MEDICAL IMAGING COMPUTING BASED ON GRAPHICAL PROCESSING UNITS FOR HIGH PERFORMANCE COMPUTING

K.Suresh[1], L.Gangadhar[2], M.Vidya[3]

[1]*Assistant Professor, Department of IT, AITS, Autonomous Institution, Rajampet, Kadapa (Dt.,) AP.*
[2]*Student of Department of IT, AITS, Autonomous Institution, Rajampet, Kadapa (Dt.,) AP*
[3]*Student of Department of IT, AITS, Autonomous Institution, Rajampet, Kadapa (Dt.,) AP*

Abstract

The Design of GPU(Graphical Processing Unit) will well suitable for express the data parallel computations because GPU will specialized for parallel and today's digital images in medical are huge volume of collections in every day, however medical imaging produces demand to improve the medical diagnosis and procedures. This survey is provide graphical processing computations and hardware require to compute and give better information for diagnosis of Cancer Treatment using Radiation Therapy. It is important techniques to increase quality of medical image data clinically under pressure to make enriched data and improve accurate treatment and decrease the complications.

Keywords— *High Performance Computing, CPU, GPU, Medical Image Computing.*

----------***----------

1. INTRODUCTION

Computer assisted medical treatment is more impact on now a days to diagnosis the patients in lifesaving .visualization techniques will help the doctors to minimize the overheads and better understanding with accurate manner, most challenging role of a doctor is to perform successfully operations taking decisions, potentially lifesaving with the help of medical imaging. Medical imaging plays a vital role to minimize the overheads of clinicians all over the world. Clinicians are under pressure by evolvement of medical imaging and understanding the problems of patients in an accurate manner. This can be clearly observed in the area of medical imaging computing, the associated diagnosis and medical treatment. The visualization system needs to be clear and gives quality imaging to perform the operations in desired manner. Central Processing Unit (CPU) has serial von Neumann processor ,which is highly optimized to perform series of computations .Replacing today's computers systems are taking a new turn unlike traditional processors with Graphical Processing Unit (GPU) because GPU[2] can handle Massive parallelism to compute high performance accelerator platform for parallel computing ,especially with efficient framework with excellent performance faster and fine for demanding tasks . GPU can increase performance when compared to CPU's which are serving in medical imaging because GPU perform and give quality digital images which help doctors for in treatment.

GPU has graphics pipeline to fixed functionality with limited configuration for implement hardware efficiency this restriction helps for the programmers to allow different stages in high level programming languages. This impact to able creates unique style of each application and make different stages of rendering pipeline [3]. The high level programming languages are associated with different graphics application programming interfaces such as OpenGL, DirectX [12] [13] [14].

Modern medical imaging technologies are one of the important fields for physicians [4]. For analyse modalities with various dimensions or common attributes, the transformation need to be extracted from the imaging data for common clinical work flow to understand the patient diagnostic purpose. This is a common procedure of treatment to scan the image for quick diagnosis decisions towards more efficient image computation techniques [1]. This required to device an algorithm with computational efficiency and make use of optimum computational power of the hardware.

The rest paper is organized as follows Section 2 discussed Graphical Computing via Graphical Processing Unit and Section 3 Medical Image Context to be consider all applications of related to medical with help of GPU and finally GPU based medical imaging .

2. GPU COMPUTING

Intel introduced the technology which allows CPU to slow down, suspend or shut down [3]. The demand of hand held devices also increased in early 90's. To minimize power consumption scientists started paying attention on sequential improvement of hardware and software both. It has combined to increase the performance and battery lifetime of hand held devices [4].In terms of development of personal computers power management there are three specifications power

management, System design and Application. These are 3-interlocked tracks that helped in the development [5]. In terms of computing systems the power and performance management mainly highlights computer architecture, operating systems, CAD, compilers etc. Here, the description starts from operating system first. Operating systems have basically two major roles, providing an abstraction and managing resources like CPU time, memory allocation, disk quota etc. On OS level an efficient power performance can be enhanced by Power Management on IO Devices, Low-Power Scheduling for Processors or by Reducing Memory Power [6].

The era of the principles of Moore's Law is a single core processor is rapidly closing due to various things including the memory and clock rate. To overcome these limitations, industry trends have moved to multicore and many-core processors. Continued development of multicore and massively multiprocessing architectures in recent years holds great promise, such as IBM, AMD [19] and Oracle-SUN are presenting prototypes with as many as 80 cores that can theoretically achieve more than 1 Teraflops of GPU computing performance. The current is a 100-core processor available for general-purpose and high-performance computing for the years; the GPU has evolved from a highly specialized pixel processor to more flexible, highly programmable architecture that can perform a wide range of data parallel operations [4]. Next coming years later, the task of transforming the geometry was also moved from the CPU to the GPU, one of the first steps toward the modern graphics pipeline. Because the processing of vertices and pixels is inherently parallel, the number of dedicated processing units increased rapidly, allowing commodity PCs to render ever more complex 3-D scenes [22] in tens of milliseconds.

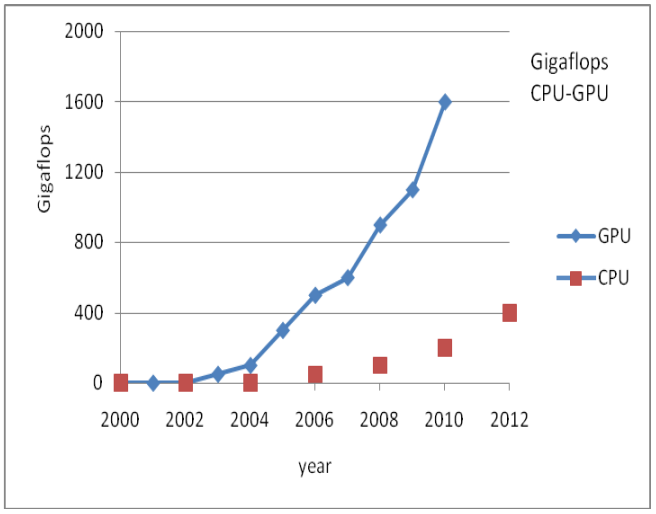

Fig 1 a. Comparison of CPU and GPU in terms of Gigaflops

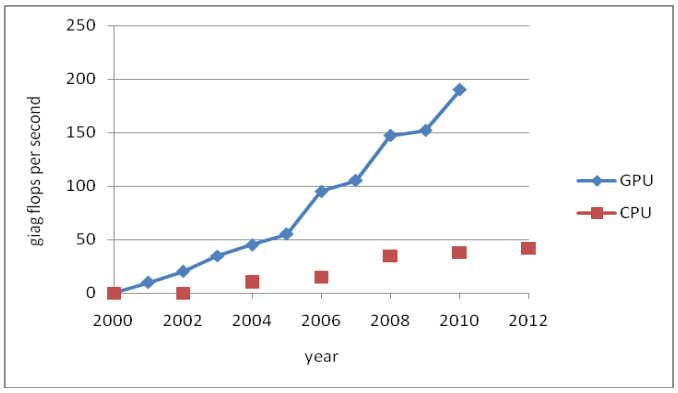

Fig 1.b. Performance Comparison in terms of Bandwidth

As shown in Fig.1 (a) and 1. (b) Of these focus since 2000, the number of compute cores in GPU processors has doubled roughly every 16 months. Over the same period, GPU cores have become increasingly sophisticated and versatile, enriching their instruction set with a wide variety of control flow mechanisms, support for double-precision floating-point arithmetic, built-in mathematical functions, a shared-memory model for inter thread communications, atomic operations, and so forth. In order to sustain the increased computation throughput, the GPU memory bandwidth has doubled every 19 months, and recent GPUs can achieve a peak memory bandwidth of 408 GB/s . With more computing cores, the peak performance of GPUs, measured in billion floating-point operations per second (GFLOPS)[5], has been steadily increasing .

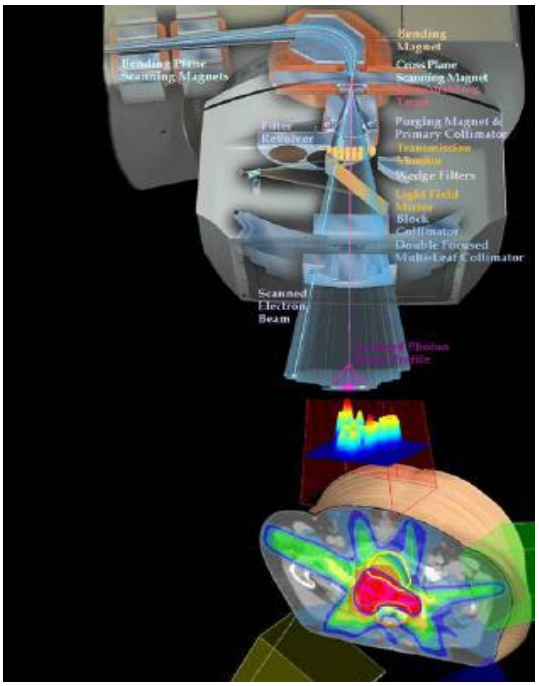

Fig 2 GPU based Technologies for Medical Image

2.1 Objectives

1. To develop high level parallel design for Reducing the radiation from CT scans.
2. Efficient implementation corresponding to patterns that can be easily reconstruction CT scan and reduces radiation by 35-70x cpu takes around 2hours take process, but CUDA[11][22] take 2minutes and clinically practical.
3. Developing the adaptation mechanism that dynamically Operating on a beating heart, only 2% of surgeons can operate on a beating heart patient stands to lose1 point of IQ every 10minutes with heart stopped. GPU enables real time motion compensation to virtually stop beating heart for surgeons.
4. To define a virtualisation of the parallel software in the form of low energy consumption.

GPU programming model is different from CPU programming because it has unique hardware supporting, however, whilst speedup over CPU code, to achieve a scalable high performance code hardware resources efficiently is still a difficult task.

Main bottlenecks are GPU handling massive threaded architecture is used to hide memory latency

3. MEDICAL IMAGING CONTEXT

In coming Surgical intervention and clinical diagnostic systems will be benefit from modern modalities and visualization tools based on GPU technologies for example Cancer Treatment consuming time on Conventional Radiotherapy it took days but based GPU technology it took less time within hours .

Throughout the years, medical science has evolved towards providing a better and longer life for every human being. Medicine has taken technological advances which make available new tools and methods. Medical imaging is one important field fruitfully used nowadays. Doctors can now have visualization equipment that gives them more information for diagnosis. Ultrasound visualization and magnetic resonance imaging are broadly used examples of medical imaging. Such techniques require intensive computation power that may imply trade-offs on quality for achieving a reasonable performance based on Figure 2 and Figure 3. However, the surfacing of general purpose graphic processing units leads to a new software paradigm that can handle a larger bulk of intense computation requirements. Medical imaging is broadly used as a diagnosis tool. X-rays and MRI give doctors a better understanding of the condition of the patient and help them decide the best way to treat them. There is still a lot of research currently focused on improving these techniques. Engineers are working on new and improved algorithms for processing the information and providing doctors with better visualization tools[8].

The continued development of multicore and massively multiprocessing architectures in recent years holds great promise for interventional setups. In particular, massively multiprocessing graphics units with general-purpose programming capabilities have emerged as front runners for low-cost high-performance processing. HPC, in the order of 1 TFLOPS, is available on commodity single-chip graphics processing units (GPUs) with power requirements not much greater than an office computer. Multi-GPU systems with up to eight GPUs can be built in a single host and can provide a nominal processing capacity of eight TFLOPS with less than 1,500 W power consumption under full load

GPU implementations to be more challenging than multicore CPU implementation and in terms of achievable performance gain[6].

Hardware and architectural complexities in designing multicore systems aside, perhaps as big a challenge is an overhaul of existing application design methodologies to allow efficient implementation on a range of massively multicore architectures. As one quickly might find, direct adaptation of existing serial algorithms is more often than not neither possible due to hardware constraints nor computationally justified.

An important amount of the available algorithms require very intense computations over huge amounts of data. When these algorithms are processed by CPU implementations, the time consumed is too long because of the sequential nature of the CPU architecture. I.e. consecutively all computations are executed on each piece of data even if such pieces of data have no dependencies between them. This approach may still be useful for procedures like MRI visualization where the time to get results is not so critical. However, when we talk about interventions, the response time plays an important role.

A fresh approach consists of taking advantage of the intrinsic parallel nature of the processing (Since same computations may be done on different pieces of data at the same time).GPGPU implementation is a promising approach for achieving medical imaging with a better time performance with no quality sacrifices) from the parallel nature of image processing.

The improvement in performance due to the usage of GPGPUs may allow to get results in shorter periods of time. The processing time reduction may give room for larger data sets which contain information at a higher resolution. In conclusion, the introduction of a GPGPU[16] implementation for the medical imaging algorithms can produce better quality images in less time. Furthermore, medical imaging for interventions require a real time (RT) application with hard teal time constraints and low latency.

RT medical imaging implies a major challenge for engineers. While diagnosis visualizations (e.g. Radiographies) are softly constrained in time, visualization tools for interventions must be very precise. Visualization tools that help doctors on medical procedures have very tight constraints as human lives are at stake during the procedures. There is no room for delays or

imprecise timing and doctors expect a good visualization quality. In this scenario, a GPGPU implementation becomes more than an alternative for getting results in a shorter time; it becomes the option for meeting the real-time constraints that medical interventions require while keeping a good quality.

Medical imaging techniques such as CT, MRI and x-ray imaging are a crucial component of modern diagnostics and treatment [20]. As a result, many automated methods involving digital image processing have been developed for the medical field. Image segmentation is the process of ending the boundaries of one or more objects or regions of interest in an image.

Modern graphics processing units a high performance platform for accelerating the variation level set method, which, in its simplest sense, consists of a large number of parallel computations over a grid. NVIDIA's CUDA framework for general purpose computation [15] on GPUs was used in conjunction with three different NVIDIA GPUs to reduce processing time by11 . This speedup was sufficient to allow real-time segmentation at moderate cost.

Geometry usually a collection of triangles is generated by an application on the host processor, and handed onto a graphics processor to be rendered. The first stage of the pipeline operates on triangle vertices, mapping them from three-dimensional scene space to two-dimensional display space. Since scenes can consist of millions of triangles, and each mapping operation is completely independent, parallel hardware is important.

Early GPUs consisted entirely of fixed function hardware and thus served little purpose when they were not being used to generate graphics. The only programmability was in what scene and texture data were passed to them [12]. Over time, programmability has been Apply texture (optional) Rasterizing (convert geometry to pixels) Composite (combine fragments into image) added to GPUs to enable more inventive and life-like effects[8]. Both the vertex and fragment processing engines were made programmable to allow dynamic scene transformation and innovative lighting techniques. In each new hardware release, the features of the programmable portion of the hardware were expanded, allowing longer programs, more complex control, and higher precision. NVIDIA's G80 hardware was significant on this front, as it used programmable execution cores as its basis, almost entirely eliminating fixed-function units[10]. Additionally, it debuted unified shader architecture: rather than having separate programmable units for vertex and fragment operations, the same collection of processing elements could operate on any data. This unified pool of processing resources makes GPUs much more capable of general-purpose workloads than they were previously.

4. GPU COMPUTING TRENDS

The GPU is consuming the a part of high performance computing it is more flexible ,accurate and required complete designing portion of High performance computing, actually GPU erand CPU are different types of processor that can operate parallel or asynchronous that is it can operate simultaneously so it can help get heterogeneous computing. Based on CPU and GPU is better resource managing to the efficient tasks with multiple resources, each resource performance tasks in which best suited.

The CPU then continues to perform CPU-side calculations simultaneously as the GPU processes its operations, and only synchronizes with the GPU when its results are needed. There is also support for independent streams, which can execute their operations simultaneously as long as they obey their own streams order. Current GPUs support up-to 16 concurrent kernel launches [18], which means that we can both have data parallelism, in terms of a computational grid of blocks, and task parallelism, in terms of different concurrent kernels. GPUs furthermore support overlapping memory copies between the CPU and the GPU and kernel execution. This means that we can simultaneously copy data from the CPU to the GPU, execute 16 different kernels, and copy data from the GPU back to the CPU if all these operations are scheduled properly to different streams.

Medical imaging techniques such as CT and MRI scans generate very large sets of data. The large volume of data present results in high computation times, so GPUs have frequently been used reduce this time. Computed Tomography (CT) scans are a method of capturing three dimensional images of body structures. These scans are done by rotating an x-ray capture system around the target, capturing a large sequence of two-dimensional radiographic images. Computer 38 systems are used to construct a volumetric image from these scans.

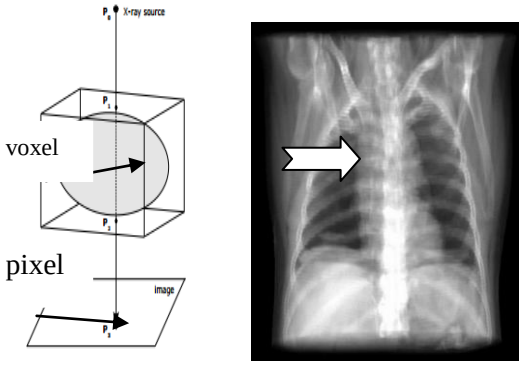

Fig 3 GPU based Radiographs

GPU-based radiographs become digital pixel for constructing the traditional and new era GPU based Radiography based on fig.3 show the GPU based radiographs require 1/10 of CPU

based Construction,Cone Beam CT reconstruction application. This application used an NVIDIA GeForce 8800GT GPU [9], and was able to compute a complete 5123pixel reconstruction in 12.61 seconds. This compares favourably with an earlier CPU-based result of 201 seconds.

Another application of GPU-based Cone Beam CT reconstruction [23], demonstrated a speedup from 178 seconds with a CPU implementation to 53 seconds using an NVIDIA Quadro FX 4500 GPU. This GPU used an earlier architecture and thus had lower performance than the GeForce 8800GT used in [23].In interactive random walk-based image segmentation algorithm, implemented both on a GPU and a standard CPU. In this approach, a user places several seeds at locations inside and outside the segmentation target. A pixel is classified as part of the segmentation target if a random walk from that pixel is more likely to arrive at a target seed than a non-target seed. This algorithm's performance varies based on the number and placement of seeds, and this is rejected in segmentation time. CPU-based segmentation time is vary from 3to83 seconds across differing images, segmentation targets, and seed placement, while equivalent GPU-based segmentation times vary from 0.3{1.5seconds}.

Table 1 Different Clinical Tasks to Human Body with Techniques

Clinical Task	***Techniques***
Image acquisition	CT, MRI, US
Visualization of scans in their entirety	Direct volume rendering
Multi-modality rendering, segmentation of different tissues or organs	Tagged volume rendering
Visualization of brous tissue	Fiber tracking
Grouping of anatomically similar structures	Clustering
Visualization of vessels	Model-based reconstruction, implicit visualization
Associate preoperative data with patient in operation room	Tracking, registration algorithms, intra-operative imaging
Medical simulations, Training	Virtual and mixed reality

5. CONCLUSIONS

Research in this area continues to be motivated by the need to minimize the overhead mismatched diagnose. the race for higher clock-rates within the CPU industry has come to an end and processors started to become rather “wider than faster”, research results in general purpose computation on graphics hardware are an important factor towards a possible adoption of GPU technology by CPU manufactures. Novel processor designs with a multitude of cores started to be available (IBM cell processor) or announced, which will be interesting platforms for algorithms that require a rather hybrid processor design. At this point, developers still have to decide for a platform for massively parallel processing, be that a single GPU, or multiple processors organized in clusters. As OpenCL [21] receives broad support from all processor manufacturers, it appears as an emerging standard for programming parallel architectures. Such standardization might allow the reduction of processor specific programming efforts.

REFERENCES

[1] U. Pietrzyk, K. Herholz, A. Schuster, H.-M.v. Stockhausen, H. Lucht, W.-D. Heiss, Clinical applications of registration and fusion of multimodality brain images from PET, SPECT, CT,and MRI, Eur. J. Radiol. 21 (3) (1996) 174–182.

[2] R. Shams, P. Sadeghi, R.A. Kennedy, R.I. Hartley, A survey of medical image registration on multicore and the GPU, IEEE Signal Process. Mag. 27 (2) (2010) 50–60.

[3] J. Tsao, Interpolation artifacts in multimodality image registration based on maximization of mutual information,IEEE Trans. Med. Imaging 22 (7) (2003) 854–864,doi:10.1109/TMI.2003.815077.

[4] J.P.W. Pluim, J.B.A. Maintz, M.A. Viergever, Interpolation artefacts in mutual information-based image registration,Comput. Vis. Image Underst. 77 (9) (2000)211–232.

[5] GPGPU, Website of general purpose computation on theGPU, http://www.gpgpu.org.

[6] J.D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger,A.E. Lefohn, T.J. Purcell, A survey of general-purposecomputation on graphics hardware, Comput. Graph. Forum26 (1) (2007) 80–113.

[7] A. Thall, Extended-precision floating-point numbers for GPU computation, in: SIGGRAPH’06: ACM SIGGRAPH 2006Research Posters, ACM, New York, NY, USA,2006,p.52.doi:http://doi.acm.org/10.1145/1179622.1179682.

[8] C. Vetter, C. Guetter, C. Xu, R. Westermann, Non-rigid multi-modal registration on the GPU, in: Proceedings of SPIE Medical Imaging, vol. 6512, 2007, pp. 651228-1–651228-8.

[9] Z. Fan, C. Vetter, C. Guetter, D. Yu, R. Westermann, A.Kaufman, C. Xu, Optimized GPU implementation of

learning-based non-rigid multi-modal registration, in: Proceedings of SPIE Medical Imaging, 2008, pp.69142Y-1–169142Y-10.

[10] V. Podlozhnyuk, Histogram calculation in CUDA, Tech. rep.,NVidia, 2007.

[11] R. Shams, N. Barnes, Speeding up mutual information computation using NVIDIA CUDA hardware, in: DICTA'07:Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, IEEE Computer Society, Washington, DC, USA, 2007, pp. 555–560.

[12] R.J. Rost, OpenGL Shading Language, 2nd edition, Addison-Wesley Professional, 2006.

[13] J. Neider, T. Davis, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Release 1, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993.

[14] D. Blythe, The Direct3D 10 system, ACM Trans. Graph. 25 (3) (2006) 724–734, doi:http://doi.acm.org/10.1145/1141911.1141947.

[15] W.R. Mark, R.S. Glanville, K. Akeley, M.J. Kilgard, Cg: a system for programming graphics hardware in a C-like language, in: SIGGRAPH'03: ACM SIGGRAPH, ACM Press, New York, NY, USA, 2003, pp. 896–907,doi:http://doi.acm.org/10.1145/1201775.882362.

[16] I. Buck, T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston, P. Hanrahan, Brook for GPUs: stream computing on graphics hardware, ACM Trans. Graph. 23 (3) (2004) 777–786,doi:http://doi.acm.org/10.1145/1015706.1015800.

[17] J. Owens, Streaming architectures and technology trends, in: GPU Gems 2, Addison-Wesley Professional, 2005, pp. 457–470.AMD,CALSDK,http://ati.amd.com/developer/.

[18] G.C. Sharp, N. Kandasamy, H. Singh, M. Folkert, GPU-based streaming architectures for fast cone-beam ct image reconstruction and demons deformable registration, Phys.Med. Biol. 52 (19) (2004) 5771–5783.

[19] Khronos Group, The OpenCL specification, http://www.khronos.org/opencl/.

[20] R. Strzodka, M. Droske, M. Rumpf, Fast image registration in DX9 graphics hardware, J. Med. Inform. Technol. 6 (2003) 43–49.

[21] A. Köhn, J. Drexl, F. Ritter, M. Koenig, H.-O. Peitgen, GPU accelerated image registration in two and three dimensions, in: Bildverarbeitung für die Medizin, Informatik Aktuell, Springer, 2006, pp. 261–265.

[22] N. Courty, P. Hellier, Accelerating 3D non-rigid registration using graphics hardware, Int. J. Image Graph. 8 (1) (2008) 1–18.

[23] P. Muyan-Özc¸elik, J.D. Owens, J. Xia, S.S. Samant, Fast deformable registration on the GPU: a CUDA implementation of demons, in: The 2008 International Conference on Computational Science and its Applications,ICCSA 2008, IEEE Computer Society, 2008, pp. 223–233.