A Lattice Implementation of High-precision IIR Filter for X-ray CT Reconstruction

Kazuhisa Hayashi, Hiroshi Tsutsu, Yoshitaka Morikawa, Nobuo Funabiki

Abstract— **The authors previously proposed *forward* and *backward one round (FBOR) method* to accelerate the filtering process by realizing the *discrete Hilbert transform (DHT)* function using an IIR filter in the convolution backprojection (CB) method that is a typical reconstruction algorithm for the X-ray CT. The FBOR method cascades causal and anticausal allpass filters to realize the linear phase DHT, and implements them by parallel ladders of allpass filters. Unfortunately, this method has to adopt the DHT that has the brickwall transition at the zero frequency and thus, requires high sensitivity coefficients in order to reduce distortions of a reconstructed image. As a result, it needs to rely on double precision floating points to avoid cancellations of significant digits. In this paper, we propose a *successive-scan one-round (SSOR) method* of implementing the DHT by cascading *primary lattice allpass filters* that allows the use of single precision floating points. We confirm the effectiveness of our proposal through simulations.**

Index Terms—**Allpass filter, Computerized tomography (CT), Hilbert transform, IIR filter, Lattice filter**

I. Introduction

THE X-ray computerized tomography (CT) has been a powerful diagnosis tool in medical fields, and the three dimensional CT (3D-CT) has been developed in recent years. In order to move ahead this 3D-CT on, a lot of researchers around the world are intensively studying this technology [1][2].

A conventional 3D-CT device such as the helical CT generates a 3D image by stacking a number of 2D reconstruction images. Thus, the speed up of the 2D image reconstruction algorithm is important to realize a realtime 3D-CT technology, which is called the 4D-CT.

As a typical 2D image reconstruction algorithm for the X-ray CT, the *convolution back-projection* (CB) method has been widely known [3] [4]. The CB method consists of the two processes. The first one is the filtering process of applying a high frequency emphasis ramp filter to each projection data by convolutions. The other one is the backprojection process of calculating the reconstruction pixel values from all the filtered projection data by interpolations. Because the backprojection process usually consumes the longer computation cost than the filtering process, a lot of studies have been spent on the speed up of the former process. However, when we use the *tree structured filter bank method* among the speed up technologies, we cannot ignore the computation cost of the filtering process [5] [6].

To speed up the filtering process along with the tree structured filter bank, the authors have recently proposed an algorithm using an IIR filter for the high frequency emphasis of a signal [7]. This algorithm composes a high frequency emphasis filter by using a difference filter and a *discrete Hilbert transform* (*DHT*) with the linear phase, and implements the filter as a parallel ladder of allpass filters.

In the application to non-realtime image processing, the DHT can turn out to the zero phase by adopting both the causal allpass filter and the anticausal allpass filter. Besides, the initial values of anticausal allpass filters are calculated by using the finite signal range to connect the anticausal allpass filter to the output of the causal allpass filter that has the infinite range. This method is called the *forward-backward one-round* (*FBOR*) *method*. Unfortunately, this method has to adopt the DHT that has the brickwall transition at the zero frequency and thus, requires high sensitivity coefficients in order to reduce distortions of a reconstructed image. As a result, it needs to rely on double precision floating points to avoid cancellations of significant digits, which become a serious disadvantage in terms of the computation speed.

In this paper, we propose a *successive-scan one-round* (*SSOR*) *method* of implementing the DHT by cascading *primary lattice allpass filters* that allows the use of single precision floating points. In this method, we have to connect the causal and the anticausal primary filters at the edge of the input signal after the former filters execution. In our proposal a ramp filter is composed by a cascade connection of a differentiator and a linear phase DHT that is implemented by a parallel-ladder of allpass filters. We confirm the effectiveness of our proposal through simulations.

The rest of this paper is organized as follows Section II introduces the ramp filter function for CT reconstruction, as well as DHT. Section III reviews the implementation of allpass filters in the FBOR method and Section IV proposes the SSOR method. Section V discusses the evaluation results through simulations, and lastly Section VI concludes this paper with future works.

II. RAMP FILTER USING ALLPASS FILTER

A. Convolution Back-projection Method

When $\{p_{i\pi/M}(m\tau)\}_{i\in Z_M, m\in Z}$ represents the discrete time projected data with sampling pitch τ along the direction with angle $i\pi/M$ from the horizontal axis, the reconstruction in the convolution backprojection (CB) method is described as follows:

The authors are with the Graduate School of Natural Science and Technology, Okayama University, 3-1-1 Tsushimanaka, Okayama-shi, 700-8530, Japan,(e-mail:hayashi_k@hitachizosen.co.jp).
(e-mail:tsutsu@trans.cne.okayama-u.ac.jp)
(e-mail:morikawa@trans.cne.okayama-u.ac.jp)
(e-mail:funabiki@cne.okayama-u.ac.jp)

$$f(\mathbf{r}) = \frac{1}{M}\sum_{i=0}^{M-1} \hat{q}_{i\pi/M}(r_1 \cos i\,\pi/M + r_2 \sin i\,\pi/M), \tag{1a}$$

$$q_{i\pi/M}(n\tau) = \tau \sum_{m\in Z} r((n-m)\tau)p_{i\pi/M}(m\tau) \tag{1b}$$

where $\mathbf{r}=(r_1,r_2)^{\mathrm{T}}$ represent the reconstruction lattice points with the unitary sampling pitch and $\hat{q}_{i\pi/M}(\cdot)$ represents the interpolated value at the point · from the sequence $q_{i\pi/M}(n\tau)$ in (1b). (1a) and (1b) represent the backprojection and the convolution processes in the CB method, respectively. $\{r(n\tau)\}_{n\in Z}$ is the ramp filter sequence whose frequency response is given by:

$$R(e^{j\tau\omega}) = |\,\omega\,|, \quad |\,\omega\,| \le \pi/\tau. \tag{2}$$

In the application, the ramp filter $R(e^{j\tau\omega})$ is often modified [1] to the high-frequency suppressed version as

$$R(e^{j\tau\omega}) = (2/\tau)|\sin(\tau\omega/2)|. \tag{2$'$}$$

B. Ramp Filter using DHT

The ramp filter (2′) can be written by the multiplication of the sine and the sign functions as follows:

$$R(e^{j\tau\omega}) = D(e^{j\tau\omega})H(e^{j\tau\omega}) \tag{3}$$

where $\quad D(e^{j\tau\omega}) = (2/\tau)\sin(\tau\omega/2),$ (4a)

and

$$H(e^{j\tau\omega}) = \begin{cases} 1 & \text{when } 0 < \tau\omega < \pi \\ 0 & \text{when } \tau\omega = 0 \\ -1 & \text{when } -\pi < \tau\omega < 0 \end{cases}. \tag{4b}$$

$H(e^{j\tau\omega})$ in (4b) is called the *Hilbert transform*. $D(e^{j\tau\omega})$ is easily implemented by a difference filter as mentioned later, but $H(e^{j\tau\omega})$ is not as it is approximately realized by modifying the causal halfband lowpass filter $F_{\mathrm{HB}}(z)$ [8] of the order K with allpass filters as follows.

$$F_{\mathrm{HB}}(z) = 0.5\{A^{(0)}(z^2) + z^{-1}A^{(1)}(z^2)\} \tag{5a}$$

where $A^{(0)}(z)$ and $A^{(1)}(z)$ are the causal allpass filters given by

$$A^{(0)}(z) = \prod_{k=0}^{K^{(0)}-1} \frac{p_k^{(0)} + z^{-1}}{1 + p_k^{(0)} z^{-1}}, \quad A^{(1)}(z) = \prod_{k=0}^{K^{(1)}-1} \frac{p_k^{(1)} + z^{-1}}{1 + p_k^{(1)} z^{-1}} \tag{5b}$$

where $\quad K^{(0)}=K-\lfloor 0.5K \rfloor, \quad K^{(1)}=\lfloor 0.5K \rfloor$ (5c)

and $0<p_k^{(0)}<1$ $(0\le k<K^{(0)})$, $0<p_k^{(1)}<1$ $(0\le k<K^{(1)})$. (5d)

Since $F_{\mathrm{HB}}(z)$ in (5a) is not the zerophase as it is, we first modify $F_{\mathrm{HB}}(z)$ to get its zerophase version by multiplying $F_{\mathrm{HB}}(z)$ with its conjugate $F_{\mathrm{HB}}(z^{-1})$. Then, we shift it in the positive direction of $\tau\omega$ by $\pi/2$, remove 0.5 from it, and double it. Lastly, we have an approximation of the Hilbert transform $H(e^{j\tau\omega})$:

$$\begin{aligned} H(z) &= 2\{F_{\mathrm{HB}}(-jz^{0.5})F_{\mathrm{HB}}(jz^{-0.5}) - 0.5\} \\ &= 0.5j\,z^{0.5}\{z^{-1}A(-z) - A(-z^{-1})\} \end{aligned} \tag{6}$$

where $A(z)=A^{(1)}(z)A^{(0)}(z^{-1})$.[1] As $D(z)=(1/j\tau)z^{-0.5}(z-1)$, the transfer function with the frequency response in (2′) can be approximated by the following filter:

$$R(z) = 0.5\,(z-1)/\tau \cdot \{z^{-1}A(-z) - A(-z^{-1})\}. \tag{7}$$

For the software implementation, we moreover a little modify (7). Instead of the definition of (5b), we define $P(z)$, $P^{(0)}(z)$ and $P^{(1)}(z)$ as follows:

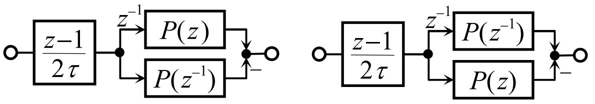
$$P(z) = P^{(1)}(z)P^{(0)}(z^{-1}) \tag{8a}$$

$$P^{(0)}(z) = \prod_{k=0}^{K^{(0)}-1} \frac{1 - p_k^{(0)} z}{1 - p_k^{(0)} z^{-1}}, \quad P^{(1)}(z) = \prod_{k=0}^{K^{(1)}-1} \frac{1 - p_k^{(1)} z}{1 - p_k^{(1)} z^{-1}}. \tag{8b}$$

A simple manipulation derives the next expression for $R(z)$:

$$R(z) = \begin{cases} \dfrac{z-1}{2\tau}\{z^{-1}P(z) - P(z^{-1})\} & \text{when } K \text{ is even} \\ \dfrac{z-1}{2\tau}\{z^{-1}P(z^{-1}) - P(z)\} & \text{when } K \text{ is odd} \end{cases}. \tag{9}$$

Fig.1 shows the block diagram of (9). The both cases consist of allpass filters $P(z)$ and $P(z^{-1})$. In what follows, we call $P^{(1)}(z)$ and $P^{(0)}(z^{-1})$, which constitute $P(z)$, *forward* and *backward allpass filters*, respectively.



(a) When K is even. (b) When K is odd.

Fig.1 Block diagram of ramp filter (9).

III. Implementation of Allpass Filter

In the last section, we derived the transfer function (9) for the ramp filter. The allpass filter $P(z)$ is composed of the forward filter and the backward allpass filter $P^{(1)}(z)$, $P^{(0)}(z^{-1})$, which are both IIR filters. Thus, for the implementation of (9), we have to connect those IIR filters.

Fortunately, the measured projection data sequences are finite, so in the outsides of the data, the signal values are considered to be zero. In this case, we can estimate the outside response of the forward filter in the Z domain to calculate the initial values for the backward filter. Here we review the implementation of the connection between the forward and backward allpass filters, called the *FBOR method* [7].

A. Outline of FBOR Method

In this method, we use the allpass filters $P^{(1)}(z)$ and $P^{(0)}(z^{-1})$ in their expansion forms, which are

$$P^{(s)}(z) = \frac{1 + \sum_{k=1}^{K^{(s)}} c_k^{(s)} z^k}{1 + \sum_{k=1}^{K^{(s)}} c_k^{(s)} z^{-k}} \quad (s = 0,\ 1). \tag{10}$$

[1] Whenever the denominator and the numerator polynomials of a transfer function $A(z)$ are each other reciprocal, $A(z)A(z^{-1})=1$ is always true so that its amplitude characteristic is 1 for all frequencies. For this reason, a filter with this property is called allpass filter.

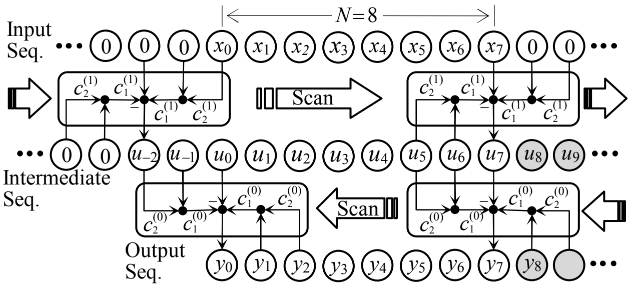

N=8
Input Seq.
0 0 0 x0 x1 x2 x3 x4 x5 x6 x7 0 0
Scan
Intermediate Seq.
0 0 u-2 u-1 u0 u1 u2 u3 u4 u5 u6 u7 u8 u9
Scan
Output Seq.
y0 y1 y2 y3 y4 y5 y6 y7 y8

Fig.2 Schematic action of the allpass filters on an input sequence of length $N=8$ (filter order $K=4$).

The forward iteration starts from $n=-K^{(1)}=-2$ and since $n=N$ runs with free input. Thus, when $n\geq N$ the intermediate output u_n depends only on $u_{N-1},\cdots,u_{N-K^{(1)}}$. The backward iteration starts from $n=\infty$ and stops when $n=0$. But, when $y_N,\cdots,y_{N+K^{(1)}-1}$ are calculated beforehand, the iteration starts only at $n=N-1$.

Note that because (10) is an expansion of (8b), the poles of (10) are $\{p_k^{(s)}\}_{0\leq k\leq K^{(s)}}$ $(s=0,1)$.

Now, we transcribe $P^{(1)}(z)$ and $P^{(0)}(z^{-1})$ in the time domain:

$$u_n = x_n + \sum_{k=1}^{K^{(1)}} c_k^{(1)}(x_{n+k} - u_{n-k}) \quad (n++) \tag{11a}$$

and
$$y_n = u_n + \sum_{k=1}^{K^{(0)}} c_k^{(0)}(u_{n-k} - y_{n+k}) \quad (n--) \tag{11b}$$

where for the forward filter $P^{(1)}(z)$ the input and output sequences are respectively $\{x_n\}_{n\in Z_N}$ and $\{u_n\}_{n\in Z}$, and for the backward filter $P^{(0)}(z^{-1})$ they are respectively $\{u_n\}_{n\in Z}$ and $\{y_n\}_{n\in Z}$. Note that for the table computation, we must repeat the iterations with an increment of n for the forward filter and with a decrement for the backward filter, as shown in Fig.2. It shows the schematic action of $P^{(1)}(z)$ and $P^{(0)}(z^{-1})$ on a finite input sequence of length $N=8$ for the filter order $K=4$. From the figure, we see it is no need to prepare separate registers for $\{x_n\}$, $\{u_n\}$ and $\{y_n\}$. This is the reason why we rewrote as the mixed form (8) in Section II.

When we apply the forward filter with the zero padding outside the finite duration, the first nonzero output appears at $n=-K^{(1)}$ and thus, we can start the iteration at $n=-K^{(1)}$. On the other hand, after $n=N$, the difference equation turns to the input free one $u_n=-\sum_k c_k^{(1)} u_{n-k}$ $(n\geq N)$, whose solution is only the so-called "initial condition dependent (ICD)" one.

B. Initial Value Calculation for Backward Filter

The ICD solution, including the edge values, is expressed [7] in the Z domain as

$$U(z) = \sum_{k=1}^{K^{(1)}} u_{N-k} z^{-N+k} + \sum_{k=1}^{K^{(1)}} c_k^{(1)} u_{N-k} \frac{z^{-N}}{D^{(1)}(z)} \tag{12}$$

where the inner edge values $\{u_{N-k}\}$ have been calculated by the scanning when of $n<N$ and $D^{(1)}(z)$ is the denominator polynomial of $P^{(1)}(z)$.

The backward filter $P^{(0)}(z^{-1})$ acts on $\{u_n\}$ with a decrement of n, as shown in Fig.2. However, we can skip the scanning the region $n\geq N$, because $\{u_n\}_{n\geq N-K^{(1)}}$ are written by (12), and so $\{y_n\}_{n\geq N}$ can be calculated through the inverse Z transform:

$$y_n = \frac{1}{2\pi j}\oint_{|z|=1} P^{(0)}(z^{-1})U(z)z^{-n+1}dz \quad (n\geq N). \tag{13}$$

However, it is suffice to calculate y_n's in the range of $N\leq n<N+K^{(0)}-1$, because we scan with a decrement of n to get the output $\{y_n\}_{0\leq n<N}$.

When we express the connection from $\{u_{N-l-1}\}_{0\leq l<K^{(1)}}$ to $\{y_{N+k}\}_{0\leq k<K^{(0)}}$ as $y_{N+k}=\sum_{l=1}^{M}\gamma_{k,l}\cdot u_{N-l-1}$, the transfer gains $\gamma_{k,l}$ are given [7] using the residue calculations of (13) as follows:

$$\gamma_{k,l} = \sum_{m=0}^{K^{(1)}-1}\left[(p_m^{(1)})^k \sum_{s=0}^{l} c_s^{(1)} (p_m^{(1)})^{l-s} z^{K^{(1)}-K^{(0)}} \right.$$
$$\left. \prod_{s=0(s\neq m)}^{K^{(0)}-1} \frac{p_m^{(1)}-p_s^{(0)}}{1-p_s^{(0)}p_m^{(1)}} \prod_{s=0(s\neq m)}^{K^{(1)}-1} \frac{1}{p_m^{(1)}-p_s^{(1)}} \right]$$
$$(0\leq k<K^{(0)}, 0\leq l<K^{(1)}) \tag{14}$$

Because almost poles $\{p_m^{(1)}\}$ are close to 1, the calculation of $1/(p_m^{(1)}-p_s^{(1)})$ in (14) loses the precision due to the cancellation of significant digits. Additionally, in the computation of expansion form coefficients $\{c_m^{(0)}, c_m^{(1)}\}$, we also lose the precision, and the poles inside the unit circle may roll over it, resulting in the instable iterations.

IV. PROPOSAL OF SSOR METHOD

In this section, we propose the *successive-scan one-round* (*SSOR*) method for the direct implementation of (8).

A. Outline of SSOR method

Let u_n^k denote the output sequence of the k-th forward first order allpass filter of $P^{(1)}(z)$ in (8b). Then the input is u_n^{k-1} and its action to the input can be written by the following difference equation:

$$u_n^k = u_n^{k-1} + p_k^{(1)}(u_{n-1}^k - u_{n+1}^{k-1}) \quad (n++, 0\leq k<K^{(1)}) \tag{15a}$$

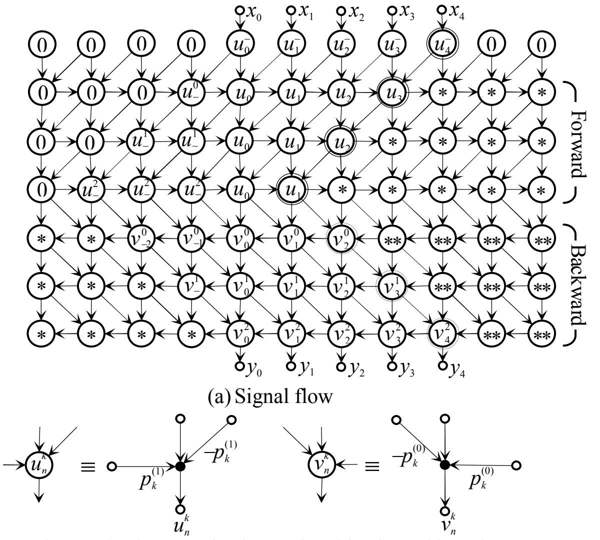
where $u_n^{-1}=x_n$ $(0\leq n<N)$ is the given input sequence to $P(z)$. Let v_n^k denote the output sequence of the k-th backward first order allpass filter of $P^{(0)}(z^{-1})$ in (8b). Then, the backward filter works in the time domain as

$$v_n^k = v_n^{k-1} + p_k^{(0)}(v_{n+1}^k - v_{n-1}^{k-1}) \quad (n--, 0\leq k<K^{(0)}) \tag{15b}$$

where $v_n^{-1}=u_n^{K^{(1)}-1}$ is the output sequence of $P^{(1)}(z)$ and $v_n^{K^{(0)}-1}$ $(0\leq n<N)$ results in the objective output y_n.

Fig.3 illustrates the above actions when the input signal length is $N=5$ and the allpass filter order is $K=6$. The upper half corresponds to the forward filter $P^{(1)}(z)$ and the lower half does to the backward filter $P^{(0)}(z^{-1})$. The computational elements are shown in Fig.3 (b) and their action will be apparent by referring (15a) and (15b). The input data are transferred to the first line with the zero value padding and renamed $\{u_n^{-1}\}$. The first filter $(1-p_0^{(1)}z)/(1-p_0^{(1)}z^{-1})$ acts on $\{u_n^{-1}\}$ and outputs $\{u_n^0\}$ by sliding its computing element left to right. Replacing $p_0^{(1)}$ by $p_1^{(1)}$, the same scanning is performed in the next line and the similar scans are repeated line by line. Lastly, we get the output of $P^{(1)}(z)$ on the $K^{(1)}$-th line. The similar scans are performed for the backward filter $P^{(0)}(z^{-1})$ on $K^{(0)}$ lines and lastly, we get the objective output on the $K^{(0)}$-th line.

In the forward scan region, the rest states marked by "0"are kept in the left hand side, so scans may be just started after those zeros. On the right hand side, the values marked by * can be computed when the boundary values marked by ○ are given and in the result we can obtain the free running output sequence on the $K^{(1)}$-th line. The output sequence may also be expressed in the Z domain.



(a) Signal flow

(b) Basic elements for forward and backward iterations

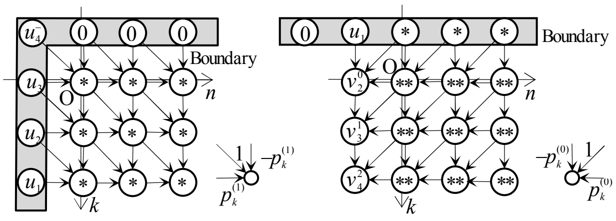
Fig.3 Schematic action of the primary allpass filters on an input sequence of length $N=5$ (filter order $K=6$).

The forward scans repeat $K^{(1)}$ times for $P^{(1)}(z)$ and the backward ones do $K^{(0)}$ times for $P^{(0)}(z^{-1})$. Zeros are padded outside the finite duration of an input. In the forward scan region, rest states marked by "0"are kept in the left hand side, so scans may be just started after zeros. On the right hand side, the values marked by * and ** are calculated when the boundary values marked by ○ are given. So the transfer matrix, from the points marked by ○ to those marked by ○ , may be pre-calculated.

The backward scans follow the forward scans. In the backward region in Fig.3, the values marked by ** are dominated only by the sequence marked by * on the last line of the forward scan. Therefore, we can get the expressions of those sequences marked by ** in the Z domain and in the result, we can obtain the transfer matrix Γ from the values marked by ○ to those by ○ through the inverse Z transform.

B. Initial Value Calculation for Backward Filters

In order to express the transfer matrix Γ in terms of the allpass filter coefficients, let us reformulate the problem in the last paragraph to the 2-D boundary problem. Fig.4(a) rearranges the nodes in the free run region of forward scans in Fig.3 such that $u^k_{K^{(1)}-1-k}$ $(-1\leq k<K^{(1)})$ line up on the vertical line of the axis k, and at the same time the time index n is shifted line by line. The connection of the nodes on the k-th line is shown at the bottom right. From Fig.4(a) it is known that the boundary values propagate the nodes in the line-variant manner. Fig.4(b) also shows the boundary problem for the backward free run regions marked by ** in Fig.3.



(a) For forward scans

(b) For backward scans

Fig.4 Illustration as two-dimensional boundary problems in the regions marked by * and ** in Fig3.

Now, let us obtain the transmittance $\gamma_{k,l}$ from the l-th line boundary point in Fig.4(a) to the k-th line point of $n=-1$ in Fig.4(b). Note that the value at the former point, computed in the forward scan, is $u^l_{N\text{-}2\text{-}l}$ $(-1\leq l<K^{(1)})$, and that at the latter point, computed in the backward scan, is $v^k_{N-K^{(0)}+k}$ $(-1\leq k<K^{(0)})$. Suppose the unit impulse exists only at the l-th line boundary. Then, there exist two kind propagations. One is the impulse that propagates just to the rightside neighbor node with the weight $p_l^{(1)}$ followed the right-direction propagation with the transfer function $1/(1-p_l^{(1)}z^{-1})$ and vertical propagation to the last line with the transfer function:

$$\prod_{s=l+1}^{K^{(1)}-1}(z^{-1}-p_s^{(1)})\big/(1-p_s^{(1)}z^{-1}). \qquad (16a)$$

The other is the impulse that propagates just to the right-downside neighbor node with the weight 1, followed by the right-direction propagation with the transfer function $1/(1-p_{l+1}^{(1)}z^{-1})$ and the vertical propagation to the last line with the transfer function:

$$\prod_{s=l+2}^{K^{(1)}-1}(z^{-1}-p_s^{(1)})\big/(1-p_s^{(1)}z^{-1}). \qquad (16b)$$

Therefore, when $0<l<K^{(1)}-1$, the contribution $R_l(z)$ of the supposed impulse to the last line is given in the Z domain:

$$\begin{aligned} R_l(z) &= \frac{p_l^{(1)}}{1-p_l^{(1)}z^{-1}}\prod_{s=l+1}^{K^{(1)}-1}\frac{z^{-1}-p_s^{(1)}}{1-p_s^{(1)}z^{-1}}+\frac{1}{1-p_{l+1}^{(1)}z^{-1}}\prod_{s=l+2}^{K^{(1)}-1}\frac{z^{-1}-p_s^{(1)}}{1-p_s^{(1)}z^{-1}} \\ &= \frac{1-p_l^{(1)}p_{l+1}^{(1)}}{(1-p_l^{(1)}z^{-1})(1-p_{l+1}^{(1)}z^{-1})}\prod_{s=l+2}^{K^{(1)}-1}\frac{z^{-1}-p_s^{(1)}}{1-p_s^{(1)}z^{-1}}. \qquad (17a) \end{aligned}$$

However, when $l=-1$, the first horizontal propagation does not exist, so

$$R_{-1}(z)=\frac{1}{1-p_0^{(1)}z^{-1}}\prod_{s=1}^{K^{(1)}-1}\frac{z^{-1}-p_s^{(1)}}{1-p_s^{(1)}z^{-1}}, \qquad (17b)$$

and when $l=K^{(1)}-1$, only the first horizontal propagation does exist, so

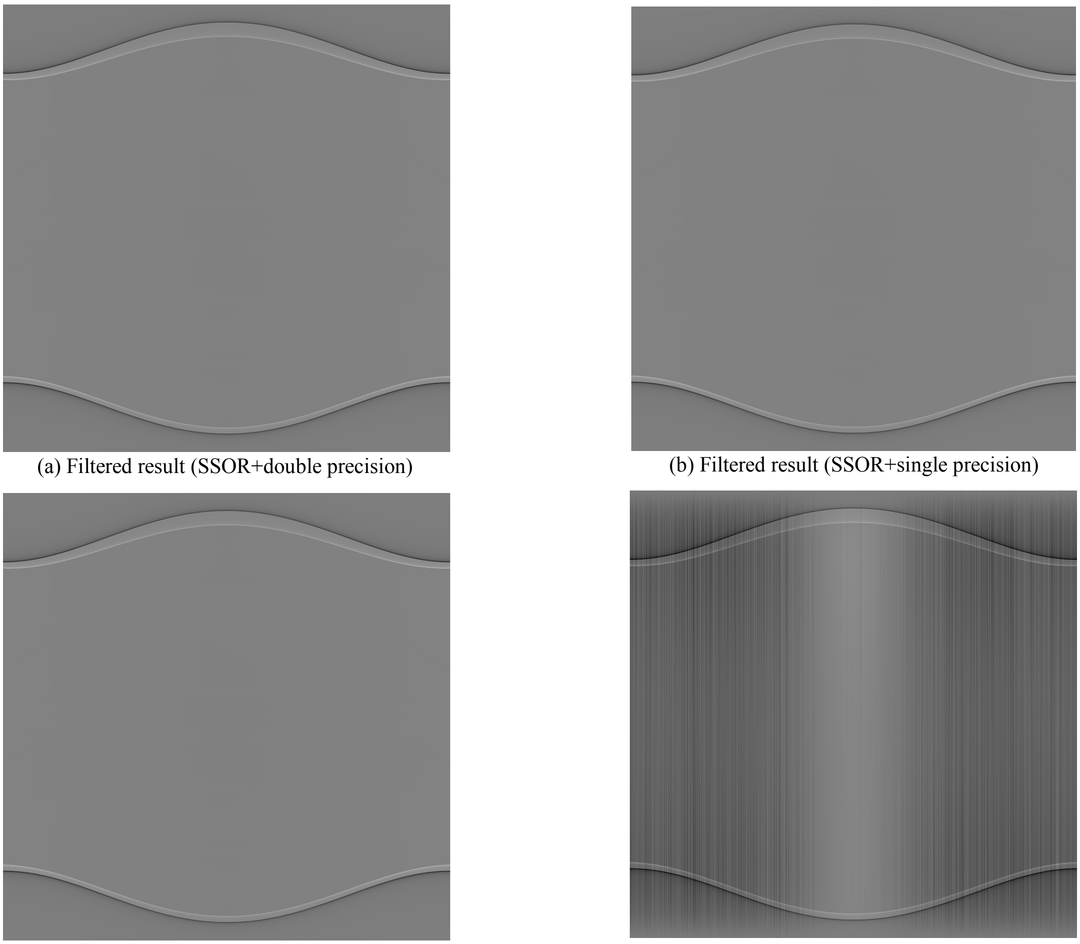
$$R_{K^{(1)}-1}(z)=\frac{1}{1-p_{K^{(1)}-1}^{(1)}z^{-1}}. \qquad (17c)$$

Supposing the values of absent poles, $p_{-1}^{(1)}$ and $p_{K^{(1)}}^{(1)}$, are zero, (17b) and (17c) are imported in (17a) and we can reference (17a) for $0\leq l<K^{(1)}$.

Fig.4(b) implies all the responses on the k-th line to the boundary sequence, whose Z transform is $R_l(z)+\delta_{l,K^{(1)}-1}z$, can be given as $\{R_l(z)+\delta_{l,K^{(1)}-1}z\}\prod_{s=0}^{k}(z-p_s^{(0)})/(1-p_s^{(0)}z)$, where $\delta_{\cdot,\cdot}$ is the Kronecker delta. Therefore, when we evaluate its inverse Z transform at $n=-1$, we have the expression of $\gamma_{k,l}$:

$$\gamma_{k,l}=\frac{1}{2\pi j}\oint_{|z|=1}\left(R_l(z)+\delta_{l,K^{(1)}-1}z\right)\prod_{s=0}^{k}\frac{z-p_s^{(0)}}{1-p_s^{(0)}z}dz. \qquad (18)$$

Because the pole in a unit circle is $\{p_k^{(1)}\}_{l\leq k\leq K^{(1)}}$, the transfer gains $\gamma_{k,l}$ of (18) is computable using the residue calculations as follows:



(a) Filtered result (SSOR+double precision)

(b) Filtered result (SSOR+single precision)

(c) Filtered result (FBOR+double precision)

(d) Filtered result (FBOR+single precision)

Fig.5 Comparison of calculation precision on the filtering process of projection data.
Fig. 5(a) – (d) show the filtering results to the projection data by the proposed SSOR method and the conventional FBOR method both with two precision. In these projection images, the horizontal axis of the image is angle from horizontal direction in cross-sectional image, and the vertical axis is radial direction from the center in cross-sectional image. The number of projections is set 1024, and the number of sampling points by one projection line is set 1024 (τ=1).

$$\begin{aligned}\gamma_{k,l} = (1-p_l^{(1)}p_{l+1}^{(1)})\Bigg[& \frac{(p_l^{(1)})^2}{p_l^{(1)}-p_{l+1}^{(1)}}\prod_{s=l+2}^{K^{(1)}-1}\frac{1-p_s^{(1)}p_l^{(1)}}{p_l^{(1)}-p_s^{(1)}}\prod_{s=0}^{k}\frac{p_l^{(1)}-p_s^{(0)}}{1-p_s^{(0)}p_l^{(1)}} \\ & +\frac{(p_{l+1}^{(1)})^2}{p_{l+1}^{(1)}-p_l^{(1)}}\prod_{s=l+2}^{K^{(1)}-1}\frac{1-p_s^{(1)}p_{l+1}^{(1)}}{p_{l+1}^{(1)}-p_s^{(1)}}\prod_{s=0}^{k}\frac{p_{l+1}^{(1)}-p_s^{(0)}}{1-p_s^{(0)}p_{l+1}^{(1)}} \\ & +\sum_{m=l+2}^{K^{(1)}-1}\Bigg\{\Bigg(\frac{(p_m^{(1)})^2}{(p_m^{(1)}-p_l^{(1)})(p_m^{(1)}-p_{l+1}^{(1)})}\prod_{s=l+2,s\neq m}^{K^{(1)}-1}\frac{1-p_s^{(1)}p_m^{(1)}}{p_m^{(1)}-p_s^{(1)}} \\ & +\delta_{m,K^{(1)}-1}p_m^{(1)}\Bigg)\prod_{s=0}^{k}\frac{p_m^{(1)}-p_s^{(0)}}{1-p_s^{(0)}p_m^{(1)}}\Bigg\}\Bigg]. \qquad (19)\end{aligned}$$

Thus, we have the initial value for the backward scans by the following:

$$v_{N-K^{(0)}+k}^{k} = \sum_{l=0}^{K^{(0)}-1}(\gamma_{k,l}\cdot u_{N-1-l}^{l}) \qquad (20)$$

V. Evaluation by Simulation

In general, the sensitivity coefficients of the IIR filter with the lattice form is lower than the direct form. Therefore, in this section, we evaluate the accuracy of the proposed SSOR method by comparing the FBOR method by using the single precision and the double precision floating point arithmetic at filtering process.

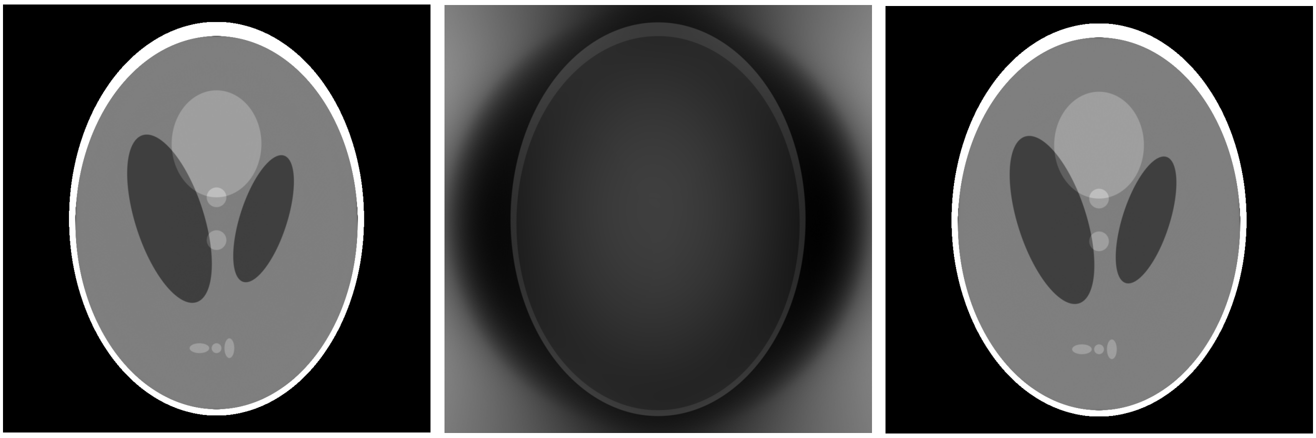
A. Design of DHT

DHT in (6) is designed using the method of bilinear transformation from the analog prototype filter that is adopted the Elliptic filter in order to have a characteristic of brickwall in the transition band [8]. The design parameters are selected as follows experimentally: half of filter order is K=9 (the full order of allpass filter is 19th), and transition band width of DHT is $2\pi\times 2.0\times 10^{-5}$. Although a ripple arises with an Elliptic filter, it is $\delta = 9.0\times 10^{-4}$ negligibly small. As a result, filter coefficients obtained by this are shown in a TABLE I. Most of the filter coefficients are close to 1, therefore, the accuracy may be lost in the calculation of ($p_m^{(1)}- p_s^{(1)}$) in (14) of the FBOR method.

TABLE I COEFFICIENTS OF ALLPASS FILTER CONSTITUTES DHT

$p_0^{(0)}$	0.18566495333171432	$p_0^{(1)}$	0.5282824098880475
$p_1^{(0)}$	0.77724225721229034	$p_1^{(1)}$	0.90473322777988785
$p_2^{(0)}$	0.96097801349957346	$p_2^{(1)}$	0.98431448486802842
$p_3^{(0)}$	0.99378410291412467	$p_3^{(1)}$	0.99765544416900143
$p_4^{(0)}$	0.99940014964476187		

This table shows the set of coefficients of allpass filter with order K = 9 in (8b) which constitutes DHT in (6) using this simulation. Since the ramp filter in (2') is well approximated in this application, DHT has characteristic of brickwall in transition band. Most the filter coefficients are close to 1.



(a) Reconstructed Image (SSOR method) (b) Reconstructed Image (FBOR method) (c) Reconstructed Image (FFT method)

Fig.6 Comparison of the calculation accuracy by reconstructed image.
Fig. 6(a) – (c) show the reconstruction results by the SSOR method, by the FBOR method and by the conventional method (FFT method) processed by FFT (Fast Fourier Transform) using a FIR filter with the single precision floating point calculation respectively. The reconstructed images by SSOR method and FFT method are 50 times amplified in values, while that the amplitude of the reconstructed images by FBOR method is about 6 times. Although the time of filtering process of the SSOR method is twice the FBOR method, it can be performed by less than 1/10 of the FFT methods.

B. *Experiment method*

In this simulation, the cross-sectional data with 1024×1024 pixels called the Head Phantom data of Shepp and Logan [9] is used. The number of projections is set 1024, and the number of sampling points by one projection line is set 1024 (τ=1). The format of the floating point used is that a single precision floating point has 32 bits in total, 1 bit for sign, 8 bits for exponent, and 23 bits for significant parts. A double precision floating point has 64 bits in total, 1 bit for sign, 11 bits for exponent, and 52 bits for significant parts. Moreover, we performed the simulation using the computer whose CPU is Genuine Intel$^{(R)}$ T1300 at 1.66GHz, memory is 1.24GB RAM, and compiler is Visual C++.

C. *Experiment results*

Fig. 5 (a)-(d) show the filtering results to the projection data by the proposed SSOR method and the conventional FBOR method both with the two precisions. With the double precision floating point calculation, there is no difference between both the filtered images in Fig. 5 (a) and (c). However, with the single precision floating point calculation, the SSOR method can provide the similar filtered image in Fig. 5 (b) as with the double precision, whereas the FBOR method cannot do it where the output image is sustained by a lot of lines in Fig. 5 (d).

Then, we compare the accuracy of the reconstructed image. Fig. 6 (a)-(c) show the reconstruction results by the SSOR method, by the FBOR method, and by the conventional method (FFT method) processed by FFT (Fast Fourier Transform) using a FIR filter with the single precision floating point calculation, respectively. The reconstructed images by the SSOR method and the FFT method are 50 times amplified in the values, while that the amplitude of the reconstructed images by the FBOR method is about 6 times. These results indicate that even with the single precision floating point calculation, our SSOR method can provide the similar high-quality image to the FFT method.

TABLE II PROCESSING TIME FOR FILTERING PROCESS

method	double precision	single precision
SSOR	0.274 sec	0.306 sec
FBOR	0.148 sec	0.144 sec
FFT	3.254 sec	-

This table shows the processing time for only filtering process. The processing time of the SSOR method is less than 1/10 of the FFT method, and it takes twice the time that the method FBOR. At the double and the single precision, there is no great difference so much.

Finally, the result of the processing time for the filtering process is shown in TABLE II. Although the time of filtering process of the SSOR method is twice of the FBOR method, it can be performed by less than 1/10 of the FFT methods. Besides, the double and the single precision, there is no great difference so much.

V. CONCLUSION

In this paper, we proposed a *successive-scan one-round (SSOR) method* of implementing the discrete Hilbert transform (DHT) function by cascading primary lattice allpass filters to allow the use of single precision floating points, aiming at the application to compose a ramp filter in convolution backprojection method for X-ray CT. We confirmed the effectiveness of our proposal through simulations. In future works, we will study the optimization of the coefficients for the filter, and improvement in the speed by an efficient program since the scan of the SSOR method is 4 times that of the FBOR method in filtering process.

REFERENCES

[1] L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," J. Opt. Soc. Am. A, vol.1, no. 6, pp. 612-619, 1984.

[2] E. Masahiro, "Recent progress in medical imaging technology," IEICE Trans. (D-II), vol. J87-D-II, no.1, pp.3-18, Jan. 2004.

[3] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, 1982.

[4] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[5] J. Murakami, K. Mizowaki, and Y. Morikawa, "Reconstruction algorithm for computerized tomography using tree-structured filter bank," IEICE Trans. (D-II), vol.J84- DII, no.3, pp.580-589, 2001.

[6] K. Ueda, H. Morimoto, J. Murakami, and Y.Morikawa, "Accuracy improvement of CT reconstruction using tree-structured filter bank," IEICE Trans. (D), vol.J92- D, no.7, pp.1056-1065, 2009.

[7] H. Tsutsu, K. Hayashi, H. Morimoto, J. Murakami, and Y. Morikawa, "An implementation of IIR Hilbert transformer for CT reconstructions," IEICE Tech. Report, vol. 110, no. 121, MI2010-43, pp.33-37, 2010.

[8] P. P. Vaidyanathan, *Multirate systems and Filter banks*, Prentice Hall, 1993.

[9] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," IEEE Trans. Nucl. Sci., vol. NS-21, no.3, pp.21-43, 1974.