Grenze International Journal of Engineering and Technology, Jan Issue

Performance Comparison of Parallel Computing Techniques for Processing CT Scan Images

Priyanka Bagul[1] and Dr. Vandana Inamdar[2]
[1]MES College of Engineering, Pune, Maharashtra, India
priyanka.awasare@mescoepune.org
[2]College of Engineering, Pune, Maharashtra, India
vhj.comp@coep.ac.in

***Abstract*—In recent era, a shift is observed towards the use of GPUs instead of CPUs. This is due to the ever increasing processing power, excellent programming support and high performance to price ratio. Along with this, medical imaging is booming and it plays an important role in the diagnostics and treatment planning. However, the process of 3D image reconstruction is computationally very taxing and very data heavy. In this paper, two parallel computing platforms are investigated 1. Multi-core using OpenMP and, 2. Many-core using Compute Unified Device Architecture (CUDA) enabled Graphics Processing Unit (GPU). These approaches are evaluated based on the performance against sequential execution of image preprocessing function. For illustration purpose, a case study of conversion of 2D CT scan images to 3D reconstruction image is implemented on intel i3, 8^{th} generation CPU with 4 cores using OpenMP. The same case study is implemented on CUDA enabled Nvidia Jetson TX2 GPU. We achieved speedup factor of 3 using OpenMP, whereas speedup factor of 107 using GPU for 1000 CT images.**

***Index Terms*— Graphics Processing Unit (GPU), Parallelism, CT images, CUDA, OpenMP.**

I. INTRODUCTION

Image processing is gaining more and more importance in the medical field. Processing medical images aids physicians in extracting the information for diagnosing diseases, surgical investigation, treatment and follow-up of disease and deciding better rehabilitation plans as well. At the same time, processing any medical image such as X-ray, Computed Tomography (CT), Magnetic Resonance Image (MRI), Positron Emission Tomography (PET) and ultrasound requires huge memory and very high computation capability. As extraction of relevant information is one of the complex tasks, it requires advanced computational systems to process. Also the run-time required to extract accurate and consistent image-based features must be reduced by system. [5]).

This leads to the search of various high computing platforms which are available. There is rapid growth in parallel computation application in medical imaging and image processing fields [28]. To achieve better robustness, efficiency, effective- ness and speed of execution, parallel computing techniques are exploited to make use of the maximum available performance in the computational architecture adopted [5].

Performing multiple processes simultaneously is termed as parallel processing or computing. But now, multicore processor technology (Intel Quad core and Octa core) has also evolved. This delivers 4 (Quad-core) and 8(Octa-core) complete cores within a single processor. Multicore processor technology has enhanced processing powers of servers [13].

Grenze ID: 01.GIJET.7.1.9_1

On the other side, over a past few years, the GPU's have been steadily upgraded for computation of data in parallel as a high performance platform. This parallelism is also used to process medical images to boost quality and to reduce time required to process all the data heavy tasks. Also now GPUs have efficient framework with outstanding price to performance ratio while providing support for a wide range of applications. GPU's have been used to accelerate numerous applications including medical image processing [19], [8], [2], [13].
This paper presents the performance comparison of parallel computing techniques such as multicore (Intel quad core) and many-core (GPU) for processing medical images. A case study of converting 2-D CT images to 3-D point cloud data is taken to show the comparison. The paper is organized as follows: Section II presents the literature survey in which the role of parallel computing platforms in processing the various medical images is described. Section III explains the case study of generation of 3-D point cloud data from set of 2-D CT scan images. Section IV gives implementation of the case study on two different parallel computing platforms. Section V provides result tables generated after execution of case study. Section VI concludes the results and comment on parallel performance comparison of both these parallel computing platforms.

II. RELATED WORK

Currently, there is rapid growth in using parallel computation in various medical imaging and image processing fields [28]. This section of paper is covering different medical imaging techniques including MRI, CT, ultrasound, PE, Optical tomography etc. The related work is presented in two subsections: A. applications using multicore platform using OpenMP B. applications implemented using many core platform using GPU.

A. Applications using OpenMP

Lei Hung et. al. [15] have used OpenMP on a 6 core TI-based Tomahawk platform for the medical ultrasound ap- plication. To achieve better image quality and to perform heavy computations for image processing they implemented different cache configurations. And with this they presented the performance impacts of data locality. M. D. Jones et. al. [17] implemented the Ordered-Subjects-Expectation- Maximization (OSEM) algorithm with a hybrid Message Passing Interface (MPI) and OpenMP approach, to improve parallel efficiency (PE) of the 3-dimensional (3D) PET image reconstruction OSEM algorithm. They observed PE improvement of approx.in percentage 7 to 17 verses pure MPI approach. Jian Guan et. al. [14] presented a multi-GPU implementation of the multilevel fast multipole algorithm (MLFMA) based on the hybrid OpenMP-CUDA parallel programming model for computing electromagnetic scattering of 3D conducting object. With this, compared to 8 threaded CPU-based MLFMA, the OpenMP-CUDA-MLFMA methods can achieve from 5 to 20 total Speedup ratios. Further, Li Jiajiang et. al. [16] examined effects of scheduling strategy and nested parallelism, the number of threads and loop transformation on performance of OpenMP programs. The use of OpenMP to exploit for medical imaging applications limits because of performance degradation due to load imbalance. They mentioned that the performance can also be degraded by following factors: switching overhead, scheduling overhead and synchronization, thread creation. With growing medical imaging data and with constraints of OpenMP, more computational power is needed.

B. Application using GPU

Massive computing in high resolution is mainly dominated by GPU[19]. As compared to CPUs, GPUs have ability to solve data heavy parallel problems with lesser run-time, making it less costly and energy efficient [13]. Anders Eklund et. al. [2] presented basic image processing operations and algorithms used to process individual modalities like CT, PET, MRI, ultrasound etc using GPU. With increasing size of medical imaging datasets, authors highlighted that the use of GPU in medical imaging applications becomes necessary for the processing. In 21*st* century, many efforts had been made to achieve General Purpose computing on GPU (GPGPU). Image reconstruction in MRI is one of the computational heavy applications. Wang H. et. al. [8] carried out a detailed survey for MRI applications in the image reconstruction schemes using GPU computing. Authors presented that GPUs are gaining importance due to its high computing capability and affordability. Christos G. et. al. [6] have presented a MR physics simulator for MRI that takes no assumptions with respect to the underlying pulse sequence and for complex large scale analysis on single computer. They developed a simulation platform in MATLAB and CUDA-c. With this they achieved speedup factor of about 228 compared to sequential execution on CPU and speedup factor of 31 to 115 compared to OpenMP considering thread count from 2 to 8 respectively.
Textural analysis is another important procedure in medical image processing which provides the valuable information of image recognition. Feature extraction, region segmentation and classification are involved in

pipeline of this procedure. Hasin- Yi T. et al [12] proposed a GPU acceleration for feature extrac- tion of MR brain images using the gray- level co-occurrence matrix (GLCM). They achieved more than 25 - 105 times of speedup for single precision on GeForce GTX 1080 for variable size of region of interests (ROIs). Another challenging and demanding task is noise reduction in MR brain images. Herng-Hua C et. al. [10] demonstrated an automatic trilateral filter for brain MRI restoration using GPU based acceleration. They used Intel Xeon(R) CPU-E5-2620 V3, 2.40GHz with a Tesla K40 GPU card for experimentation. They observed 757 times speedup for complete image volume of 256x256x256 pixels with 0.5 sec of execution time. Another field of medical imaging is non-Cartesian parallel MRI. In this application, the most computationally dense and time taxing method is image reconstruction. To enable real time reconstruction of sensitivity encoded radial imaging, Thomas Sangild S. et. al. [21] presented a reconstruction algorithm using GPU. Mikhail S. et. al. [7] found that ray- casting is commonly used volume rendering techniques for high fidelity requirements of medical imaging. They mapped ray-casting technique to CPU, GPU and many core architectures. They showed that parallel implementation of ray-casting gives 5.8X performance improvement on quad core Nehalem for 750x750x1000 dataset in 2.5sec. Whereas using Nvidia GTX280 GPU achieved from 5X to 8X speedup. Also when same is implemented on 16- core Intel Larrabee, it gives 10X speedup over single core Harpertown.

Medical ultrasound imaging modality is another real time diagnostic medical imaging technique. Sigurd S. et.al. [20], Idzenga et. al. [22] and Heyden K. H. et. al. [9] have used GPU acceleration to improve image quality and the performance in time. Authors of paper[9] have presented the performance com- parison of CPU and GPU for one of the advanced ultrasonic imaging algorithm Synthetic Aperture (SA). They achieved 50X times more speedup as compared to CPU by using Nvidia GTX480 GPU and 8.85X times speedup by using Nvidia 9600M GT GPU. They have also commented on power and performance trade-off where GTX480 consumes less power but combination of multi GPU gives better power-performance trade-off. Authors of paper [20] implemented a COLE algorithm using CPU and GPU keeping more focus on dynamic simulations which involve moving point scatterers. Authors have also addressed the data transfer latency issue between CPU and GPU by storing the complete required data in the GPU memory. With this they achieved 18.6X times speedup by using Nvidia K4000M GPU and 51.8X times speedup using Nvidia TITAN X GPU. In paper [22], authors have investigated the performance difference between two approaches for parallelization such as OpenMP interface on multi-CPU system and CUDA on GPU by executing 2-D normalized cross correlation (NCC) based algorithm. This algorithm is used to estimate deformation of tissue from radio frequency ultrasound data accurately. It is very time taxing procedure. Authors achieved a Speedup of 132 by executing NCC using OpenMP approach with 8 threads, whereas with Nvidia Tesla K20 GPU they achieved max. Speedup of 376.

In application such as CT reconstruction, filtering and backprojection take more computation time [11], [26]. Authors of [11] have achieved overall execution time to compute Feldkamp-Davis-Kress (FDK) reconstruction on-the-fly 30 fps (frames per second) by using GPU. Whereas, authors of [26] simulated that the GPU was 35 times faster per iteration in terms of computation time as compared to CPU.

Since reconstructed images from CT with monochromatic synchrotron radiation suffer from imaging artifacts resulting from non-uniform transmission due to strongly varying object thickness [3]. Paper [3] presented filtered back projection (FBP) algorithm used to reconstruct the image from Com- puted Laminography (CL). CL provides artifact reduced high resolution images. Speedup was achieved ranging from 1 to 8.02 times for number of threads from 1 to 20 when FBP implemented using Intel Xeon E5540 CPU. When the same is implemented on Nvidia GeForce GTX680 GPU, they achieved 10x faster reconstruction results.

To obtain desired information from medical image modali- ties like CT, CL, ultrasonic, MRI, PET etc. for diagnosing and deciding therapy planning [18] requires complex computations and time. To offload the computational burden from CPU, parallel computation can be used. This paper demonstrates two parallel platform techniques by implementing a case study of 3-D image reconstruction from CT images.

III. CASE STUDY

Ref. [24] presented a research work for development of patient specific implants from non-invasive imaging techniques by reverse engineering and additive manufacturing techniques. Authors developed an efficient algorithm for generation of free form surface from non-invasive CT scan images.They used image processing techniques and reverse engineering approach to extract point cloud data from CT scan images. 3-D point cloud data is obtained from CT images in DICOM3.0 (Digital Imaging and Communications in Medicine) format. Point cloud data is estimated by using image processing techniques [27] like segmentation, noise filtering, edge detection etc.

Fig. 1 shows methodology adopted for generation of point cloud data from CT images. These grey scale images had been acquired from GE ProSpeed CT scanner in DICOM3.0 format of 512x512 pixel size with data collection diameter 250mm with slice thickness of 5mm. This is implemented on CPU. The major challenges encountered for modelling are handling huge number of data points, noise removal, differentiation between inner and outer boundary profile points of bone and detection of nearest points on one surface[27], [24], [23], [25].

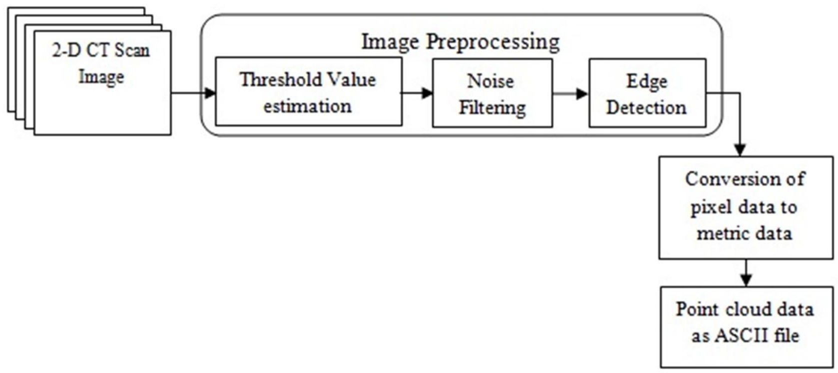

Fig. 1. Estimation of Point Cloud Data

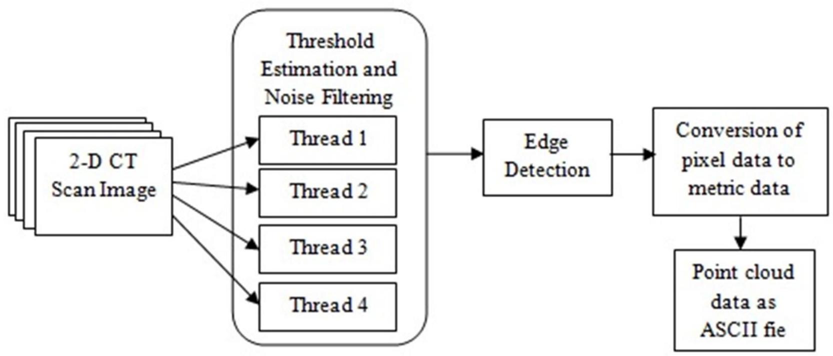

Fig. 2. Block diagram of implementation using OpenMP

IV. IMPLEMENTATION ON MULTICORE USING OPENMP

OpenMP provides richer set of directives that capture a wide range of parallelization strategies that are not strictly limited to shared memory[4]. OpenMP is portable and easy to implement program interface that allows the program to run in parallel heterogeneously on different cores of CPU. For implementation of the case study we used Intel i3, 8*th* generation CPU with 4 cores. As shown in Fig 1., in image processing, to perform operations like noise filtering and edge detection, we need to convolve the image and need to perform convolution on every pixel of image. In case study like 3-D reconstruction we need to perform convolution on every pixel of every CT image which becomes computationally heavy and time consuming for CPU. Hence we executed this function parallel using OpenMP.

Algorithm 1

for each image row in input image **do for** each pixel in image row **do**
set accumulator to 0
for each kernel row in kernel **do**
for each element in kernel row **do**
if element position is corresponding* to pixel position **then**
multiply element value corresponding* to pixel value
add result to accumulator
end if end for
end for end for
end for

The algorithm 1 shows process of convolution. While executing this algorithm using OpenMP, every CT image is processed on separate core. The system on we executed this has 4 cores, hence 4 CT images are getting processed in parallel. This is done by *convImagePll()* function. Each input image is assigned a CPU core by using *pragma omp parallel numthreads()* directive of OpenMP. Fig. 2 summarises the parallel implementation of image processing using OpenMP.

V. Implementation on Manycore using GPU

While implementing code on GPU, *gpucudaConvImage()* function is written which performs the convolution on device (GPU). Host (CPU) is sending image data to device. By defining the block size of 100x100, each image data is then given to single processor(SP) of device as shown in Fig. 3. The convolution code is then executed and the resultant data returned back to host. The image size is 512x512 and there are 256 cores present in Nvidia Jetson TX2 GPU. As more number of images are getting processed compared to OpenMP approach due to availability of more number of cores in GPU, the execution time gets reduced significantly by this approach.

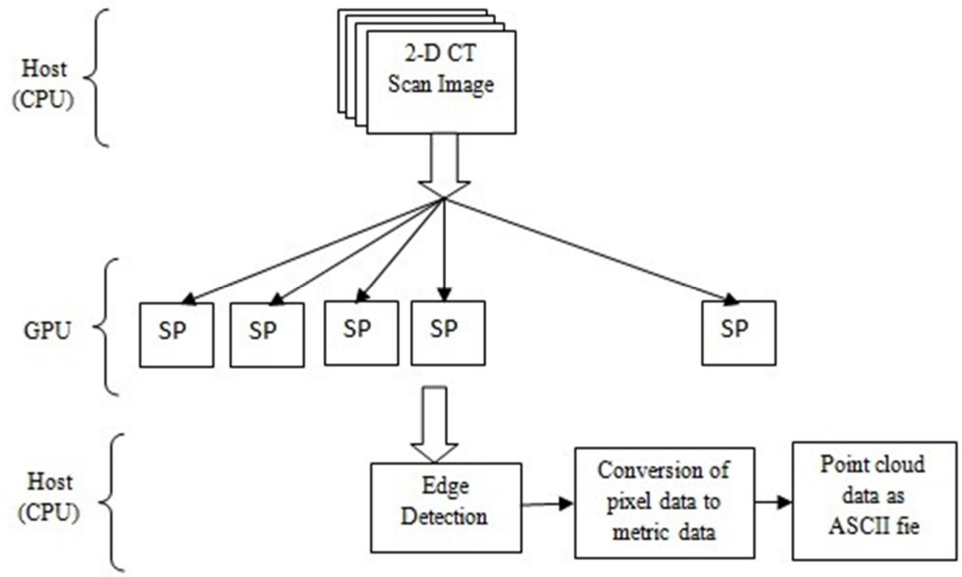

Fig. 3. Block diagram of implementation using GPU

VI. Results

According to Amdahl's law[1] of diminishing returns, the speedup factor is calculated for both above mentioned techniques. Table I shows the performance measure using OpenMP and Table II shows the speedup achieved using GPU. The comparison is made between sequential and parallel execution of code using both techniques. We executed the code for different numbers of CT images ranging from 5 to 1000. While implementing using OpenMP, the number of input images does not affect the speedup factor much. We achieved speedup factor of 2.14 for 5 number of input images whereas speedup factor of 3.09 for 1000 input CT images. Table II shows significant change in speedup factor as number of CT input images varies. We achieved 74.5 speedup factor for 5 CT input images and speedup factor of 107.3 for 1000 CT input images.

TABLE I. Performance Measure using OpenMP

Sr. No.	No. of Images	Sequential Execution time in msec.	Parallel Execution time in msec.	Speed-up Factor
1	5	98.871	46.071	2.146057
2	10	198.555	61.785	3.213644
3	25	495.348	16.025	3.091134
4	50	989.561	317.841	3.113384
5	100	1977.199	621.732	3.180147
6	250	4941.481	1592.979	3.102038
7	500	9880.918	3251.773	3.038625
8	1000	19763.58	6380.584	3.097457

TABLE II. PERFORMANCE COMPARISON OF CPU AND GPU

Sr. No.	No. of Images	Time to compute result on CPU in msec	Time to compute result on GPU in msec.	Speed-Up Factor	Execution time ratio of GPU to CPU
1	5	231.343	3.101	74.589	0.00134
2	10	440.370	5.046	87.258	0.01146
3	25	1068.448	10.826	98.692	0.01013
4	50	2114.749	20.356	103.887	0.00962
5	100	4210.366	39.524	106.525	0.00938
6	500	20952.05	197.659	106.000	0.00943
7	1000	41897.66	390.345	107.334	0.00931

VII. CONCLUSION

The computation time involved in 3-D reconstruction of CT images is reduced by using parallel techniques. GPU achieves better performance compared to OpenMP. The speedup we achieved, reduces the overall execution time significantly. At the same time few challenges such as managing parallelism across various cores and transfer of data between CPU and cores are faced.

ACKNOWLEDGMENT

Authors are thankful to Dr. V. N. Chougule for sharing information about case study and CT scan images we used for demonstrating purpose.

REFERENCES

[1] Gene M. Amdahl. “Validity of the single processor approach to achieving large scale computing capabilities” *IEEE Solid-State Circuits Society Newsletter*, 12, 2007.

[2] Daniel Forsberg Anders Eklund, Paul Dufort and Stephen LaConte. “Medical image processing on gpu-past, present and future” *Elsevier- Journal on Medical Image Analysis*, 17(8):1073–1094, Dec 2013.

[3] Lukas Helfen Daniel Ha¨nschke Tilo Baumbach Anton Myagotin, Alexey Voropaev. “Efficient volume reconstruction for parallel-beam computed laminography by filtered backprojection on multi-core clus- ters”. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 22(12):5348– 5361, DECEMBER 2013.

[4] Alejandro Duran Michael Klemm Sergi Mateo Bel- Lido Stephen L. Olivier Christian Terboven Bronis R. De Supinski, Thomas R. W. Scogland And Timothy G. Mattson. “The ongoing evolution of OpenMP” *Proceedings of the IEEE,* 106:2004 – 2019, Nov. 2018.

[5] Joa˜o Manuel R. S. Tavares Carlos A. S. J. Gulo Antonio C. Sementille. “Techniques of medical image processing and analysis accelerated by high-performance computing: A systematic literature review.” *Springer Journal of Real-Time Image Processing*, 16, 16 November 2017.

[6] A. V. Chalkias Anthony H. Aletras Christos G. Xanthis, Ioannis E. Venetis. Mrisimul: “A GPU-based parallel approach to MRI simula- tions” *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 33(3):607–617, March 2014.

[7] Mikhail Smelyanskiy et. al. “Mapping high-fidelity volume rendering for medical imaging to CPU, GPU and many-core architectures” *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, 15(6):1563–1570, NOVEMBER/DECEMBER 2009.

[8] Yuchou Chang Dong Liang Haifeng Wang, Hanchuan Peng. “A survey of GPU-based acceleration techniques in MRI reconstructions.” *Quantitative Imaging in Medicine and Surgery*, 8(2):196–208, March 2018.

[9] Billy Y.S. Yiu Alfred C.H. Yu Hayden K.-H. So, Junying Chen. “Medical ultrasound imaging: to GPU or not to GPU?” Technical report, University of Hong Kong, IEEE Computer Society, Sept.-Oct. 2011.

[10] Audrey Haihong Gallogly Herng-Hua Chang, Cheng-Yuan Li. “Brain MR image restoration using an automatic trilateral filter with GPU-based acceleration.” *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 65(2):400–413, Feb. 2018.

[11] Markus Kowarschik Holger Scherl, Benjamin Keck and Joachim Hornegger. “Fast GPU-based CT reconstruction using the Common Unified Device Architecture (CUDA).” In *IEEE Nuclear Science Symposium Conference Record,* pages 4464–4466, 2007.

[12] CHE-LUN HUNG 1 HSIN-YI TSAI1, HANYU ZHANG1 and GEY- ONG MIN. “GPU-accelerated features extraction from magnetic res- onance images.” *IEEEAccess SPECIAL SECTION ON ADVANCED SIGNAL PROCESSING METHODS IN MEDICAL IMAGING,* 5, Nov. 2017.

[13] Siau-Chuin Liew Hui Liang Khor and Jasni Mohd. Zain. “A review on parallel medical image processing on GPU.” In *4th International Conference on Software Engineering and Computer Systems (ICSECS)*, Kuantan, Malaysia, 19-21 Aug. 2015 2015.

[14] Jian-Ming Jin Jian Guan, SuYan. "An OpenMP-CUDA implementation of multilevel fast multipole algorithm for electromagnetic simulation on multi-GPU computing systems." *IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION*, 61(7):3607–3616, JULY 2013.

[15] Hangjun Yi-Barbara Chapman Sunita Chandrasekaran Lei Huang, Eric Stotzer. "Parallelizing ultrasound image processing using OpenMP on multicore embedded systems." In *IEEE Global High Tech Congress on Electronics*, Shenzhen, China, 18-20 Nov. 2012 2012.

[16] WANG Dingxing ZHENG Weiming LI Jianjiang, SHU Jiwu CHEN Yongjian. "Analysis of factors affecting execution performance of OpenMP programs." *TSINGHUA SCIENCE AND TECHNOLOGY*, 10(3):304–308, 3 June 2005.

[17] C. P. Bhole M. D. Jones, R.Yao. "Hybrid MPI-OpenMP programming for parallel OSEM PET reconstruction." *IEEE TRANSACTIONS ON NUCLEAR SCIENCE*, 53(5):2752–2758, OCTOBER 2006.

[18] Felix Ritter ; Tobias Boskamp ; A. Homeyer ; Hendrik Laue ; Michael Schwier ; Florian Link ; H.-O. Peitgen. "Medical image analysis." *IEEE Pulse*, 2:60–70, Nov.-Dec. 2011.

[19] Dr. V. S. Inamdar Priyanka M. Bagul, Saurabh R. Chiraniya. "Redefining the role of rc in the field of GPU: A survey." In *International Conference on Information Processing (ICIP)*, Pune, India, 16-19 Dec. 2015 2015.

[20] Sigurd Storve and Hans Torp. "Fast simulation of dynamic ultrasound im- ages using the GPU." *IEEE TRANSACTIONS ON ULTRASONICS, FER- ROELECTRICS, AND FREQUENCY CONTROL*, 64(10):1465–1477, Oct. 2017.

[21] Tobias Schaeffter Michael Schacht Hansen Thomas Sangild Sørensen, David Atkinson. "Real-time reconstruction of sensitivity encoded ra- dial magnetic resonance imaging using a graphics processing unitgpu- accelerated features extraction from magnetic resonance images." *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 28(12):1974–1985, Dec. 2009.

[22] Willem Vermin Jan Menssen Chris L. de Korte Tim Idzenga, Evghenii Gaburov. "Fast 2-D ultrasound strain imaging: The benefits of using a GPU." *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 61(1):207–213, January 2014.

[23] B. B. Ahuja V. N. Chougule, A. V. Mulay. "Three dimensional point cloud generations from CT scan images for bio-cad modeling." In *International Conference on Additive Manufacturing Technologies*, Shenzhen, China, 7-8 October 2013.

[24] B. B. Ahuja V. N. Chougule, A. V. Mulay. "Development of patient specific implants for minimum invasive spine surgeries (miss) from non-invasive imaging techniques by reverse engineering and additive manufacturing techniques." *Elsevier-12th GLOBAL CONGRESS ON MANUFACTURING AND MANAGEMENT*, 2014.

[25] B. B. Ahuja V. N. Chougule, A. V. Mulay. "Conversions of CT scan images into 3D point cloud data for the development of 3D solid model using B-rep scheme." In *Conference: Proceedings of the Intern ational Conference on Precision, Meso, Micro and Nano Engineering*, Calicut, Keala, INDIA, December 2013 December 2013.

[26] Jieun Jeong Van-Giang Nguyen and Soo-Jin Lee. "GPU-accelerated iterative 3D CT reconstruction using exact ray-tracing method for both projection and backprojection." In *IEEE Nuclear Science Symposium and Medical Imaging Conference*, Seoul, South Korea, 27 Oct.-2 Nov. 2013 2013.

[27] Bharatkumar Ahuja Vikas Chougule, Arati Mulay. "Methodologies for development of patient specific bone models from human body CT scans" *Springer Journal of The Institution of Engineers (India)*, Series C(99):413–418, 22 June 2016.

[28] Aly A. Farag YasserM. Kadah, Khaled Z. Abd-Elmoniem. "Parallel computation in medical imaging applications." *International Journal of Biomedical Imaging*, 2011, January 2011.