Motion Compensated Fan-Beam Reconstruction for Nonrigid Transformation

Taguchi Katsuyuki, Kudo Hiroyuki

IEEE transactions on medical imaging
Volume 27
Number 7
Page range 907-917
Year 2008-07
(C) 2008 IEEE
URL http://hdl.handle.net/2241/100109
doi: 10.1109/TMI.2008.925076
Motion Compensated Fan-Beam Reconstruction for Nonrigid Transformation

Katsuyuki Taguchi*, Member, IEEE, and Hiroyuki Kudo, Member, IEEE

Abstract—We develop an approximate fan-beam algorithm to reconstruct an object with time-dependent nonrigid transformation such as the heart. The method is in the form of derivative back-projection filtering with compensation of affine transformations on a local basis. Computer simulations showed the proposed method significantly reduces image artifact due to nonrigid motion. Therefore, with very little motion artifact, the proposed method allowed us to reconstruct images from projections over about one motion cycle, resulting in reduced image noise level down to 40% of the current level.

Index Terms—Computed tomography (CT), motion compensation.

I. INTRODUCTION

CARDIOVASCULAR disease remains the leading cause of death in the western world, placing an ever-increasing burden on both private and public health services [1]. The electrocardiogram-gated cardiac X-ray computed tomography (CT) imaging is a promising noninvasive technique for early detection and characterization of various signs of cardiovascular disease such as fatty vulnerable soft plaque (atherosclerosis) in coronary arteries, perfusion defect in myocardial, etc., [2]. However, there are two major problems with the current cardiac technique: large radiation dose to patients and insufficient temporal resolution [3]. The typical radiation dose with cardiac CT is 10–15 mSv, which is 3–5 times as large as a standard chest CT scan. The current temporal resolution is merely 80–165 ms in contrast to the minimum requirement of 10–30 ms to observe the beating heart motion without motion artifact. The current technique uses the electrocardiogram signals to select projection data acquired in a time window that is placed within a cardiac cycle with relatively slow motion (e.g., diastole). Images are then reconstructed by neglecting the cardiac motion within the time window, resulting in blurring and artifacts in the reconstructed images. Also, this technique uses only 10%–30% of the acquired data—within the cardiac time window—and throws away the rest of “off-phase” data, resulting in unnecessary radiation dose to patient if the tube current is not prospectively modulated. Recently, a method to turn off the X-ray for off-phase has been proposed [4]. This method will reduce the dose significantly at the expense of the functional (motion) information. Our approach has a potential to achieve the largest dose reduction while substantially improving the quality of 3-D and 4-D (4-D = 3-D plus time) images.

Therefore, the long-term goal of this research is to develop time resolved, low-dose cardiac CT imaging [3]. Specifically, we will develop an iterative algorithm that estimates the time-dependent motion vector field of the heart from the measured projection data and integrates it into the image reconstruction process. The motion will be estimated by maximizing the agreement between the acquired 4-D projection data and the reconstructed time-resolved 4-D images. The quality of the image will be significantly improved since the motion is compensated. In addition, lower tube current could be utilized since all of the acquired data will be used to reconstruct any cardiac phase of interest. We estimate the reduction of radiation dose to the patient will be 50%–75% of the current level [3]. A reduction of noise or dose may be achieved by using a nonlinear processing of projection data [5]–[7], which can be combined with the proposed method for further reduction of noise or dose or both.

Toward this long-term goal, one can incorporate a time-dependent motion vector field into a system matrix and solve this 4-D reconstruction problem using a numerical (statistical) algorithm. Gilland et al. developed a method that jointly estimates and compensates for the motion of the object for single photon emission CT [8], [9]. We aim at developing a similar iterative joint method; however, we want to replace the numerical reconstruction method by an analytical method as it could provide a faster convergence in the iteration process due to its linear nature. In this study, we will develop a method to reconstruct time-resolved 2-D images of dynamic objects from fan-beam projections assuming the time-dependent 2-D motion vector field is known. The algorithm will be integrated into the iterative process later.

Cardiac motion estimation is indeed a very challenging ill-posed problem. There have been many methods proposed and tested with clinical cases using optimization methods, optical flow methods, and continuum theory [10]–[16].

For respiratory motion correction in lung CT, Crawford’s approximate fan-beam algorithm [17] compensates a global expansion/translation of the object. The algorithm was a ramp filtering (FBP) with a change of variables to take a global motion model into account. Roux et al. [18] developed an exact fan-beam algorithm for a global time-dependent affine transformation by incorporating transformation into Noo’s derivative filtered backprojection with Hilbert transform kernel (DFBP)
algorithm [19]. Recently, Desbat et al. extended Roux’s motion model; his algorithm allows any deformations that maintain acquisition lines as straight lines [20], [21].

Most of nonrigid transforms such as cardiac or respiratory motion do not satisfy this restriction—lines become curves with deformations. For respiratory motion compensation, Ritchie et al. [22] applied Crawford’s algorithm on a local basis by changing the motion model for each pixel. Despite its global nature of the ramp filtering, the reconstructed images were in good quality demonstrating significantly reduced motion artifact. This shows the potential of the correction based on locally accurate motion model, filtering, and backprojection.

We hypothesize that by applying the correction locally, we can approximately reconstruct cardiac images with an acceptable level of error. Our approach to cardiac reconstruction is similar to Ritchie’s with an effort to make filtering process much more accurate. We incorporate an affine transformation into a derivative backprojection filtering (DBPF) algorithm [23], [24].

The structure of this paper is as follow. In Section II, we discuss a rigid motion and derive a basic algorithm, DABPF. In Section III, we outline DAxBPF, i.e., how to apply DABPF on a local basis to compensate for nonrigid motion. Sections IV and V present schemes and results of computer simulations, respectively, followed by discussion and conclusion in Section V.

II. RIGID MOTION

In this section, we will derive an exact algorithm, DABPF, from DBPF algorithm to reconstruct a dynamic object with globally defined time-dependent affine transformation.

A. Notations

We use the right hand coordinate system with \( \mathbf{x} = (x, y) \). The time-dependent 2-D deforming object is defined by \( f_0(\mathbf{x}_0) \), \( \mathbf{x}_0 \in \mathbb{R}^2 \), at the reference time \( t = t_0 \), and \( f_1(\mathbf{x}_1) \), \( \mathbf{x}_1 \in \mathbb{R}^2 \), at time \( t \). The time-dependent affine operation \( \Gamma_t \) projects a point \( \mathbf{x}_t \) at time \( t \) to a point \( \mathbf{x}_0 \) at time \( t_0 \)

\[
\Gamma_t(\mathbf{x}_t) = A_t \mathbf{x}_t + B_t = \mathbf{x}_0
\]

where

\[
A_t = \begin{bmatrix} a_{11}(t) & a_{12}(t) \\ a_{21}(t) & a_{22}(t) \end{bmatrix} \quad \text{and} \quad B_t = \begin{bmatrix} B_{1}(t) \\ B_{2}(t) \end{bmatrix}.
\]

We suppose that this affine deformation is invertible, that is, \( \forall t, \det A_t \neq 0 \), where \( \det A_t \) is the determinant of matrix \( A_t \). The object can thus be written as

\[
f_1(\mathbf{x}_1) = f_0(\Gamma_t(\mathbf{x}_t)) = f_0(\mathbf{x}_0).
\]

Fig. 1. Example of true and virtual source trajectories with rigid motion. Straight lines remained as lines after being transformed from time \( t \) to \( t_0 \).

The true circular source trajectory \( \mathbf{s}_t(\lambda) \) and fan-beam projections \( g_t(\lambda, \omega_t) \) can be noted as

\[
\mathbf{s}_t(\lambda) = (-R_\theta \cos \lambda, -R_\theta \sin \lambda)^T
\]

\[
g_t(\lambda, \omega_t) = \int_{0}^{\infty} f_t(\mathbf{s}_t(\lambda) + \omega_t d\lambda)
\]

where \( \lambda \in \mathbb{R} \) is a parameter of the source trajectory, \( \omega_t \) is a unit vector along a ray from \( \mathbf{s}_t(\lambda) \), and \( R_\theta \) is the distance from the source to the rotation axis. Note \( \lambda \) is monotonically increasing function of time, e.g.,

\[
\lambda = \omega t.
\]

B. DABPF Algorithm

Let a 2-D function \( \mathbb{I}_{0, \theta}(\mathbf{x}_0) \) be a Hilbert transformed 2-D image \( f_0(\mathbf{x}_0) \) at the reference time \( t_0 \), where \( \theta \) is a unit vector that defines the direction of 1-D finite inverse Hilbert transform [23] (see also Appendix). We want to first obtain \( \mathbb{I}_{0, \theta}(\mathbf{x}_0) \) from measured projections \( g_t(\lambda, \omega_t) \). We transform a circular source orbit \( \mathbf{s}_t(\lambda) \) around the object \( f_t \) into a virtual source trajectory \( \mathbf{s}_0(\lambda) \) around the object \( f_0 \) at \( t_0 \) (Fig. 1)

\[
\mathbf{s}_0(\lambda) = \Gamma_t(\mathbf{s}_t(\lambda)).
\]

The projection data with the virtual path can then be obtained by

\[
g_0(\lambda, \omega_0) = ||A_t \omega_t|| g_t(\lambda, \omega_t)
\]

where

\[
\omega = \frac{A_t \omega_t}{||A_t \omega_t||}.
\]
Proof: Using (3) inside the integral of the right-hand side of (5) and the linearity, the object \( f \) at time \( t \) can be noted by

\[
f_t(s_\phi(\lambda) + i\phi) = f_0(\Gamma_x[s_\phi(\lambda) + i\phi])
= f_0(\Gamma_x[s_\phi(\lambda)] + i \times \Gamma_x[i\phi - \Omega])
= f_0(\Gamma_x[s_\phi(\lambda)] + i \times (A_{\Theta} \phi + B_{\Theta} - B_{\phi}))
= f_0(s_\phi(\lambda) + iA_{\Theta} \phi).
\]

(10)

Inserting (10) into (5) and changing a variable, \( l = \hat{l}/\|A_{\Theta} \phi\| \), we get

\[
g_h(\lambda, \Omega) = \int_0^\infty f_0(s_\phi(\lambda) + lA_{\Theta} \phi) dl
= \frac{1}{\|A_{\Theta} \phi\|} \int_0^\infty f_0\left(s_\phi(\lambda) + \frac{lA_{\Theta} \phi}{\|A_{\Theta} \phi\|}\right) dl
= \frac{1}{\|A_{\Theta} \phi\|} \times \int_0^\infty f_0\left(s_\phi(\lambda) + l\Omega\right) dl
= \frac{1}{\|A_{\Theta} \phi\|} \times g_h(\lambda, \Omega)
\]

(11)

where \( \Omega \) is given by (9).

The derivative of the virtual projection data, (8), can then be calculated by using product rule and (6)

\[
\frac{\partial}{\partial \lambda} g_h(\lambda, \Omega) = \frac{\partial}{\partial \lambda} \|A_{\Theta} \phi\| g_h(\lambda, \Omega)
+ \|A_{\Theta} \phi\| \frac{\partial}{\partial \lambda} g_h(\lambda, \Omega)
= \frac{1}{\omega} \left( \frac{\partial A_{\Theta}}{\partial \phi} \phi \right) g_h(\lambda, \Omega)
+ \|A_{\Theta} \phi\| \frac{\partial}{\partial \lambda} g_h(\lambda, \Omega)
\]

(12)

where

\[
\frac{\partial}{\partial \lambda} g_h(\lambda, \Omega) = \lim_{\varepsilon \to 0} \frac{g_h(\lambda + \varepsilon, \Omega) - g_h(\lambda, \Omega)}{\varepsilon}.
\]

(13)

The following procedures are the same as DBPF algorithms (see [23] and Appendix). From here, we drop the argument \( \lambda \) from vectors \( s_\phi \), \( s_0 \), and \( \Omega \) for more readability. The differentiated projections are backprojected from the virtual source trajectory \( s_\phi \)

\[
b_0(s_\phi) = \int_{\lambda_1}^{\lambda_2} u(s_\phi \cdot u_0, u_0) \cdot s_\phi(\theta, \omega) \frac{\partial}{\partial \lambda} g_h(\lambda, \Omega) d\lambda
\]

(14)

where

\[
\theta = \theta_0 - [u_0 \cdot s_\phi] \cdot \theta_0
\]

(15)

\[
\omega = \frac{[u_0 - \theta_0]}{\|u_0 - \theta_0\|}
\]

(16)

\( \lambda_1, \lambda_2 \) are both ends of the backprojection range, and \( w \) for \( \lambda_i \) is a normalized redundancy weight defined as

\[
\sum_{i=0}^{\infty} u(s_\phi \cdot u_0, u_0) = 1.
\]

(17)

Finally, we obtain \( f_0(s_\phi) \)

\[
f_0(s_\phi) = \frac{1}{2\pi} H^{-1}(b_0(s_\phi))
\]

(18)

where \( H^{-1}(\cdot) \) is the 1-D finite inverse Hilbert transform along \( \phi \) direction [23]. We call this method DABPF algorithm. Note that globally defined affine transformations preserve straight lines as straight lines (see Fig. 1); thus, DABPF is exact and mathematically the same as Roux’s DAFBP.

DABPF algorithm starts with measured projections and a known time-dependent globally defined affine transform. We summarize DABPF algorithm by the following four steps that are similar to DBPF algorithm [23], [24]. Consider a \((u, v)\) coordinate system (Fig. 2), rotated by angle \( \theta \) from the \((x, y)\)-axes. Two coordinate systems are related by \( u(x, y) \) and \( v(x, y) \).

Step 1) Define a region-of-interest (ROI) and a set of the Hilbert lines \( L(v) \) such that

a) every point \( \phi = (x, y) \) inside the ROI lies on only one Hilbert line \( L(v(x, y)) \);

b) all line integral projection data passing through any object point on each Hilbert line are measured.

In Fig. 2, we show two examples on how to choose the set of the Hilbert lines.

Step 2) Along each Hilbert line \( L(v(x, y)) \), we compute the derivative backprojection by (12)-(14) to obtain the \( b_0(u, v) \). 1-D Hilbert transformed image of \( f \) along \( \phi \).

Step 3) Along each Hilbert line \( L(v(x, y)) \), we invert the Hilbert transform by using the inversion formula of finite Hilbert transform (23) and (18) to obtain a function \((u, v)\), which is the same as \( \tilde{f}(x, y) \) but is represented on the \((u, v)\) coordinates.

Step 4) From the relation \( f(x, y) = \tilde{f}(u(x, y), \phi(x, y)) \), we obtain \( f(x, y) \). This step is not necessary if \( \theta = 0 \).

C. Data Sufficiency Condition

The data sufficiency condition for DABPF is, as a combination of those for DAFBP method [18], [19] and DBPF method [23]: a point \( s_0 \) at the reference time \( t_0 \) can be accurately reconstructed from fan-beam projections acquired during a time-dependent affine deformation, provided every line passing through any object point on the filtering line of the point \( s_0 \) has an intersection with the virtual source trajectory. Fig. 4 provides a graphical description of the condition.

III. NONRIGID MOTION

In this section, we first approximate a nonrigid motion by a spatially varying affine transformation. We will then propose an
Proposed DAxBPF applies DABPF on a local basis to approximate the curve by its tangent, corresponding $s_0(\lambda)$, and locally fitted affine transform for each $x_0$.

The deformation and temporal samples at time $t_0$ could have different affine transformations; or a time-dependent motion vector $f_0(x,y)$ can then be approximated by an affine transform.

A spatially varying affine transform for each small region-of-interest (ROI) at time $t_0$ can then be approximated by an affine transform.

Applying it for each ROI can then be approximated by an affine transform for each element of matrix and vector of affine field within the ROI can then be approximated by an affine transform. Applying it for each $x_t$ will generate spatially varying affine transformations.

A. Spatially Varying Affine Transform

We consider two scenarios how a nonrigid transformation of the object is given: sparsely defined time-dependent affine transformations; or a time-dependent motion vector field.

The first scenario starts with a finite sets of time-dependent affine transformations $\Gamma_{tk}(x_{tk})$ given only for discrete spatial locations $x_{tk}$, $m = 1, \ldots, M$, and temporal samples $f_{tk}$, $k = 1, \ldots, K$. We then spatially and temporally interpolate each element of matrix and vector of affine transform [see (2)] by, e.g., a cubic spline interpolation, to obtain an affine transformation for each small region-of-interest (ROI) at time $t$ (that corresponds to each projection data). An example is shown in Fig. 5. We focus on this scenario in this paper.

The second scenario starts with a time dependent motion vector field. The time dependent deformation $M_t(x_t)$, defined by a time-dependent motion vector for each location, projects a point $x_0$ at time $t$ to a point $x_0$ at time $t_0$. The deformation $M_t(x_t)$ has to be bijective (one-to-one and onto) to be invertible. For a small ROI centering $x_t$, a vector field within the ROI can then be approximated by an affine transformation.

B. DAxBPF Algorithm

We apply DABPF algorithm to compensate for the space- and time-dependent affine transform $\Gamma_{tkm}$ on a local basis (see Fig. 3). Theoretically, each image pixel $x_t$ could have different affine transformation $\Gamma_{tkm}(x_t)$, $m = 1, \ldots, N_x$, where $N_x$ is the number of pixels along $(x,y)$-axes. As a pixel-specific compensation is computationally too expensive, we propose the following approach called DAxBPF.

DAxBPF first reconstructs a finite set of images $f_{0,m}(x_0)$, $m = 1, \ldots, M(\ll N_x)$, at the reference time $t_0$ by using DABPF (steps 1–4 in Section II-B) to compensate for $\Gamma_{tkm}$.

The second scenario starts with a time dependent motion vector field. The time dependent deformation $M_t(x_t)$, defined by a time-dependent motion vector for each location, projects a point $x_0$ at time $t$ to a point $x_0$ at time $t_0$. The deformation $M_t(x_t)$ has to be bijective (one-to-one and onto) to be invertible. For a small ROI centering $x_t$, a vector field within the ROI can then be approximated by an affine transformation.

We then spatially and temporally interpolate each element of matrix and vector of affine transform [see (2)] by, e.g., a cubic spline interpolation, to obtain an affine transformation for each small region-of-interest (ROI) at time $t$ (that corresponds to each projection data). An example is shown in Fig. 5. We focus on this scenario in this paper.

A. Spatially Varying Affine Transform

We consider two scenarios how a nonrigid transformation of the object is given: sparsely defined time-dependent affine transformations; or a time-dependent motion vector field.

The first scenario starts with a finite sets of time-dependent affine transformations $\Gamma_{tk}(x_{tk})$ given only for discrete spatial locations $x_{tk}$, $m = 1, \ldots, M$, and temporal samples $f_{tk}$, $k = 1, \ldots, K$. We then spatially and temporally interpolate each element of matrix and vector of affine transform [see (2)] by, e.g., a cubic spline interpolation, to obtain an affine transformation for each small region-of-interest (ROI) at time $t$ (that corresponds to each projection data). An example is shown in Fig. 5. We focus on this scenario in this paper.

The second scenario starts with a time dependent motion vector field. The time dependent deformation $M_t(x_t)$, defined by a time-dependent motion vector for each location, projects a point $x_0$ at time $t$ to a point $x_0$ at time $t_0$. The deformation $M_t(x_t)$ has to be bijective (one-to-one and onto) to be invertible. For a small ROI centering $x_t$, a vector field within the ROI can then be approximated by an affine transformation.

We then spatially and temporally interpolate each element of matrix and vector of affine transform [see (2)] by, e.g., a cubic spline interpolation, to obtain an affine transformation for each small region-of-interest (ROI) at time $t$ (that corresponds to each projection data). An example is shown in Fig. 5. We focus on this scenario in this paper.

A. Spatially Varying Affine Transform

We consider two scenarios how a nonrigid transformation of the object is given: sparsely defined time-dependent affine transformations; or a time-dependent motion vector field.

The first scenario starts with a finite sets of time-dependent affine transformations $\Gamma_{tk}(x_{tk})$ given only for discrete spatial locations $x_{tk}$, $m = 1, \ldots, M$, and temporal samples $f_{tk}$, $k = 1, \ldots, K$. We then spatially and temporally interpolate each element of matrix and vector of affine transform [see (2)] by, e.g., a cubic spline interpolation, to obtain an affine transformation for each small region-of-interest (ROI) at time $t$ (that corresponds to each projection data). An example is shown in Fig. 5. We focus on this scenario in this paper.

The second scenario starts with a time dependent motion vector field. The time dependent deformation $M_t(x_t)$, defined by a time-dependent motion vector for each location, projects a point $x_0$ at time $t$ to a point $x_0$ at time $t_0$. The deformation $M_t(x_t)$ has to be bijective (one-to-one and onto) to be invertible. For a small ROI centering $x_t$, a vector field within the ROI can then be approximated by an affine transformation.

We then spatially and temporally interpolate each element of matrix and vector of affine transform [see (2)] by, e.g., a cubic spline interpolation, to obtain an affine transformation for each small region-of-interest (ROI) at time $t$ (that corresponds to each projection data). An example is shown in Fig. 5. We focus on this scenario in this paper.
and as Step 5, a weighted summation is then applied to images 
\( f_{0,m}(x_0) \) with

\[
 f_0(x_0) = \frac{\sum_{m=1}^{M} W_{0,m}(x_0)f_{0,m}(x_0)}{\sum_{m=1}^{M} W_{0,m}(x_0)} \tag{19}
\]

where \( W_{0,m} \) is a spatially varying weight which corresponds to spatially changing elements of affine transformation (Fig. 5). Note that the filtering direction can be independently defined for each ROI.

### C. Data Sufficiency Condition

Similar to DABPF algorithm, and as Pack et al., empirically stated in [25], the data sufficiency condition for DABPF is as follows: a point \( x_0 \) at the reference time \( t_0 \) can be reconstructed from fan-beam projections acquired during a time-dependent nonrigid deformation with better quality, provided every line passing through any object point on the filtering line of the point \( x_0 \) has an intersection with the virtual source trajectory. The reconstructed point \( x_0 \) is with better quality than if the condition is not satisfied. Unfortunately, satisfying the condition in itself does not guarantee that the resulting quality of the image is sufficient.

### IV. EVALUATION METHODS

Computer simulations were performed to assess the performance of the proposed algorithms in terms of the following aspects with nonrigid motion: 1) overall motion artifact, 2) spatial resolution, 3) nonperiodic motion pattern, 4) the effect of feathering in reconstruction, 5) image noise, 6) tradeoff between image noise and spatial resolution, and 7) off-synchronized motion.

#### A. General Methods

We used a 5-ball phantom defined in Table I. Unless otherwise mentioned, the motion of the phantom was as follows: one motion cycle period governed all of balls; the ball #2 in the north rotated over 60° about the origin while the others over 30°; all balls contract to 2:1; all balls translate over (40 mm, 20 mm). Specifically, elements of affine transformation in (2) are defined as

\[
 A_t = C_t \text{Rot}(\varphi_t) \tag{20}
\]

\[
 C_t = c_0 + c_1 \cos \left( \frac{2\pi t}{T_{RR}} \right) \tag{21}
\]

\[
 \varphi_t = \varphi_0 + \varphi_1 \cos \left( \frac{2\pi t}{T_{RR}} \right) \tag{22}
\]

\[
 B_{i,t} = B_{i,1} \sin \left( \frac{2\pi t}{T_{RR}} \right) ; \quad i = 0, 1
\]

\[
 B_{0,1} = -20; \quad B_{1,1} = -10 \tag{23}
\]

where \( \text{Rot}(\varphi) \) is a rotation matrix for an angle of \( \varphi \), \( T_{RR} \) is a time period for one motion cycle, \( (\varphi_0, \varphi_1) \) for the north ball and for the others were \( (30°, -30°) \) and \( (15°, -15°) \), respectively. Three \( T_{RR} \)’s that correspond to 60, 63, and 67 cycles-per-minute (bpm) were used; their projection ranges were \( 6\pi/T_{RR} \), 5.71\( \pi/T_{RR} \), 5.37\( \pi/T_{RR} \), respectively, with a gantry rotation time of \( 1/3s/(2\pi) \approx 333.3\text{ms/rev} \). At 60 bpm, the object motion cycle and projection rotation cycle are in full-sync: \( T_{RR} \) corresponds to \( \pi/7.2 \); 72 bpm and 90 bpm are next fully-synchronized conditions. \( T_{RR} \) to \( 5\pi \) and \( 4\pi \), respectively. Taguchi et al. showed that the effect of object motion simply depends on the extent of such synchronization [26]. Thus, studying the motion cycles of 60, 63, and 67 bpm will be sufficient to see the effect of synchronization from a full-sync to a severe off-sync. Among them, we used the motion with 63 bpm intensively.

As the affine elements for the north ball were different from those for the others, we employed the method described in Section III-A and obtained smoothly changing affine transformations (Fig. 5). DABPF used the affine transformation defined for balls #1, #3, #4, and #5.

Fan-beam projection data were generated over \( 8\pi \) with \( 1/3s/(\pi) \approx 333.3\text{ms/rev} \) and monochromatic X-ray at 80 keV. The other scan conditions were similar to a clinical X-ray CT scanner: Equiangular 672 samples over 52.14°; 1160 projections per \( 2\pi \); the source-to-isocenter distance of 570 mm. An example of generated projection data is shown in Fig. 5.

Circular images were reconstructed with a matrix of \( 512^2 \) over a diameter of 500-mm range. During the image reconstruction with DABPF and DABPF methods, when the range of backprojection for an \( x \) was \( n_\lambda \pi, n_\lambda \pi \in N^\ast \), a rectangular weight with a height of \( 1/n_\lambda \) was used for the normalized redundancy weight \( w \). When the backprojection range was a fraction of \( \pi \), \( (n_\lambda + 2\beta)\pi, 0 < \beta < 1 \), a trapezoidal weight was used such that \( 1/n_\lambda \) in the middle region and linearly reduced over \( \beta\pi/2 \) to zero at both edges. For \( H^{-1}(\cdot) \), \( \beta \) was fixed at (1.0), horizontally from left to right; and known pixel value \( f_0(x_0) = 0 \) for \( ||x_0|| > 220 \text{mm} \) was used to obtain the offset value (see (17) of [23]). A Shepp–Logan filter [27] and Parker weight [28] was used for FBP method.

The “true images” were obtained as follows. At each time \( t \), projection data over \( 8\pi \) was obtained without phantom motion; and an image was reconstructed by FBP with a \( 1/8 \) weight.

#### B. Overall Motion Artifact and Spatial Resolution

The 5-cylinder phantom with a 63 bpm motion was used. Images at \( t = 0 \), \( T_{RR}/4 \), \( T_{RR}/2 \), and \( 3T_{RR}/4 \), were reconstructed by FBP, DABPF, and DABPF, respectively.

The profiles of the images at \( t = 0 \) and \( T_{RR}/2 \) were obtained to assess the accuracy of pixel values and the spatial resolution.
C. Backprojection Range and Feathering

The projection data with 63 bpm was used. In order to evaluate the effect of a nonperiodic object motion, three images were reconstructed by DAXBPF from projections over 2\(\pi\), 4\(\pi\), and 6\(\pi\) (corresponding to 0.35, 0.70, and 1.05 cycles). A rectangular redundancy weight was applied in each case.

At the presence of the object motion, a feathering weight is often applied to reduce the effect of inconsistency in projection data [29]–[31]. We assessed the ability of such feathering weights to reduce the motion artifacts. We used a trapezoidal weight and reconstructed images from projections over 2\(\pi\), 4\(\pi\), and 6\(\pi\), respectively, and compared images with a rectangular weight.

D. Image Noise

The projection (line integral) data with 63 bpm was converted to the transmitted X-ray intensity with the incident flux of 5000 photons per ray. Poisson noise was then added and the noisy intensity data were then log-converted back to line integrals. Image reconstruction was employed by FBP and DAXBPF with projections over 2\(\pi\), 4\(\pi\), and 6\(\pi\), respectively.

E. Noise-Resolution Tradeoff

We evaluated the tradeoff between the image noise and spatial resolution at the expanded and contracted phases with and without motion compensation.

We used three projection data under three motion conditions: a) the heart was beating; the heart was stationary either at b) \(t = 0\) (the expanded phase) or at c) \(t = T/2\) (the contracted phase). One image each at \(t = 0\) and \(T/2\) was reconstructed using DAXBPF algorithm. DAXBPF reduces to DBPF without motion correction. An index of the spatial resolution was then obtained as follows. A horizontal profile near the north ball was obtained in each image (Fig. 6, arrows); the inverse of the maximum gradient of the profile was obtained and normalized by that of the stationary expanded object (\(t = 0\)).

Poisson noise was added to projection data using the same procedure as Section IV-D. For each of the three projection sets, we generated 20 realizations of noisy projection data (60 data sets in total). One noisy image each at the expanded and contracted phases was reconstructed from each of noisy data from motion condition a); one noisy image each at the stationary expanded object (condition-b) and c). An index of the image noise was then obtained as follows. The standard deviation for each pixel was computed over the 20 noise realizations. These standard deviations were then averaged over the square region of 110 mm \(\times\) 110 mm (see Fig. 6) and normalized by the average standard deviation obtained from the stationary expanded object (condition-b). We also computed the standard deviation of standard deviation values over the entire image.

We obtained a tradeoff curve of spatial resolution and noise with Gaussian filter strength as the parameter along the curve. To each image, we applied a 21 \(\times\) 21 Gaussian filter with various strength and measured the above two indexes for noise and spatial resolution.

F. Off-Synchronized Motion

We assessed the effect of two kinds of off-synchronizations, inter-off-sync and intra-off-sync, both of which induces an inconsistency between projections at the beginning and the end of backprojection range. In the first experiment, the object motion cycle and the scanner’s projection angle were off-sync (inter-off-sync). The cycle of the entire object motion was chosen as 60, 63, and 67 bpm corresponding to a scanner’s rotation of 6\(\pi\)/cycle (fully synced), 5.71\(\pi\)/cycle, 5.37\(\pi\)/cycle, respectively. Images were reconstructed from projections over 6\(\pi\) using DAXBPF.

In the second experiment, the north ball and the other balls were off-sync (intra-off-sync). The north ball followed 60 and 67 bpm while the others did 63 bpm. Images were reconstructed by DAXBPF from data over 6\(\pi\) and 6\(\pi\).

V. Evaluation Results

A. Overall Motion Artifact and Spatial Resolution

Fig. 6 shows images reconstructed by FBP, DABPF, and DAXBPF together with the true phantom images with a narrow window width (0.054/0.18 corresponding to 300 H.U./0 H.U.) to enhance the artifact.

Images of FBP exhibited strong shading/whitening artifact and distorted shape of balls. Images of DABPF showed almost no artifact except for severely smeared north ball. The artifact did not spread out to large area as FBP. Images of DAXBPF presented no visible artifacts in all motion phases.

Horizontal profiles at \(t = 0\) and \(T_{RR}/2\) shown in Fig. 7 confirmed the above observations: FBP provided shifted or distorted profiles; DABPF also showed distorted profiles; the edge of DAXBPF was slightly blurred at \(t = 0\) but values in flat regions were accurate.

We made two observations on edge profiles of DAXBPF. First, they were more blurred than that of FBP in general; and second, the sharpness of edges was phase dependent. DAXBPF provided sharper edge profiles at \(t = T_{RR}/2\) (contracted phase)
than those at \( t = 0 \) (expanded phase)—in fact, the profiles at \( t = T_{RR}/2 \) are very close to the “truth,” FBP without object motion.

We believe common causes of the two phenomena must be the derivative process with respect to \( \lambda \) and the motion compensation process. When derivative is calculated by differentiating adjacent samples as in (13) and if those samples are spatially separated in a larger distance than adjacent rays used in the ramp filtering of FBP, the derivative process reduces the spatial resolution. This is a common problem with other methods that use a differentiating step [32]–[34].

The interval of samples is transformed as \( n_0 = A_t n_t \) during the motion compensation process. The intrinsic Nyquist frequency \( f_{Ny} \) defined by the sampling intervals along \( n_t \) at \( t \) is then converted to \( f_{Ny}/||A_t n_t|| \) at the reference time \( t_0 \). The image at the expanded phase \( t = 0 \) is reconstructed with \( ||A_t n_t|| > 1 \) for almost all projections, which results in the reduced Nyquist frequency, so as the spatial resolution of images. In contrast, the image at the contracted phase \( t = T_{RR}/2 \) is reconstructed with \( ||A_t n_t|| < 1 \), thus, shows the improved spatial resolution. If the scale of the object is not isotropic (i.e., scales along \( x \)- and \( y \)-axes are different from each other), it may cause a phase-dependent asymmetric point spread function.

### B. Backprojection Range and Feathering

Fig. 8(a)–(c) shows images reconstructed by DAxBPF from \( 2\pi \), \( 4\pi \), and \( 6\pi \), respectively. The effect of nonperiodic motion patterns during the backprojection range was seen as shadings (arrow) and white cloud (curved arrow) in Fig. 8(a) (\( 2\pi \)) and 7b (\( 4\pi \)). Notice, however, a very narrow window width is used to enhance the shading, which is merely 0.002 cm\(^{-1}\) or a 3.2% of the difference between the north ball and ball #1 (on center). The relative rotating motion between the north ball and the rest of the phantom over \( 2\pi \) was \( 21^\circ \), which was twice as large as the rotating motion over \( \pi(10.5^\circ) \). Comparing Fig. 8(a) with the FBP in Fig. 6, we found the proposed local correction worked noticeably well to reduce the effect of nonrigid motion.

The visibility of the shading artifact was significantly reduced by using a slight overlap (\( 0.2\pi \)) with a trapezoidal feathering weight [Fig. 8(d)–(f)]—although the reduced pixel value remained.

### C. Image Noise

Fig. 9 shows reconstructed noisy images at \( t = 0 \). In order to compare image noise with the same point spread function, a \( 7 \times 7 \) Gaussian filter was applied to the image reconstructed by FBP to produce Fig. 9(a). A square ROI was used at the isocenter to measure the standard deviation of pixel values, which was normalized against that of filtered FBP image. The normalized standard deviation value of DAxBPF method decreased with increasing a distance from the center from 0 to 100 mm. There was no angular dependency.

Fig. 10(a) shows four tradeoff curves. Four observations were made: 1) when the object was stationary, images at the...
two phases had different noise level at the same spatial resolution; 2) when the object was beating, images at the two phases had different spatial resolutions; 3) at the expanded phase, the tradeoff curve for the beating phantom is inferior to that for the stationary phantom; and 4) at the contracted phase, the tradeoff curve for the beating phantom indicates a better performance than for the stationary phantom.

The first observation is attributed to the difference in phantom size. Larger objects attenuate X-ray photons more, and produce noisier projection data (and reconstructed images) for a given incident X-ray photon flux. And discussion in Section V-A explains the mechanism of the second observation.

A further study is required to understand the nature of the tradeoffs, observations 3 and 4, in detail. However, we think the logarithm conversion process may be the major cause. Photon noise in acquired X-ray intensity data is Poisson distributed in this study (it will be compound Poisson with additive electronic noise in actual data), which is log-converted for line integrals and then scaled during the motion compensation process of DABPF. This nonlinear process may introduce a favorable tradeoff for DABPF when images at the contracted phase were reconstructed. Since the gain/loss of tradeoff curves must depend on the shape and the motion of the object, we cannot definitively conclude that this property is or is not an advantage of DABPF when applied to clinical patient data.

### E. Off-Synchronized Motion

Fig. 11 shows difference images with and without inter-off-synchronization from the image of FBP with a stationary object at $t = 0$. The motion cycle of the entire object and the rotation cycle of the scanner were in full-sync (a: 60 bpm), moderately off-sync (b: 63 bpm), and largely off-sync (c: 67 bpm). Both images with out-of-sync were almost identical to a fully-synchronized case (60 bpm), demonstrating the robustness of DABPF algorithm.

Fig. 12 presents difference images with intra-off-sync from the image of FBP with the stationary object. A negligible level of derivation of pixel values ($\pm 0.0002$ cm$^{-1}$, 1 H.U., $\pm 0.1\%$) was observed (arrows), which was in fact reduced by applying the feathering.

### VI. DISCUSSION AND CONCLUSION

We developed an exact DABPF algorithm that compensates for a time-dependent affine transformation of the object. We also developed an approximate algorithm, DABPF, to reduce the effect of non-rigid motion of the object. The results showed that an approximate DABPF algorithm, which applies DABPF on a local basis, reconstructs images with good quality. DABPF significantly reduces the motion artifact compared with the current cardiac algorithm (FBP with Parker weight). The proposed DABPF also reduces the image noise when it uses projections over a larger range than Parker weight does. A 60% noise reduction from the current level, if directly converted, implicates an 83% dose reduction from the current level (from 10–15 mSv to 1.7–2.6 mSv) to obtain images with the current noise level. This could have a substantial impact on the public health.

Our results show that the reconstructions from projections over $2\pi$ is likely to be acceptable; yet images over $6\pi$ or quasi-one cycle is better in terms of the intensity of artifacts and image noise. The use of data over one heart cycle is practically possible in clinical settings. The current cardiac CT protocols choose a helical pitch between 0.2–0.33 (corresponding to $6\pi$–$10\pi$ for each point $\varphi$) depending on the patient heart rate, such that each point $\varphi$ is covered at least over one heart cycle.

The affine motion model does not preserve the mass of the object as it deforms. It proportionally changes the mass with the scale. This motion model may not be a right one for various
motions of human body. However, as the heart expands and contracts, iodine-mixed blood comes into and goes out of chambers (e.g., left ventricle), which drastically increases and decreases the total mass. Thus, in fact, affine transform is a better model for cardiac imaging than those which preserve the mass.

A natural question one may ask is, “Why does DAXBPF work well for nonrigid motion?” Unfortunately, we have been unsuccessful in providing a mathematical explanation. Intuitive reasoning is a slow and quasi-periodic object motion relative to the projection angle. Let us consider an object which consists of a finite set of delta functions and assume a nonrigid motion simply moves their locations, \( f(x) = \sum_m \delta(x_m) \), where \( m \) is an index for delta functions. The filtered parallel projection data can then be decomposed into a finite set of sinograms. With nonrigid motion, the backprojection (integration) process introduces an error at a point \( x \) as each sinogram is radially shifted due to the motion of the delta function at \( x_m \), for each time \( t \). The effect of such shift introduces inaccurate values, weights and integration range. If the motion within the integration range is drastic, the effect is significant. However, if the motion is slow and smooth within the angular range, the error remains small. A quasi-periodic motion over the angular range also reduces the inconsistency between the beginning and the end of angular range, thus, reduces the artifact. We tested phantoms with various off-sync and nonperiodic motions reasonable in cardiac imaging (Figs. 6, 8, and 9) and obtained good results.

The above intuitive explanation seems applicable to other analytical reconstruction algorithms with different filtering kernels. For example, applying the local compensation approach to Roux’s and Desbat’s DAFBP algorithms [18], [20], [21] would yield DAXFBP algorithm. It would be interesting to study whether such DAXFBP provides as good images as DAXBPF. We plan to implement DAXFBP and evaluate its performance.

The proposed DAXBPF will provide good images even at the presence of transverse truncation as DBPF does [23], [24]. As DAXBPF do not require projection data outside of the region-of-interest and its filtering lines, it also has a potential to reduce necessary radiation dose to the patient in cardiac scan.

The disadvantage of DAXBPF is a computational expense. The current FBP requires \( O(N^3) \) calculations for the ramp filtering process and \( O(N^3) \) calculations for backprojection process. If \( M \) sets of motion models are used along each filtering line, DAXBPF will first reconstruct \( M \) images, and then apply a weighted summation. Thus, DAXBPF requires \( O(MN^3) \) for finite inverse Hilbert transform and \( O(MN^3) \) for backprojection process. An acceleration process such as fast Fourier transform is not applicable for finite inverse Hilbert transform. Therefore, DAXBPF algorithm will be approximately \( M \) times computationally demanding than the current FBP methods. The appropriate number of motion models in one dimension, \( M \), depends on the complexity of the motion, thus, is of interest in future works. From literatures for modeling and estimation of cardiac motion [10]–[16], we assume 10–30 would be sufficient to cover the human heart (100 mm) with which the nonrigid motion over 3–10 mm range is approximated by one affine motion model.

Future works are as follows. The image quality with different nonrigid motion patterns needs to be further evaluated. The phantom we used in this study was not too simple; however, as the algorithm remains approximate, one has to carefully examine the performance of DAXBPF method with more complex motions, such as 4-D NCAT phantom or clinical cardiac data.

The spatial resolution of image needs to be improved. We believe, as discussed in Section V-A, one of its major causes is the derivative (or differentiation) step with respect to \( \lambda \). One can use a new scheme recently proposed by Noo et al. [35] with a chain rule: it does not require a significant change from the current formulae. Another solution is to use integration by parts as Zou et al. did in [36], which, however, requires a significant change of reconstruction formulae. Moreover, there is a concern as we expect an increased spatial variation of image noise as \( \|z_0 - z_0(\lambda)\|^2 \) weight will replace \( \|z_0 - z_0(\lambda)\|^{-1} \) weight.

The extension of the proposed fan-beam DAXBPF method to cone-beam reconstruction is relatively straightforward. The derivative of cone-beam projections is taken with respect to the source parameter, \( \lambda \), followed by a weighted cone-beam back-projection and a 1-D finite inverse Hilbert transform. The optimal filtering direction of 1-D Hilbert transform is an interest of the research. Candidates include an actual PI-line formed by virtual trajectory and a virtual PI-line [37] that does not connect two source points in either virtual or actual trajectories.

In conclusion, we have developed fan-beam reconstruction algorithms using derivative backprojection filtering approach to compensate for a time-dependent deformation. DAXBPF method for a global affine transformation is exact. DAXBPF for a nonrigid transformation are approximate, which showed promising results with computer simulations.

**Appendix**

For the sake of completeness, we rederive with consistent notations fan-beam DBPF algorithm for an arbitrary orbit.

In the reference time or for the stationary object, parallel-beam projection can be expressed as

\[
p(r, \phi) = \int_{-\infty}^{\infty} f(r_{\rho} + r' \rho_{\perp}) \, dr'
\]

where \( \rho = (\cos \phi, \sin \phi) \), and \( \rho_{\perp} = (-\sin \phi, \cos \phi) \) (see Fig. 13).
The DBPF algorithm for a parallel beam projections [23] can be written as

\[
p(r; \phi) = \lim_{\epsilon \to 0} \frac{p(r + \epsilon, \phi) - p(r, \phi)}{\epsilon}
\]

(25)

\[
b(\xi) = \int_{\phi_0}^{\phi_0} w(r, \phi) \text{sgn}(\theta - \eta) p'(r, \phi) d\phi
\]

(26)

\[
f(\xi) = -\frac{1}{2\pi} H^{-1}(b(\xi))
\]

(27)

where \(H^{-1}(\cdot)\) is the 1-D inverse Hilbert transform along \(\theta\) direction [23] and \(w\) is a normalized redundancy weight that satisfies

\[
\sum_{i=0}^{\infty} [w(r, \phi + 2i\pi) + w(-r, \phi + (2i + 1)\pi)] = 1.
\]

(28)

Parallel- and fan-beam projections are related by

\[
p(r; \phi) = g(\lambda; \alpha) [g(\lambda; \eta_{\alpha}) = \eta_{\alpha}]\eta_{\alpha} = 0.
\]

(29)

Taking a derivative with respect to \(\lambda\) of both sides of (29) yields

\[
p'(r; \phi) = \frac{g'(\lambda; \alpha)}{g'(\lambda; \eta_{\alpha})} [g(\lambda; \eta_{\alpha}) = \eta_{\alpha} = 0].
\]

(30)

For simplicity, we use a prime sign to indicate the derivative with respect to \(\lambda\) (e.g., \(g' \equiv \partial g/\partial \lambda\)). This rebinning formula (30) can only hold for nonzero \(g'(\lambda; \eta_{\alpha})\); thus, the tangent vector \(g'(\lambda)\) must not be orthogonal to \(\eta_{\alpha}\). Any source points \(g(\lambda)\) and \(g(\lambda_i)\) on the line \(U\) can be parameterized by a distance \(q\)

\[
\xi + q\eta_{\alpha} = g(\lambda)
\]

(31)

where \([q] = \|\xi - g(\lambda)\|\). Differentiating each side of (31) with respect to \(\lambda\) yields

\[
\frac{dq}{d\lambda} \eta_{\alpha} + q \frac{d\phi}{d\lambda} = g'(\lambda).
\]

(32)

Thus

\[
g'(\lambda; \eta_{\alpha}) = \frac{d\phi}{d\lambda} \eta_{\alpha} + \frac{d\phi}{d\lambda} = \frac{g'(\lambda; \eta_{\alpha})}{\|\xi - g(\lambda)\|}.
\]

(33)

Inserting (30) into (25) and (33) into (26) yields a fan-beam BPF formulae for an arbitrary source orbit

\[
g'(\lambda, \alpha) = \lim_{\epsilon \to 0} g(\lambda + \epsilon, \alpha) - g(\lambda, \alpha)
\]

(34)

\[
bq(\xi) = \int_{\lambda_1}^{\lambda_2} w(s, \eta_{\alpha}, \eta_{\alpha}) \text{sgn}(\theta - \eta) g'(\lambda, \alpha) d\lambda
\]

(35)

where

\[
\eta_{\alpha} = \frac{g(\xi - s) - \alpha}{\|\xi - s\|},
\]

\[
\alpha = \frac{\xi - s}{\|\xi - s\|},
\]

(36)

(37)

and the sum of the redundancy weights to rays from \(\lambda_i\) that belong to the same line passing through a point \(\xi\) is 1

\[
\sum_{i=0}^{\infty} w(s, \eta_{\alpha}, \eta_{\alpha}) = 1.
\]

(38)

ACKNOWLEDGMENT

The authors would like to thank W. P. Segars for his help with CT simulation program, and F. Noo, T. Flohr, G. Lauritsch, and J. Xu for their discussion.

REFERENCES


