Image Processing Application Using Parallel Computing

Vengayil Nayana Murali[1], Rahul C[2]

[1]M. Tech Scholar, L.B.S College of Engineering, Kasargod, India

[2]Assistant Professor, Department of Computer Science, L.B.S College of Engineering, Kasargod, India

Abstract: Image processing has been the rapidly developing area and a promising field for research. Different kinds of images are processed for use in various fields like medical imaging, satellite imaging, military purpose etc. If implemented using multi core CPU or GPU the computation speed is increased. CUDA is the software platform that supports GPUs by Nvidia. In this survey a few image processing applications are discussed. The results are compared by implementing those in CPU as well as GPU.

Keywords: Image processing, GPU, parallel computing, CUDA, PCA

1. Introduction

Since the invention of digital cameras, there has been a wide tendency to process the images. The raw data from the image sensor is converted into color-corrected image in a standard image file format. Many algorithms that involve mathematical operations have been developed and is still being developed to process those images for a variety of applications. Inorder to process the images any form of signal processing can be used where the input is an image, a series of images, or a video and the output can be either an image or a set of parameters related to the image. Computer graphics and computer vision are closely related to image processing. Many techniques are used in image processing such as image editing, image restoration, principal component analysis, linear filtering, wavelets etc. The image to be processed can be of 2-dimensional or 3 dimensional.

Image editing encompasses the processes of altering images. The pixels are changed to enhance the image. Sometimes the image may contain noise or it may be corrupted due to camera misfocus. The process of taking such noisy/corrupt image and estimating the clean, original image is known as image restoration. PCA is a mathematical tool which transforms a number of correlated variables into a number of uncorrelated variables. These uncorrelated variables are called principal components. The first principal component is taken to be along the direction with maximum variance. The second principal component lies in the subspace perpendicular to that of the first principal component. The third principal component is taken in the maximum variance direction in the subspace perpendicular to the first two and so on. The PCA [9] was initially applied for multivariate data reduction and was first developed for a linear transformation based on linear algebra. Wavelets are a mathematical tool that can be used to extract information from many different kinds of data like audio signals or images.

The computations in which calculations are carried out simultaneously are known as parallel computing. Larger problems are divided into smaller problems where they are carried out simultaneously. Parallelism was mainly employed in high performance computing [7]. Parallel computing can be classified according to the hardware support for parallelism, as multi-core and multi-processor computers having multiple processing elements within a single machine.

A Graphics Processing Unit (GPU) is a specialized electronic circuit that has highly parallel structure and is used in computer graphics and image processing. They are more efficient than general purpose CPU for algorithms that has to process large blocks of data in parallel. It can be either a video card or can be embedded in certain motherboard. Nvidia was the company that popularized the term GPU in 1999. These GPU when used for general programming is termed as General Purpose GPU (GPGPU). Computer graphics processing involves linear algebra matrix. Inorder to do general purpose computation on GPU, several new programming language and platforms have been built. Nvidia and AMD have their software platforms, Compute Unified Device architecture (CUDA) [8] and Stream SDK respectively. OpenCL is a framework for writing programs that execute on platforms consisting of CPUs and GPUs. The compute intensive portions of the application are sent to the GPU while the remainder of the code still runs on the CPU. During the computation the data elements are mapped to parallel processing threads.

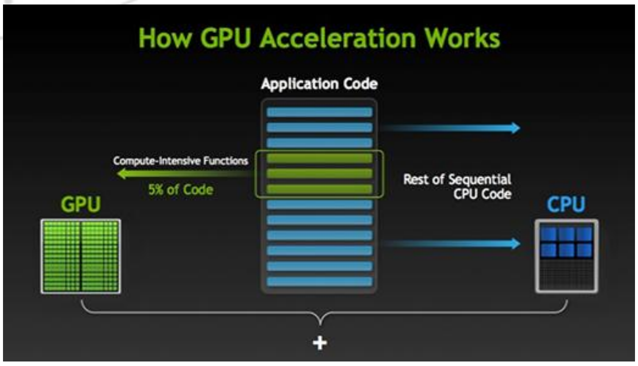

Figure 1: Working of GPU

The GPU has been used in processing wide variety of image processing applications. Many image processing applications in the field of medical science, satellite imagery have made use of parallel computing capability of GPU. High throughput and relatively low cost are the main advantages of GPU. Most of the modern laptops and desktops have GPU embedded in it. Even there are systems that use multiple GPUs. The invention of GPGPU has brought a wide change in how general problems can be solved with the help of GPU that is intended mainly for graphics. There have been many image processing applications developed with the help of GPU.

2. Related Works

William Chapman et.al, [1] in their paper presented a design for parallel processing of synthetic aperture radar (SAR) data using one or more GPU. This design supports real-time reconstruction of a 2-dimensional image from a matrix of echo pulses and their corresponding response values. Here a dual partitioning scheme is used that divides the output image into tiles and the input matrix into a set of pulses. The pair of image tile and pulse set is distributed to thread block in a GPU. Results were generated with Nvidia Tesla C1060. This design has achieved a peak throughput of 136 Gflops/s. The throughput can be increased by distributing the pulse matrix across multiple GPUs.

Syam Sujith Maddikonda and Shanmugha Sundaram G.A has discussed in their paper [2] about the relevance of digital beam forming (DBF) in space-borne Radar satellites for enhancement to quality imaging. The DBF technique helps to resolve issues like wide area coverage and resolution. Since the SAR data is computationally intensive to process, high performance computing is necessary. The process of comparison has been performed on both Intel core(TM) 2 Duo CPU and an NVIDIA Tesla C1060 GPU. The raw instrument data was collected by the satellite sensor and is the Level 0. This contains certain area on the ground called swath and has to be run through a SAR processor. This data is usable only after considerable correction called Level 1. Level 1 is further divided into Level1.1 and Level1.5. Level1 will have a small variation of pixel values based on the brightness measured by each sensor. The data contains amplitude and phase information and hence standard image processing functions will not work on SAR data. Because standard image data is in the visible region of thermal wavelength and processors are designed to read this. SAR imagery is derived from an active sensor at very different wavelength ranges (microwave region). The data is obtained as a result of the interaction between the RF energy and the target. So to process Level 1.5 data specialized software package is required. During the process of calibration of SAR image, the linear amplitude image is converted into radiometrically calibrated power image. Graphical user interface of MapReady software provide interactive way to convert SAR data to user friendly formats.

Tim Idzenga et.al in their work [3] parallelized the normalized cross correlation (NCC) based algorithm for estimating the deformation of tissues from radio frequency ultrasound data. Two approaches for parallelization was investigated: OpenMP interface on multi-CPU system and CUDA on a GPU. A maximum speedup factor of 132 was achieved on OpenMP approach and on Nvidia Tesla K20 achieved a speedup of 376. Number of templates selected from the deformation image can be of the order of ten thousand. Each NCC based matching operation between the template and the image is performed on a separate processor. In this paper they have used 1, 4, or 8 CPUs for parallelization. A GPU consists of several multi processors each containing 8 stream processors that operate in parallel. Kernels are launched using CUDA in GPU. Inorder to parallelize strain estimation algorithm, all templates from the pre-deformation image are selected and places in a 3-dimensional matrix. The same is done for post-deformation ultrasound image. Then both the template as well as the image matrix is transferred from the host to the GPU. Each launched thread contains NCC-based matching. Within each thread the denominator of NCC was calculated using sum table approach and the numerator in the time-domain. Thread produces 2-D NCC matrix describing the match between template and image. The results from all the threads are collected in a 3D matrix with the third dimension as the template ID. This is then transferred back to the host for detecting the best match.

Seung-Hun Yoo et.al, [4] has proposed an approach to accelerate multi-scale image fusion speed on GPU. It is implemented using CUDA software platform. Multi-scale image fusion algorithms have 3 basic procedures: multiscale transform, fusing and inverse multiscale transform. Commonly used multiscale transforms are pyramid and discrete wavelet transform. The Occupancy Calculator (OCC) provided by the Nvidia considers all factors to suggest the best method to decompose the data. The number of thread processors and block in the GPU can have a significant impact on the overall performance. They only use global memory without shared memory because many operations in algorithms need not share data. Before the execution of operations the entire image is transferred to the GPU.

Quantitative retrieval has been a growing area in remote sensing due to the rapid development of remote instruments and retrieval algorithm. Jia Liu et.al, [5] has presented two efficient implementations of aerosol optical depth (AOD) retrieval algorithm from the moderate resolution imaging spectroradiometer (MODIS) satellite data. Two high performance architecture were employed: multi-core processors and GPU. To retrieve a 10-year aerosol optical depth (AOD) dataset at 1-km spatial resolution, the synergetic retrieval of aerosol properties model from the MODIS data (SRAP-MODIS) is used. This algorithm employs a set of nonlinear equations. In the multi-core implementation of the AOD retrieval OpenMP has been used. AOD retrieval procedure uses a set of equations to be solved to calculate the AOD from the solution and the input parameter for each pixel of the prospected images. To solve the equation using Broyden's method and perform the AOD calculations, each pixel is treated entirely by one thread within a parallel for loop. To tune the performance OpenMP's scheduling strategy is carefully investigated. Unbalance of the threads happens since the algorithm has a pixel based nature. So to overcome this imbalance, a dynamic scheduling strategy was used with OpenMP. For the GPU implementation Tesla K20 from Nvidia is used. The source image data is loaded to CPU.GPU is initialized and memory is allocated. Now the data is transferred from host to GPU. From the GPU it defines and initializes variables in GPU. The nonlinear equations are solved using Broyden's method. AOD calculation based on Angstrom's turbidity formula is done and the results are stored in global memory. Now the data will be transferred from GPU to the host and the AOD results will be written.

Damien Vintache et al, has proposed [6] iterative reconstruction for transmission tomography on GPU using Nvidia CUDA. Normally the iterative reconstruction algorithm for X-ray CT image reconstruction suffers from their high computational cost. By the use of GPU, the processing times can be reduced. The ordered subsets convex (OSC) algorithm, an iterative reconstruction algorithm for transmission tomography, has been developed with CUDA. In iterative algorithm 3-D image is reconstructed from 2-D projection. The OSC algorithm tries to improve an image estimate to a solution. After iterating through all angles of a subset, the estimated attenuation coefficients are updated for each voxel of the volume.

Two geometrical operators of the forward projection and backward projection have been written to implement OSC algorithm. Forward projection is the geometrical transformation from the volume space to the detector plan space and backward projection is the geometrical transformation from detector plane space to the volume space. Above two and estimated volume correction are the expensive operations of OSC algorithm and parallelizable task. Three kernels have been implemented to perform parallel computation corresponding to forward projection, backward projection and estimated volume update operation. Using texture has given a speedup factor of 7.5 compared to unoptimized CPU version.

3. Conclusion

Many image processing applications has been developed and still research is happening in this area. GPU acts as a computing coprocessor for the application of image processing so that valuable amount of time is saved. Researchers have leveraged the computing capability of GPU as well as multi-CPU system.

References

[1] William Chapman, Sanjay Ranka, Sartaj Sahni, Mark Schmalz, Uttam Majumdar, "Parallel Processing Techniques for the Processing of Synthetic Aperture Radar Data on GPUs" IEEE conference paper
[2] Syam Sujith Maddikonda, Shanmugha sundaram G.A, "SAR Image Processing using GPU" International Conference on Communication and Signal Processing, April 3-5, 2014, India
[3] Tim Idzenga, Evghenii Gaburov, Willem Vermin, Jan Menssen, and Chris L. de Korte, "Fast 2-D Ultrasound Strain Imaging: The Benefits of Using a GPU" IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 61, no. 1, January 2014
[4] Seung-Hun Yoo, Jin-Hyung Park, Chang-Sung Jeong, "Accelerating Multi-scale Image Fusion Algorithms using CUDA", 2009 International Conference of Soft Computing and Pattern Recognition
[5] Jia Liu, Dustin Feld, Yong Xue, *Senior Member, IEEE*, Jochen Garcke, and Thomas Soddemann, "Multicore Processors and Graphics Processing Unit Accelerators for Parallel Retrieval of Aerosol Optical Depth From Satellite Data: Implementation, Performance, and Energy Efficiency", IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing, Vol. 8, No. 5, May 2015
[6] Damien Vintache, Bernard Humbert, David Brasse," Iterative Reconstruction for Transmission Tomography on GPU Using Nvidia CUDA", Tsinghua Science And Technology Issn 1007-0214 02/20 pp11-16 Volume 15, Number 1, February 2010
[7] Sharanjit Singh, Parneet Kaur and Kamaldeep Kaur, "Parallel Computing in Digital Image Processing" International Journal of Advanced Research in Computer and Communication Engineering, Vol 4, issue 1, January 2015
[8] Eric Olmedo, Jorge de la Calleja, Antonio Benitez and Ma. Auxilio Medina, "Point to Point Processing of digital Images Using Parallel Computing", International Journal of Computer Science Issues, Vol 9, Issue 3, No. 3, May 2012.
[9] Pramod Kumar Pandey, Yaduvir Singh and Sweta Tripathi, "Image Processing Using Principal Component Analysis", International Journal of Computer Applications(0975-8887) Volume 15, No. 4, February 2011.