Fast local reconstruction by selective backprojection for low dose in dental computed tomography [*]

YAN Bin(闫镔)[1)]   DENG Lin(邓林)[2)]   HAN Yu(韩玉)   ZHANG Feng(张峰)

WANG Xian-Chao (汪先超)   LI Lei(李磊)[3)]

National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450002, China

Abstract: The high radiation dose in computed tomography (CT) scans increases the lifetime risk of cancer, which becomes a major clinical concern. The backprojection-filtration (BPF) algorithm could reduce the radiation dose by reconstructing the images from truncated data in a short scan. In a dental CT, it could reduce the radiation dose for the teeth by using the projection acquired in a short scan, and could avoid irradiation to the other part by using truncated projection. However, the limit of integration for backprojection varies per PI-line, resulting in low calculation efficiency and poor parallel performance. Recently, a tent BPF has been proposed to improve the calculation efficiency by rearranging the projection. However, the memory-consuming data rebinning process is included. Accordingly, the selective BPF (S-BPF) algorithm is proposed in this paper. In this algorithm, the derivative of the projection is backprojected to the points whose x coordinate is less than that of the source focal spot to obtain the differentiated backprojection. The finite Hilbert inverse is then applied to each PI-line segment. S-BPF avoids the influence of the variable limit of integration by selective backprojection without additional time cost or memory cost. The simulation experiment and the real experiment demonstrated the higher reconstruction efficiency of S-BPF.

Key words: local reconstruction, dental CT, selective backprojection, short scan

PACS: 87.59.-e, 07.85.-m   **DOI:** 10.1088/1674-1137/38/10/108201

1 Introduction

X-ray computed tomography (CT) has been widely used in dental CT for diagnosis. One of the objectives of modern CT is to reduce the radiation dose because of the health risks caused by X-rays. In dental CT, projection data are collected by irradiating the brain. The global reconstruction algorithm requires projection without truncated data to obtain the image of dens, which means the brain must be irradiated by a higher radiation dose. However, the local reconstruction algorithm can deal with truncated projection data [1].

Recently inspired by the ATRACT algorithm [2] which can restrain the truncation artifacts to some extent by using local filter and global filter, Wang [3] proposed an filtration-backprojection (FBP) algorithm based on the Radon inversion transform (RIT) and achieved fast reconstructing without reducing the quality of the reconstruction. But the ability to restrain the truncation artifacts is limited.

The backprojection-filtration (BPF) algorithm can be used for local reconstruction. This algorithm was developed for image reconstruction on PI-line segments [4] in a helical cone-beam scan, and it provided a strategy for reconstructing the exact region of interest (ROI) using truncated data [5–11]. Because the circular scanning trajectory is easy to implement and control in practice, it is widely used in CT [12]. Based on the concept of virtual PI-line and virtual circular orbit, Yu et al. modified the BPF algorithm to reconstruct images in a circular cone-beam scan [13]. The modified algorithm is an excellent work for low-dose dental CT because it can deal with the local reconstruction in the short scan. However, the modified algorithm is not a good choice for practical dental CT because of its low reconstruction efficiency and poor parallel performance [14]. The tent backprojection-filtration (T-BPF) algorithm [14] has been recently developed to obtain a fast reconstruction from truncated

data in circular cone-beam CT. This algorithm improves the reconstruction efficiency and parallel performance by making projection data rebinned into a tent-like parallel-beam format from the cone-beam format. And it has the same parallel performance as the FDK (Feldkamp, Davis, Kress) algorithm [15]. However, data rebinning is included in the T-BPF algorithm. Data rebinning not only introduces errors caused by trilinear-interpolation but also uses a considerable amount of memory to store the projection data acquired by the detector at all views and the rebinned projection data.

In this work, S-BPF was developed for fast reconstruction from truncated data in circular cone-beam CT. In S-BPF, the derivative of the projections is firstly backprojected to the points whose x coordinate is less than that of the source focal spot to obtain the differentiated backprojection (DBP). The finite Hilbert inverse [16] is then applied to each PI-line segment to reconstruct images from the DBP. Compared with the T-BPF algorithm, S-BPF avoids the influence of the variable integration interval by selective backprojection instead of data rebinning. Only one view projection is loaded at any time in S-BPF. Therefore, the reconstruction efficiency of S-BPF is considerably higher than that of the BPF algorithm. The memory cost of S-BPF is also considerably less than that of the BPF and T-BPF algorithms.

Similar to the BPF and T-BPF algorithms, S-BPF can also use the projection data acquired in a short scan, which can reduce the dose of radiation and the time of data acquisition under the same sampling frequency, compared with algorithms in a full scan. S-BPF can also deal with the truncated projection data, which can reduce the dose of radiation and achieve local reconstruction, compared with the global reconstruction algorithm.

This paper is organized as follows. Section 2 describes the geometrical structure, recalls the BPF and T-BPF algorithms, and introduces the S-BPF algorithm in detail. Section 3 demonstrates the advantages of S-BPF based on the results of reconstruction in the simulation experiment and the real experiment by comparing with the original BPF algorithm and the T-BPF algorithm. Finally, Section 4 gives the conclusion.

2   Methods

This section describes the geometrical structure for the circular cone-beam scan, recalls the BPF and T-BPF algorithms, and describes S-BPF in detail.

2.1   Geometrical structure

The cone-beam geometry is shown in Fig. 1. $Oxyz$ represents a Cartesian coordinate system of the world. Z is the rotation axis of the scanner. $S(\beta)$ is the position of the source focal spot in the Oxy plane and is uniquely determined by the angular parameter β. The

position of the source focal spot can be expressed using the following equation:

$$S(\beta) = R \cdot (\sin\beta, \cos\beta, 0), \quad \beta \in [\beta_{\text{start}}, \beta_{\text{end}}], \qquad (1)$$

where R is the distance between the origin of the Cartesian coordinate system of the world and the source, and β_{start}, β_{end} correspond to the starting and ending points of the circular orbit respectively. Projection data are collected on a flat-panel detector. (a,b) represents the local Cartesian coordinate system of the detector. The distance between the source and the detector is fixed, and the origin of the Cartesian coordinate system of the detector is at the line which is determined by the source and the origin of the Cartesian coordinate system of the world. Every X-ray and its projection can be uniquely determined by $P(\beta,a,b)$.
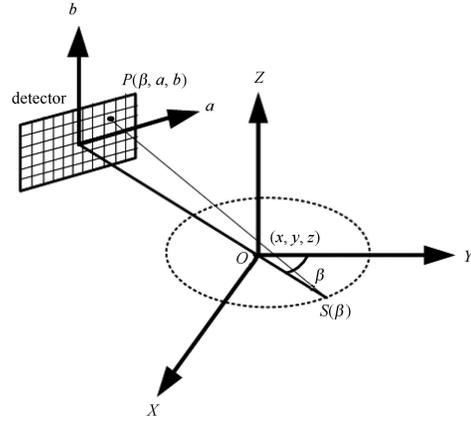

Fig. 1.   The geometrical structure for circular cone-beam scan.

2.2   BPF and T-BPF

The DBP is described, which is the first of the two steps in the BPF algorithm [11] and it can be obtained by backprojecting the derivative of the projections, as shown in Formula (2):

$$b(\vec{r}) = \int_{\beta_{\text{start}}}^{\beta_{\text{end}}} P'(\beta, a_{\text{r}}, b_{\text{r}}) \mathrm{d}\beta, \qquad (2)$$

where $\vec{r}$ denotes any point on a PI-line, $P'(\phi, a_{\text{r}}, b_{\text{r}}) = \partial P(\phi, a_{\text{r}}, b_{\text{r})}/\partial a_{\text{r}}$, and $[\beta_{\text{start}}, \beta_{\text{end}}]$ denotes the view-angle integration interval of backprojection. In general, we make every PI-line parallel to the Y axis, and every PI-line can be uniquely determined by $\text{PI}(x,z)$ (the same applies to T-BPF and S-BPF). Therefore, the integration interval of backprojection is the same for PI-line segments on the plane $x = k$, where k is an arbitrary constant. The relationship between the PI-line and scan angle is shown in Fig. 2. β_{start} and β_{end} are determined by the plane $x=k$, as shown in Formula (3):

$$\begin{cases} \beta_{\text{start}} = \arcsin(x/R) \\ \beta_{\text{end}} = \pi - \arcsin(x/R). \end{cases} \qquad (3)$$
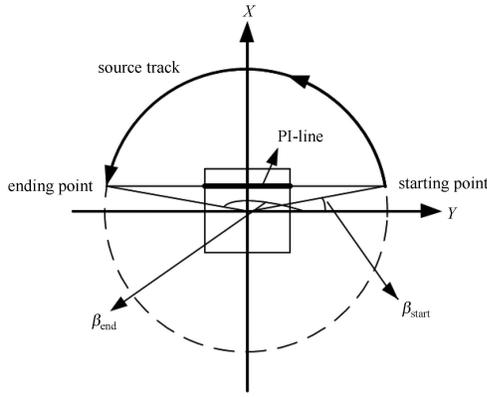
Fig. 2.   Relation between the PI-line and scan angle in BPF (bottom view for Fig. 1).

The object function $f(\vec{r})$ and $b(\vec{r})$ have the following relation [11, 17, 18]:

$$b(\vec{r}) = -2\pi \mathrm{H}f(\vec{r}), \qquad (4)$$

where $\mathrm{H}f$ presents the Hilbert transform of f. The object function $f(\vec{r})$ can be obtained by applying the inverse of the finite Hilbert transform [16]. The BPF algorithm for CBCT is performed as follows:

1) To load all projections and determine the derivative of the projection.

2) To select a plane $x=k$ for reconstruction and determine the accordant β_{start} and β_{end}.

3) To obtain the DBP of the plane $x=k$, and backproject the derivative of the projection that corresponds to $\beta \in [\beta_{\mathrm{start}}, \beta_{\mathrm{end}}]$ on the points that are in the plane.

$$b(\vec{r}) = \int_{\beta_{\mathrm{start}}}^{\beta_{\mathrm{end}}} P'(\beta, a_{\mathrm{r}}, b_{\mathrm{r}}) \mathrm{d}\beta, \; \overrightarrow{r} \in \{(x,y,z)|x=k\}. \qquad (5)$$

4) To perform the finite Hilbert inverse on those PI-lines in the plane $x=k$. Repeat step 2 if not all planes are backprojected.

To improve parallel performance and reconstruction efficiency, T-BPF is performed by firstly rebinning the cone-beam data to tent-like parallel-beam data and then applying the BPF-type algorithm to reconstruct images.

A virtual detector is introduced to describe the formula of rebinning conveniently. It is placed on the center of rotation and parallel to the real detector. $P(\beta, m, n)$ denotes the projection data acquired by the virtual detector and (m,n) is the position on the virtual detector; The rebinned projection data can be parameterized by $P_{\mathrm{tent}}(\theta, s, t)$, where (s,t) is the position after rebinning. The rebinning formula is implemented using the following equation:

$$P_{\mathrm{tent}}(\theta, s, t) = P\left(\theta - \arcsin\frac{s}{R}, \frac{sR}{\sqrt{R^2 - s^2}}, \frac{tR^2}{R^2 - s^2}\right). \quad (6)$$

After rebinning, the cone-beam data are arranged into parallel-beam data on a virtual rectangular detector

plane shown in Fig. 3. In the image reconstruction from the parallel-beam projection data, the integration limit of view-angle for backprojection is fixed (from 0 to π) for every PI-line, and the DBP can be described as

$$b(\vec{r}) = \int_0^{\pi} P'_{\mathrm{tent}}(\theta, a_{\mathrm{r}}, b_{\mathrm{r}}) \mathrm{d}\theta. \qquad (7)$$

The T-BPF algorithm for CBCT is performed as follows:

1) To load all projections.

2) To rearrange the cone-beam projection to tent-like parallel-beam projection and determine the derivative of the rebinned projection.

3) To obtain the DBP of the $f(x)$, and backproject the derivative of the rebinned projection that corresponds to $\beta \in [0, \pi]$ to $f(x)$.
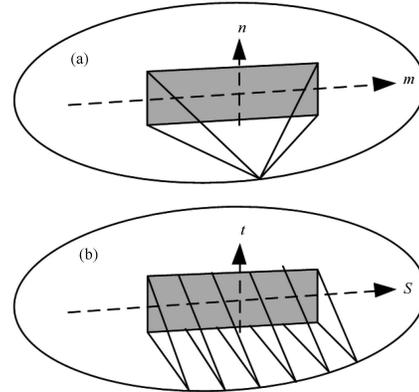
4) To carry out the finite Hilbert inverse on the PI-line.

Fig. 3.   Cone-beam projection data at a view (a) tent-like parallel projection data at a view after rebinning (b), where (t,s) are the coordinates on the virtual detector in T-BPF.

2.3   S-BPF

Backprojection is the most time-consuming part in the BPF algorithm. T-BPF improves backprojecting efficiency by rearranging projection data because the variable limit of integration of backprojection is converted to a fixed limit of integration. However, T-BPF introduces the data rebinning process. In this paper, we proposed S-BPF, which improves the backprojecting efficiency by selective backprojection without rearranging projection data, reading projection repeatedly from hard disk, or loading all projections beforehand. S-BPF has greatly reduced memory cost compared with the BPF and T-BPF algorithms. S-BPF also greatly reduces the time cost compared with the BPF algorithm.

We modified Formula (7) to avoid the influence of the variable limit of integration for the backprojection.

Formula (7) can be described in detail using the following equation:

$$b(x,y,z) = \int_{X_{\mathrm{min}}}^{X_{\mathrm{max}}} \delta(\tilde{x} - x) \cdot \int_{\beta_{\mathrm{start}}}^{\beta_{\mathrm{end}}} \int_{Y_{\mathrm{min}}}^{Y_{\mathrm{max}}} \delta(\tilde{y} - y)$$

$$\times \int_{Z_{\min}}^{Z_{\max}} \delta(\tilde{z}-z)P'(\beta,a_{\mathrm{r}},b_{\mathrm{r}})\mathrm{d}\tilde{z}\mathrm{d}\tilde{y}\mathrm{d}\beta\mathrm{d}\tilde{x}, \tag{8}$$

where $\delta(x)$ is pulse function; $X_{\min}$, $X_{\max}$, $Y_{\min}$, $Y_{\max}$, $Z_{\min}$, and $Z_{\max}$ denote the minimum and maximum of objects in the x, y, and z directions, respectively. To make the limit of integration the same for every PI-line, Formula (8) is transformed as follows:

$$b(x,y,z) = \int_{X_{\min}}^{X_{\max}} \delta(\tilde{x}-x)\cdot \int_0^{2\pi} u(\beta-\beta_{\mathrm{start}})u(\beta_{\mathrm{end}}-\beta)$$
$$\times \int_{Y_{\min}}^{Y_{\max}} \delta(\tilde{y}-y)\int_{Z_{\min}}^{Z_{\max}} \delta(\tilde{z}-z)P'(\beta,a_{\mathrm{r}},b_{\mathrm{r}})$$
$$\mathrm{d}\tilde{z}\mathrm{d}\tilde{y}\mathrm{d}\beta\mathrm{d}\tilde{x}, \tag{9}$$

where $u(x)$ is the step function. In this way, integrating factors $\mathrm{d}x$, $\mathrm{d}y$, $\mathrm{d}z$, and $\mathrm{d}\beta$ are unattached. Thus, the order of integration can be changed as

$$b(x,y,z) = \int_0^{2\pi} \int_{X_{\min}}^{X_{\max}} \int_{Y_{\min}}^{Y_{\max}} \int_{Z_{\min}}^{Z_{\max}} u(\beta-\beta_{\mathrm{start}})u(\beta_{\mathrm{end}}-\beta)$$
$$\times\delta(\tilde{x}-x)\delta(\tilde{y}-y)\delta(\tilde{z}-z)P'(\beta,a_{\mathrm{r}},b_{\mathrm{r}})\mathrm{d}\tilde{z}\mathrm{d}\tilde{y}\mathrm{d}\tilde{x}\mathrm{d}\beta. \tag{10}$$

$x<R\sin\beta$ can be obtained from the geometry relation between the angle β_{start} β_{end} and PI-line as Formula (3). Thus, Formula (10) can be simplified as follows:

$$b(x,y,z) = \int_0^{2\pi} \int_{X_{\min}}^{X_{\max}} \int_{Y_{\min}}^{Y_{\max}} \int_{Z_{\min}}^{Z_{\max}} u(R\sin\beta-x)\delta(\tilde{y}-y)$$
$$\times\delta(\tilde{z}-z)P'(\phi,a_{\mathrm{r}},b_{\mathrm{r}})\mathrm{d}\tilde{z}\mathrm{d}\tilde{y}\mathrm{d}\tilde{x}\mathrm{d}\beta. \tag{11}$$

No point is backprojected when the source is at the broken line in Fig. 4. Thus, the projection obtained when the source is at the broken line does not contribute to reconstruction, so the limit of integration of β can be cut short and Formula (11) can be written as follows:

$$b(x,y,z) = \int_\psi^{\pi-\psi} \int_{X_{\min}}^{X_{\max}} \int_{Y_{\min}}^{Y_{\max}} \int_{Z_{\min}}^{Z_{\max}} u(R\sin\beta-x)\delta(\tilde{y}-y)$$
$$\times\delta(\tilde{z}-z)P'(\phi,a_{\mathrm{r}},b_{\mathrm{r}})\mathrm{d}\tilde{z}\mathrm{d}\tilde{y}\mathrm{d}\tilde{x}\mathrm{d}\beta, \tag{12}$$

where $\psi=\arcsin(X_{\min}/R)$. Compared with the BPF algorithm, S-BPF is performed in such a way that plane $x=k$ is selected by the projection of every view instead of the way the projection is selected by plane $x=k$.

The shadow in Fig. 4 is backprojected by the derivative of the projection that corresponds to β. Only one projection is loaded at any time, and every projection is loaded only one time in memory. After backprojecting the derivative of the projection that corresponds to $\beta\in[\psi,\pi-\psi]$ (where $\psi=\arcsin(X_{\min}/R)$), every PI-line is backprojected by the derivative of the projection that corresponds to $\beta\in[\beta_{\mathrm{start}},\beta_{\mathrm{end}}]$, and the DBP of $f(x)$ is obtained. The S-BPF algorithm for CBCT is performed as follows:

1) Determine the ψ.

$$\psi=\arcsin(X_{\min}/R). \tag{13}$$

2) Load a projection ($\beta \in [-\psi,\pi+\psi]$), and get the derivative of the projection.

$$P'(\beta,a,b)=\frac{\partial}{\partial a}P(\beta,a,b). \tag{14}$$

3) Backproject the derivative of the projection to the PI-line, where $x<R\sin\beta$.

$$b(x,y,z) = \int_{X_{\min}}^{R\sin\beta} \int_{Y_{\min}}^{Y_{\max}} \int_{Z_{\min}}^{Z_{\max}} \delta(\tilde{y}-y)\delta(\tilde{z}-z)$$
$$\times P'(\phi,a_{\mathrm{r}},b_{\mathrm{r}})\mathrm{d}\tilde{z}\mathrm{d}\tilde{y}\mathrm{d}\tilde{x}. \tag{15}$$

4) Proceed to step 5) if all projections ($\beta\in[-\psi,\pi+\psi]$) are used or return to step 2).
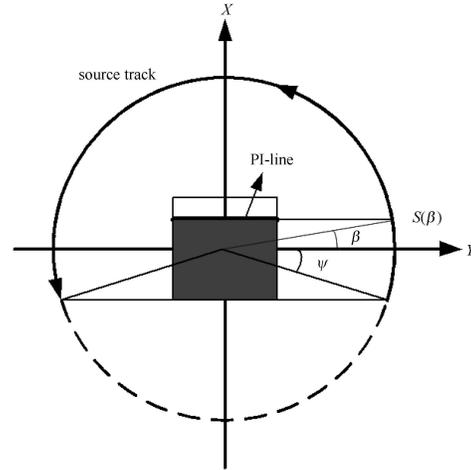
5) Carry out the finite Hilbert inverse on the PI-line.

Fig. 4.   Relationship between the PI-line and scan angle in S-BPF (bottom view for Fig. 1).

$$f(x,y,z)=\frac{1}{2\pi}\frac{1}{\sqrt{(y_{c2}-y)(y-y_{c1})}}\times\left[\int_{y_{c1}}^{y_{c2}}\frac{\mathrm{d}y_{\mathrm{c}}}{\pi(y-y_{\mathrm{c}})}\sqrt{(y_{c2}-y_{\mathrm{c}})(y_{\mathrm{c}}-y_{c1})}b(x,y_{\mathrm{c}},z)+C\right], \tag{16}$$

where y_{c1} and y_{c2} denote the y coordinates of the two end points of the PI-line. The constant C is obtained by

$$C=\frac{2\pi P_\pi(x,z)-\int_{y_{c1}}^{y_{c2}}\int_{y_{c1}}^{y_{c2}}\frac{\sqrt{(y_{c2}-y')(y'-y_{c1})}}{\sqrt{(y_{c2}-y_{\mathrm{c}})(y_{\mathrm{c}}-y_{c1})}}\frac{b(x,y',z)}{\pi(y_{\mathrm{c}}-y')}\mathrm{d}y'\mathrm{d}y_{\mathrm{c}}}{\int_{y_{c1}}^{y_{c2}}\frac{1}{\sqrt{(y_{c2}-y_{\mathrm{c}})(y_{\mathrm{c}}-y_{c1})}}\mathrm{d}y_{\mathrm{c}}}, \tag{17}$$

where $P_\pi(x,z)$ denotes the line integral of the object function along the PI-line PI(x,z), it can be obtained from the projection data when the PI-line is in the mid-plane ($z=0$). But for off-mid-planes ($z \neq 0$), we use Formula (18) as follows:

$$P_\pi(x,y)$$
$$= \frac{P(\arcsin(x/R),a_1,b_1)+P(\pi-\arcsin(x/R),a_2,b_2)}{2}, \quad (18)$$

where (a_1,b_1) and (a_2,b_2) indicate the position of the projection from the view angle $\arcsin(x/R)$ and $\pi-\arcsin(x/R)$ for the point $(x,(y_{c1}+y_{c2})/2,z)$, which is the midpoint on the PI-line segment.

3 Implementation and experiment

To testify the advantages of the S-BPF method presented above, the three algorithms were implemented in C language for the simulation experiment and the real experiment. The experiments were performed by using the CPU(Inter(R) Xeon(R) X5450 @3.00 GHz).

3.1 Simulation experiment

The standard 3D Shepp-Logan head phantom [19] was used for reconstruction to compare S-BPF with the T-BPF and BPF algorithms. The length, width, and height of the head phantom were 7.14331 mm. The distance between the source and the rotation axis of the scanner was 477 mm and the distance between the source and the detector was 1265 mm. The cone-beam projection data were acquired from a flat-panel detector with 500 pixels×500 pixels and 0.148 mm per pixel. 360 projection views were uniformly distributed over the 2π circular trajectory. However, all the algorithms only used the views in the range of π plus twice the cone angle. The final reconstruction was done on a 256×256×256 grid.

The root mean square error (RMSE) [20] was introduced to evaluate the reconstruction results, and it can be written as follows:

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}[f_t(i)-f_0(i)]^2}, \quad (19)$$

where f_t and f_0 denote the reconstructed and the reference images of voxels.

Figure 5 presents the model of Shepp-Logan head phantom and the reconstruction results of three algorithms for the model. The three orthogonal planes ($x=0$, $y=0$ and $z=0$) of 3D images reconstructed by the BPF, T-BPF and S-BPF algorithms were compared. At the same time, the corresponding profiles on the middle horizontal line are shown in Fig. 6.
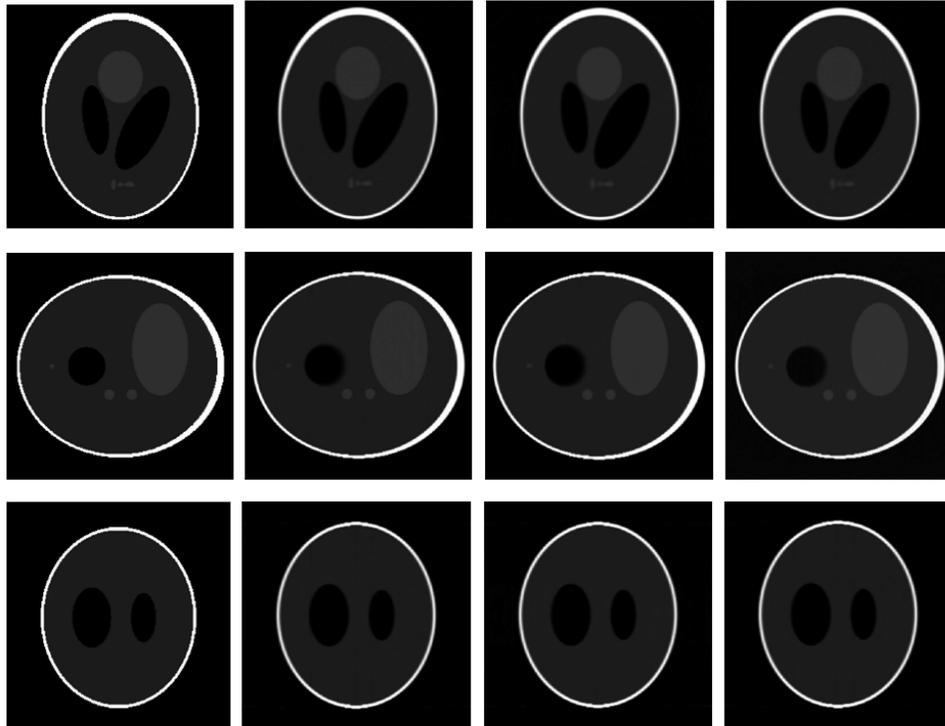

Fig. 5.   2D slices in 3D images reconstructed using the BPF, T-BPF, and S-BPF algorithms. The 128th slices of the plane $z=0$, plane $x=0$ and plane $y=0$ are represented in the first, second, and third rows respectively. The slices of the Shepp-Logan model and the reconstruction results of the BPF, T-BPF, and S-BPF algorithms are represented in the first, second, third, and fourth columns respectively.

Figure 6 and Table 1 show no significant differences among the reconstruction results of the original BPF algorithm, the T-BPF algorithm and S-BPF. However, in the case of serial computing, the S-BPF is more efficient and memory saving than the BPF and T-BPF algorithms.
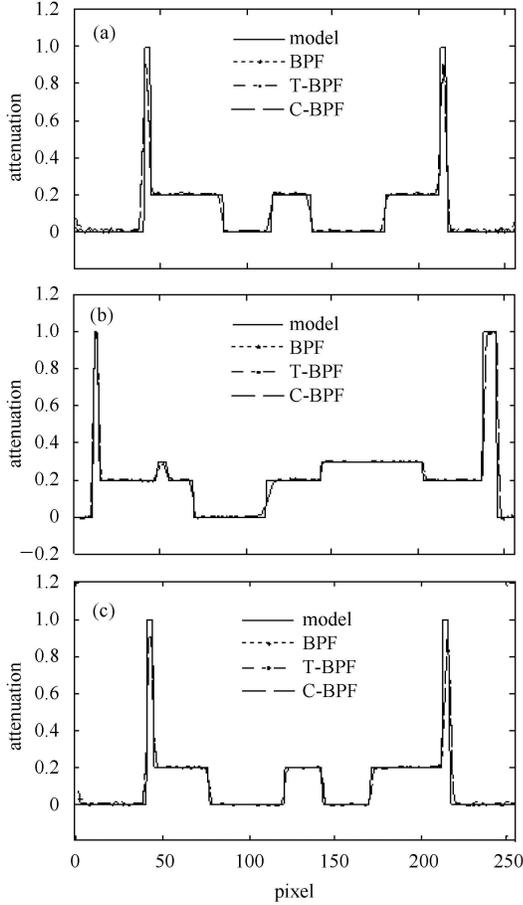

Fig. 6.   Profiles of the images shown in Fig. 5 along the horizontal middle lines. (a), (b), and (c) are the profiles of the first, second, and third rows of Fig. 5, respectively.

Table 1.   Reconstruction results of BPF, T-BPF, and S-BPF algorithms.

algorithm	size	time/min	memory/Mb	RMSE
BPF	256×256×256	16.05	417.36	0.0708
T-BPF	256×256×256	12.18	753.51	0.0701
S-BPF	256×256×256	11.71	49.23	0.0708

In BPF algorithms, four circles are present in its backprojection (three are dimensional direction circles and one is a view-angle circle). The integration limit of the view-angle is determined by the location of the plane $x=k$. Thus, a dimensional direction circle, must be implemented before the view-angle circle, which produces more calculations of the trigonometric function and the intermediate variable that corresponds to the y coordinate or z coordinate in backprojection. The above operations account for the low reconstruction efficiency of the BPF algorithm. The limit of integration for backprojection (about π) is required by every plane. Therefore, all projections must be loaded in the memory before; otherwise, every projection will be read repeatedly from the hard disk [21].

The T-BPF algorithm avoids the influence of the variable limit of integration for the backprojection by projection data rebinning. Thus, the T-BPF algorithm has a higher reconstruction efficiency than the BPF algorithm. Compared with the BPF algorithm, additional memory is used to keep the rebinned projection and the memory cost for the T-BPF algorithm is about twice over that for the BPF algorithm.

In S-BPF algorithms, the backprojection implementation not only reduced many repetitive calculations of trigonometric function, but also invoked no additional intermediate variables that corresponded to the y coordinate or z coordinate in the backprojection without projection data rebinning. The memory required by the S-BPF is considerably less than that required by the BPF and T-BPF algorithms because all projections for the BPF and T-BPF algorithms must be loaded to eliminate repeated readings of the projection data [21]. Only one view of the projection data is loaded for the S-BPF algorithm at any time.

Contrast experiments of parallel performance for the three algorithms are performed on the Tesla C1060 GPU in the environment of CUDA 4.4 runtime API.

No difference can be observed between the images reconstructed by CPU and GPU (Fig. 7 and Fig. 9). As shown in Table 2, the time costs for parallel performance by the BPF, T-BPF and S-BPF algorithm are 3.245, 0.705, and 0.677 s respectively. The parallel performance of the BPF algorithm is poor because of relations between a spatial direction circle and the scanning angle circle in the backprojection. The parallel performance of the T-BPF algorithm is better than that of

Table 2.   Contrast experiments of CPU and GPU.

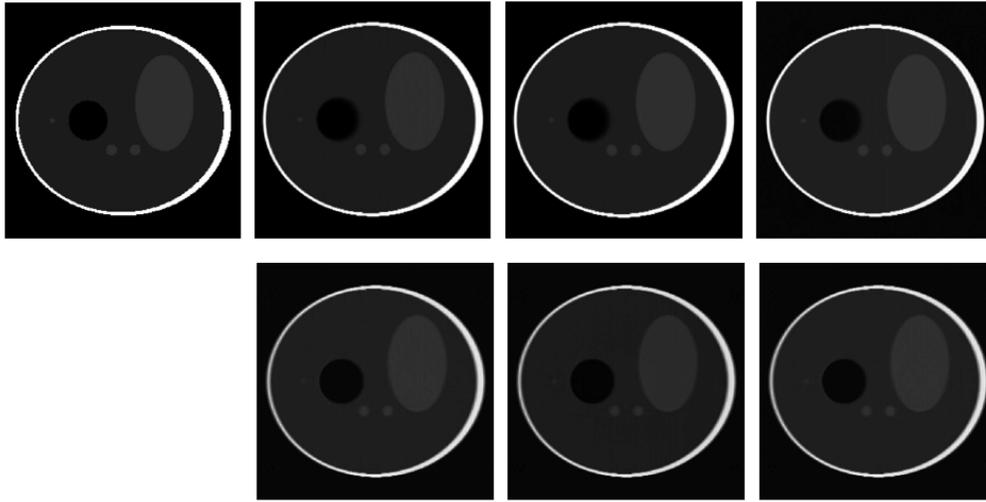| algorithm	size	reconstruction time cost (CPU)/s	reconstruction time cost (GPU)/s	speedup factor
BPF	256×256×256	963.7	3.245	297
T-BPF	256×256×256	730.8	0.705	1036
S-BPF	256×256×256	705.6	0.677	1043

Fig. 7.   Reconstructed slice of the three algorithms by parallel computing (GPU). The Shepp-Logan model and the reconstructed slices of the BPF, T-BPF, and S-BPF algorithms are represented in the first, second, third, and fourth columns, respectively. On the right, the reconstructed slices using CPU and GPU are represented in the first and second rows, respectively.
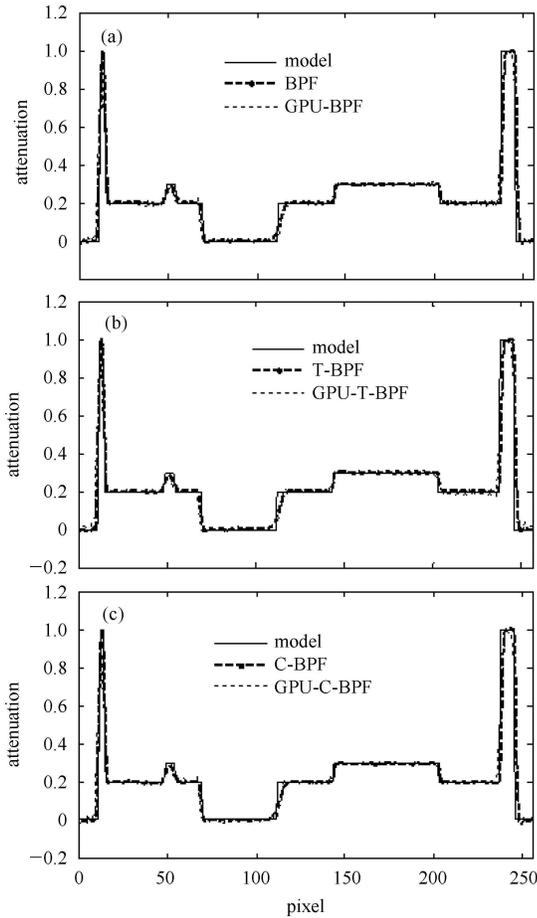

Fig. 8.   Profiles of the reconstructed slices along the horizontal middle lines. (a), (b), and (c) represent the profiles of the slices reconstructed using the BPF, T-BPF, and C-BP algorithms, respectively.

the BPF algorithm because the four circles are independent in backprojection implementation. The parallel performance of S-BPF is slightly better than that of the T-BPF algorithm because no data rebinning is included and the relations between the scanning angle circle and a spatial direction circle is eliminated by neglecting some PI-lines in backprojection implementation.

3.2   Real experiment

Reconstructions of real data were performed for three algorithms in the case of serial implementation and the real data were acquired with the cone-beam CT system which mainly consisted of the flat-panel detector (Varian 4030E, USA) with a pixel size of 0.127 mm, the object holder (which can be circumvolved) and the X-ray source (Hawkeye 130, Thales, France).

Experiments were carried out to reconstruct an ROI of the head phantom and check the capability of the S-BPF method (Fig. 9). The distance between the source and the rotation axis of the scanner was 678 mm and the distance between the source and the detector was 1610 mm. The projection data were acquired from the flat-panel detector with 3200 pixels ×2304 pixels and 0.128 mm per pixel. 360 projection views were uniformly distributed over the 2π circular trajectory. The size of the reconstructed ROI was 700 pixels ×1200 pixels ×500 pixels with 0.107 mm per pixel. The projection view used by the three algorithms varied from ψ to $180-\psi$. The distance between the rotation axis of the scanner and the ROI center was 450 pixels (48.15 mm). Thus, the following equation applies:

$$\psi=\arcsin\left(\frac{(450-700/2)\times0.107}{678}\right)\times\frac{180}{\pi}=9.080°. \quad (20)$$
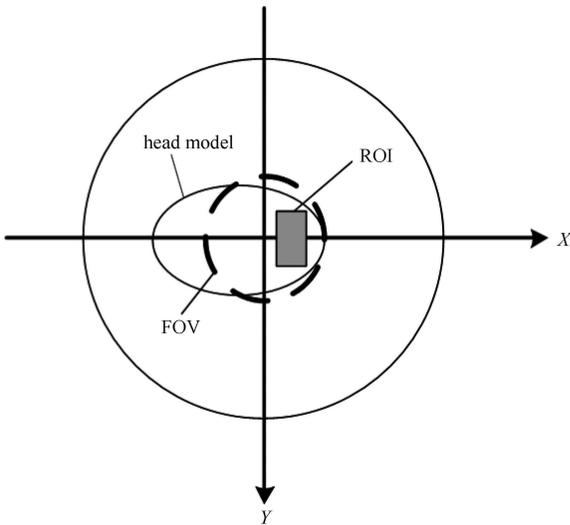
Fig. 9.   The bottom view for ROI.

The data acquired on the detector at any view angle were not used completely. We only used the middle 1600 pixels in the b axe direction. The data type of the projection data was changed to unsigned short from float.

Figure 10 shows the correctness of S-BPF again by re-constructing the ROI from the truncated data. As shown in Fig. 10 and Table 3, the three algorithms with trun-cated data result in reconstructed images with no signifi-cant differences, and the advantages on memory cost and time cost of S-BPF are evident. In the case of compara-ble reconstruction quality, the reconstruction efficiency of S-BPF is the highest, and its memory cost is the least among the three algorithms. In practice, the memory cost of an algorithm counts. If the memory cost is too high to be satisfied, some strategies must be introduced, such as lowering the spatial resolution of the reconstruc-tion images or decomposing the backprojection by the limiting of integration of spatial direction. However, the two resolution results have lower reconstruction quality or greater time cost.

Large scale reconstruction is needed to reconstruct the high spatial resolution or large size of ROI. To achieve large scale reconstruction, BPF and T-BPF require con-siderable amounts of memory, which cannot be satis-fied because of limited hardware. For the BPF algo-rithm, reading projections from the hard disk repeatedly or block-dividing backprojection must be carried out. Block-dividing backprojection also involves the reading projection from the hard disk repeatedly. Therefore, the
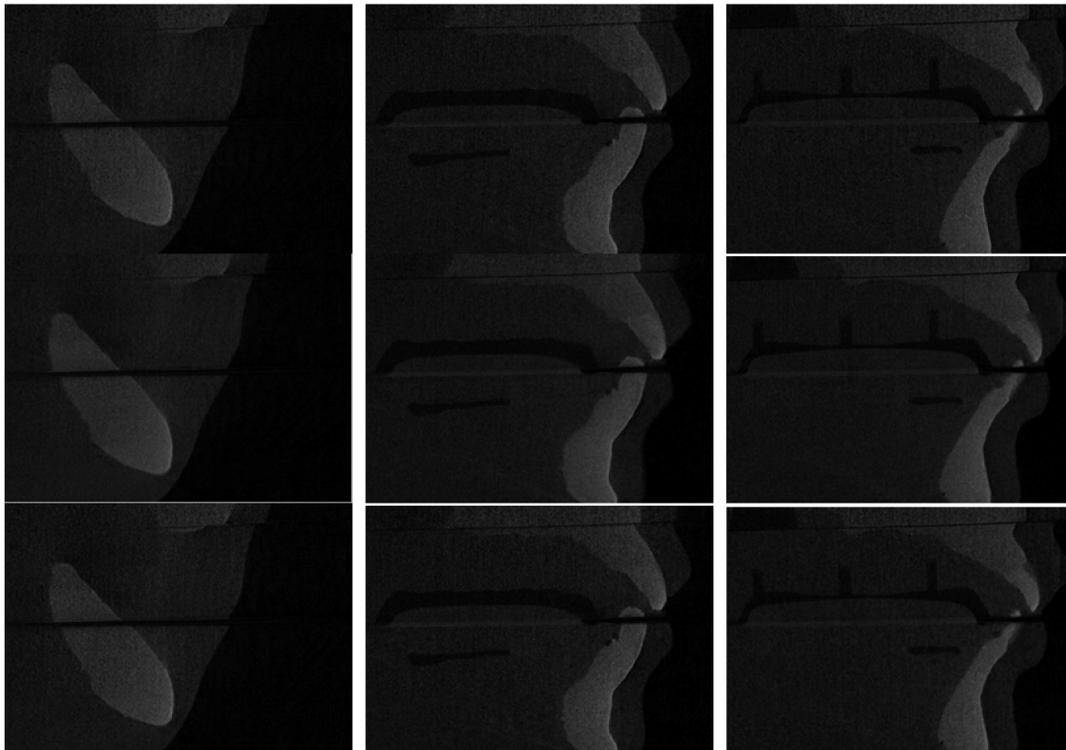

Fig. 10.   2D slices in 3D images reconstructed using the BPF, T-BPF, and S-BPF algorithms. The slices of the plane y=26.86 mm, plane y=54.85 mm and plane y=61.83 mm are represented in the first, second, and third rows respectively. The reconstruction results of the BPF, T-BPF, and S-BPF algorithms are represented in the first, second and third columns respectively.

Table 3.   Reconstruction results of BPF, T-BPF, and S-BPF algorithms.

algorithm	size	time/min	memory/GB
BPF	700×500×1200	668.54	8.54
T-BPF	700×500×1200	249.73	15.77
S-BPF	700×500×1200	213.38	1.67

BPF algorithm must utilize considerably higher time cost to avoid the high memory cost in the case of large scale reconstruction. For the T-BPF algorithm, reading and writing projection on the hard disk must be carried out repeatedly. The T-BPF algorithm must use block-dividing data-rebinning (the block is divided by the index of pixel on the detector). Block-dividing data-rebinning involves many trigonometric functions or the use of some memory to store the results of the trigonometric functions that are computed beforehand. The rebinned projection must be stored on a hard disk after data rebinning on a block and be read for backprojection. The required processes will cost longer time to obtain the DBP. In a word, the T-BPF algorithm must also use a considerably higher time cost to avoid large memory cost in case of large scale reconstruction. However, no additional process for decomposing the backprojection is needed for the S-BPF algorithm because of a low memory cost for S-BPF. Thus, S-BPF can reconstruct considerably faster than the BPF and T-BPF algorithms in the case of large scale reconstruction.

4   Conclusion

In the work, S-BPF was developed for image reconstruction from the truncated data acquired in a short scan. Two steps are implemented to carry out S-BPF: firstly, the derivative of projections is backprojected to the points whose x coordinate is less than that of the source focal spot to obtain the DBP; then, the finite Hilbert inverse is applied on each PI-line segment to reconstruct images from the DBP. The S-BPF algorithm avoids the influence of the variable limit integration by selective backprojection without projection data rebinning and the need to load all projection data in advance. Compared with the BPF and T-BPF algorithms, S-BPF has a higher reconstruction efficiency, better parallel performance and lower memory cost. The approach can also be applied to practical CT systems and achieve a fast local reconstruction from the truncated data acquired in a short scan. But only a few projections are allowed to be truncated and the accurate reconstruction cannot be reached in the case of that one or two end points of the PI-line are in the object, which is to be reconstructed. This case can be deal with by DBP-POCS [22, 23] and SVD-THT [24–26] and the method of obtaining DBP in this paper can also be used for the two kinds of algorithm to improve reconstruction efficiency.

References

1 Adel Faridani et al. Contemporary Mathematics, 2000, **278**: 29
2 Frank Dennerlein. Cone-beam ROI Reconstruction using the Laplace Operator. 11th international meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine. Germany: Potsdam, 2011. 80
3 WANG Xian-Chao et al. Chin. Phys. B, 2012, **21**: 118702
4 Danielsson P E, Edholm P, Seger M. Towards Exact 3D-Reconstruction for Helical Cone-Beam Scanning of Long Objects. A new Detector Arrangement and a New Completeness Condition Proc. 1997 Int. Meet. on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine. America: Pittsburgh, 1997.141
5 ZOU Yu, PAN Xiao-Chuan. Phys. Med. Biol., 2004, **49**: 941
6 ZOU Yu, PAN Xiao-Chuan. Phys. Med. Biol., 2004, **49**: 383
7 ZOU Yu, PAN Xiao-Chuan. Phys. Med. Biol., 2004, **49**: 2717
8 ZHUANG Ting-Liang et al. Phys. Med. Biol., 2004, **49**: 5489
9 YU Heng-Yong et al. Med. Phys., 2005, **32**: 3305
10 Clackdoyle R et al. IEEE Trans. Nucl. Sci., 2004, **51**: 2570
11 Noo F, Clackdoyle R, Jed D P. Phys. Med. Biol., 2004, **49**: 3903
12 FU Jian et al.IEEE Trans. Nucl. Sci., 2010, **57**: 1071
13 YU Li-Feng et al. IEEE Trans. Med. Imag., 2006 **25**: 869
14 WANG Xian-Chao et al. IEEE Trans. Nucl. Sci., 2013 **60**: 174
15 Feldkamp A L, Davis L C, Kress J W. J Opt. Soc. Amer. A, 1984, **1**: 612
16 Susumu Okada, David Elliott. Math. Nachr., 1991, **153**: 43
17 Emil Y Sidky, ZOU Yu, PAN Xiao-Chuan. Phys. Med. Biol., 2005 **50**: 1643
18 ZOU Yu, PAN Xiao-Chuan, Emil Y Sidky. J. Opt. Soc. Amer., 2005, **22**: 2372
19 Shepp L A, Logan B F. IEEE Trans. Nucl. Sci., 1974, **NS-21**: 21
20 HAN Xiao et al. IEEE Trans. Med. Imag., 2011, **30**: 606
21 WANG Xian-Chao et al. J. X-ray Sci.Tech., 2012, **20**: 69
22 YE Yang-Bo, WEI Yu-Chuan, WANG Ge. International Journal of Biomedical Imaging, 2007, **1**: 2
23 LI Liang et al. J. X-ray Sci. Tech., 2009, **17**: 135
24 YU Heng-Yong, YE Yang-Bo, WANG Ge. J. X-ray Sci. Tech., 2008, **16**: 243
25 Katsevich. A Inverse Problems, 2010, **26**(11): 115011
26 JIN Xin et al. IEEE Trans. Med. Imag., 2012, **31**(11): 2108