Studies on Artifacts of the Katsevich Algorithm for Spiral Cone-beam CT

Hengyong Yu
College of Communication Engineering, Hangzhou Dianzi University
Hangzhou, Zhejiang 310018, China
yuhengyong@hotmail.com
Ge Wang
Department of Radiology, University of Iowa
Iowa City, Iowa 52242, USA
ge-wang@uiowa.edu

ABSTRACT

Recently, Katsevich proposed an exact filtered backprojection (FBP) algorithm to solve the long object problem. Although the Katsevich algorithm is theoretically error-free, it may produce image artifacts if its numerical implementation is not well designed. To evaluate the Katsevich algorithm, we implement it in planar detector geometry. Then, we study four types of image artifacts associated with the numerical implementation of this algorithm, which are "***texture***", ***"streak"***, ***"shadow"*** and ***"winkle"*** artifacts. The ***"texture"*** artifacts appear if the endpoints of the PI-Line are not treated appropriately. The ***"streak"*** artifacts are caused by an inadequate number of filtering lines Q, which can be reduced by setting $0.3N < Q < 0.7N$. In the step of Hilbert filtering, the aliasing effect results in ***"shadow"*** artifacts if the window is not truncated properly. The ***"wrinkle"*** artifacts are due to both the interpolation and difference operators, which may be minimized using more sophisticated interpolation methods, increasing projection data and using a more accurate derivative formula. Our results should be valuable to apply the Katsevich algorithm optimally for practical spiral cone-beam CT.

Keywords: Computed tomography (CT), image artifacts, Katsevich algorithm, spiral cone-beam, filtered backprojection (FBP)

1. INTRODUCTION

Spiral cone-beam computed tomography (CT) is a major area of medical imaging research. Generally speaking, the spiral cone-beam scanning mode offers significant advantages in image quality than other modes such as multiple circular scans for reconstruction of a long object. The two important components for spiral cone-beam CT are data acquisition and image reconstruction. As various types of detectors are under rapid development[1], spiral cone-beam reconstruction algorithms are continuously improved. Wang *et al.* categorized cone-beam algorithms into tree types: exact, approximate and iterative algorithms[2]. To compensate for image artifacts introduced in approximate reconstruction for large cone angles, exact reconstruction algorithms become increasingly important[3,4]. Although the exact approach is theoretically error-free, it is still subject to image artifacts due to the discrete nature of numerical implementation[5]. Therefore, it is desirable to study various types of image artifacts associated with specific numerical implementations of an exact image reconstruction algorithm.

Recently, mathematician Katsevich proposed an exact image reconstruction formula from longitudinal truncated cone-beam projections[6-9]. It is widely regarded as a significant breakthrough in solving the long object problem. Based on this milestone achievement, several modified algorithms are reported[10-16]. Independent of the work by Noo *et al.*[17], we implemented the Katsevich algorithm in planar detector geometry[18]. Then, we were motivated to study the artifact behaviors of the Katsevich algorithm. In this paper, we focus on the numerical implementation of formula (2.13) in Reference [7]. In section II, we review the Katsevich algorithm and summarize each step of its numerical

Developments in X-Ray Tomography IV, edited by Ulrich Bonse, Proc. of SPIE Vol. 5535
(SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.559286

implementation. In Section III, we demonstrate four types of image artifacts, explain their causes and suggest corrective means. In Section IV, we conclude this paper.

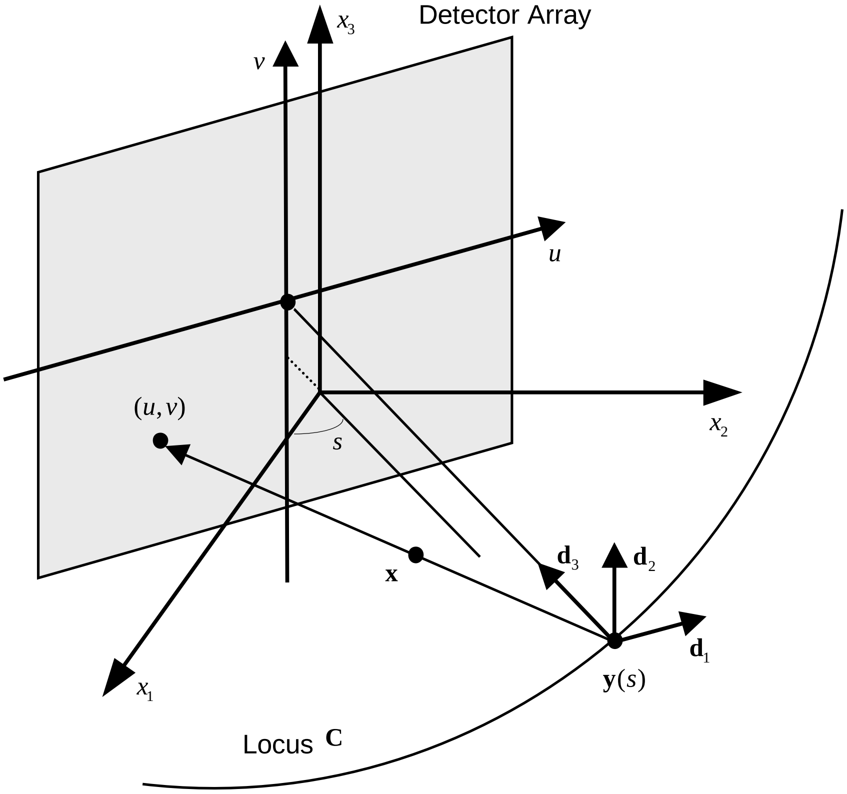


Figure 1. Local coordinate system for cone-beam projection measurement on a detector plane

2. IMPLEMENTATION OF THE KATSEVICH ALGORITHM

2.1. KATSEVICH FORMULA IN PLANAR DETECTOR GEOMETRY[18]

As shown in Fig.1, let a helical scanning locus be defined as

$$\mathbf{C} := \left\{ \mathbf{y} \in \mathbb{R}^3 : y_1 = R\cos(s), y_2 = R\sin(s), y_3 = \frac{sh}{2\pi}, s \in \mathbb{R} \right\} \qquad (1)$$

and let

$$\mathbf{d}_1 := (-\sin(s), \cos(s), 0) \qquad (2)$$

$$\mathbf{d}_2 := (0,0,1), \qquad (3)$$

$$\mathbf{d}_3 := (-\cos(s), -\sin(s), 0). \qquad (4)$$

represent a local coordinate system with cone-beam data measured on a planar detector array parallel to $\mathbf{d}_1$ and $\mathbf{d}_2$ at a distance D from $\mathbf{y}(s)$. A detector position in the array is given by a pair of values (u,v), which are signed distances along $\mathbf{d}_1$ and $\mathbf{d}_2$ respectively. Let (u,v) =(0, 0) correspond to the orthogonal projection of $\mathbf{y}(s)$ onto the detector array. Since D is constant, the equispatial cone-beam projection can de denoted as $g(s,u,v)$. In this local coordinate system, Katsevich's formula[7] is implemented in the following three steps[18]:

(S1) Computing of the first derivative

$$D_g(s,u,v) = \left(\frac{\partial}{\partial s} + \frac{D^2+u^2}{D}\frac{\partial}{\partial u} + \frac{uv}{D}\frac{\partial}{\partial v} \right) g(s,u,v) . \tag{5}$$

(S2) Hilbert filtering of the first derivative

(S2.1) Pre-weighting: $\Psi^{(1)}(s,u,v) = D_g(s,u,v)\frac{D}{\sqrt{D^2+u^2+v^2}}$ (6)

(S2.2) Hilbert filtering along the line $L(s_1)$:

$$\Psi^{(2)}(s,u,v) = \int_{-\infty}^{\infty} \Psi^{(1)}(s,\tilde{u},\tilde{v})\frac{1}{(\tilde{u}-u)}d\tilde{u} , \tag{7}$$

where $(\tilde{u},\tilde{v})$ denote the local coordinate of a variable point on the line $L(s_1)$ determined by (u,v) with

$$v = \frac{Dh(s_1-s)}{2\pi R} + \frac{h(s_1-s)ctg(s_1-s)}{2\pi R}u . \tag{8}$$

(S2.3) Post-weighting: $\Psi^{(3)}(s,u,v) = \frac{\sqrt{D^2+u^2+v^2}}{D}\Psi^{(2)}(s,u,v)$ (9)

(S3) Backprojection of the filtered data into the 3D object support.

$$f(\mathbf{x}) = -\frac{1}{2\pi^2}\int_{s_b(\mathbf{x})}^{s_t(\mathbf{x})} \frac{1}{|\mathbf{x}-\mathbf{y}(s)|}\Psi^{(3)}(s,u^*,v^*)ds \tag{10}$$

$$u^* = \frac{D(\mathbf{x}-\mathbf{y}(s))\bullet\mathbf{d}_1}{(\mathbf{x}-\mathbf{y}(s))\bullet\mathbf{d}_3} , \tag{11}$$

$$v^* = \frac{D(\mathbf{x}-\mathbf{y}(s))\bullet\mathbf{d}_2}{(\mathbf{x}-\mathbf{y}(s))\bullet\mathbf{d}_3} . \tag{12}$$

Where $s_b(\mathbf{x})$ and $s_t(\mathbf{x})$ denote parametric values corresponding to the PI-Line segment determined by $\mathbf{x}$.

2.2. NUMERICAL IMPLEMENTATION

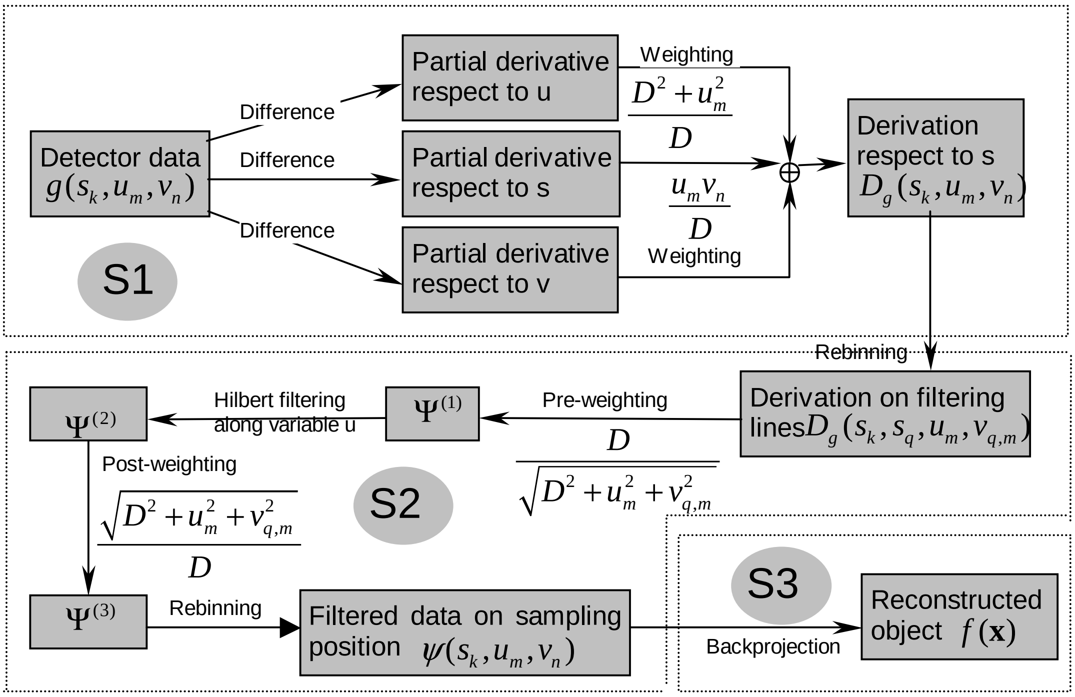
To implement the Katsevich algorithm, cone-beam projections are uniformly sampled at the source and detector positions with intervals Δs , Δu , Δv for s , u and v , respectively. The sampled data are denoted as:

$$g(s_k,u_m,v_n) \quad 0 \le k < K, 0 \le m < M, 0 \le n < N , \tag{13}$$

where k , m and n are the indexes of sampling points with respect to s , u and v , respectively. In practice, m and n are indexes of detector units, with Δu Δv being the detector size. As summarized in Fig. 2, the discrete counterparts of the above three steps are as follows.

(S1) First derivative:

$$D_g(s_k,u_m,v_n) \approx \frac{g(s_{k+1},u_m,v_n)-g(s_{k-1},u_m,v_n)}{2\Delta s} + \frac{D^2+u_m^2}{D}\frac{g(s_k,u_{m+1},v_n)-g(s_k,u_{m-1},v_n)}{2\Delta u} ,$$
$$+\frac{u_m v_n}{D}\frac{g(s_k,u_m,v_{n+1})-g(s_k,u_m,v_{n-1})}{2\Delta v} . \tag{14}$$



Partial derivative respect to u
Weighting
$\frac{D^2+u_m^2}{D}$
Difference
Detector data $g(s_k,u_m,v_n)$
Difference
Partial derivative respect to s
Derivation respect to s $D_g(s_k,u_m,v_n)$
$\oplus$
$\frac{u_m v_n}{D}$
Difference
Weighting
Partial derivative respect to v
S1
Rebinning
$\Psi^{(2)}$
Hilbert filtering along variable u
$\Psi^{(1)}$
Pre-weighting
Derivation on filtering lines$D_g(s_k,s_q,u_m,v_{q,m})$
$\frac{D}{\sqrt{D^2+u_m^2+v_{q,m}^2}}$
Post-weighting
$\frac{\sqrt{D^2+u_m^2+v_{q,m}^2}}{D}$
S2
S3
$\Psi^{(3)}$
Rebinning
Filtered data on sampling position $\psi(s_k,u_m,v_n)$
Reconstructed object $f(\mathbf{x})$
Backprojection

Figure 2. Flowchart for the numerical implementation of the Katsevich algorithm.

(S2) Hilbert filtering:

This step is more complicated than other steps. For any $s\in\mathbb{R}$, $s_1-s\in[-\pi+0.5\Delta,\pi-0.5\Delta]$ with $\Delta=2\cos^{-1}(r/R)$ and r being the object radius. Thus, for all s_k, the interval $[-\pi+0.5\Delta,\pi-0.5\Delta]$is uniformly sampled at $s_q,|q|\le Q$. Using (8), the following matrix is calculated:

$$v_{q,m}=\frac{Dhs_q}{2\pi R}+\frac{hs_q ctg(s_q)}{2\pi R}u_m, \qquad (15)$$

where the only exception is that $v_{0,m}=\frac{h}{2\pi R}u_m$.

(S2.1) Rebinning: From $D_g(s_k,u_m,v_n)$ into $D_g(s_k,s_q,u_m,v_{q,m})$ via linear interpolation along the direction of v.

(S2.2) Pre-weighting: $\Psi^{(1)}(s_k,s_q,u_m,v_{q,m})=D_g(s_k,s_q,u_m,v_{q,m})\frac{D}{\sqrt{D^2+u_m^2+v_{q,m}^2}}$, (16)

(S2.3) Hilbert Filtering: $\Psi^{(2)}(s_k,s_q,u_m,v_{q,m})\approx\sum_{m'=-M}^{M}\frac{\Psi^{(1)}(s_k,s_q,u_{m'},v_{q,m'})\Delta u}{(u_{m'}-u_m)}$. (17)

(S2.4) Post-weighting: $\Psi^{(3)}(s_k,s_q,u_m,v_{q,m})=\frac{\sqrt{D^2+u_m^2+v_{q,m}^2}}{D}\Psi^{(2)}(s_k,s_q,u_m,v_{q,m})$. (18)

(S2.5) Rebinning: From $\Psi^{(3)}(s_k, s_q, u_m, v_{q,m})$ into $\psi(s_k, u_m, v_n)$ via linear interpolation along the direction of v.

(S3) Backprojection

Assuming k_b and k_t are nearest integers satisfying $|s_b(\mathbf{x}) - s_{k_b}| < 0.5\Delta s$ and $|s_t(\mathbf{x}) - s_{k_t}| < 0.5\Delta s$, then

$$f(\mathbf{x}) = -\frac{\Delta s}{2\pi^2} \sum_{k=k_b}^{k=k_t} \frac{\psi(s_k, \tilde{u}(s_k, \mathbf{x}), \tilde{v}(s_k, \mathbf{x}))}{|\mathbf{x} - \mathbf{y}(s_k)|}, \tag{19}$$

where

$$\tilde{u}(s_k, \mathbf{x}) = \frac{D(\mathbf{x} - \mathbf{y}(s_k)) \bullet \mathbf{d}_1}{(\mathbf{x} - \mathbf{y}(s_k)) \bullet \mathbf{d}_3}, \tag{20}$$

$$\tilde{v}(s_k, \mathbf{x}) = \frac{D(\mathbf{x} - \mathbf{y}(s_k)) \bullet \mathbf{d}_2}{(\mathbf{x} - \mathbf{y}(s_k)) \bullet \mathbf{d}_3}. \tag{21}$$

2.3. PARAMETER SELECTION

In the above numerical implementation of the Katsevich algorithm, the four basic discrete parameters are K, M, N, Q. K represents the number of total projections. An appropriate K is selected based on the requirement on the desired resolution and the length of the interval of s. M and N define the detector size, and $2Q+1$ is the number of Hilbert filtering lines on each projection.

The Katsevich algorithm is numerically evaluated using the 3D Shepp-Logan phantom[19]. In the following discussion, unless otherwise stated, all the parameters should remain the same as in Table 1.

Table 1. Cone-beam imaging parameters in the numerical simulation.

Scanning radius (R)	75cm
Source to detector distance (D)	150cm
Helical pitch (h)	25cm
Object radius (r)	25cm
Number of projections per turn	600
Scanning range (K=2101)	$s \in [-3.5\pi, 3.5\pi]$
Detector size (M=300,N=80)	300×80
Detector width (Δu)	0.356cm
Detector height (Δv)	0.439cm
Number of filtering lines (Q=0.65N)	105
Reconstruction matrix	$256 \times 256 \times 256$

3. FOUR TYPES OF ARTIFACTS

In the numerical simulation, we find four types of artifacts associated with Katsevich algorithm, which are respectively referred to as ***Texture***, ***Streak***, ***Shadow*** and ***Wrinkle*** artifacts. In the following, we report the appearance, cause and remedy for each of them.

3.1 TEXTURE ARTIFACTS

First, we discuss ***"texture"*** artifacts as illustrated in Fig. 3. It is well known that a fixed voxel $\mathbf{x}$ belongs to one and only one PI-Line, and it can be reconstructed from data collected within the source angular interval defined by the PI-Line. Significant "***Texture***" artifacts appeared when the endpoints of the PI-Line were not treated appropriately. Recall that the two endpoints are parameterized by continuous angular values $s_b(\mathbf{x})$ and $s_t(\mathbf{x})$. In the numerical implementation of the backprojection step, we used discrete counterparts s_{k_b} and s_{k_t}. The "***Texture***" artifacts in Fig 3 were caused by the truncation errors from $s_b(\mathbf{x})$ and $s_t(\mathbf{x})$.

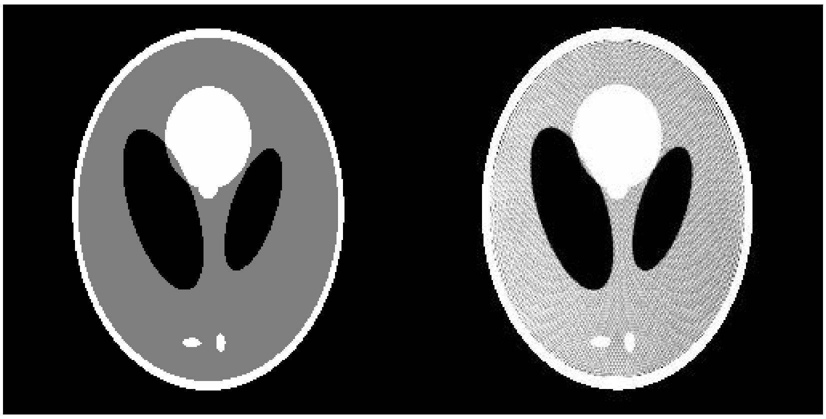


Figure 3. Illustration of texture artifacts. With the left being the original slice at z=-6.25cm, the right contains texture artifacts. The gray-scale interval [1.015, 1.025] is mapped to [0, 255].

To reduce ***"Texture"*** artifacts, we can approximate $s_b(\mathbf{x})$ as an interpolation between $s_{(k_b-1)}$ and s_{k_b}, and $s_t(\mathbf{x})$ as an interpolation between $s_{(k_t+1)}$ and s_{k_t} Here we assume that they satisfy $s_{(k_b-1)} < s_b(\mathbf{x}) < s_{k_b}$ and $s_{k_t} < s_t(\mathbf{x}) < s_{(k_t+1)}$. Therefore, formula (19) becomes:

$$f(\mathbf{x}) = -\frac{\Delta s}{2\pi^2} \sum_{k=k_b-1}^{k=k_t+1} \frac{\psi(s_k, \tilde{u}(s_k,\mathbf{x}), \tilde{v}(s_k,\mathbf{x}))\lambda(k)}{|\mathbf{x}-\mathbf{y}(s_k)|}, \tag{22}$$

with the coefficients

$$\lambda(k) = \begin{cases} 0.5\lambda_b^2 & k = k_b - 1 \\ 0.5 + \lambda_b - 0.5\lambda_b^2 & k = k_b \\ 1 & k_b + 1 \le k \le k_t - 1, \\ 0.5 + \lambda_t - 0.5\lambda_t^2 & k = k_t \\ 0.5\lambda_t^2 & k = k_t + 1 \end{cases} \tag{23}$$

$$\lambda_b = \frac{s_{k_b} - s_b(\mathbf{x})}{\Delta s}. \tag{24}$$

$$\lambda_t = \frac{s_t(\mathbf{x}) - s_{k_t}}{\Delta s} \tag{25}$$

3.2 STREAK ARTIFACTS

In the numerical implementation, M and N are the numbers of detectors along horizontal and vertical directions, and $2Q+1$ is the number of Hilbert filtering lines on each projection. Since the Hilbert filtering is performed on a group of non-horizontal lines, as shown in Fig. 4, N row derivative data are rebinned into $2Q+1$ filtering lines. It is important to select a suitable value for Q. When Q was too small relative to N, ***"Steak"*** artifacts appeared, as shown in Fig. 5. In the slice z=-6.25 cm, it is observed that dark streak artifacts emerged from the centers of the internal ellipsoids. In our experiments, this type of artifacts can be effectively reduced by setting $0.3N < Q < 0.7N$.

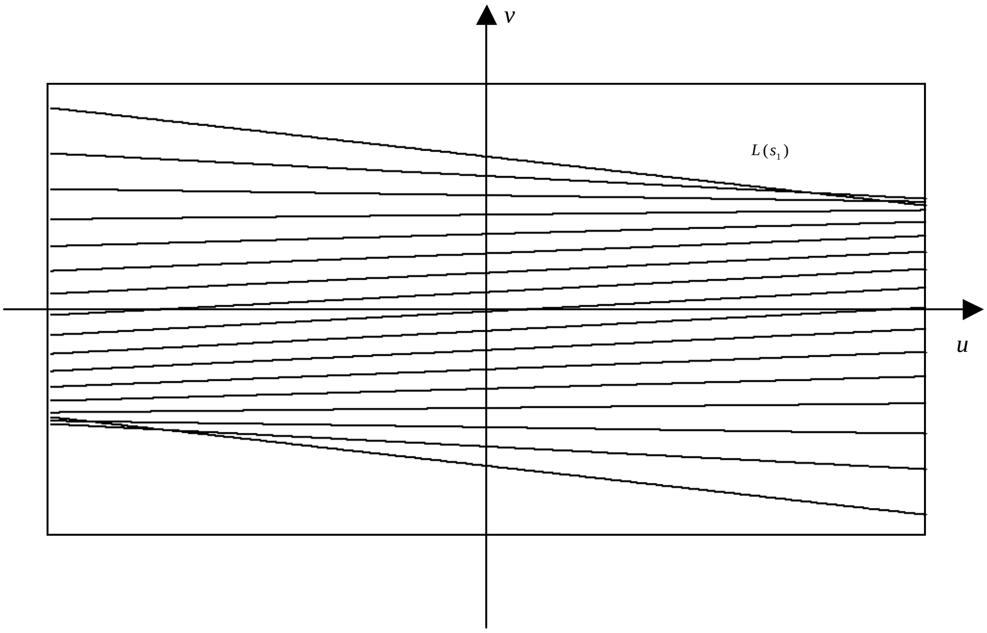


Figure 4. Illustration of Hilbert filtering lines

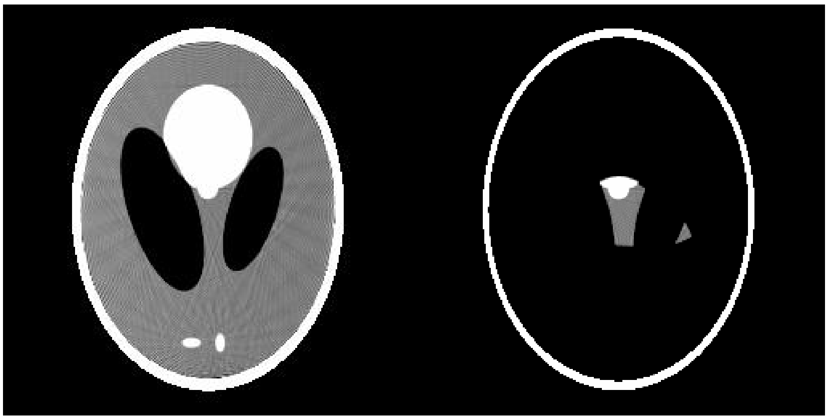


Figure 5. Illustration of steak and shadow artifacts. The left is reconstructed slice at z=-6.26cm with steak artifacts using Q=0.13N. The right is slice with shadow artifacts using a window width L=M. The gray-scale interval [1.015, 1.025] is mapped

3.3 SHADOW ARTIFACTS

The discrete Hilbert transform (17) can be performed in the Fourier domain using FFT. However, the ideal Hilbert transform is not numerically feasible because it has an infinite duration. Hence, we truncated it using a window function with a finite duration L . Various window functions may yield different artifacts. Among typical windows including rectangular, Hamming, Hanning and Blackman functions, we found that the rectangular window is the best in terms of the image quality. Because the discrete Hilbert transform (17) performs a linear convolution but the FFT method corresponds to a circular convolution, the convolution from FFT are not always equal to the results from (17) without compensating for aliasing. Hence, even given a rectangular window, substantial "***shadow***" artifacts may occur if the duration L is not well designed. For example, as shown in Fig5, serious shadow artifacts appeared when the window duration L was equal to the number of detector column M . To overcome the aliasing induced "***shadow***" artifacts, the duration L must satisfy $L \geq 2M - 1$. Considering the computing speed and the antisymmetry of discrete Hilbert transform kernel, we suggest the following two rules:

(a) Select an odd number L satisfying $L \geq 2M - 1$;
(b) Extend the data length to a nearest power of 2 with zero padding.

3.4 WINKLE ARTIFACTS

In the filtering step, we have to resort the derivative data along the filtering lines before filtering and estimate the filtered data at the corresponding detector positions using interpolation. Also, we need interpolate the filtered data on any voxel projection position in the backprojection step (Formula (22)). All of these interpolation operators may cause artifacts. In the simulation, we evaluated two common interpolation methods in the backprojection step. They are nearest-neighbor interpolation and bi-linear interpolation. In the nearest-neighbor interpolation, backprojection is done only according to the nearest detector position. In the bi-linear interpolation, backprojection is performed linearly combining data of the four adjacent detector positions. As shown in Fig. 6, the nearest-neighbor interpolation produced ***"Winkle"*** artifacts while bi-linear interpolation largely eliminated them. Furthermore, we can further suppress this type of artifacts using the following additional means:

(a) **Utilize a more sophisticated interpolation method:** A more complex interpolation method may further suppress winkle artifacts. However, the improvement is limited up to the resolution inherent in the projection data. Specially, bi-linear interpolation should be adequate if the projection parameters K, M and N are not too small.

(b) **Increase data sampling rates:** Increasing projection data can accordingly decrease the interpolation error. It involves the number of projections per turn and the detector size M and N (inversely proportional to the detector size parameters Δu and Δv). The right of Fig. 6 shows reconstructed slice at z=-6.25cm with K =4550, M =700 and N =180. Theoretically, K, M and N can be infinite, but it is not feasible in practice.

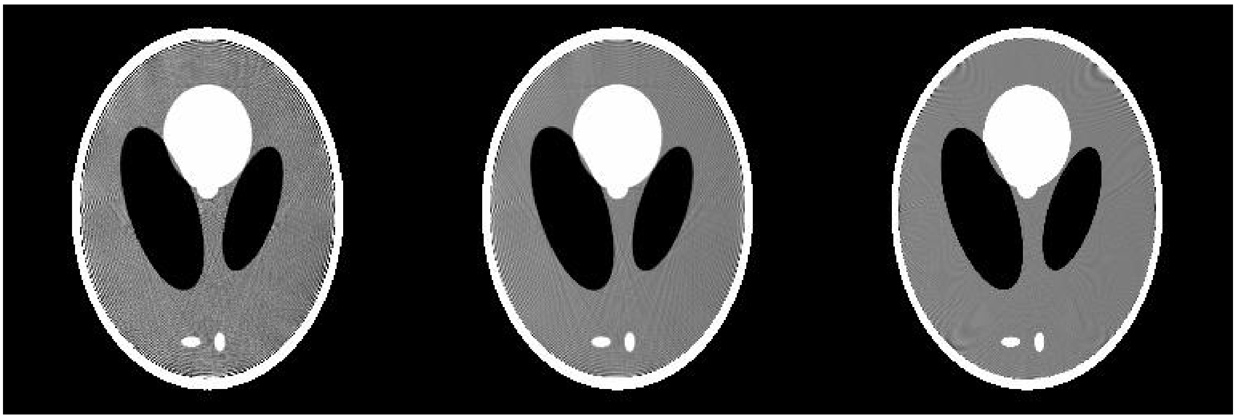


Figure 6. Illustration of winkle artifacts. The left and middle images are reconstructed slices at z=-6.25cm using nearest-neighbor interpolation and bi-linear interpolation, respectively. The right is the counterpart using more projection data. The gray-scale interval [1.015, 1.025] is mapped to [0, 255].

(c) **Apply a more suitable derivative formula:** We used formula (14), which is a discrete version of formula (5), to compute the derivative of projection data. Since the winkle artifacts are related to the three difference operators in formula (14), modifying difference operators can suppress these artifacts. Note that the definition of the cone-beam data derivative is:

$$D_g(s,\boldsymbol{\beta}) = \lim_{\zeta \to 0} \frac{g(s+\zeta,\boldsymbol{\beta}) - g(s,\boldsymbol{\beta})}{\zeta} \qquad (26)$$

where $\boldsymbol{\beta}$ is a fixed unit vector in the global coordinate system. In our local coordinate system illustrated in Fig.1, it can be directly formulated as:

$$D_g(s_{k+0.5},u_m,v_n) \approx \frac{g(s_{k+1},u^+,v^+)-g(s_k,u^-,v^-)}{\Delta s} \tag{27}$$

$$u^+ = \frac{Du_m \cos(0.5\Delta s) + D^2 \sin(0.5\Delta s)}{D\cos(0.5\Delta s) - u_m \sin(0.5\Delta s)} \tag{28}$$

$$v^+ = \frac{Dv_n}{D\cos(0.5\Delta s) - u_m \sin(0.5\Delta s)} \tag{29}$$

$$u^- = \frac{Du_m \cos(0.5\Delta s) - D^2 \sin(0.5\Delta s)}{D\cos(0.5\Delta s) + u_m \sin(0.5\Delta s)} \tag{30}$$

$$v^- = \frac{Dv_n}{D\cos(0.5\Delta s) + u_m \sin(0.5\Delta s)} \tag{31}$$

Shown as the left of Fig. 7, formula (27) greatly suppressed winkle artifacts. However, we must point out that it also introduced serious blurring when the projections number was not sufficiently large, in which case formula (14) still worked well. Clearly, although formula (27) avoids two difference operators, it uses two interpolation operators, usually blurring some image details.

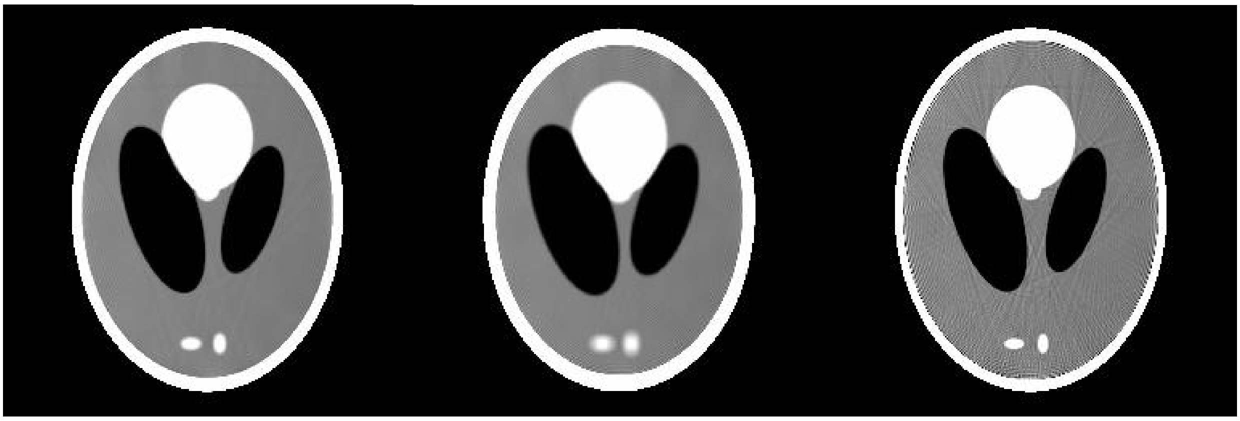


Figure 7. Impact of formula (27). The left is reconstructed slice at z=-6.25cm using the same parameters as in Table 1. The middle is reconstructed slice using 300 projections per turn and the blurring is more evident. Theright presents reconstructed slice using the same parameters as that for the middle based on formula (14), producing little blurring. The gray-scale interval [1.015, 1.025] is mapped to [0, 255].

4. CONCLUSION

Despite the exactness of the Katsevich reconstruction formula in the continuous domain, significant artifacts may appear when it is numerically implemented. Based on the implementation in planar detector geometry, we have studied four types of such artifacts, including "***texture***", ***"streak"***, ***"shadow"*** and ***"winkle"*** artifacts. Also, we have reported a number of effective means to suppress these four types of artifacts. Our results should be valuable to apply the Katsevich algorithm optimally for practical spiral cone-beam CT.

ACKNOWLEDGEMENTS

We would like to thank Dr. Shiying Zhao for valuable discussions. This work was supported in part by the NIH grant R01 EB002667.

REFERENCES

1. M. J. Yaffe and J. A. Rowlands, “X-ray detectors for digital radiology,” *Physics in Medicine and Biology,* 42:1-39, 1997.
2. G. Wang, C.R. Crawford, W.A. Kalender, “Multi-row-detector and cone-beam spiral/helical CT”, *IEEE Trans. Med. Imaging* 19(9):817-821, 2000.
3. K. C. Tam, S. Samarasekera, and F. Sauer; Exact cone-beam CT with a spiral scan; *Physics in Medicine and Biology*, vol. 43, pp. 1015–1024, 1998.
4. M. Defrise, F. Noo, and H. Kudo; A solution to the long-object problem in helical cone-beam tomography; *Physics in Medicine and Biology*, vol. 45, pp. 623–643, 2000.
5. S. W. Lee, G. Cho, and G. Wang, “Artifacts associated with implementation of the Grangeat formula”, *Med. Phys*. 29(12):2871-2880, 2002.
6. A. Katsevich; “Theoretically exact fbp-type inversion algorithm for spiral CT,” *SIAM J. Appl.Math,* vol. 62, pp. 2012-2026, 2002.
7. A. Katsevich, “Improved exact FBP algorithm for spiral CT”, *Adv. Appl. Math*, 32:681-697, 2004.
8. A. Katsevich, “Analysis of an exact inversion algorithm for spiral cone-beam CT”. *Physics in Medicine and Biology*. 47: 2583-2597, 2002.
9. A. Katsevich., et al. “Evaluation and empirical analysis of an exact FBP algorithm for spiral cone-beam CT”, *Proceedings of SPIE* Vol 5032 II: 663-674, 2003, San Diego.
10. Y. Zou and X. Pan, “Three-term exact fbp reconstruction in cone-beam helical CT”, in 2003 *IEEE Nuclear Science Symposium Conference Record*, in press, 2004.
11. Y. Zou and X. Pan, “Exact image reconstruction on PI-lines in helical cone beam CT,” *Physics in Medicine and Biology*. 49: 941-959, 2004.
12. Y. Ye, J. Zhu and G. Wang, “Minimum Detection Windows, PI-Line Existence, and Uniqueness for Helical Cone-Beam Scanning of Variable Pitch”, *Medical Physics;* 31(3):566-572, 2004.
13. Y. Ye, J. Zhu and G. Wang, “Geometric Studies on Variable Radius Spiral Cone Beam Scanning,” *Medical Physics*, 31(6):1473-1480, 2004.
14. H. Yu, Y. Ye and G. Wang, “Katsevich-type algorithms for variable radius spiral cone-beam CT”, *Proceedings of SPIE*, Vol 5535, August 2004.
15. Y. Ye and G. Wang, “Cone beam reconstruction along a 3D spiral of a variable radius and/or variable pitch”, *Proceedings of SPIE*, Vol 5535, August 2004.
16. S. Zhao, H. Yu and G. Wang, “A family of Katsevich-type FBP algorithms for spiral cone-beam CT”, *Proceedings of SPIE*, Vol 5535, August 2004.
17. F. Noo, J. Pack, and D. Heuscher, “Exact helical reconstruction using native cone-beam geometries”, *Physics in Medicine and Biology*., 48:3787-3818, 2003.
18. H. Yu and G. Wang, “Studies on Implementation of the Katsevich Algorithm for Spiral Cone-beam CT”, *Journal of X-ray Science and Technology*; 12(3), 2004.
19. G. Wang, T. H. Lin, P. C. Cheng, D. M. Shinozaki, “A general cone-beam reconstruction algorithm”. *IEEE Trans. Med. Imaging*:12(3):486-496, 1993.