi-GPU system, there is a piece of data that occupies M address spaces to be processed. Assuming that there are N GPUs in the system, you only need to divide the data into M/N evenly, so that the N GPUs are separated and processed in parallel. In data-based distribution, there are two distribution methods: one is the derivation/collection method, and the other is the striping/blocking method[13].

$$y^{'} = b_{\infty} + b_{10} + b_{01}x + b_{20}x \tag{3}$$

$$(x, y) = F(x, y) \tag{4}$$

Multi-GPU parallel computing can also be realized in the way of stream processing. Stream is a way to manage CUDA concurrency. By default, CUDA creates an execution stream inside a GPU. If you do not specify a number, the default is 0.

All data transmission and Kernel calls are queued in this stream and executed in sequence in the queue. By explicitly creating and using multiple execution streams, more work can be done per unit time and the application can be executed faster[14-16].

The easiest way to use multiple GPUs in a single CUDA program is to implicitly create a stream for each context of each device, and use cudaSetDevice() to change the current device to use multiple GPUs. CUDA causes the context state to change through cudaMalloc() when creating the context, thereby introducing the creation of the context environment. GPU device drivers can make a single CUDA application use multiple devices by providing multiple contexts for the application in the device driver. Queuing tasks on each device, thereby improving application performance by increasing the number of GPUs in the system. through simulation without distortion of the main dynamic characteristics of the system.

C. Image Depth Processing

There are generally two methods for processing digital images, one is based on the spatial domain, and the other is based on the time domain: In the spatial-based processing method, first consider the entire digital image as all pixels on the plane. And then directly process these pixels. Therefore, in the spatial processing method of digital images, the processing algorithm mainly considers three aspects: pixel-level processing, feature-level processing, and target-level processing. This feature makes the airspace processing method has a great advantage in GPU acceleration[17-19].

The frequency-domain processing method is different from the spatial-domain processing method. In the frequency-domain processing method, the Fourier transform of the digital image is first required to obtain the spectrum of the image to be processed. After the spectrum is obtained, the subsequent processing is performed, and the processed result is re-processed. Perform inverse transformation to obtain the final processing result. Frequency-domain processing algorithms generally have high-density calculations, which are based on the processing advantages of GPUs. And the CUDA platform contains two powerful calculation libraries: CUDA FFT and CUDA BLAS libraries. These two computing libraries provide great help for GPU-accelerated digital image processing based on frequency domain processing[20-22].

III. EXPERIMENT

Using CUDA technology, the concurrency of different applications is shown below.

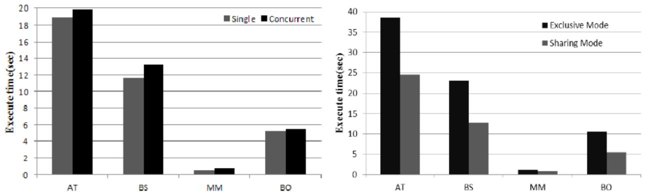


Fig. 3. concurrency of different applications

The images of first set of experimental test comparison results of the test star map are shown below.

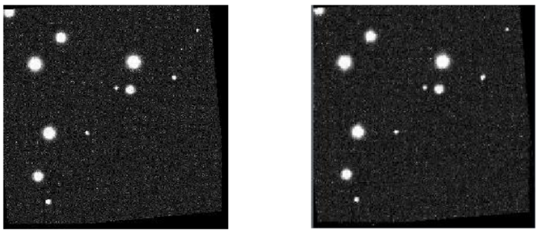


Fig. 4. First test star map

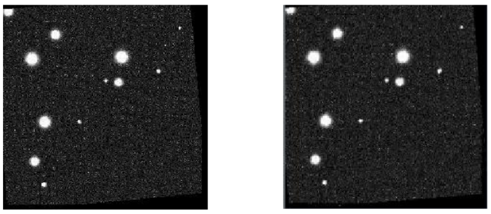


Fig. 5. The first set of star maps after registration

The images of second set of experimental test comparison results of the test star map and Power spectrum of input image are shown below.

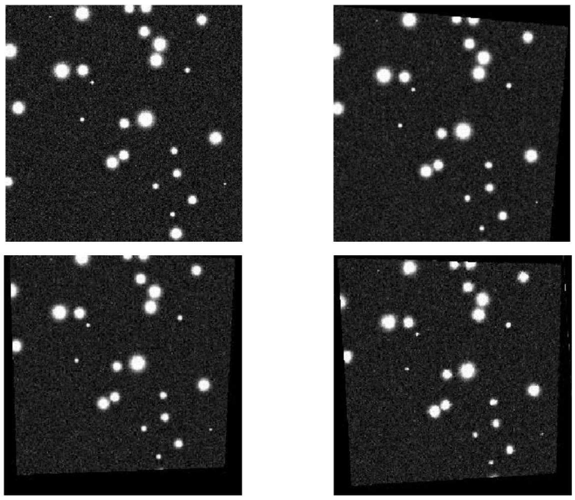


Fig. 6. The Second set of star maps after registration

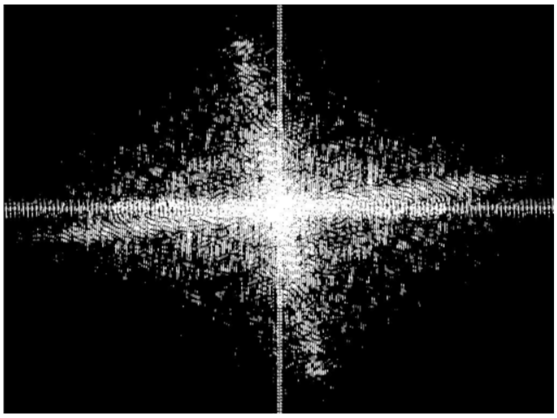


Fig. 7. Power spectrum of input image

IV. CONCLUSION

This article first introduced a digital image processing algorithm based on GPU and CPU, and learned a parallel computing method based on Multi-GPU and CPU+GPU hybrid heterogeneous programming. Then the Multi-GPU architecture and cluster mode are introduced in detail, and the parallel computing of heterogeneous systems is implemented on Multi-GPU. On this basis, the image depth processing method based on CPU+GPU parallel computing is introduced. And the feasibility of this method is verified in the experimental simulation.

REFERENCES

[1] CUDA C Programming Authoritative Guide [M] (United States) Cheng Runwei (John Cheng) waiting, Li Liang translation. Beijing: Machinery Industry Press. 2017.5

[2] Lu Huibin,Zhao Yanfang,Zhao Yongjie.Image dehazing based on the combination of bright channel and dark channel[J].Journal of Optics,2018,38(11):233-240.

[3] Wang Ru. Research and implementation of real-time image defogging method based on dark channel prior[D]. Xi'an: Xidian University, 2018.

[4] Zhang Wei. Synchronization control of fractional order financial hyperchaotic system based on new sliding mode method [J]. JOURNAL OF INNER MONGOLIA AGRICULTURAL UNIVERSITY (NATURAL SCIENCE EDITION), 2019, 040 (002): 89-93

[5] Xu Huan. Research on Real-time Restoration Technology of Atomized Degraded Optical Image[D]. Chengdu: University of Chinese Academy of Sciences (Institute of Optoelectronic Technology, Chinese Academy of Sciences), 2018.

[6] Zhang Minhua. Research and implementation of defogging algorithm optimization based on convolutional neural network[D]. Xi'an: Xidian University, 2018.

[7] Ya He K M, Sun J, Tang X. Single image haze removal using dark channel prior[C]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, USA: IEEE Computer Society, 2009: 1956-1963.

[8] CUDA C Programming Authoritative Guide [M] (United States) Cheng Runwei (John Cheng) waiting, Li Liang translation. Beijing: Machinery Industry Press. 2017.5

[9] GPU programming and optimization: mass high-performance computing[M] Fang Minquan, Zhang Weimin, Fang Jianbin. Beijing: Tsinghua University Press, 2016. 9

[10] Tan R T. Visibility in bad weather from a single image[C] // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Alaska, USA: IEEE Computer Society, 2008: 1-8.

[11] He K M, Sun J, Tang X. Guide image filtering[C] // Proceedings of European Conference on Computer Vision. Crete, Greece: Springer, 2010: 1-14.

[12] Yang Yan, Bai Haiping, Wang Fan. Single image adaptive defogging algorithm based on guided filtering[J]. Computer Engineering, 2016, 42(1): 265-271.

[13] Zeng Jiexian, Yu Yonglong. Image edge preserving and defogging algorithm combining bilateral filtering and dark channel[J]. Journal of Image and Graphics, 2017, 22(2): 0147-0153.

[14] Zeng Jiexian, Yu Yonglong. Image edge preserving and defogging algorithm combining bilateral filtering and dark channel[J]. Journal of Image and Graphics, 2017, 22(2): 0147-0153.

[15] Sun Yanan, Gao Yuanyuan, Song Hongjun et al. Fast image defogging method combining bilateral filtering and corrosion processing[J]. Computer Engineering and Applications, 2017, 53(1):178-182， 194.

[16] Mao Xiangyu, Li Weixiang, Ding Xuemei. Single image defogging algorithm based on sky segmentation[J]. Journal of Computer Applications, 2017, 37(10): 2916-2920.

[17] Zhang Baoshan, Yang Yan, Chen Gaoke, Zhou Jie. Dehazing algorithm combining histogram equalization and dark channel prior [J]. Sensors and Microsystems, 2018, 37(03): 148-152.

[18] Kim T K, Paik J K, Kang B S. Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering[J]. IEEE Transactions on Consumer Electronics, 1998, 44(1):82-87.

[19] Gu Zhenfei,Zhang Dengyin.A foggy image enhancement method based on the variational Retinex model[J].Journal of China University of Mining & Technology,2018 ,47(06) :1386-1394.

[20] Liu Yang, Zhang Jie, Zhang Hui. Research and application of an improved Retinex algorithm in image defogging[J].Computer Science,2018, 45(S1): 242-243+251.

[21] Geethu H , Shamna S , Kizhakkethottam J J . Weighted Guided Image Filtering and Haze Removal in Single Image[J]. Procedia Technology, 2016, 24:1475-1482.

[22] Narasimhan S G , Nayar S K . Contrast restoration of weather degraded images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(6):720-724.