JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

PERFORMANCE OF THREE DIMENSIONAL IMAGE PROCESSING METHOD BASED ON DATA LAYERING NORMALIZATION USING TENSORFLOW

Mahendra Kumar B[1], Rambabu Arjunarao Vatti[2], Dr.K.A.Ansal[3], Akula Chandra Sekhar[4]

[1]Assistant Professor, Department of Master of Computer Applications, Dayananda Sagar College of Engineering, SM hills, Kumaraswamy Layout, Bengaluru, India-560111.

[2]Professor, Electronics and Communication Engineering, Bharat Institute of Engineering and Technology, Hyderabad, India.

[3]Associate Professor ECE, Saintgits College of Engineering, Kottayam, Kerala-686532, India.

[4]Professor, Avanthi Institution of Engineering and Technology, cherukupally (V), Vizianagaram District, Andhra Pradesh, India.

[1]Mahen778@gmail.com, [2]rambabuvatti.india@gmail.com, [3]ansal.ka@saintgits.org, [4]chanduforu@gmail.com

ABSTRACT: Three dimensional image processing is used in different image process real time applications such medical imaging, satellite communication, underwater photography and industries. Imagery algorithms are used in day to day life. The huge data and complexity, the processing of images are tedious process and couldn't get effective decision. Image processing algorithms are parallel processing and we need to use multicore CPUs and Graphics Processing units. The high resolution image processing are pulse compression, synthetic aperture, tomography and underwater photography. So processing above three dimensional images, we implement data layering normalization in 2D plane. The dynamic range three dimensional images are selected and processed by using data layering normalization. The proposed method has been analyzed and tested using TensorFlow. Here we use multicore deep learning GPUs test method normalization process and verified by using experimental data. This method provides strong robustness, image background equalization and decision making capabilities.

KEY WORDS: Three dimensional image processing, Tensorflow, data layering, normalization, Tensorflow, deep learning

I. INTRODUCTION

Image processing is major thrust area and many of the industries are using various image processing techniques such as image enhancement, restoration, compression and multimedia application. These internet world huge volumes of multimedia data are used in various applications like social media, medical field and research industries. The variety of image analytics tools are used to process images and making effective decision. Nowadays, robot process automation, deep learning, machine learning, virtual reality and augmented learning are played vital and next few year they are rolling the world [1].

Google TensorFlow is the platform and building model for machine learning, deep learning and neural network applications. This tools has optimization, image pre processing, inference generation and residue analysis. This tool is modular extensible design so we will use for specific applications. This tool will be used in heterogeneous environments like multicore processing, GPUs, Server computing and Tensor processing units. This is deep learning framework with Python API and open source application by Google [2].

Tensorflow has computation graph, edges and operational nodes. This is arbitrary dimensional arrays and process multiple inputs & outputs. Tensors are used to process one node to another node [3]. The edges are used to find the flow and computations. Computational graph is generated in each stage for verifying performance. This is high order processing tool so we will get effective decision making results [4][5].

In this paper we implement image processing algorithm to evaluate three dimensional image using GPUs. We evaluate several image operations like image gradient calculation, edge detection, image transformation and segmentation. This paper has following sections, section II describes related works, section 3 describe methodologies and models, section 4 explains implementation and results and section 5 gives conclusion and discussion.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

II. RELATED WORKS

Three dimensional imaging sonar has wide band signals and emits at grazing angles. In this case, we got high resolution images. Babu Zho et al, depth dimension, navigation dimension and azimuth coordination are three factors affect image quality [6]. The ultra wide band frequency signals have large glancing angles. To achieve good performance in high resolution images we need to implement data layered method. Cong et al, the images has different depth and medium environments so the image region is unbalanced stage. So we can't make effective decision and extract information from images is tedious.

Jiang et al, the following reasons to affect the image quality 1. Imaging sonar imbalance in moving objects, 2. Transmitting data from one node to another node has reduce the quality, 3. Glancing angle has affect overall system process [7]. According to various survey, Kirchhoff model, perturbation model and scattering models are used at various degrees. Camera angle and position is major factor. The numerical aperture calculation is used for finding scattering and far-field images.

Damier et al, the deep learning architecture has five software frameworks and it has extensibility, hardware utilization and speed. This is multi threaded GPU model and flexible neural network computation. Emir et al, the comparative study of GPU acceleration and deep learning software are trained by neural network and two CPU and two GPU platforms for testing. In this case, distributed graph algorithm is suggested for multiple GPUs. TensorFlow has better scalability and provides comparative results [9].

The TensorFlow is developed by C++ and which has API for implementing other programming languages. Nowadays Python API is used for programming. Several third party vendors are used various programming languages for coding. TensorFlow is a computational graph model and it is flexible and distributed processing. TensforFlow has placeholders and variable tensors. The computational graph is generated by TensorFlow and creates queue nodes with dependencies. The queue has executed with dependencies and generates unsolved dependencies. This is parallel processing model so each node is assigned to devices and find the computational factor [10][11].

TensorFlow run as simulation graph and kernel is used to complete the operation. GPU implementations are done by using variety of processor and non processor model. This is abstract interface model and implements hardware, backend and domain specific compilers. NVIDIA cards, CUDA tool kit are used in development environment. Based on above literatures, we implement data layering model for analysing performance of three dimensional image processing using TensorFlow.

III. MODEL AND METHODS

Parallel processing has become important for handling high performance computing. The amount data, images and signals are increased rapidly. GPUs based computer graphics are used for parallel processing applications. CPUs are time consuming and tedious to handle images. The real time image processing platforms NVIDIA and OpenGL are used for writing codes and TensorFlow is used for computational graphs. This is GPUs digital signal processors and field programmable gate arrays for increase the performance. The below Fig 1 shows three dimensional image data in cylindrical share and represented using 2D plane.

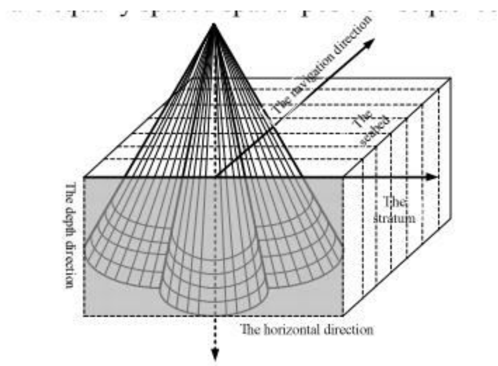

Figure 1: Three dimensional image model

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

This above model represented as P(x,y,z) and echo distance r with respect to t. The data converted to horizontal and vertical sequences. 256x256 pixels for 2D images, 5x5 for kernel size, for high resolution input images 4096x4096 and kernel size is 11x11. If the 3D image can be represented in 2D plane means 64x64x64 image size, kernel size is 5x5x5.

In this some characteristics are addressed. Edge detection is the first step of image processing. We used canny and gradient methods are used. 512x512 and 1024x1024 image data are selected for initial stage. Image gradient operation is applied for processing colour and gray scale image. 3x3 kernel size is applied for finding edges. Canny edge detector is applied for extracting structural information. Gaussian filtering is applied for smoothing and sharpening images. The length of the kernel is 5x5 and integer matrix is set as 1000x1000 and 10000x10000.

Image resizing is applied for processing the images. Data layering in 2D plane is used to resize the pixels. Here, we used 1024x1024 and 2048x2048 pixel for image data processing. K-means image segmentation techniques are applied and form the cluster group as 100x100 pixels. Image deblurring is the convolution process and optimization process is applied using 512X512 and 1024x1024 pixels. 45 and 90 degrees image rotation is applied for processing image data.

Partial differential equation is applied for image smoothing. For the 2D data plane, 512x512 and 1024x1024 image dataset is used for processing. Here quantization and dampling factor is calculated. To solve this problem, we used reflective compensative techniques to apply normalization operation. The following are the steps to process two set of images.

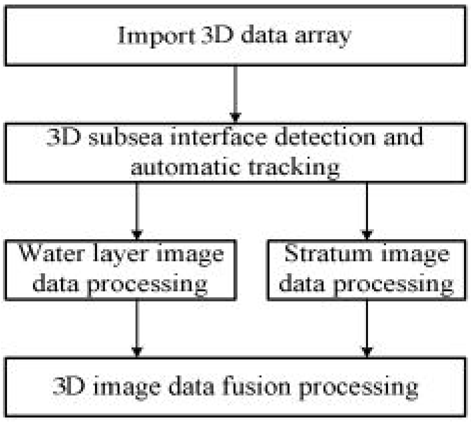

Figure 2: Image data fusion – two set of 3D images

Input 1: The water layer images are used to processing our proposed method. Here, length and apertures are calculated using interference factor. The attenuation and depth direction is measured.

$A_h = R \text{ X } (1-\cos(\pi/2))$

Input 2: Stratum images are selected from reflected objects. Azimuth classification is applied for finding depth and normalization process. Here, normalization process is applied for suppress the image and find the lossy factor.

$B_h = R \text{ X } (1+\sin(\pi/2))$

The converted 2D array is guided and applied statistical process for echo data. The array of sealed reference energy is calculated by using below equation.

$Rh = \sum \frac{image(Ah,Bh)}{N}$

$N = Rh / Max(Ah, Bh)$ and $Image_n = Image_n \text{ X } Rh$

Apply deep convolution formula and find attenuation energy, the energy balance is calculated by

$Image(n,R) = Image_n + \sum image(Ah,Bh) + \sum N$

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

For the above formula image data fusion is applied for measuring edges, skip part and particle. The DCT transformation is applied to find depth and position

$DCT_{n,R}$ = ((1/N)(0.5) X image(Ah,Bh)) / N-1

Inverse operation

IDCTn,R = $((1/N)^{0.5}$ / image(Ah,Bh))

Based on above formula we converted Python API code and tested the accuracy using TensorFlow.

IV. ACTION DATA PROCESSING USING TENSORFLOW

The experiments are performed using 64bit GNU Intel processor and GPU Tesla K80. The seabed data is selected and below figure 3 shows that water layer and stratum part. The interference factor is calculated and echo factors are adjusted.

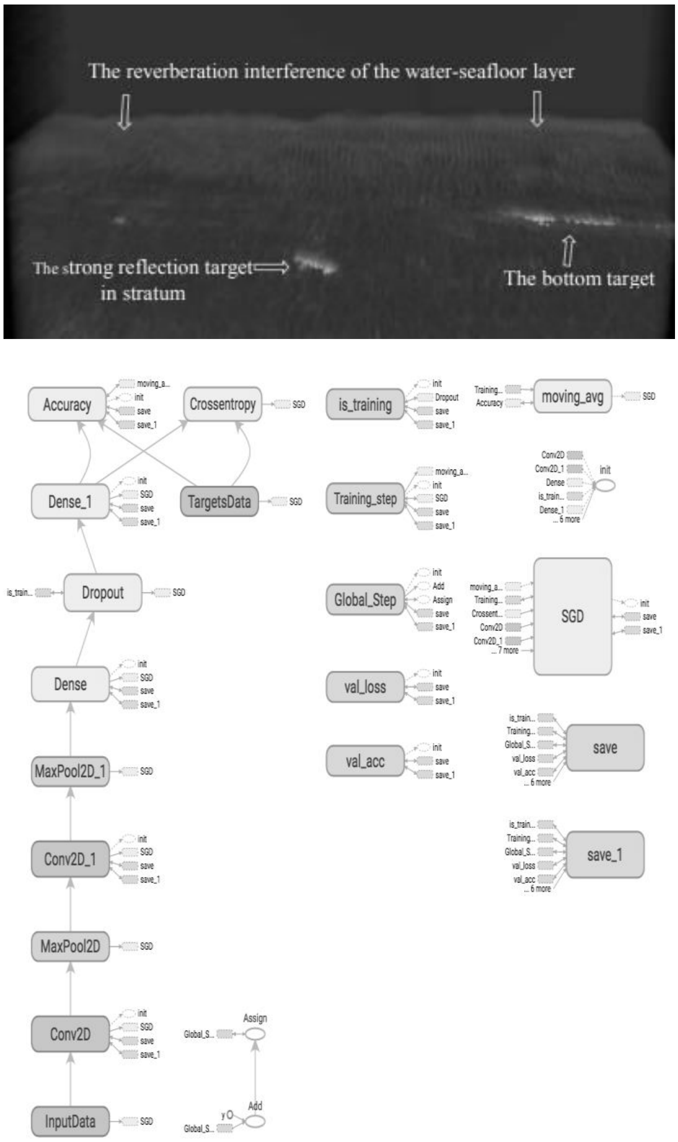

Figure 3: Input 1,2 and TensorFlow Computational Graph

Based on above computation graph results, detection and tracking data are suppressed. The echo energy and azimuth number is calculated and tested various time intervals. The below graph shows that comparison of results of azimuth and normalized values.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

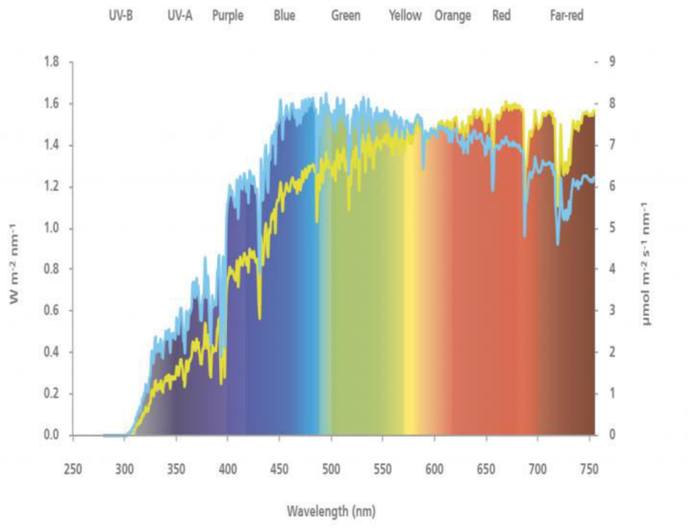

Figure 4: Normalized values and azimuth number results

The above graph shows the result of normalized values of input image size as 1024X1024 and kernel size is 3x3. The same coordinate result are tested in CPU and below table shows the performance of our algorithm.

Processor	CPU	GPU
Cores	2	2 X 2048
Clock Frequency (max)	2.5 GHz	756Mhz
Global Memory	2GB	2 X 1 TB
Power	90W	200W
Canny	0.45	0.31
Image Blurring	0.56	0.48
Gradient	0.78	0.45
2D Convolution	0.98	0.56
PDE	0.78	0.54
3D Convolution	0.92	0.62

Table 1: Comparison results of CPU and GPU our proposed system

From these experiments, GPU is performed very well and simulation results are shown good performance. TensorFlow gives computational graph and compared the CPU performance also. Based on the above results, the following are taken into account, 1. This method is effectively classified water level and stratum image data, 2. Found edges, side band and low band values, 3. Eliminated hidden and noisy elements, 4. Interference factor is calculated at each level, 5. Performance are compared and discussed. Our proposed method provides three dimensional images are processed by data layering normalization using TensorFlow.

V. CONCLUSION

In this paper, a three dimension image processing method is proposed and data layering normalization method is applied. TensorFlow platform is used to generate convolution matrix and test performance. Three dimensional volume of image data is selected and applied image processing techniques used data layering normalization method. In this method, we used azimuth number, canny operator and PDE for energy balancing, image background variance, gray scale modelling, gradient operations and image quality. CPU and GPU performance are compared with various factors. The computation is calculated at different time intervals and parallel devices. In future, same method is applied for huge volume of dataset and multiple GPUs.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

VI. REFERENCES

1) Bibo Zhu,Hanlei Jiang,Weihua Cong, “A three-dimensional image processing method based on data layered two-dimensional normalization”, 978-1-7281-1708-9/19/$31.00 ©2019 IEEE

2) Damir Demirović, Emir Skejić, Amira Šerifović-Trbalić, “Performance of some image processing algorithms in TensorFlow”, Study of Caffe, Neon, Theano, and Torch for Deep Learning“. CoRR, abs/1511.06435. 2015.

3) J. Lawrence,J. Malmsten, A. Rybka at all. “Comparing TensorFlow Deep Learning Performance Using CPUs, GPUs, Local PCs and Cloud.” 2017.

4) S.Manikandan, M.Chinnadurai, D.Maria Manuel Vianny and D.Sivabalaselvamani, "Real Time Traffic Flow Prediction and Intelligent Traffic Control from Remote Location for Large-Scale Heterogeneous Networking using TensorFlow", International Journal of Future Generation Communication and Networking, ISSN: 2233-7857, Vol.13, No.1, (2020), pp.1006-1012.

5) Y. Kochura, S. Stirenko, O. Alienin, M. Novotarskiy and Y. Gordienko, "Comparative analysis of open source frameworks for machine learning with use case in single-threaded and multi-threaded modes," 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, 2017, pp. 373-376.

6) S. Shi, Q. Wang, P. Xu and X. Chu, "Benchmarking State-of-the-Art Deep Learning Software Tools," 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, 2016, pp. 99-104.

7) Manikandan S, Chinnadurai M, Thiruvenkatasuresh M.P, Sivakumar M. (2020). "Prediction of Human Motion Detection in Video Surveillance Environment Using Tensor Flow", International Journal of Advanced Science and Technology, 29(05), 2791 - 2798.

8) BIbo ZHU, HanLei Jiang,Weihua Cong. A Three-dimensional Seabed Tracking Method Based on Joint Judgment of Dual Domain Features， IEEE ICSPCC2018.

9) Peizhen Zhang,Xiukun Li,Buried target acoustic scattering characteristics and its experiment. Acoustics Journal, 2018;43(6):935- 942.

10) ShengChun Piao,"Marine Environment and Hydro acoustic Propagation Characteristics", Academic Seminar on Underwater Acoustics and Navigation Technology. 2012.8.

11) M. Abadi, P. Barham, J. Chen, at all.“TensorFlow: a system for largescale machine learning”. In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16). USENIX Association, Berkeley, CA, USA, 265-283., 2016

12) CUDA, http://docs.nvidia.com/cuda/