Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 3 Issue 10, October - 2016

A Novel Backprojection of Katsevich Algorithm Based on Rotation Symmetry of Adjacent Slices and Implementation on GPU

Yan Zhang
School of Electronic and Information Engineering
Harbin Institute of Technology (ShenZhen)
ShenZhen, P.R.CHINA
ianzh@hit.edu.cn

HaiGang Zheng
School of Electronic and Information Engineering
Harbin Institute of Technology (ShenZhen)
ShenZhen, P.R.CHINA
hackenzheng@126.com

***Abstract*—The Katsevich algorithm can obtain exact reconstruction image as an exact reconstruction method with the filtered back-projection. However, the intensive computation required by this algorithm prohibits its clinical use. Optimization of back-projection step is vital as it is the most time-consuming. In this paper, a novel back-projection based on rotation symmetry of adjacent slices is proposed which sets slice step size multiple of x-ray source step size on z-axis first, then calculate back-projection using decision boundary to determine integration range. The decision boundary calculation can been speed up by using rotation matrix derived from rotation symmetry. Besides, the graphics processing unit(GPU) is used to accelerate the algorithm. Both time complexity and space complexity of the proposed method are better than PI line method. The simulation results show that our method actually have better performance than PI line based back-projection, especially on memory.**

Keywords—Cone-beam CT; Katsevich; back projection; GPU;

I. INTRODUCTION

Cone-beam Computed Tomography(CBCT) has been widely used in stomatology and radiology medical diagnosis as it has advantages of scanning speed, reconstructed image resolution and utilization of x-rays. As a kind of exact CBCT reconstruction algorithm, Katsevich algorithm was proposed by katsevich in 2002 based on Grangeat theory[1]. The algorithm is a breakthrough for CBCT reconstruction. It improved computing efficiency and accuracy as it requires minimal number of scan times under the same system environment. Noo et al. in 2003[2] and G.Wang et al. in 2004[3] both explained the algorithm implementation in detail and summarized the relation between image quality and CBCT system parameters.

Recent years, the Katsevich algorithm acceleration based on GPU has been widely studied as GPU has higher performance and lower developing difficulty. There is no report about acceleration of Katsevich algorithm on graphics processing unit(GPU) until 2010 Yan et al.[4] published their results in the literature. In their research, GPGPU technique and Tam window based back-projection method were used. They proposed an overscan formula and volume blocking method to reconstruct large volume in limited device memory. Wu et al.[5] proposed a more precise overscan formula and utilized PI line instead of Tam window to determine the integration range. The quality of the reconstructed images in Wu's is better than Yan's. In 2015, Bardino et al.[6] reported that they can accelerate both FDK and Katsevich algorithm on GPU with different architecture, such as AMD and NVIDIA. Their importance is the implementation and integration with Python, PyCUDA and PyOpenCL.

Though the PI line method is more stable and the reconstructed image has less artefacts than Tam window based method, an array whose size is the same as the voxels needs to be maintained to keep the PI lines for each voxel and massive no-linear equations need to be solved while back-projecting. Here, we proposed a novel back-projection method which utilizes the rotation symmetry of adjacent slices and reconstruct Shepp-Logan model on GPU with the new method.

The first part of the article is the introduction, the Katsevich algorithm based on PI line is given in the second part, the proposed novel back-projection based on rotation symmetry of adjacent slices is described in the third part, and the simulation results on GPU are presented on the fourth part. The last part is the conclusion.

II. KATSEVICH ALGORITHM BASED PI LINE

The Katsevich algorithm includes two parts which are 1-D shift-invariant filtering of a derivative of the two consecutive projections and back-projection[7]. The general back-projection of Katsevich algorithm utilizes PI segment to determine the integration range which ensure the projection of the reconstructed voxel locates in Tam window. The whole Katsevich reconstruction formation can be written as follows:

$$f(x) = -\frac{1}{2\pi}\int_{s_b}^{s_t} \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}(s)\|} * \int_0^{2\pi} \frac{\partial}{\partial q} Df\left(\boldsymbol{y}(q), \boldsymbol{\theta}(s,\boldsymbol{x},\gamma)\right)\Big|_{q=s} \frac{d\gamma}{\sin\gamma} ds \quad (1)$$

Where x represents the voxel to be reconstructed, y(s) is the x-ray source and s is the rotation angle between y(s) and the start position.

In Katsevich algorithm's first part, there are five steps that are differential, length correction weighting, forward height rebinning, 1-D Hilbert transform and backward height rebinning. And there are PI segment calculation and back-projection accumulation in the second part.

For each voxel x, the coordinate in global system is $[x_1, x_2, x_3]^T$, and is $[r\cos\gamma, r\sin\gamma, x_3]^T$ in cylinder system, the corresponding PI segment is $I_{PI(x)} = [s_b, s_t]$, whose start angle s_b satisfies

$$x_3 = h[(\pi - 2\arctan(\frac{r\sin(\gamma - s_b)}{R - r\cos(\gamma - s_b)})) * (1 + \frac{r_2 - R_2}{2R(R - r\cos(\gamma - s_b))} + s_b)] \quad (2)$$

Also, the bottom of PI segment s_b should be in the follow interval

$$\frac{x_3}{h} - \pi \le s_b \le \frac{x_3}{h} \quad (3)$$

We can obtain the value of s_b by solving no-linear equation (2) and (3), then the top of PI segment s_t can been calculated as

$$s_t = s_b + \pi - 2\arctan(\frac{r\sin(\gamma - s_b)}{R - r\cos(\gamma - s_b)}) \quad (4)$$

After getting the PI segment, the reconstructed voxel x's volume density is

$$f(\mathrm{x}) \approx \sum_k \frac{\rho(\mathrm{s_k}, \mathrm{x})\Delta \mathrm{s}}{2\pi v^*(\mathrm{s_k}, \mathrm{x})} g_f^F(\mathrm{s_k}, \mathrm{u}^*(\mathrm{s_k}, \mathrm{x}), \mathrm{w}^*(\mathrm{s_k}, \mathrm{x})) \quad (5)$$

Where $\rho(\mathrm{s_k}, \mathrm{x})$ is the smooth factor to reduce the artifacts caused by the endpoints of PI segment.

III. BACKPROJECTION BASED ON ROTATION SYMMETRY OF ADJACENT SLICES

During the CBCT reconstruction process, not all projections are useful for a slice to be reconstructed. For a frame projection, the corresponding valid slices range is computable according to the over-scan formula[4], also with the valid interval of the fixed x-ray source corresponding to one slice. The rays of cone beam projecting the Tam window's top curve and bottom curve will intersect with slice plane and we name the intersection as decision boundary. The decision boundary is illustrated in Fig.1(a). If the reconstructed voxel is between the decision boundary and the x-ray source, its projection will locate in Tam window according to the cone beam cover theroy[8]. This has the same effect as PI segment. The decision boundary can be used to determine if the voxel utilize current projection data.

A. Construct the Rotation Symmetry of Adjacent Slices

The adjacent slices' gap in z-axis is Δz , the step size of adjacent x-ray sources is Δs ,in z-axis direction is Δh .Set $\Delta z = m * \Delta h$, m is positive integer. Slice Z_i and Z_{i+1} are adjacent which the subscript i represents the i-th slice. And there are corresponding x-ray source S_n and S_{n+m} which the subscript n represents the n-th projcetion. The slice Z_i is in the cone beam cover of S_n. Then the decision boundary formed by Z_i and S_n is similar to the one formed by Z_{i+1} and S_{n+m}, which is illustrated in Fig.1(b). We can prove the two decision boundary equations are the same in reference system.

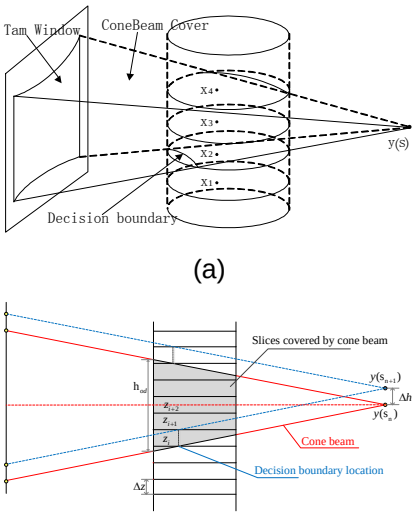

Fig.1. *Decision boundary and rotation symmetry between adjacent slices: (a)3-D projection demonstration, (b)2-D view*

B. Decision Boundary and Rotation Matrix

The arbitrary x-ray source's coordinate in the reference system is

$$\begin{cases} u = -R\cos(s)\sin(s_0) + R\sin(s)\cos(s_0) \\ v = \frac{h}{2\pi}s - \frac{h}{2\pi}s_0 \\ w = R\cos(s)\cos(s_0) + R\sin(s)\sin(s_0) + R \end{cases} \quad (6)$$

The rays which intersect with detector forming the Tam window's top curve and bottom curve are the part of lines connecting current x-ray source $y(s_0)$ and other x-ray sources $y(s)$. κ represents the line passing through $y(s_0)$ and $y(s)$, its parameter equation is as follow

$$\begin{cases} u = Rt(-\cos(s)\sin(s_0) + \sin(s)\cos(s_0)) \\ v = \frac{ht}{2\pi}(s - s_0) \\ w = Rt(\cos(s)\cos(s_0) + \sin(s)\sin(s_0) - 1) + 2R \end{cases} \quad (7)$$

The parameter equtation of reconstructed slice plane Z in the reference system is as formula (8)

$$v = z - \frac{h}{2\pi} s_0 \tag{8}$$

The decision boundary is the intersection of κ line and plane Z, so its formula is

$$\begin{cases} u = \dfrac{R(2\pi z - hs_0)}{h(s - s_0)} \sin(s - s_0) \\ w = \dfrac{R(2\pi z - hs_0)}{h(s - s_0)} (\cos(s - s_0) - 1) + 2R \end{cases} \tag{9}$$

Where R is helical radius and h is helical pitch. Let s_0 denote current x-ray source location. s represents other arbitrary point on the helical trajectory and the parametric variable. For arbitrary s_0, the range of s-s_0 is fixed. So set another parameter variable α= s- s_0.

In section III, we have set $\Delta z = m * \Delta h$ and there are adjacent slices Z_i and Z_{i+1}. X-ray source S_n and S_{n+m} will project decision boundary a and b in slice plane Z_i and Z_{i+1} correspondingly. Their parameter equation are as follow

$$\begin{cases} u_i = \dfrac{R(2\pi z_i - hs_n)}{h * \alpha} \sin(\alpha) \\ w_i = \dfrac{R(2\pi z_i - hs_n)}{h * \alpha} (\cos(\alpha) - 1) + 2R \end{cases} \tag{10}$$

$$\begin{cases} u_{i+1} = \dfrac{R(2\pi z_{i+1} - hs_{n+\mathrm{m}})}{h * \alpha} \sin(\alpha) \\ w_{i+1} = \dfrac{R(2\pi z_{i+1} - hs_{n+m})}{h * \alpha} (\cos(\alpha) - 1) + 2R \end{cases} \tag{11}$$

For adjacent x-ray source, there is

$$\Delta h = \Delta s * h / (2\pi) \tag{12}$$

So, there is derivation

$$2\pi z_{i+1} - hs_{n+m} = 2\pi(z_i + \Delta z) - h(s_n + m\Delta s) = 2\pi z_i - hs_n \tag{13}$$

Substituting (13) into equtation (11) we can find the intersection a and b's parameter equations are equal. This is the main content about rotation symmetry of adjacent slices.

All the above derivations are illustrated in reference system. While in the implementation, the decision boundary formula should been written in global system as formula (14) and (15) to simplify the calculation. Using 1-D array to denote the equation's discretization. The array index represents abscissa and corresponding value represents ordinate.

$$\begin{cases} x_i = (w_i - R)\cos(s_n) - u_i \sin(s_n) \\ y_i = (w_i - R)\sin(s_n) + u_i \cos(s_n) \end{cases} \tag{14}$$

$$\begin{cases} x_{i+1} = (w_{i+1} - R)\cos(s_{n+m}) - u_{i+1} \sin(s_{n+m}) \\ y_{i+1} = (w_{i+1} - R)\sin(s_{n+m}) + u_{i+1} \cos(s_{n+m}) \end{cases} \tag{15}$$

In addition, the intersection b can been denoted by intersection a. There is a rotation between them. The abscissa x_{i+1} has follow relation.

$$\begin{aligned} x_{i+1} &= (w_i - R)\cos(s_n + m\Delta s) - u_i \sin(s_n + m\Delta s) \\ &= \cos(m\Delta s)[(w_i - R)\cos(s_n) - u_i \sin(s_n)] \\ &\quad - \sin(m\Delta s)[(w_i - R)\sin(s_n) + u_i \cos(s_n)] \\ &= \cos(m\Delta s)x_i - \sin(m\Delta s)y_i \end{aligned} \tag{16}$$

Also, the ordinate y_{i+1} can been denoted as

$$y_{i+1} = \sin(m\Delta s)x_i + \cos(m\Delta s)y_i \tag{17}$$

Let matrix A be the intersection a's parameter equation and matrix B represents intersection b. There is a rotation between A and B. Let matrix Y be the rotation matrix. Therefore, Y is can been written as (18). Based on rotation matrix Y, we can accelerate decision boundary calculation.

$$Y = \begin{Bmatrix} \cos(m\Delta s) & -\sin(m\Delta s) \\ \sin(m\Delta s) & \cos(m\Delta s) \end{Bmatrix} \tag{18}$$

Assume the reconstruction volume is N*N*N, the average iteration times for calculate PI segment is n, and k frames of projections are needed for reconstructing one slice. The time complexity and space complexity analysis are illustrated in Table 1. Here, the time complexity refer to calculating pi-line and decision boundary. In theory, decision boundary based method has less time and space complexity while the computational complexity of back projection accumulation are almost the same.

TABLE I. TIME AND SPACE COMPLEXITY COMPARISON

Method	Time	Memory
PI Line	nN^3	$2N^3$
Decision boundary	kN^2	kN^2

IV. SIMULATION RESULTS

First, we set m=1, which represents the slice step size and x-ray source step size in z-axis direction are equal. Other primary parameter are set as Table 2. The simulation is carried on a machine with a 3.6GHz Intel i7-4790 CPU, 16GB DDR3 memory and a 2GB GeForce GTX 960 graphic card. Both PI line method and our method were implemented on GPU to reconstruct 256*256*256 volume size. The time and memory consuming comparison during back projection are shown in Table 3. The memory doesn't include projection frame data. The time unit is second and the memory unit is MB. It shows that our method has great advantage on memory consumption to reconstruct the same size volume.

TABLE II. CBCT SIMULATION PARAMETER

Parameter	PI-line	This paper
FOV in z axes	-0.25/-0.22	-0.378/-0.122
slice step size Δz	0.00012	0.001
projections per turn	256	256
pitch of helix P	0.219	0.256
source step size Δh	0.00086	0.001

TABLE III. TIME AND MEMORY CONSUMPTION

Volume	Method	Time	Memory
256^3	PI-line	4.536	192
	This paper	4.322	96

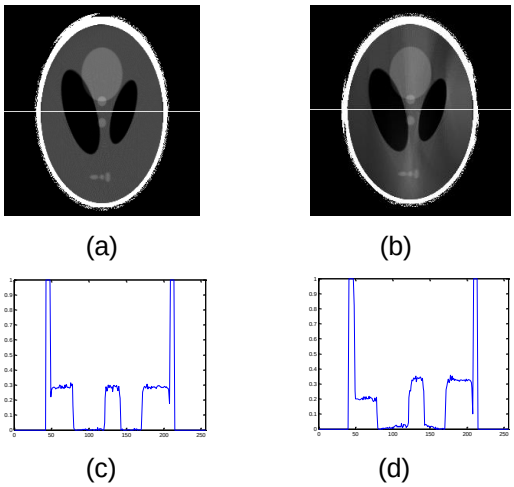

Fig.2. *The reconstruction results and profiles for the white line. (a)(c)PI line method (b)(d) our method*

V. CONCLUSION

In this paper, a novel back-projection method for multiple slices reconstruction based on rotation symmetry of adjacent slices is proposed. Using decision boundary instead of PI segment to determine integration range can obtain better performance of computation time and memory. A rotation matrix based on rotation symmetry is derived to accelerate decision boundary computation, therefore accelerate multiple slices reconstruction. Our experiment on GPU has shown that the algorithm can be applied to practical use and reduce time and memory consumption.

ACKNOWLEDGMENT

This work was supported in part by the technology development and innovation design fund of Nashan, Shenzhen (Grant No. KC2013JSCX0041A), the scientific development fund of Shenzhen (Grant No. JC201005260168A), the Strategic emerging industry development fund of Shenzhen (Grant No. JCYJ20130329161328512), and the Science and Technology Planning Project of Guangdong Province (Grant No. 2013B090600105).

REFERENCES

[1] KATSEVICH A, Theoretically exact FBP-type inversion algorithm for spiral CT, SIAM Journal on Applied Mathematics, Vol.62,No.6, 2002,pp.2012-2026.

[2] NOO F, PACK J, Heuscher D. Exact helical reconstruction using native cone-beam geometries, Physics in Medicine and Biology, Vol.38, No.23, 2003pp.3787-3818.

[3] YU HengYong, WANG Ge. Studies on implementation of the Katsevich algorithm for spiral cone-beam CT, Journal of X-Ray Science and Technology, Vol.12, No.2, 2004pp. 97-116.

[4] YAN GuoRui, TIAN Jie, et al. Fast Katsevich algorithm based on GPU for helical cone-beam computed tomography, IEEE Transactions on Information Technology in Biomedicine, Vol.14, No.4, 2010pp. 1053-1061.

[5] WU DuFan, LI Liang, ZHANG Li, et al. A Practical GPU-based Implementation of The Katsevich Algorithm, Chinese Journal of Stereology and Image Analysis, Vol.16, No.4, 2011pp. 355-359.

[6] Bardino J, Rehr M, Vinter B. Cph CT Toolbox: A performance evaluation. HPCS, 2015 International Conference on. IEEE, 2015: 35-46.

[7] Wunderlich A J. The Katsevich inversion formula for cone-beam computed tomography[J]. Department of Mathematics, MS, 2006, 127.

[8] YANG JianSheng, KONG Qiang, ZHOU Tie, et al. Cone beam cover method: An approach to performing backprojection in Katsevich's exact algorithm for spiral cone beam CT, Journal of X-ray science and technology, Vol.12, No.4, 2004pp. 199-214.