JOURNAL OF NANO- AND ELECTRONIC PHYSICS
Vol. **6** No 3, 03050(4pp) (2014)

ЖУРНАЛ НАНО- ТА ЕЛЕКТРОННОЇ ФІЗИКИ
Том **6** № 3, 03050(4cc) (2014)

The Use of GPUs for Solving the Computed Tomography Problem

A.E. Kovtanyuk[1,2,*]

[1] *Far Eastern Federal University, 8, Sukhanova Str., 690950 Vladivostok, Russia*
[2] *Institute for Applied Mathematics, 7, Radio Str., 690041 Vladivostok, Russia*

(Received 19 May 2014; revised manuscript received 03 July 2014; published online 15 July 2014)

Computed tomography (CT) is a widespread method used to study the internal structure of objects. The method has applications in medicine, industry and other fields of human activity. In particular, Electronic Imaging, as a species CT, can be used to restore the structure of nanosized objects. Accurate and rapid results are in high demand in modern science. However, there are computational limitations that bound the possible usefulness of CT. On the other hand, the introduction of high-performance calculations using Graphics Processing Units (GPUs) provides improving quality and performance of computed tomography investigations. Moreover, parallel computing with GPUs gives significantly higher computation speeds when compared with (Central Processing Units) CPUs, because of architectural advantages of the former. In this paper a computed tomography method of recovering the image using parallel computations powered by NVIDIA CUDA technology is considered. The implementation of this approach significantly reduces the required time for solving the CT problem.

Keywords: Electronic imaging, Computed tomography, Parallel computing, Graphics processing units.

PACS numbers: 81.70.Tx, 02.30.Zz

1. INTRODUCTION

The main limitations on the development of numerical methods for solving complex problems are the huge amount of time required for calculations and the inaccessibility and high cost of powerful computing cluster systems.

Despite the work of leading research organizations in the development of computing systems to improve the cost / benefit ratio of CT investigations, the demand for higher performance remains. In this case, an innovative comprehensive development environment, CUDA, created by NVIDIA provides a way forward [1]. The main new feature of CUDA is the option to send C, C++ and Fortran code directly to the Graphics Processing Units (GPUs), with no assembly language required. Furthermore, this technology is available to anybody with graphics processing units compatible with CUDA.

A classic example of an unrealized task was the project of TechniScan Medical Systems [2]. The goal of the company was to recover a 3D image of the mammary glands of patients to diagnose diseases. The raw data for the image was obtained using ultrasonography. However, the use of CT methodology was not possible due to the high requirements on system resources, until NVIDIA released high performance GPUs, with CUDA.

The introduction of computed tomography (CT) has allowed researchers to reconstruct 2D and 3D representations of the objects of analysis by stratified non-destructive influence. This method is widespread in medicine, microscopy, petrophysics and geophysics. But the computational complexity of CT is still extremely high.

In CT the level of detail of an object depends on the step size of the grid chosen for scanning and calculating. For two-dimensional images the computational complexity of the algorithm is equal to $O(n^3)$, where n is the dimension of the grid chosen. In medicine and microscopy the level of detail of a research object is crucial if it is to be used to make informed decisions. Thus, to obtain the two-dimensional internal structure of an object with a diameter equal to 1 meter and with 10^{-3} mm step size the order of the number of calculations is 10^{18}, which is unacceptable for modern CPUs (Central Processing Units), even taking into account multi-core systems. Note, that filtering noise caused by the scattering by various algorithm (See, for example, [3-5]) requires additional computational cost.

The implementation of computed tomography data processing on GPUs using NVIDIA CUDA provides a significant improvement to performance because of features of the algorithm. Another example of the successful introduction of computing with CUDA is the Procter & Gamble Company. Its research center modeled the efficiency of surface-active agents (surfactants). The researchers, by using two NVIDIA Tesla GPUs, archieved performance equal to the supercomputer Cray XT3 with 128 processors, as well as the IBM BlueGene / L with 1024 processors [6]. In terms of economic advantages, it is important to understand that the energy consumption is proportional to the fourth power of the frequency of a processor [7]. Hence, if the clock frequency increases twofold, the electric energy consumption will raise 16 fold, as well as the heat released.

In recent years, the growth rate of the CPU frequency has dropped significantly. Therefore, there has been a new trend to increase the number of cores. In terms of the number of cores, it is clear that GPU's are far superior to CPU's, for example AMD Opteron 6386S with 16 cores and 2.8 GHz is inferior to NVIDIA GeForce GTX690 with 3072 CUDA cores and total frequency equal to 915 MHz. AMD and Intel recognize that the race to increase the number of cores will end soon, preferring heterogeneous systems, combining the Central and Graphics Processing Unit on one chip.

* kovtanyuk.ae@dvfu.ru

The article was reported at the International Conference «The Advanced Technology, Equipment and Analytical Systems for Materials», Kursk, 13-14 May, 2014

2. COMPUTATIONAL ALGORITHM FOR IMAGE RECOVERY OF COMPUTED TOPOGRAPHY PROBLEMS

Let us consider in more detail the reconstruction algorithm of an object of research. The mathematical basis of the method was proposed by I. Radon in 1917 [8], which was based on the idea of recovering multivariate functions by its integral characteristics. However, the world's first X-ray tomography was demonstrated by G. Hounsfield [9] in 1972, since by that time there were X-ray machines performing a large number of high quality images, as well as a computer that could handle these images.

According to Radon's methods, the structure of the object can be reconstructed from a series of parallel cross-sections with penetrating radiation. Suppose that we need to determine the density distribution of matter $f(x_1,x_2)$ in the cross section of the object. The object is irradiated by a beams in the various directions. The reduction of the rays is proportional to the density of the matter. The residual intensity of the rays passing through the object along the straight line is fixed on the opposite side of the emitter. Suppose that as a result of irradiating at various angles, a set of integral projections of the function f was obtained. The task of reconstructing the internal structure of the object is simplified to the inversion of the Radon transform.

To restore the two-dimensional structure of the object under study from data of irradiation, we will use the parallel scheme of scanning. In this case, irradiation occurs at different angles of inclination of the studied object. For a fixed angle, radiation sources and detectors are located at the parallel lines (see Fig.1). For single angle, the number of radiographic lines is equal to $2q + 1$. The number of radiographic angles is equal to p.

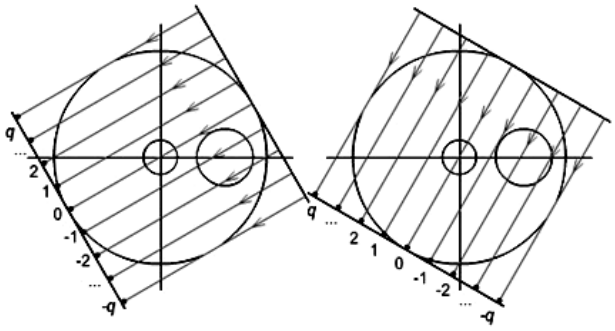

Fig. 1 – Parallel scanning scheme

As a result of scanning, a two dimensional array of data $g_{j,k}$ is obtained, where $j = 1,...,p$ is an index indicated the number of angle of irradiation, and $k = -q,...,q$ is an index of radiographic line.

To solve the problem of computed tomography, we apply the algorithm of convolution and back projection, which consists of two steps.

Step 1. Calculating the convolution.

For all indices $j = 1,..., p$, and $k = -q,..., q$ to calculate the value of the convolution

$$v_{j,k} = h\sum_{i=-q}^{q} w_b(h(k-i))g_{j,i}), \quad k = -q,...,q.$$

Here, h is a distance between two parallel radiographic lines, w_b is a filter, b is a given spectral width [5]. For example, b is equal to π / h in the case, if this value is less than the number of angles of irradiations, and

$$w_b(h(k-i)) = 1/\pi^2 h^2(1-4(k-i)^2).$$

Step 2. Calculating the back projection.

For each point of reconstruction we calculate discrete back projection

$$f(x) = \frac{2\pi}{p}\sum_{j=1}^{p}((1-u)v_{j,k} + uv_{j,k+1}),$$

where

$$u = \frac{s}{h} - k\,,\ s = x_1\cos\phi + x_2\sin\phi,\ \ \phi = j\pi / p\,,$$

$$k \le \frac{s}{h} \le k+1\,.$$

Some aspects of the implementation of the algorithm on GPUs are presented in the next section.

3. USING GRAPHICS PROCESSING UNITS FOR PARALLEL IMPLEMENTATION OF THE RECONSTRUCTION ALGORITHM

According to steps 1-2 of the reconstruction algorithm, the solving the CT problem reduces to a serial computation of matrixes $v_{j,k}$ and the function $f(x)$ on a given grid with dimension $N \times N$. An important computational feature of the method is the independence of computing $v_{j,k}$ for each index. Similarly, this property holds for computing matrix $f_{m,n}$, which is a discrete representation of $f(x)$ on a given grid. Thus, the calculation of each element of the matrix $v_{j,k}$ and the function $f(x)$ can be done independently. From the point of view of effective use of the CUDA technology, those arrays are distributed into groups. One of the simplest ways is a partition by the rows or columns. Hence, each row or column of the matrixes $v_{j,k}$ and $f_{m,n}$ is calculated by blocks. Grouped items are transferred for execution in the streaming multiprocessor CUDA (SM) GPU. The presence of many SMs allows processing blocks in parallel. Furthermore, each multiprocessor includes extra scalar processors (SPs), providing additional parallelization. A feature of NVIDIA CUDA is the formation of blocks of tasks in the form of pools, called warp. The size of warp for modern GPUs that support CUDA is 32 threads. Tasks within a pool (warp) are executed in SIMD (Single Instruction Multiple Data) style, i.e. each thread inside the warp be able to carry out only one instruction. Thus, providing for each element of the matrixes $v_{j,k}$ and $f_{m,n}$ a sequence of instructions required to obtain the values, CUDA ena-

bles the same operation for the group of the elements independently and in parallel, thereby improving performance. The number of available multiprocessors and scalar processors on the GPU plays a key role in estimating the improved performance. An additional constraint is the amount of available memory, but additional fragmentation processes can eliminate this concern.

Let us consider implementation of both steps of reconstruction algorithm on GPUs for Cormack's phantom [10]. To test the program code the graphics nVidia GeForce GTX 660M (CUDA Capability 3.0) is used.

Implementing the first step.

Below are the results of testing the function that implements the first step of the algorithm for different block sizes (blockDim.x) and different number of blocks (gridDim.x) (see Table 1). Here, *timeout* is an error that occurred when it has exceeded the maximum computation time, default value is 10 seconds; *blocksize* is an error generated in excess of the allowable number of threads per block.

Table 1 – Runtime of the first step of the algorithm depending on the size and number of blocks

Test number	blockDim.x	gridDim.x	Runtime
1	1	1	*timeout*
2	2	1	8.12078
3	4	1	4.26376
4	8	1	2.33119
5	16	1	1.30424
6	32	1	0.749946
7	64	1	0.441025
8	128	1	0.257105
9	256	1	0.180325
10	512	1	0.108049
11	1024	1	0.054956
12	2048	1	*blocksize*
13	1024	2	0.0275024
14	1024	4	0.0275249
15	512	4	0.0275238
16	256	8	0.0275835
17	128	16	0.0275793
18	64	32	0.0275684
19	32	64	0.0540881
20	16	128	0.100913

To optimize the program code, we introduced temporary variables to store some intermediate results of calculations that reduce runtime in 4,226 times. In addition, to reduce the computation inside the function, all possible computations that do not depend on the indices were calculated on the host processor and placed in constant memory.

If amount of radiographic lines for single angle, $2q+1$ is less than the maximum number of threads per block, it is possible to compute the convolutions for each angle in a separate block. For this we perform the following optimization actions:

Radiographic data for single angle of irradiation must be multiples of 128 bits (see Fig. 2)

Taking into account that size of the warp is 32, determine the block size as $32s$, where $32(s-1) < 2q+1 \le 32s$.

Before computing the convolution for a single angle

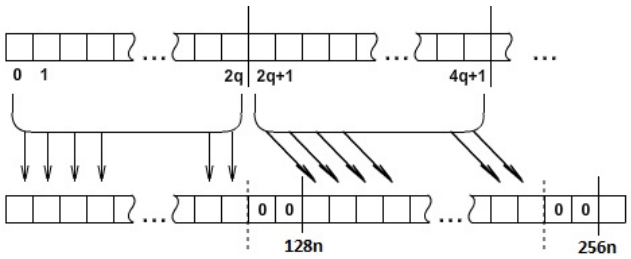

Fig. 2 – Scheme of storing data

of irradiation, radiographic data are copied from global memory to shared memory.

Implementing the second step.

Computing the back projection is parallelized similarly to the first step, except that the use of shared memory is impossible, since the reconstruction of a single pixel of image requires radiographic data of all angles of irradiations.

Note, there is ability to recover the three-dimensional structure of the object based on the two-dimensional reconstructions obtained for a set of cross sections using the considered algorithm.

Here, in sequential computations, time spent on reconstruction in a single cross-section decreases with increasing of the cross section No. (see Table 2).

Table 2 – Runtime of imaging of single cross section

No.cross section	Runtime	No.cross section	Runtime
1	0.0685953	26	0.0254455
2	0.0661115	27	0.0254391
3	0.0545759	28	0.0250839
4	0.0370371	29	0.024842
5	0.0359608	30	0.0253934
6	0.0317436	31	0.0248262
7	0.0291794	32	0.0242855
8	0.0292195	33	0.0241654
9	0.0290545	34	0.0240534
10	0.0287304	35	0.0241818
11	0.0286961	36	0.0238183
12	0.028704	37	0.0234513
13	0.0278976	38	0.023478
14	0.0277789	39	0.0236015
15	0.0277146	40	0.0233004
16	0.0276469	41	0.0230786
17	0.0274842	42	0.0231629
18	0.0268462	43	0.0230548
19	0.027262	44	0.0232344
20	0.0266465	45	0.022733
21	0.0261585	46	0.0226073
22	0.0262316	47	0.0226477
23	0.0262062	48	0.0225259
24	0.0256861	49	0.0224326
25	0.0253958	50	0.0220398

4. CONCLUSIONS

Thus, the development of computing technology on graphics processors can significantly reduce the computational time required for image reconstruction of CT or Electronic Imaging problems. In addition, the described approach can effectively restore the image on a grid of a higher dimension, thereby improving the accuracy and quality of research in applied sciences. However, the development of algorithms to efficiently use shared and global memory requires more attention.

ACKNOWLEDGEMENT

This work was supported by the Scientific Fund of the Far Eastern Federal University, by the Scientific Program "Far East", and by the Russian Foundation for Basic Research (project 13-01-00275).

REFERENCES

1. R. Farber, *CUDA Application Design and Development* (Elsevier Inc.: 2011).
2. T. Valich, *Tom's Hardware US*, http://www.tomshardware.com/news/Nvidia-Tesla-Techniscan,5653.html (2008).
3. D.S. Anikonov, A.E. Kovtanyuk, I.V. Prokhorov, *Transport Equation and Tomography* (Utrecht-Boston: VSP: 2002).
4. A.E. Kovtanyuk, I.V. Prokhorov, *J. Inv. Ill-Posed Probl.* **14** No 6, 609 (2006).
5. A.E. Kovtanyuk, I.V. Prokhorov, *Num. Analysis Appl.* **1** No 1, 46 (2008).
6. D. Sanders, E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming* (New York: Addison-Wesley Publishiong: 2011).
7. A.V. Boreskov, A.A. Kharlamov, *The basics of working with the CUDA technology* (Moscow: DMK: 2010).
8. F. Natterer, *The mathematics of computerized tomography* (New York: John Wiley and Sons Inc.: 1986).
9. G.N. Hounsfield, *Br. J. Radiol.* **46**, 1016 (1973).
10. A.M. Cormack, *J. Appl. Phys.* **34**, 2722 (1963).