Journal of Innovative Optical Health Sciences
Vol. 3, No. 1 (2010) 39–43

FAST RECONSTRUCTION METHOD BASED ON COMMON UNIFIED DEVICE ARCHITECTURE (CUDA) FOR MICRO-CT

GUOTAO QUAN, TANGYOU SUN and YONG DENG*
Britton Chance Center for Biomedical Photonics
Wuhan National Laboratory for Optoelectronics —
Huazhong University of Science and Technology
Wuhan 430074, P. R. China
**ydeng@mail.hust.edu.cn*

Three-dimensional image reconstruction with Feldkamp, Davis, and Kress (FDK) algorithm is the most time consuming part in Micro-CT. The parallel algorithm based on the computer cluster is capable of accelerating image reconstruction speed; however, the hardware is very expensive. In this paper, using the most current graphics processing units (GPU), we present a method based on common unified device architecture (CUDA) for speeding up the Micro-CT image reconstruction process. The most time consuming filtering and back-projection parts of the FDK algorithm are parallelized for the CUDA architecture. The CUDA-based reconstruction speed and image qualities are compared with CPU results for the projecting data of the Micro-CT system. The results show that the 3D image reconstruction speed based on CUDA is ten times faster than the speed with CPU. In conclusion the FDK algorithm based on CUDA for Micro-CT can reconstruct the 3D image right after the end of data acquisition.

Keywords: Micro-CT; FDK; CUDA; GPU.

1. Introduction

As the development of small animal image and drug evaluation, Micro-CT[1] has been a hot research field. Because of the high spatial resolution (about 50–100 micron) it is so popular in bone disease research.[2,3] In the past the main obstacle which limits the development of Micro-CT is (1) low precision of equipment and (2) long reconstruction time. But as the development of high precision detection equipment and micro focus X-ray cube, Micro-CT had got rid of the limitation of equipment. So the only problem left is the huge time consuming in reconstruction.

The most popular algorithm in Micro-CT reconstruction is Feldkamp, Davis, and Kress (FDK) which was proposed in 1984 to reconstruct cone-beam (CB) projections measured with a circular orbit of the X-ray cube opposite to the detector. It has been developed into various extensions to satisfy different application in the past 20 years. But the main problem of this reconstruction algorithm is that higher spatial resolution is more time-consuming, for example, a single experiment sample ($80 \times 80 \times 80$ mm) with 80 micro resolution needs 1–3 h for reconstruction. So it is not suitable for batch experiment especially for large number of

*Corresponding author.

experiment sample. In the past computer cluster had been introduced in Micro-CT reconstruction to speed up the reconstruction process, but it is so expensive, and not suitable for miniaturization of Micro-CT system.

Graphics processing units (GPU) was first developed for hardware control of mixed graphics and text modes, sprite positioning and display, with the huge requirement for real-time video processing and 3D game, GPU had been very efficient at managing computer graphics and as their highly parallel structure, they are more effective than general-purpose CPUs for a range of complex algorithms.[4] Recently Sharp[5] and Scherl *et al.*[6,7] had proposed a fast cone-beam CT image reconstruction method based on GPU, They change the FDK algorithm into a parallel form and utilize the stream process framework of GPU in which every parallel reconstruction loop had been processed in a single stream processor and every stream processor can process the data at the same time. But their research focuses on traditional CT reconstruction. We presented a fast Micro-CT FDK algorithm based on common unified device architecture (CUDA). Comparing with reconstruction results based on CPU, we found that more than ten times faster speed can be achieved. In this paper, first, the detail of our method will be introduced, then some reconstruction results will be shown in the second section, at last we will give conclude and predict the future of this method.

2. Method

In 1984, Feldkamp, Davis, and Kress first introduced the FDK reconstruction algorithm, which was time-efficient and low requirement of computer comparing with traditional algebra reconstruction technology (ART). In this method, an X-ray source is rotated around the interest region of sample in a circular orbit, and an area detector placed opposite to the source receives the transmission X-ray at each angle. Figure 1 shows the schematic of geometric configuration. FDK[8–10] algorithm is the most popular algorithm in Micro-CT reconstruction. The basic process of FDK algorithm includes filter and back-projection.

2.1. *Filter*

Projection images taken from Micro-CT system must first be high pass-filtered. This step is independent from back-projection step. The filtered signal

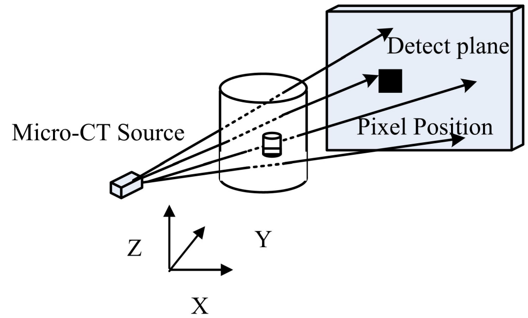


Fig. 1. FDK back-projection geometry.

can be described as:

$$p_f(\theta, a, b) = \left(\frac{R^2}{\sqrt{R^2 + a^2 + b^2}} p_b(\theta, a, b) \right) \times g(a), \tag{1}$$

where R is the source-center distance, $\frac{R^2}{\sqrt{R^2+a^2+b^2}}$ is the production of two cosine factors of the fan and cone-angle and $g(a)$ is a high pass filter. a and b are pixel coordinate in images. θ is the rotation angle at each projection image. $p_b(\theta, a, b)$ is the image received by Micro-CT system. $p_f(\theta, a, b)$ is the filtered image. This step had been described in Fig. 2

2.2. *Back-projection*

After filter step, images at each projection angle must be back-projected along the X-ray direction (see Fig. 1). The sum of back-projection result at each projection angle reveals the real volume of

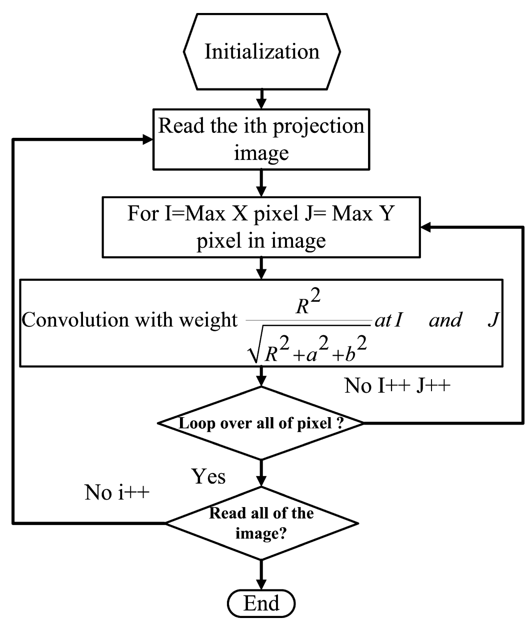


Fig. 2. Filter step flow chart.

interest. It can be described as Eq. (2):

$$f(x_v, y_v, z_v) = \sum_\theta \frac{R^2}{U(x_v, y_v, \theta)} p_f(\theta, a(x_v, y_v, \theta), b(x_v, y_v, z_v, \theta)), \qquad (2)$$

where:

$$U(x_v, y_v, \theta) = R + x_v \cos\theta + y_v \sin\theta,$$

$$a(x_v, y_v, \theta) = R\frac{x_v \sin\theta - y_v \cos\theta}{R + x_v \cos\theta + y_v \sin\theta},$$

$$b(x_v, y_v, \theta) = z_v \frac{R}{R + x_v \cos\theta + y_v \sin\theta},$$

R is the source-center distance, x_v and y_v are the reconstruction coordinates of real volume of interest, θ is the rotation angle at each projection image, $p_f(\theta, a(x_v, y_v, \theta), b(x_v, y_v, z_v, \theta)) p_f(\theta, a, b)$ is the filtered image, a and b are pixel coordinate in images, and $f(x_v, y_v, z_v)$ is the reconstructed pixel value of volume. This step had been described in Fig. 3. Because steps 1 and 2 require much more time in traditional FDK based on CPU, we make the FDK algorithm parallelized for the CUDA architecture. The flow chart of filter algorithm is shown in Fig. 4 and the flow chart of back-projection step is shown in Fig. 5.

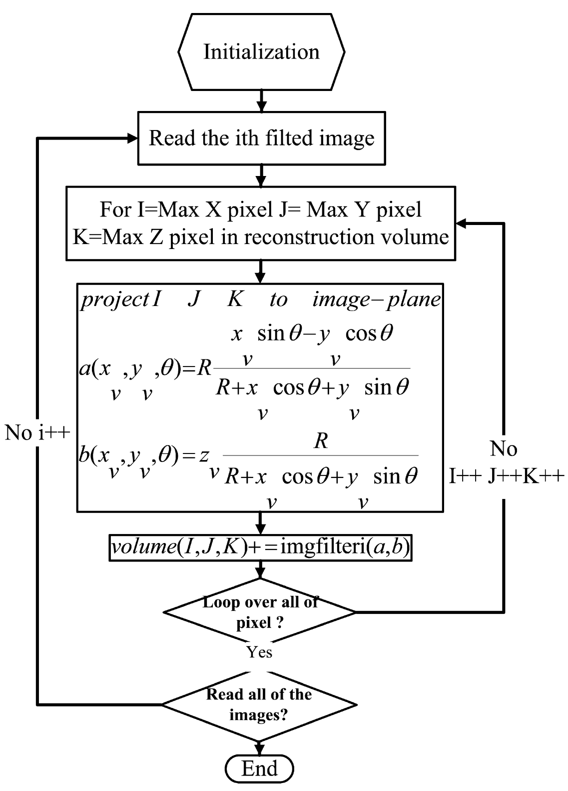


Fig. 3. Back-projection flow chart.

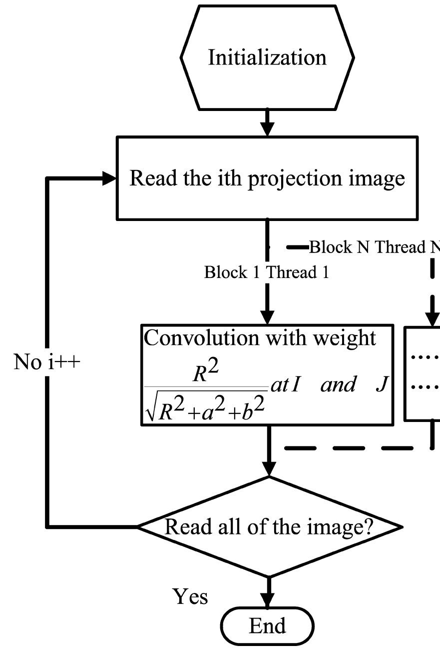


Fig. 4. Flow chart of filter based on CUDA.

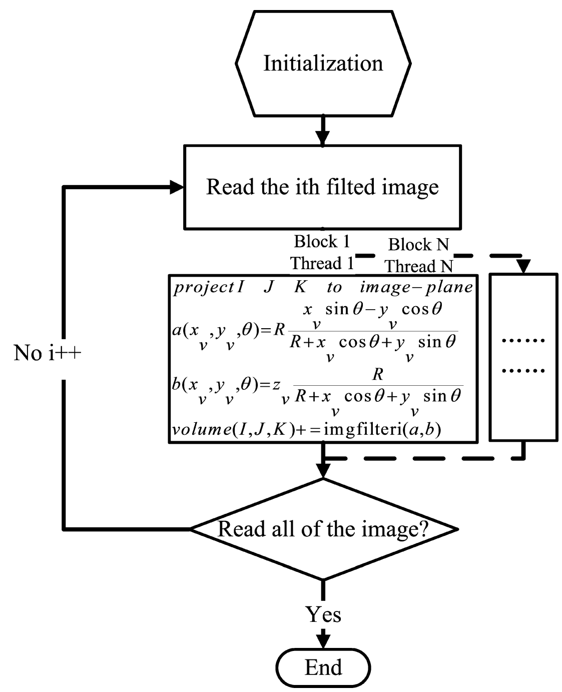


Fig. 5. Flow chart of back-projection based on CUDA.

3. Result

The FDK based on CPU had been tested on Intel(R) Core(TM) i7 CPU 920@2.67Ghz platform with 6GB RAM, and the FDK based on CUDA had been tested on NVIDIA GeForce GTX 295.

3.1. *Filter result*

First we use our Micro-CT to receive 205 projection images with 1.8 degree interval. Then 205 images (1024×1024 pixels) had been processed on both CPU and GPU. The result shows that more than 20 times faster speed had been achieved (Table 1).

3.2. *Back-projection result*

In order to show clearly the performance of our FDK based on CUDA, we first process the 200 projection images with weight filter shown in filter step. Then we reconstruct 800 slices of 500×500, 800×800, and 1000×1000 pixels for a mouse's filtered projection images. The performance of our FDK based on CUDA is shown in Table 2.

To evaluate the error of our method, a slice of mice had been reconstructed. Figures 6(a) and 6(b) are the reconstruction slice of mice based on CPU and CUDA, respectively. The absolute error of two images is shown in Fig. 7, the results show that the error between reconstruction images based on CPU and CUDA is very small.

A whole 3D reconstruction image of mice is shown in Fig. 8 with the slice reconstructed above with 3313.27 s consuming in CPU or 241.40 s consuming in GPU.

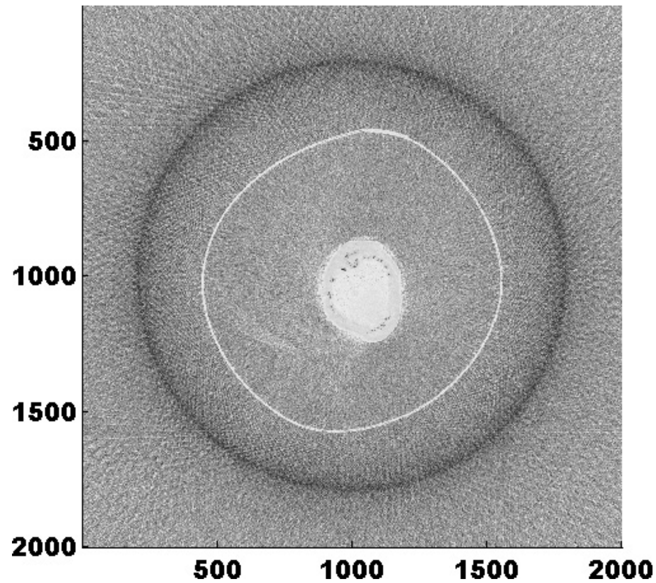
From the result above we can see that in the past the whole slices of small animal would take about 1 h to reconstruct, but with our method based on CUDA the reconstruction process can be finished in few minutes, especially for a batch experiment which will process large number of experiment samples this method will show its tremendous power of speed. On the other hand the error between our method and traditional method based on CPU is so small, which proves the validity of our method.

Table 1. Time-consuming in filter.

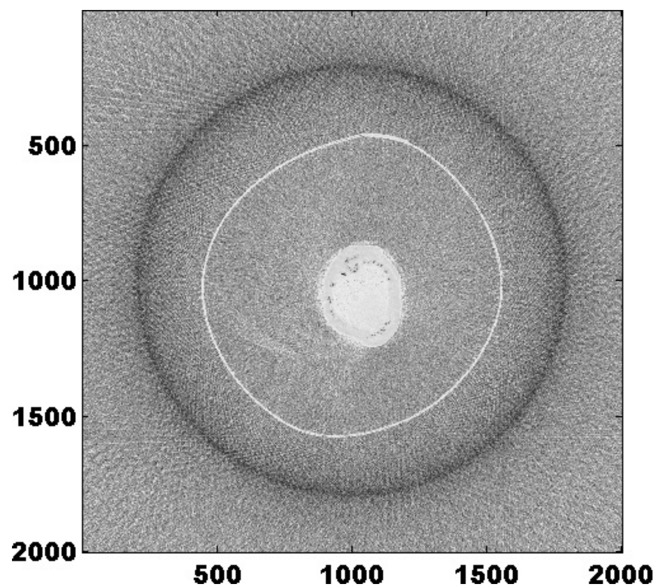
Images for filter (1024 × 1024)	Time on CPU (s)	Time on CPU (s)	Speedup
205	985.9	46.7	21.096

Table 2. Time-consuming in back-projection.

Pixel	500 × 500	800 × 800	1000 × 1000
Time on CPU (s)	881.84	2261.62	3313.27
Time on GPU (s)	75.32	164.42	241.40
Speedup	11.7	13.8	13.8



(a) Back projecting result with CPU



(b) Back projecting result with GPU

Fig. 6. Slice of reconstruction mice.

4. Conclusion

In this paper we proposed a fast CUDA based FDK method for Micro-CT. Comparing with the FDK method based on CPU we prove the validity and performance of our method. Due to the low requirement of this method (only an high performance GPU), it may push forward the miniaturization of Micro-CT system, and as the result of fast speed of our method, a batch of experiment samples can be reconstructed in short time, which will save much more time in Micro-CT experiments. But

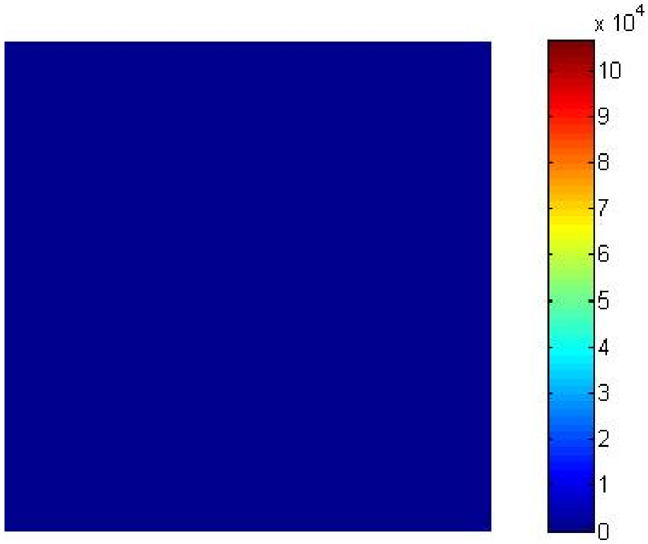


Fig. 7. Absolute error of Figs. 6(a) and 6(b).

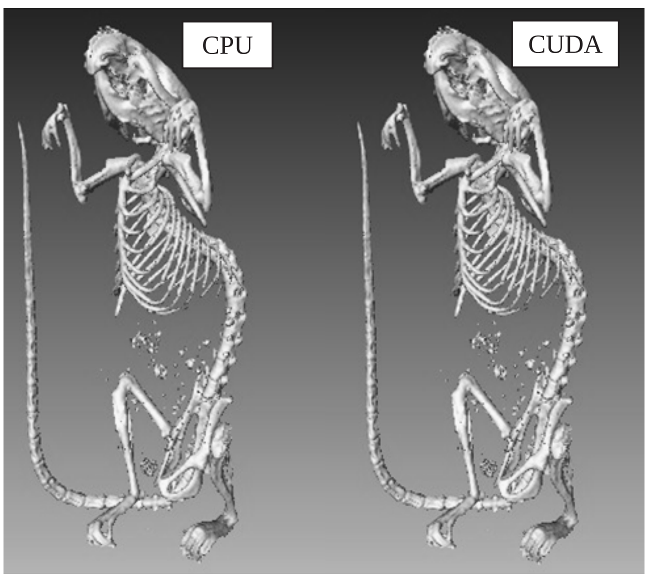


Fig. 8. Reconstruction images of mice.

our method only speeds up the FDK algorithm of Micro-CT for about 13 times, that is not so fast comparing with other application based on CUDA (molecular dynamics, video process, and so on). Because we do not utilize the full power of stream processor and lots of time had been consumed in data transform between RAM of CPU and RAM of GPU.

Acknowledgments

We gratefully acknowledge the financial support by National Nature Science Foundation of China (Grant No. 30070261).

References

1. M. Kachelrieb, "Micro-CT," *Handb Exp Pharmacol* **185**(1), 23–52 (2008).
2. K. J. Batenburg, J. Sijbers, "Discrete tomography from micro-CT data: Application to the mouse trabecular bone structure — art. no. 614240," *Medical Imaging 2006: Physics of Medical Imaging*, Pts 1–3, 6142, 14240–14240 (2006).
3. I. Lima, M. J. Anjos, M. L. F. Farias *et al.*, "Bone diagnosis by X-ray techniques," *Eur. J. Radiol.* **68**(3), S100–S103 (2008).
4. J. Fung and S. Mann, "Using graphics devices in reverse: GPU-based image processing and computer vision," *Proc. IEEE Int. Conf. Multimedia & Expo.*, Hannover, Germany, June 23–26 (2008).
5. G. C. Sharp, N. Kandasamy, H. Singh *et al.*, "GPU-based streaming architectures for fast cone-beam CT image reconstruction and demons deformable registration," *Phys. Med. Biol.* **52**(19), 5771–5783 (2007).
6. H. Scherl, B. Keck, M. Kowarschik, J. Hornegger, "Fast GPU-based CT reconstruction using the common unified device architecture (CUDA)," *IEEE Nuclear Science Symposium Conference Record* **6**(1), 4464–4466 (2007).
7. G. R. Yan, J. Tian, S. P. Zhu *et al.*, "Fast cone-beam CT image reconstruction using GPU hardware," *J. X-Ray Sci. Technol.* **16**(4), 225–234 (2008).
8. T. Rodet, F. Noo, M. Defrise, "The cone-beam algorithm of Feldkamp, Davis, and Kress preserves oblique line integrals," *Med. Phys.* **31**(7), 1972–1975 (2004).
9. P. S. Cho, "Cone beam image reconstruction," in *Imaging in Radiotherapy*, A. L. Boyer, J. D. Hazle, Eds., pp. 459–474, Advanced Medical Publishing, Madison, Wisconsin (1998).
10. G. Wang, M. Jiang, S. W. Lee, H. Liu, E. Hoffman, "Cone-beam reconstruction for micro-CT," *2002 IEEE International Symposium on Biomedical Imaging, Proceedings*, 384–387 (2002).