박사 학위논문

Ph. D. Dissertation

일반화된 1차원 총변동을 이용한 내부 단층 촬영법과 Bedrosian 항등식을 통한 다해상도 구현

Interior tomography using 1D generalized total variation penalty and multiscale implementation by Bedrosian identity

KAIST
1971
KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
한국과학기술원

이 민 지 (李 旼 芝 Lee, Min Ji)

바이오및뇌공학과

Department of Bio and Brain Engineering

KAIST

2016

일반화된 1차원 총변동을 이용한 내부 단층 촬영법과 Bedrosian 항등식을 통한 다해상도 구현

Interior tomography using 1D generalized total variation penalty and multiscale implementation by Bedrosian identity

KAIST

Interior tomography using 1D generalized total variation penalty and multiscale implementation by Bedrosian identity

Advisor : Professor Ye, Jong Chul

by

Lee, Min Ji

Department of Bio and Brain Engineering

KAIST

A thesis submitted to the faculty of KAIST in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Bio and Brain Engineering . The study was conducted in accordance with Code of Research Ethics[1].

2015. 10. 8.

Approved by

Professor Ye, Jong Chul

[Advisor]

[1]Declaration of Ethical Conduct in Research: I, as a graduate student of KAIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

일반화된 1차원 총변동을 이용한 내부 단층 촬영법과 Bedrosian 항등식을 통한 다해상도 구현

이 민 지

위 논문은 한국과학기술원 박사학위논문으로
학위논문심사위원회에서 심사 통과하였음.

KAIST

2015년 10월 8일

심사위원장 예 종 철 (인)

심사위원 박 성 홍 (인)

심사위원 박 용 근 (인)

심사위원 조 승 룡 (인)

심사위원 최 명 철 (인)

DBIS
20105369

이 민 지. Lee, Min Ji. Interior tomography using 1D generalized total variation penalty and multiscale implementation by Bedrosian identity. 일반화된 1차원 총변동을 이용한 내부 단층 촬영법과 Bedrosian 항등식을 통한 다해상도 구현. Department of Bio and Brain Engineering . 2016. 66p. Advisor Prof. Ye, Jong Chul. Text in English.

ABSTRACT

In x-ray computed tomography, exact cone-beam reconstruction algorithms have been developed for various scanning trajectories. Volumes scanned by helical, saddle, and circle-plus-line trajectories can be also reconstructed exactly with these algorithms. Generally, the exact cone-beam reconstruction algorithms are divided into two types, such as filtered backprojection and backprojection filtration. Particularly, the backprojection filtration method has influenced to develop new interior tomography algorithms using the Hilbert transform relationship. One of them is to determine the region of interest uniquely when tiny subregions inside of the region of interest are known. The solution can be obtained by projections onto convex sets, but it is well-known that the convergence speed of the projections onto convex sets is slow. Hence, to overcome the limitation, we employs a parallel proximal algorithm to simultaneously consider multiple convex constraints rather than projecting on each of them sequentially as in the projections onto convex sets. Our simulation results show that the solution for the interior tomography problem can be accurately obtained using parallel proximal algorithm with a much smaller number of iterations than the projections onto convex sets. However, knowing subregions before reconstruction is impractical, so we propose a new method of interior tomography using a one-dimensional generalized total variation seminorm penalty. Unlike the existing interior tomography work with two-dimensional total variation, the proposed algorithm guarantees the exact recovery of two- and three-dimensional piecewise smooth images using a one-dimensional generalized total variation seminorm penalty. The one-dimensional seminorm penalty allow us to convert the original two- or three-dimensional problem into a set of one-dimensional problems, and it leads to a much faster reconstruction. The restrictions placed on each one-dimensional problem by our one-dimensional regularizer are much more relaxed than those imposed by the two-dimensional regularizer in previous works. For the optimization problem with total variation, we use fast gradient projection algorithm with a sublinear rate of convergence. To further accelerate the algorithm up to a level for clinical use, we propose a novel multiscale reconstruction method by exploiting the Bedrosian identity of the Hilbert transform. More specifically, we show that the high frequency parts of the one-dimensional signals can be quickly recovered analytically with the Hilbert transform because of the Bedrosian identity. This implies that computationally expensive iterative reconstruction need only be applied to low resolution images in the downsampled domain, which significantly reduces the computational burden. We show the efficacy of the proposed multiscale algorithm using circular fan-beam and helical cone-beam geometry. With a graphics processing unit implementation, we demonstrate that the speed of the algorithm can be significantly accelerated up to the level for clinical use for various acquisition geometries.

Keywords: Interior tomography, Hilbert transform, generalized TV, Bedrosian theorem, multiscale decomposition

Contents

Abstract i
Contents ii
List of Tables iv
List of Figures v

Chapter 1. Introduction **1**

Chapter 2. Analytic cone-beam reconstruction formula **4**
2.1 X-ray transform and Radon transform 4
2.2 Filtered backprojection method 5
2.2.1 General reconstruction formula 5
2.2.2 Examples of scanning geometry 9
2.3 Backprojection filtration method 13
2.3.1 Fourier analysis of differentiated backprojection 13
2.3.2 Examples of scanning geometry 16
2.4 Implementation 18
2.4.1 Filtered backprojection type 18
2.4.2 Backprojection filtration type 24
2.4.3 Programming by graphics processing unit 26
2.5 Simulation result 29

Chapter 3. Parallel proximal algorithm for interior tomography with tiny a priori knowledge **34**
3.1 Interior tomography using *a priori* knowledge 34
3.2 Proximity operator 36
3.3 Parallel proximal algorithm 36
3.4 Simulation results 38

Chapter 4. Interior tomography using 1D generalized total variation penalty and multiscale implementation by Bedrosian identity **41**
4.1 Interior tomography using 1D generalized total variation 41
4.2 Multiscale implementation by Bedrosian identity 44
4.3 Implementation 46
4.3.1 Differentiated backprojection 47
4.3.2 Multiscale decomposition 48

KAIST

4.4 Simulation results 48
4.4.1 Circular fan-beam CT 49
4.4.2 Helical cone-beam CT 49
4.5 Discussion 53
4.5.1 Comparison with extrapolation method 53
4.5.2 Downsampling Factor 54

Chapter 5. Conclusion **62**

References **63**

Summary (in Korean) **67**

KAIST

List of Tables

2.1 Computational time (seconds) for analytic cone-beam reconstructions 33

4.1 Computational time (seconds) by truncation rate for helical cone-beam CT when depth is 2. 52
4.2 Computational time (seconds) by depth for helical cone-beam CT when truncation rate is 60.9%. 52
4.3 Whole workstation environment for simulating the proposed method. 53

KAIST

List of Figures

1.1 An example of 2D interior tomography problem in a circular fan-beam CT geometry. . . . 1

2.1 Geometric parameters in Katsevich integral. . . . 5
2.2 FBP reconstruction for general trajectory. . . . 8
2.3 FBP reconstruction for circular fan-beam CT. . . . 9
2.4 FBP reconstruction for circle-plus-line cone-beam CT. . . . 10
2.5 FBP reconstruction for saddle cone-beam CT. . . . 10
2.6 Helical cone-beam CT. . . . 11
2.7 FBP reconstruction for helical cone-beam CT. . . . 12
2.8 Katsevich reconstruction for helical cone-beam CT. . . . 12
2.9 BPF reconstruction for circular fan-beam CT. . . . 16
2.10 BPF reconstruction for circle-plus-line cone-beam CT. . . . 17
2.11 BPF reconstruction for helical cone-beam CT 17
2.12 Total process of FBP reconstruction for fan-beam CT. . . . 18
2.13 Total process of FBP reconstruction for cone-beam CT. . . . 18
2.14 Cone-beam projection. . . . 19
2.15 Filtering implementation of FBP algorithm. . . . 21
2.16 Total process of BPF reconstruction for fan-beam CT. . . . 24
2.17 Total process of BPF reconstruction for cone-beam CT. . . . 25
2.18 Discrete indices for the cone-beam projection, PI line, and 3D Cartesian space. . . . 27
2.19 GPU parallelizing of differentiation. J projection images are loaded in GPU memory, and the number of J depends on a capacity of GPU. Each pixel of a differentiation image are parallelized, and for differentiation, the anterior and posterior images are necessary. . . . 28
2.20 GPU parallelizing of backprojection. J differentiated images are loaded in texture memory, and it is accessed with real number. The each thread deals with a PI line, and its number is greater than or equal to the number of PI lines to cover all PI lines at once. Each differentiated images are judged if it is included in helix segment corresponding to the PI line, and if so, it is backprojected on all voxels of the PI line. . . . 29
2.21 GPU parallelizing of Hilbert transform. The coefficients of Hilbert transform is initially saved as a constant array. A thread performs Hilbert transform of a PI line by convolution of Hilbert kernel and backprojection of the PI line. . . . 30
2.22 FBP reconstruction result of saddle CT. Transverse image and its row line profile (top), coronal image and its row line profile (middle), and sagittal image and its column line profile (bottom). . . . 31
2.23 BPF reconstruction result of circle-plus-line CT. Transverse image and its row line profile (top), coronal image and its row line profile (middle), and sagittal image and its column line profile (bottom). . . . 32

3.1 Interior problem with *a priori* knowledge. . . . 36

KAIST

3.2 Reconstruction images and line profiles after 100 iterations. 38
3.3 Error images after 100 or 200 iterations. 39
3.4 MSE convergence of POCS and PPXA. 40

4.1 Step-by-step reconstruction procedure along a PI line. 45
4.2 Flowchart of the proposed multiscale reconstruction approach for interior tomography problem. 46
4.3 The implementation of the circular fan-beam interior tomography. 47
4.4 The implementation of the helical cone-beam interior tomography. 48
4.5 Approximate ellipse boundary. 49
4.6 2D fan-beam reconstruction. For the simulation, the K of the L is 1, and the iteration number of the POCS is 100. The l and u are 0 and 0.060442, respectively, and the λ is 0.001, and N is 20. The depth of the multiscale implementation is 2. (a) Ground truth, (b) low frequency signal reconstruction, (c) high frequency signal reconstruction, and (d) the final reconstruction result. 50
4.7 3D XCAT phantom. 51
4.8 Reconstructed images by the truncated helical cone-beam CT of the 3D XCAT phantom. For the simulation, the K of the L is 1, and the iteration number of the POCS is 100. The l and u are 0 and 0.0309, respectively, and the λ is 0.001, and N is 20. The depth of the multiscale implementation is 2. From the first to third columns, each column shows the slice image when the slice number is (a) 84, (b) 130, and (c) 212, and each row is for the ideal phantom, FBP, BPF, 3D TV, and the proposed method. The last column shows the cut views indicated by the white line on the images. The number written at the corner of the reconstructed image shows the RMSE and NRMSE in the Hounsfield unit (HU) scale. 56
4.9 3D inner organ phantom. 57
4.10 Reconstructed images by the truncated helical cone-beam CT of the 3D inner organ phantom with noise. For the simulation, the K of the L is 1, and the iteration number of the POCS is 100. The l and u are 0 and 0.0604, respectively, and the λ is 0.001, and N is 20. The depth of the multiscale implementation is 2. From the first to third columns, each column shows the same (a) transverse, (b) coronal, and (c) sagittal image, and each row is for full-beam BPF, FBP, BPF, 3D TV, and the proposed method. The last column shows the cut views indicated by the white line on the images. The number written at the corner of the reconstructed image shows the RMSE and NRMSE in the HU scale. . . 58
4.11 Error images from the extrapolated projection by FBP, BPF, and the proposed method when $L_{\text{extension}} = 140$. 59
4.12 NRMSE value changes with respect to $L_{\text{extension}}$. 59
4.13 Window for interior tomography $w(x)$. (a) is for the large FOV case, and (b) is for the small FOV case. 59
4.14 Interior tomography reconstruction results with various downsampling rate for two kinds of window size. Reconstructed $f(x)$ from (a) a large FOV and (b) a smaller FOV. The spectrum $\hat{w}(\omega)$ and $\hat{g}_H(\omega)$ for (c) a large FOV and (d) a smaller FOV. The subscript numbers after f and $\hat{g}_H$ represent the downsampling rate. In (c)-(d), B refers to the Nyquist sampling rate of the original DBP data. 60

KAIST

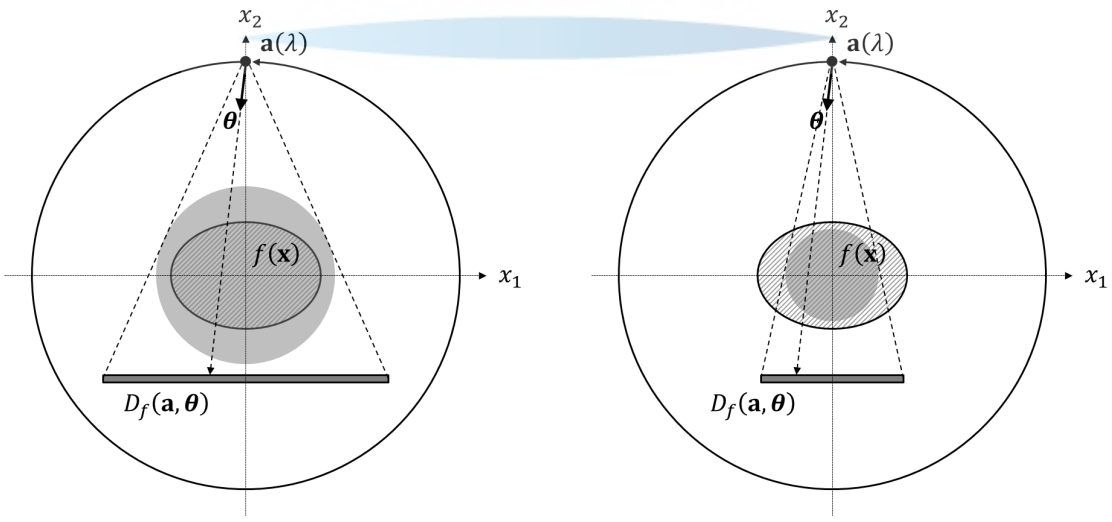
4.15 Interior tomography reconstruction result in the noisy helical cone-beam CT with varying the downsampling rate. The left most numbers indicate the downsampling rate, and for each downsampling rate, transverse, coronal, and sagittal images of the reconstruction are shown. The number written at the corner of the reconstructed image shows the RMSE and NRMSE in the HU scale. 61

KAIST

Chapter 1. Introduction

For a circular cone-beam computed tomography (CT) scanner with a two-dimensional (2D) detector, the Feldkamp-Devis-Kress (FDK) method has been used in most. However, it is an approximate reconstruction algorithm causing cone-beam artifacts. Exact cone-beam reconstruction algorithms can remove these artifacts though different scanning trajectories to cover the same field of view (FOV) with the FDK method. There are two types of the exact cone-beam reconstruction methods; filtered backprojection (FBP) method [1, 2, 3, 4, 5, 6, 7, 8] and backprojection filtration (BPF) method [9, 10, 11, 12, 13, 14, 15, 16]. Both methods consist of same reconstruction steps; derivative, one-dimensional (1D) Hilbert filtration, and backprojection, but the orders are different. The FBP method reconstructs images by backprojection of filtered derivative scanning images, and the BPF method reconstructs images by filtering backprojection of derivative images. In this research, we provide the mathematical preliminaries of the FBP and BPF reconstruction and explain how to implement the algorithms for the circular fan-beam CT and the saddle, circle-plus-line, helical cone-beam CT. We also describe the graphics processing unit (GPU) implementation of both FBP and BPF methods. The GPU provides multicore processors for parallelization and texture memories for fast interpolation [17]. The reconstruction algorithms particularly for the case of cone-beam CT need to compute a lot of repetitive tasks and interpolations, so the GPU implementation can make the reconstruction much faster than CPU.

KAIST



(a) Full-beam tomography (b) Interior tomography

Figure 1.1: An example of 2D interior tomography problem in a circular fan-beam CT geometry.

The main aim of this research is to address the classical interior problem, which is a limited data sampling problem from a 2D or three-dimensional (3D) x-ray CT where projections at each view extend only to the shadow of a circular region completely interior to the subject being scanned. Our object model is continuous, and the view angle and detector sampling are also continuous. For example, Figure 1.1 illustrates a 2D circular fan-beam source and detector systems with two types of detector sizes, where

the region of interest (ROI) area of each case is shaded in gray. While the detector in Figure 1.1(a) has a large enough size to cover the whole support of the object, the detector in Figure 1.1(b) is too small to cover the object from *any* projection view. We call the case in Figure 1.1(b) interior tomography or ROI tomography. Interior tomography is useful in reducing the radiation exposure by limiting the x-ray aperture for specific organs such as the heart. It has also been investigated to reduce the x-ray detector size for cost savings. However, due to detector truncation at *each* view, the underlying image cannot be reconstructed accurately by analytic methods, and approximate methods such as sinogram extrapolation method [18] are widely used instead. Nonetheless, this often introduces bias to the CT number in the reconstructed image.

So far, most of the prior studies [19, 20, 21, 9, 22, 23] have been concerned about ROI reconstruction problems where the projection data at some view angles are not truncated, so one can use differentiated backprojection (DBP) to reconstruct the entire ROI starting from the complete data region. On the other hand, due to detector truncation at all projection views, our problem is more restricted, and these approaches cannot directly be used. Recently, it was shown that an ROI image can be reconstructed uniquely from truncated projection data when the ROI includes a background with air [24] or the intensity of the subregions inside the ROI is known *a priori* [25, 26]. More specifically, after backprojecting the differentiated projection data on chord lines, a 2D or 3D tomography problem using fan-beam or cone-beam CT can be converted into a set of 1D problems, which makes the interior tomography problem much simpler extrapolation problem. Based on this formulation, general imaging geometries that allow perfect reconstruction have been identified [24]. For example, if the imaging ROI includes a background with air, then *a priori* knowledge of regions with air determines unique reconstruction. Another important example is that even though there exists no air interface in the ROI, if tiny regions within the ROI and boundary of the object are known *a priori*, an unique and stable solution can be computed [25, 26]. Asides from the theoretical results, the authors in [24, 25, 26] noted that the constraints due to *a priori* knowledge are convex and used projections onto convex sets (POCS) algorithm to obtain the solution by sequentially projecting on the constraint sets. They demonstrated excellent reconstruction results using POCS. However, one of the main limitations is that the POCS method finds the optimized solution by sequentially projecting the current solution onto each convex set, so its convergence speed is slow.

To accelerate the convergence speed, we employ a parallel proximal algorithm (PPXA) that has been recently proposed to address multiple convex constraints using a proximal algorithm [27]. More specifically, PPXA calculates the multiple proximal projections and computes a weighted average using Douglas Rachford splitting algorithm [27]. As the interior tomography problem can be converted into a 1D optimization problem with multiple convex constraints, it is nicely fitted into PPXA framework. In particular, the convex constraints in the interior tomography problem are just a set of indicator functions, so the corresponding proximal mappings are very simple, which makes the computational complexity of PPXA minimal. The chapter 3 explains how to apply PPXA to the interior tomography with *a priori* knowledge, and show that PPXA converges faster than POCS method. Even though the current formulation is based on 2D formulation, extension to 3D is straightforward and will be reported later.

Even if the convergence speed is accelerated by PPXA, however, it is difficult to know the intensity inside an object in general. In [28], a continuous domain singular value decomposition of the finite Hilbert transform operator characterized in [29] was used to represent an interior image with a linear combination of the eigenfunctions, after which the null space compensation was performed by using a more general set of prior subregion knowledge. Moreover, Katsevich and Wang [30] proved the general

uniqueness results for the interior problem and provided stability estimates. On the other hand, using 2D or 3D total variation (TV), the authors in [31, 32, 33] showed that a unique reconstruction is possible if the images are piecewise constant, linear or smooth. In these works, an iterative procedure to reconstruct the interior images requires multiple applications of forward and backward projections; thus, its computational complexity prohibits its routine use clinically, even though highly optimized implementations of these iterative approaches are currently available for clinical use [34].

In our previous work [35], we revisited the classical interior problem, where projections at each view extend only to the shadow of a circular region completely interior to the subject being scanned, and the object model is continuous, and the view angle and detector sampling are also continuous. In particular, we were also interested in reconstructing images that are piecewise smooth; i.e., the domain where the image is defined can be decomposed into a finite number of subdomains such that the image is a smooth function on each piece. However, our approach generalized the ideas developed in [31, 32, 33] with a simplified 1D formulation on the *chord lines*, and as a result, we were able to perfectly reconstruct new classes of functions. In particular, we showed that we can reconstruct any image that is a generalized L-spline along a collection of chord lines passing through the ROI [35]. For example, we can reconstruct images whose 1D restrictions are nonpolynomial exponential B-splines, which were not covered in [31, 32, 33].

To further accelerate the algorithm up to a level for clinical use, we proposes a multiscale reconstruction method that sequentially reconstructs the low frequency and high frequency parts of a 1D signal at different resolutions. Related frequency-split reconstruction methods have been used in the x-ray CT community for various purposes such as artifact reduction from redundant data [36] or metal artifact reduction [37]. However, in our case, the frequency-split is a natural by-product of the Hilbert transform property, which can be used for a more efficient numerical implementation. More specifically, we show that the high frequency part of the signal can be recovered *analytically* using a one-step Hilbert transform thanks to the powerful Bedrosian identify in the Hilbert transform of the product of two functions [38]. Therefore, computationally expensive iterative reconstruction need only be performed to reconstruct the low frequency part of the signal after downsampling, which enables a fast implementation. The decomposition of DBP signal into low frequency part and high frequency part can be achieved by downsampling and upsampling, and they are implemented by wavelet decomposition and reconstruction. In addition, the algorithm can be performed in parallel for each chord, which lends itself for easy implementation using a GPU. We verify the efficacy of the algorithm using various acquisition trajectories, which show that a very accurate reconstruction can be obtained quickly on a practical level using the clinical helical cone-beam CT acquisition geometry.

This Thesis is organized as follows. Chapter 2 provides the mathematical preliminaries of the analytic exact cone-beam reconstruction methods and implementation of them. Chapter 3 explains how to apply the PPXA to the interior tomography with tiny a priori knowledge. Chapter 4 describes the main idea of the interior tomography using generalized 1D TV and multiscale implementation. In this chapter, we provide the experimental results and discussions. Lastly, we conclude in chapter 5.

Chapter 2. Analytic cone-beam reconstruction formula

2.1 X-ray transform and Radon transform

In x-ray CT, the x-ray photons are attenuated when they pass through the body. The attenuation can be expressed by an exponential relationship for the linear attenuation coefficient by assuming monochromatic spectrum. The linear attenuation coefficient μ denotes the probability that an x-ray or photon will interact with the material per unit path length travelled:

$$\mu = -\frac{(\frac{dI}{ds})}{I} \ , \tag{2.1}$$

where I denotes the x-ray intensity. Assuming that scattered photons are usually rejected through scatter grid or detector collimator, the linear attenuation coefficients mainly characterizes the photo-electric event. Then, (2.1) provides us the well-known Beer-Lambert law that quantifies the x-ray intensity at s:

$$I(s) = I(0) \exp\left(- \int_0^s ds' \ \mu(s') \right) \ , \tag{2.2}$$

where $I(0)$ denotes the impinging x-ray flux before it is absorbed by the body. Then, by taking the minus logarithm, we have

$$- \log \frac{I(s)}{I(0)} = \int_0^s ds' \ \mu(s') \ . \tag{2.3}$$

Mathematically, (2.3) is often called the x-ray transform. Let the variable $\boldsymbol{\theta}$ denotes a vector on the unit sphere $\mathbb{S}^{d-1} \in \mathbb{R}^d$. The collection of vectors that are orthogonal to $\boldsymbol{\theta}$ is denoted as

$$\boldsymbol{\theta}^{\perp} = \{ \mathbf{u} \in \mathbb{R}^d : \mathbf{u} \cdot \boldsymbol{\theta} = 0 \} \ . \tag{2.4}$$

Then, the x-ray transform is formally defined as

$$D_{\mu}(\mathbf{a}, \boldsymbol{\theta}) = \int_0^{\infty} dt \ \mu(\mathbf{a} + t\boldsymbol{\theta}) \ , \tag{2.5}$$

where μ corresponds to the linear attenuation coefficient and $\boldsymbol{\theta} \in \mathbb{S}^{d-1}$ denotes the unit vector in the x-ray photon propagation direction, and $\mathbf{a} \in \mathbb{R}^d$ refers the x-ray source location.

The x-ray transform is closely related to well-known Radon transform, For a given object function $\mu(\mathbf{x})$, the Radon transform $\mathcal{R}\mu$ of this function at $(\boldsymbol{\theta}, s)$ is given by

$$\mathcal{R}\mu(\boldsymbol{\theta}, s) = \int d\mathbf{x} \ \mu(\mathbf{x}) \delta(s - \boldsymbol{\theta} \cdot \mathbf{x}) = \int_{\boldsymbol{\theta}^{\perp}} d\mathbf{x} \ \mu(s\boldsymbol{\theta} + \mathbf{x}) \ , \tag{2.6}$$

where $s \in \mathbb{R}$. Specifically, in 2D geometry, Radon transform coincides with x-ray transform. However, for 3D imaging, more sophisticated relationship holds.

Radon transform gives us an important Fourier domain relationship:

$$\mathcal{R}\mu(\boldsymbol{\theta}, \xi) = \int ds \ \mathcal{R}\mu(\boldsymbol{\theta}, s) e^{j2\pi\xi s} = \int d\mathbf{x} \ \mu(\mathbf{x}) e^{-j2\pi\xi(\boldsymbol{\theta} \cdot \mathbf{x})} \tag{2.7a}$$

$$= \hat{\mu}(\xi\boldsymbol{\theta}) \tag{2.7b}$$

where $\hat{\mu}(\cdot)$ denotes the Fourier transform of $\mu(\mathbf{x})$. This relationship is often called the projection slice theorem. In fact, the projection slice theorem tells us that the projection data corresponds to the Fourier domain data acquired in polar grid. For 2D problems, this provides us the following reconstruction formula known as the *filtered backprojection (FBP)* algorithm:

$$\mu(\mathbf{x}) = \int d\boldsymbol{\omega}\ \hat{\mu}(\boldsymbol{\omega}) e^{j2\pi\boldsymbol{\omega}\cdot\mathbf{x}} = \int d\boldsymbol{\omega} \int d\xi\ \hat{\mu}(\xi\boldsymbol{\theta})\xi e^{j2\pi\xi\cdot\mathbf{x}} \tag{2.8a}$$

$$= \frac{1}{2}\int d\boldsymbol{\theta}\ g(\boldsymbol{\theta}\cdot\mathbf{x}) \tag{2.8b}$$

where $g(\boldsymbol{\theta}\cdot\mathbf{x})$ denotes the filtered signal using the ramp filter:

$$g(t) = \int_0^\infty d\xi\ |\xi|\hat{\mu}(t)e^{j2\pi\xi t}\ . \tag{2.9}$$

For 3D case, the inversion formula for Radon transform is given by

$$\mu(\mathbf{x}) = -\frac{1}{8\pi^2}\int_{\mathbb{S}^2} d\boldsymbol{\theta}\ \ \frac{\partial^2}{\partial s^2}\mathcal{R}\mu(\boldsymbol{\theta}, s)\Big|_{s=\boldsymbol{\theta}\cdot\mathbf{x}}\ . \tag{2.10}$$

2.2 Filtered backprojection method

2.2.1 General reconstruction formula

In this section, we will review the FBP-type formula for general trajectory. This problem has been extensively studied by many researchers [1, 2, 3, 4, 5, 6, 7, 10].

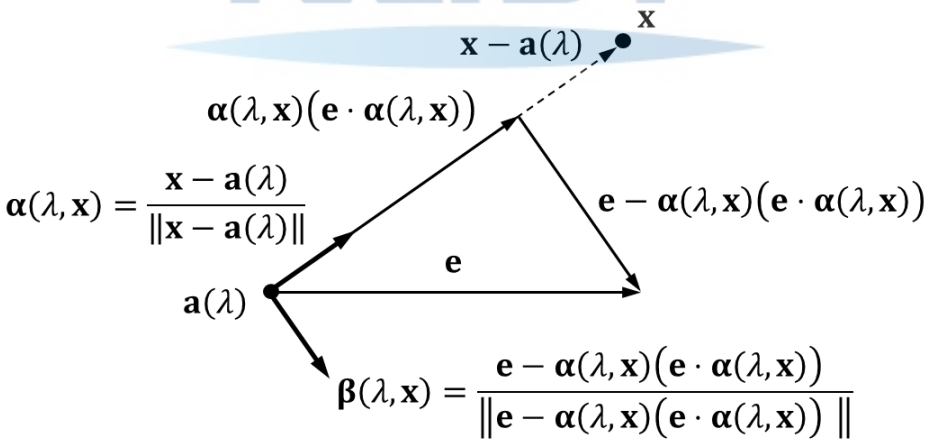


Figure 2.1: Geometric parameters in Katsevich integral.

For a given object function $f(\mathbf{x})$, its d-dimensional inverse Fourier transform from the spectrum $\hat{f}(\boldsymbol{\omega})$ is given by

$$f(\mathbf{x}) = \frac{1}{(2\pi)^d}\int d\boldsymbol{\omega}\ \hat{f}(\boldsymbol{\omega})e^{j\boldsymbol{\omega}\cdot\mathbf{x}}\ , \tag{2.11}$$

where $d = 2, 3$ are for 2D and 3D tomography problems, respectively. In addition, if $\boldsymbol{\theta}$ denote the d-dimensional unit vector and $\mathbf{a} \in \mathbb{R}^d$ is the source location, then the x-ray transform D_f maps a function on $\mathbb{R}^d$ into the set of its line integrals

$$D_f(\mathbf{a}, \boldsymbol{\theta}) = \int_0^\infty dt\ f(\mathbf{a} + t\boldsymbol{\theta})\ . \tag{2.12}$$

Then, for a given a filtering direction $\mathbf{e} \in \mathbb{S}$, the following Katsevich integral is now defined [6], which is originated from Katsevich formula for helical cone-beam reconstruction [1, 2, 3, 4, 5]:

$$\mathcal{K}(\mathbf{x}, \mathbf{e}, \lambda^-, \lambda^+) = -\frac{1}{2\pi^2} \int_{\lambda^-}^{\lambda^+} d\lambda \; \frac{1}{\|\mathbf{x} - \mathbf{a}(\lambda)\|} g(\lambda, \mathbf{x}, \mathbf{e}) \tag{2.13}$$

where

$$g(\lambda, \mathbf{x}, \mathbf{e}) = \int_{-\pi}^{\pi} d\gamma \; \frac{1}{\sin\gamma} \; \frac{\partial}{\partial \nu} D_f(\mathbf{a}(\nu), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e})) \Big|_{\nu=\lambda} \tag{2.14}$$

with

$$\boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e}) = \cos\gamma \boldsymbol{\alpha}(\lambda, \mathbf{x}) + \sin\gamma \boldsymbol{\beta}(\lambda, \mathbf{x}, \mathbf{e}) \tag{2.15}$$

and

$$\boldsymbol{\alpha}(\lambda, \mathbf{x}) = \frac{\mathbf{x} - \mathbf{a}(\lambda)}{\|\mathbf{x} - \mathbf{a}(\lambda)\|} \; , \quad \boldsymbol{\beta}(\lambda, \mathbf{x}, \mathbf{e}) = \frac{\mathbf{e} - \boldsymbol{\alpha}(\lambda, \mathbf{x})(\mathbf{e} \cdot \boldsymbol{\alpha}(\lambda, \mathbf{x}))}{\|\mathbf{e} - \boldsymbol{\alpha}(\lambda, \mathbf{x})(\mathbf{e} \cdot \boldsymbol{\alpha}(\lambda, \mathbf{x}))\|} \; . \tag{2.16}$$

The specific geometric parameters in Katsevich formula are illustrated in Figure 2.1.

Here, we analyze the mathematical meaning of Katsevich integral, which is fundamental to derive the general formula.

Theorem 2.2.1. *The Katsevich integral can be represented by*

$$\mathcal{K}(\mathbf{x}, \mathbf{e}, \lambda^-, \lambda^+) = \int d\boldsymbol{\omega} \; \hat{f}(\boldsymbol{\omega}) e^{j2\pi\boldsymbol{\omega}\cdot\mathbf{x}} \sigma(\mathbf{x}, \boldsymbol{\omega}) \tag{2.17}$$

where

$$\sigma(\mathbf{x}, \boldsymbol{\omega}) = \frac{1}{2} \operatorname{sgn}(\boldsymbol{\omega} \cdot \mathbf{e}) \left[\operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda^-, \mathbf{x})) - \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda^+, \mathbf{x})) \right] \; . \tag{2.18}$$

Proof. Using the 3D inverse Fourier transform

$$f(\mathbf{x}) = \int d\boldsymbol{\omega} \; \hat{f}(\boldsymbol{\omega}) e^{j2\pi\boldsymbol{\omega}\cdot\mathbf{x}} \; , \tag{2.19}$$

(2.55) can be represented as

$$D_f(\mathbf{a}(\lambda), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e})) = \int_0^{\infty} dt \; f(\mathbf{a} + t(\cos\gamma\boldsymbol{\alpha} + \sin\gamma\boldsymbol{\beta})) \tag{2.20a}$$

$$= \int_0^{\infty} dt \int d\boldsymbol{\omega} \; \hat{f}(\boldsymbol{\omega}) e^{j2\pi\boldsymbol{\omega}\cdot(\mathbf{a} + t(\cos\gamma\boldsymbol{\alpha} + \sin\gamma\boldsymbol{\beta}))} \; . \tag{2.20b}$$

Accordingly, we have

$$g(\lambda, \mathbf{x}, \mathbf{e}) = \int_{-\pi}^{\pi} d\gamma \; \frac{1}{\sin\gamma} \; \frac{\partial}{\partial \nu} D_f(\mathbf{a}(\nu), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e})) \Big|_{\nu=\lambda} \tag{2.21a}$$

$$= \int_{-\pi}^{\pi} d\gamma \; \frac{1}{\sin\gamma} \int_0^{\infty} dt \int d\boldsymbol{\omega} \; \frac{\partial}{\partial \nu} e^{j2\pi\boldsymbol{\omega}\cdot(\mathbf{a}(\nu) + t(\cos\gamma\boldsymbol{\alpha} + \sin\gamma\boldsymbol{\beta}))} \Big|_{\nu=\lambda} \hat{f}(\boldsymbol{\omega}) \tag{2.21b}$$

$$= \int_{-\pi}^{\pi} d\gamma \; \frac{1}{\sin\gamma} \int_0^{\infty} dt \int d\boldsymbol{\omega} \; j2\pi\boldsymbol{\omega} \cdot \dot{\mathbf{a}}(\lambda) e^{j2\pi\boldsymbol{\omega}\cdot(\mathbf{a}(\nu) + t(\cos\gamma\boldsymbol{\alpha} + \sin\gamma\boldsymbol{\beta}))} \hat{f}(\boldsymbol{\omega}) \; , \tag{2.21c}$$

and $\dot{\mathbf{a}}(\lambda)$ denotes the derivative of $\mathbf{a}(\lambda)$ with respect to λ. Making a change of variables

$$u_1 = t\cos\gamma \; , \quad u_2 = t\sin\gamma \; , \tag{2.22}$$

we have

$$\int_{-\pi}^{\pi} d\gamma \; \frac{1}{\sin\gamma} \int_0^{\infty} dt \; e^{j2\pi\boldsymbol{\omega}\cdot(t\cos\gamma\boldsymbol{\alpha} + t\sin\gamma\boldsymbol{\beta})} = \int du_1 \int du_2 \; \frac{1}{u_2} e^{j2\pi\boldsymbol{\omega}\cdot(u_1\boldsymbol{\alpha} + u_2\boldsymbol{\beta})} \tag{2.23a}$$

$$= \delta(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}) \int du_2 \; \frac{1}{u_2} e^{j2\pi u_2 \boldsymbol{\omega}\cdot\boldsymbol{\beta}} \tag{2.23b}$$

$$= j\pi \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\beta}) \delta(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}) \tag{2.23c}$$

$$= j\pi \|\mathbf{x} - \mathbf{a}(\lambda)\| \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\beta}) \delta(\boldsymbol{\omega} \cdot (\mathbf{x} - \mathbf{a}(\lambda))) \tag{2.23d}$$

KAIST

which provides us

$$g(\lambda, \mathbf{x}, \mathbf{e}) = -2\pi^2 \|\mathbf{x} - \mathbf{a}(\lambda)\| \int d\boldsymbol{\omega} \ \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\beta})(\dot{\mathbf{a}}(\lambda) \cdot \boldsymbol{\omega}) \delta(\boldsymbol{\omega} \cdot (\mathbf{x} - \mathbf{a}(\lambda))) \hat{f}(\boldsymbol{\omega}) e^{j2\pi \boldsymbol{\omega} \cdot \mathbf{x}} \ . \tag{2.24}$$

Now, the scaling property for the delta function is following:

$$\delta(\alpha t) = \frac{\delta(t)}{|\alpha|} \ . \tag{2.25}$$

In addition, let $\mathbf{a}(\lambda) \in \mathbb{R}^d$ denote a vector parameterized with respect to $\lambda \in [\lambda^-, \lambda^+]$. Then, for given vectors $\mathbf{x}, \boldsymbol{\omega} \in \mathbb{R}^d$, the delta function scaling property provides us

$$\delta(\boldsymbol{\omega} \cdot (\mathbf{x} - \mathbf{a}(\lambda))) = \sum_k \frac{\delta(\lambda - \lambda_k)}{|\boldsymbol{\omega} \cdot \dot{\mathbf{a}}(\lambda_k)|} \tag{2.26}$$

where $\{\lambda_k\}$ is given by the points such that

$$\Lambda(\boldsymbol{\omega}) = \{\lambda \in [\lambda^-, \lambda^+] : \boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda) = 0\} \tag{2.27a}$$

$$= \{\lambda \in [\lambda^-, \lambda^+] : \boldsymbol{\omega} \cdot \mathbf{x} = \boldsymbol{\omega} \cdot \mathbf{a}(\lambda)\} \ . \tag{2.27b}$$

Therefore, we have

$$\mathcal{K}(\mathbf{x}, \mathbf{e}, \lambda^-, \lambda^+) = \int d\boldsymbol{\omega} \ \hat{f}(\boldsymbol{\omega}) e^{j2\pi \boldsymbol{\omega} \cdot \mathbf{x}} \sigma(\mathbf{x}, \boldsymbol{\omega}) \tag{2.28}$$

where

$$\sigma(\mathbf{x}, \omega) = \sum_k \operatorname{sgn}(\omega \cdot \boldsymbol{\beta}(\lambda_k)) \operatorname{sgn}(\boldsymbol{\omega} \cdot \dot{\mathbf{a}}(\lambda_k)) \ . \tag{2.29}$$

KAIST

We can further simplify the expression in (2.29). Using (2.16), we have

$$\operatorname{sgn}(\boldsymbol{\beta}(\lambda_k) \cdot \boldsymbol{\omega}) = \operatorname{sgn}((\mathbf{e} - (\mathbf{e} \cdot \boldsymbol{\alpha}(\lambda_k))\boldsymbol{\alpha}(\lambda_k)) \cdot \boldsymbol{\omega}) \tag{2.30a}$$

$$= \operatorname{sgn}(\mathbf{e} \cdot (\boldsymbol{\omega} - (\boldsymbol{\alpha}(\lambda_k) \cdot \boldsymbol{\omega})\boldsymbol{\alpha}(\lambda_k))) \tag{2.30b}$$

$$= \operatorname{sgn}(\mathbf{e} \cdot \boldsymbol{\omega}) \tag{2.30c}$$

where we use (2.27a). Hence, we have

$$\sigma(\mathbf{x}, \boldsymbol{\omega}) = \operatorname{sgn}(\boldsymbol{\omega} \cdot \mathbf{e}) \sum_k \operatorname{sgn}(\boldsymbol{\omega} \cdot \dot{\mathbf{a}}(\lambda_k)) \ . \tag{2.31}$$

In addition, according to Pack and Noo [6], if the source trajectory does not have discontinuities, we have

$$\sum_k \operatorname{sgn}(\boldsymbol{\omega} \cdot \dot{\mathbf{a}}(\lambda_k)) = \frac{1}{2}[\operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda^-, \mathbf{x})) - \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda^+, \mathbf{x}))] \tag{2.32}$$

where

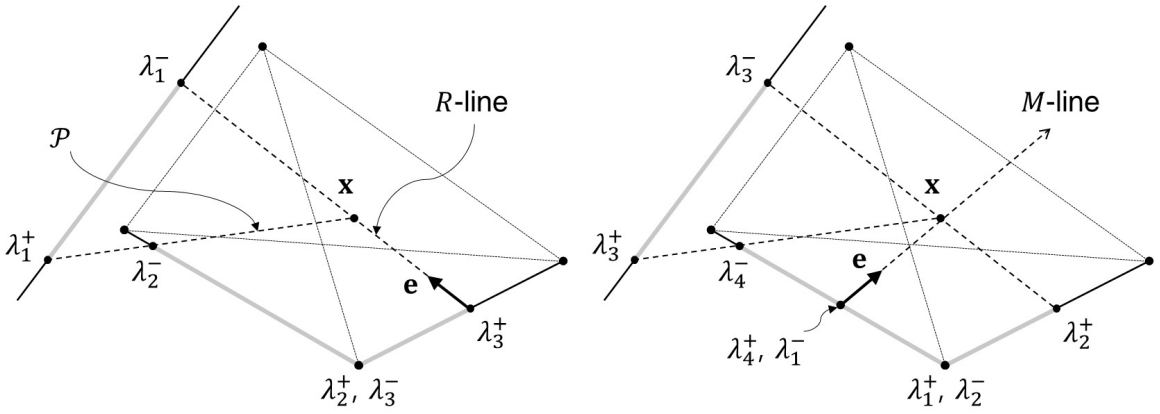
$$\boldsymbol{\alpha}(\lambda^-, \mathbf{x}) = \frac{\mathbf{x} - \mathbf{a}(\lambda^-)}{\|\mathbf{x} - \mathbf{a}(\lambda^-)\|} \ , \quad \boldsymbol{\alpha}(\lambda^+, \mathbf{x}) = \frac{\mathbf{x} - \mathbf{a}(\lambda^+)}{\|\mathbf{x} - \mathbf{a}(\lambda^+)\|} \ . \tag{2.33}$$

Finally, using (2.32), we have

$$\sigma(\mathbf{x}, \boldsymbol{\omega}) = \operatorname{sgn}(\boldsymbol{\omega} \cdot \mathbf{e}) \sum_k \operatorname{sgn}(\boldsymbol{\omega} \cdot \dot{\mathbf{a}}(\lambda_k)) \tag{2.34a}$$

$$= \frac{1}{2} \operatorname{sgn}(\boldsymbol{\omega} \cdot \mathbf{e}) \left[\operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda^-, \mathbf{x})) - \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda^+, \mathbf{x}))\right] \ . \tag{2.34b}$$

This concludes the proof. □



(a) a series of curve segments of R-line (b) a series of curve segments of M-line

Figure 2.2: FBP reconstruction for general trajectory.

By inspection, we know that the condition $\sigma(\mathbf{x}, \boldsymbol{\omega}) = 1$ leads to the exact solution. Based on this observation, Pack and Noo derived a general reconstruction formula for arbitrary trajectory [6]. More specifically, suppose a series of Q curve segments be chosen for each point $\mathbf{x}$ at which reconstruction is to be achieved (see Figure 2.2). These chosen curve segments correspond to a series of intervals $[\lambda_q^-, \lambda_q^+]$ for $q \in [1, \ldots, Q]$, and must be such that

$$k_1 \boldsymbol{\alpha}(\lambda_1^-, \mathbf{x}) = \boldsymbol{\alpha}(\lambda_Q^+, \mathbf{x}) \ , \tag{2.35a}$$

$$k_q \boldsymbol{\alpha}(\lambda_q^-, \mathbf{x}) = \boldsymbol{\alpha}(\lambda_{q-1}^+, \mathbf{x}), \quad q \in [2, \ldots, Q] \tag{2.35b}$$

with each $k_q \in \{-1, 1\}$. The sign of k_q relates to the relative positions of $\mathbf{x}$,$\mathbf{a}(\lambda_{q-1}^+)$, and $\mathbf{a}(\lambda_q^-)$ on the line that connects them. More specifically, if $\mathbf{a}(\lambda_{q-1}^+)$ and $\mathbf{a}(\lambda_q^-)$ are on the opposite sides relative to $\mathbf{x}$, then $k_q = -1$; otherwise $k_q = 1$. Based on this, Pack and Noo provided one of the most generalized reconstruction formulations:

$$\sum_{q=1}^{Q} r_q \mathcal{K}(\mathbf{x}, \boldsymbol{\alpha}(\lambda_Q^+, \mathbf{x}), \lambda_q^-, \lambda_q^+) = \begin{cases} f(\mathbf{x}), & r_Q = -1 \\ 0, & r_Q = 1 \end{cases} \tag{2.36}$$

where

$$r_q = \prod_{i=1}^{q} k_i, \quad q \in [1, \ldots, Q] \ . \tag{2.37}$$

In order to understand this, note that

$$\sum_{q=1}^{Q} r_q \mathcal{K}(\mathbf{x}, \boldsymbol{\alpha}(\lambda_Q^+, \mathbf{x}), \lambda_q^-, \lambda_q^+) = \int d\boldsymbol{\omega} \ \hat{f}(\boldsymbol{\omega}) e^{j2\pi \boldsymbol{\omega} \cdot \mathbf{x}} \sigma_{sum}(\mathbf{x}, \boldsymbol{\omega}) \tag{2.38}$$

where

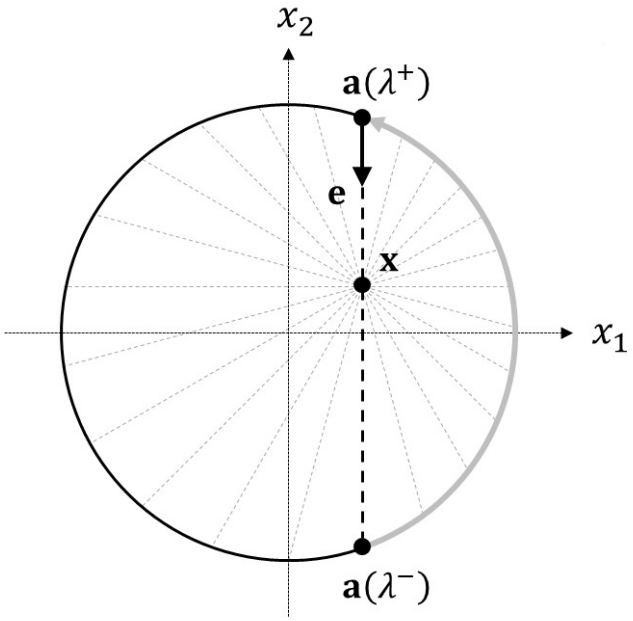
$$\sigma_{sum}(\mathbf{x}, \boldsymbol{\omega}) = \frac{1}{2} \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda_Q^+, \mathbf{x})) \sum_{q=1}^{Q} r_q [\operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda_q^-, \mathbf{x})) - \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda_q^+, \mathbf{x}))] \ . \tag{2.39a}$$

$$= \frac{1}{2}(1 - r_Q) \tag{2.39b}$$

Here, the last equality comes from Pack and Noo [6], and this results in (2.36).

2.2.2 Examples of scanning geometry

Circular fan-beam CT



KAIST

Figure 2.3: FBP reconstruction for circular fan-beam CT.

The circular trajectory in 2D fan-beam CT is defined as

$$\mathbf{a}(\lambda) = (R\cos\lambda, R\sin\lambda)^T, \quad \lambda \in [-\pi, \pi) \ , \tag{2.40}$$

where R is a radius of the circular trajectory. To apply (2.36) to the circular fan-beam CT, we choose the one curve segment $[\lambda^-, \lambda^+]$ for $\mathbf{x}$ among infinite cases of curve segments as Figure 2.3. Then, the filtering direction $\mathbf{e}$ becomes $\boldsymbol{\alpha}(\lambda^+, \mathbf{x})$, the object can be reconstructed as

$$f(\mathbf{x}) = -\mathcal{K}(\mathbf{x}, \boldsymbol{\alpha}(\lambda^+, \mathbf{x}), \lambda^-, \lambda^+) \ . \tag{2.41}$$

There are infinitely possible curve segments $[\lambda^-, \lambda^+]$ for the same $\mathbf{x}$, and as λ^+ is changed, the filtering direction $\mathbf{e} = \boldsymbol{\alpha}(\lambda^+, \mathbf{x})$ is changed either. In the 2D fan-beam case, however, even if the filtering direction $\mathbf{e}$ is changed, for all source angles $\lambda \in [0, 2\pi)$, the filtering is performed only along the detector array. Thus, the filtering process does not require the curve segments, but we still need to select curve segments $[\lambda^-, \lambda^+]$ for $\mathbf{x}$ because the curve segment is necessary for backprojection process in (2.13). As Figure 2.3, we find the $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$ whose positions in x_1-axis are same with $\mathbf{x}$.

Circle-plus-line cone-beam CT

We define the configuration of the circle-plus-line source trajectory as

$$\mathbf{a}(\lambda) = \begin{cases} \left(R, 0, \frac{H}{\phi}\lambda\right)^T, & \lambda \in [-\phi, \phi], \quad 0 \le \phi < 2\pi \\ (R\cos\lambda, R\sin\lambda, 0)^T, & \lambda \in [2\pi, 3\pi + \psi], \quad 0 < \psi < \pi \end{cases} \tag{2.42}$$

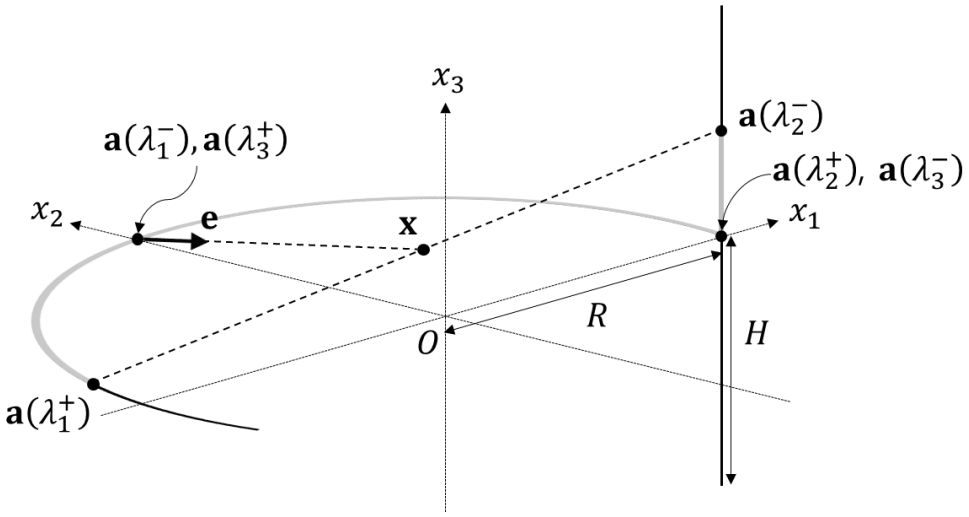


Figure 2.4: FBP reconstruction for circle-plus-line cone-beam CT.

where R is a radius of the circular trajectory and H is a half length of the line trajectory. The whole trajectory is shown in Figure 2.4. For a given $\mathbf{x}$, we can find the R-line connecting two points, of which one is on the circle trajectory part and the other is on the line trajectory part. The curve segments are indicated with gray color in Figure 2.4. The $f(\mathbf{x})$ can be reconstructed using R-line, but for the better detector efficiency, we can use M-line by splitting the circle curve segment into two parts. Now, we have the series of curve segments $\{[\lambda_1^-, \lambda_1^+], [\lambda_2^-, \lambda_2^+], [\lambda_3^-, \lambda_3^+]\}$, and the filtering direction is $\boldsymbol{\alpha}(\lambda_3^+, \mathbf{x})$. Then, the reconstruction can be accomplished as

$$f(\mathbf{x}) = \mathcal{K}(\mathbf{x}, \boldsymbol{\alpha}(\lambda_3^+, \mathbf{x}), \lambda_1^-, \lambda_1^+) - \mathcal{K}(\mathbf{x}, \boldsymbol{\alpha}(\lambda_3^+, \mathbf{x}), \lambda_2^-, \lambda_2^+) - \mathcal{K}(\mathbf{x}, \boldsymbol{\alpha}(\lambda_3^+, \mathbf{x}), \lambda_3^-, \lambda_3^+) \ . \tag{2.43}$$

Saddle cone-beam CT

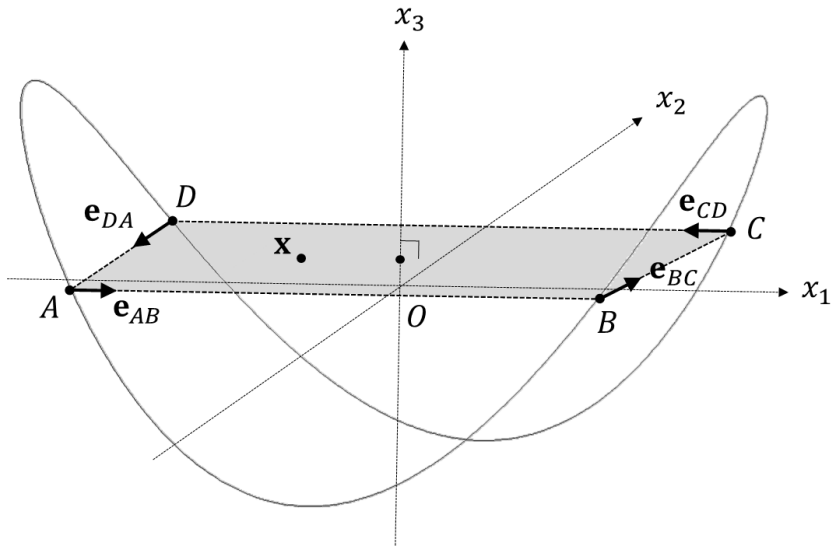


Figure 2.5: FBP reconstruction for saddle cone-beam CT.

The saddle trajectory is defined as [7]

$$\mathbf{a}(\lambda) = (R\cos\lambda, R\sin\lambda, H\cos 2\lambda)^T, \quad \lambda \in [0, 2\pi) \ . \tag{2.44}$$

Here, R is a radius of a horizontal rotation, and H is a amplitude of vertical fluctuations. Figure 2.5 shows the saddle trajectory. With this scanning trajectory, the reconstruction at $\mathbf{x}$ given by [39]

$$f(\mathbf{x}) = \frac{1}{2}\Big(\mathcal{K}(\mathbf{x}, \mathbf{e}_{AB}, \lambda_A, \lambda_B) + \mathcal{K}(\mathbf{x}, \mathbf{e}_{BC}, \lambda_B, \lambda_C) + \mathcal{K}(\mathbf{x}, \mathbf{e}_{CD}, \lambda_C, \lambda_D) + \mathcal{K}(\mathbf{x}, \mathbf{e}_{DA}, \lambda_D, \lambda_A)\Big) \ . \tag{2.45}$$

If the reconstruction point is at $\mathbf{x}$, the plane containing $\mathbf{x}$ is determined, which is perpendicular to x_3 axis, and four intersection points with the saddle trajectory are made. Then, the reconstruction at $\mathbf{x}$ is given by backprojecting filtration of derivative images with filtering direction $\mathbf{e}_{AB}$ for scanning trajectory from A to B, filtering direction $\mathbf{e}_{BC}$ for scanning trajectory from B to C, filtering direction $\mathbf{e}_{CD}$ for scanning trajectory from C to D, and filtering direction $\mathbf{e}_{DA}$ for scanning trajectory from D to A. All filtering directions are shown in Figure 2.5.

Helical cone-beam CT

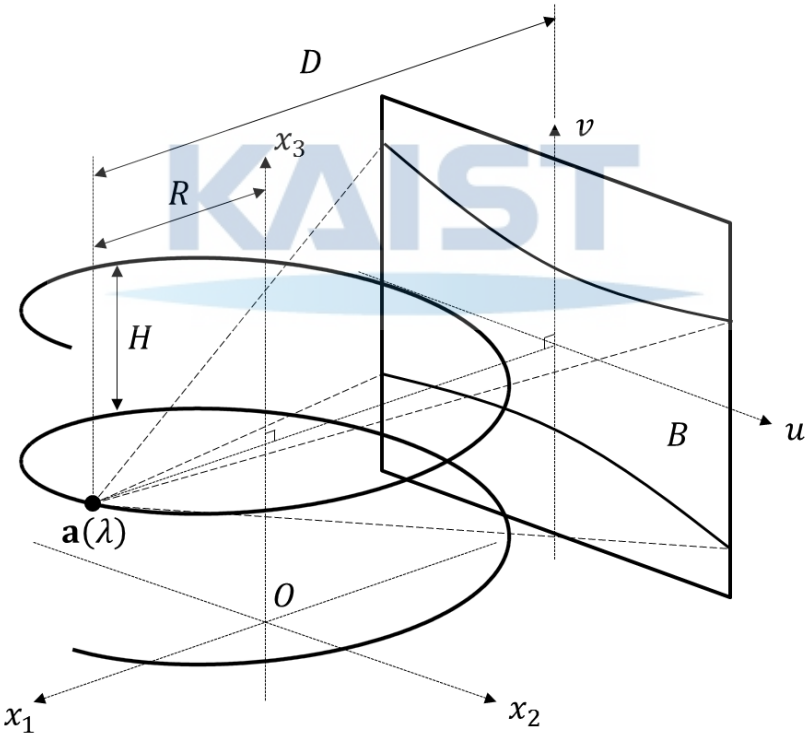


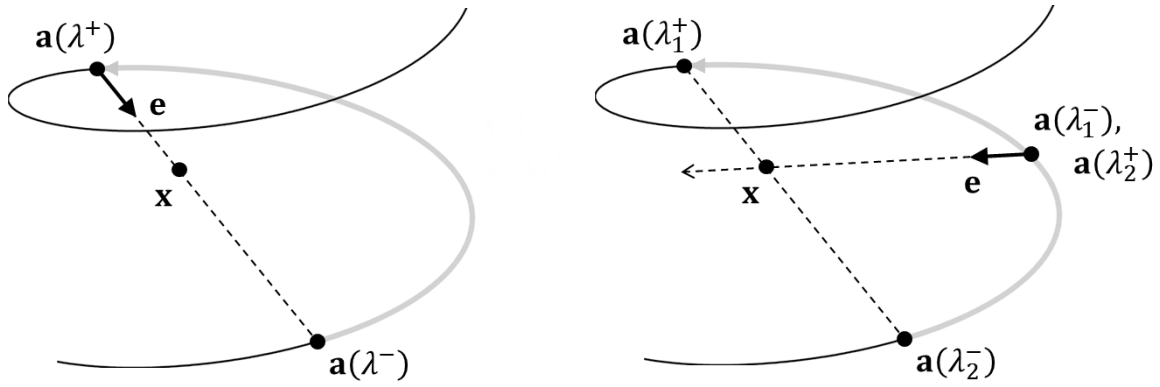
Figure 2.6: Helical cone-beam CT.

Figure 2.6 illustrates a source-detector system of a helical cone-beam CT. A helical source trajectory is expressed as

$$\mathbf{a}(\lambda) = \left(R\cos\lambda, R\sin\lambda, \frac{H}{2\pi}\lambda\right)^T, \quad \lambda \in [\lambda_{\min}, \lambda_{\max}] \ . \tag{2.46}$$

Here, R is the distance from the source to the rotation axis, and H is the pitch of the helix. The pitch of the helix refers to the translation distance along the x_3-axis during one rotation.

In helical trajectory, to reconstruct $\mathbf{x}$ inside the helix, we choose the one curve segment $[\lambda^-, \lambda^+]$ as Figure 2.7(a). Unlike the circular trajectory, the curve segment $[\lambda^-, \lambda^+]$ for $\mathbf{x} = (x_1, x_2, x_3)$ inside the



(a) (b)

Figure 2.7: FBP reconstruction for helical cone-beam CT.

helix is unique where $|\lambda^+ - \lambda^-| \leq 2\pi$, and the relationship can be explained through following equations:

$$x_1 = R[(1-t)\cos\lambda^- + t\cos\lambda^+] \,, \tag{2.47a}$$

$$x_2 = R[(1-t)\sin\lambda^- + t\sin\lambda^+] \,, \tag{2.47b}$$

$$x_3 = \frac{H}{2\pi}[(1-t)\lambda^- + t\lambda^+] \,. \tag{2.47c}$$

Here, $t \in [0,1]$ represents the relative position between $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$.

After finding the curve segment $[\lambda^-, \lambda^+]$ for $\mathbf{x}$, the object can be reconstructed by (2.36). As Figure 2.7(a), we can use the one curve segment $[\lambda^-, \lambda^+]$, then the filtering direction becomes $\mathbf{e} = \boldsymbol{\alpha}(\lambda^+, \mathbf{x})$. Otherwise, we can split the curve segment $[\lambda^-, \lambda^+]$ into two curve segments $[\lambda_1^-, \lambda_1^+]$ and $[\lambda_2^-, \lambda_2^+]$ as Figure 2.7(b), then the filtering direction becomes $\mathbf{e} = \boldsymbol{\alpha}(\lambda_2^+, \mathbf{x})$. Both filtering directions can be used for reconstruction of $\mathbf{x}$, but depending on the filtering direction, the detector efficiency can be altered [6].

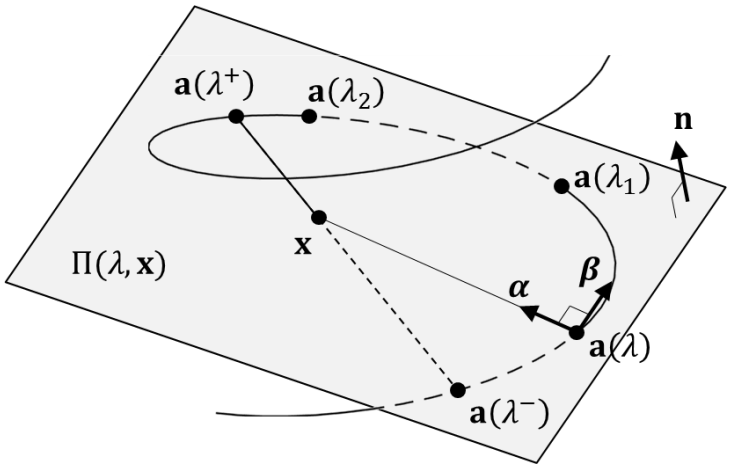


Figure 2.8: Katsevich reconstruction for helical cone-beam CT.

In (2.36), the filtering direction is chosen as $\mathbf{e} = \boldsymbol{\alpha}(\lambda^+, \mathbf{x})$, but it is not the only way to choose the filtering direction. The filtering direction $\mathbf{e}$ in (2.13) finally affect only the $\boldsymbol{\beta}(\lambda, \mathbf{x}, \mathbf{e})$ in (2.16). For the

helical cone-beam CT, Katsevich suggested different $\boldsymbol{\beta}$ which depends on only $(\lambda, \mathbf{x})$ not $\mathbf{e}$ [1]. Before explaining $\boldsymbol{\beta}$, we need to introduce the vector $\mathbf{n}$.

Suppose that λ, λ_1, and λ_2 have relationship such as

$$\lambda_1 = \frac{1}{2}(\lambda + \lambda_2), \quad \lambda - 2\pi < \lambda_2 < \lambda + 2\pi \ . \tag{2.48}$$

Then, the filtering plane $\Pi(\lambda, \mathbf{x})$ can be defined as the plane including $\mathbf{x}$, $\mathbf{a}(\lambda)$ and intersecting the helix at $\mathbf{a}(\lambda_1)$, $\mathbf{a}(\lambda_2)$ as Figure 2.8. The vector $\mathbf{n}$ is the normal vector of filtering plane $\Pi(\lambda, \mathbf{x})$, and it can be obtained as

$$\mathbf{n}(\lambda, \lambda_1) = \frac{\big(\mathbf{a}(\lambda_1) - \mathbf{a}(\lambda)\big) \times \big(\mathbf{a}(\lambda_2) - \mathbf{a}(\lambda)\big)}{\big\|\big(\mathbf{a}(\lambda_1) - \mathbf{a}(\lambda)\big) \times \big(\mathbf{a}(\lambda_2) - \mathbf{a}(\lambda)\big)\big\|} \operatorname{sgn}(\lambda_1 - \lambda), \qquad 0 < |\lambda_1 - \lambda| < \pi \ , \tag{2.49a}$$

$$\mathbf{n}(\lambda, \lambda_1) = \frac{\dot{\mathbf{a}}(\lambda) \times \ddot{\mathbf{a}}(\lambda)}{\|\dot{\mathbf{a}}(\lambda) \times \ddot{\mathbf{a}}(\lambda)\|}, \qquad \lambda_1 = \lambda \ . \tag{2.49b}$$

Here, the filtering plane $\Pi(\lambda, \mathbf{x})$ is unique for $\mathbf{x}$, and it means λ_2 is unique for $\mathbf{x}$. Thus, $\mathbf{n}(\lambda, \lambda_1)$ can be denoted as $\mathbf{n}(\lambda, \mathbf{x})$. As Figure 2.1, the $\boldsymbol{\alpha}$ is for the unit vector from the source $\mathbf{a}(\lambda)$ to reconstruction point $\mathbf{x}$, and the $\boldsymbol{\beta}$ is for the orthonormal vector to α on the filtering plane $\Pi(\lambda, \mathbf{x})$. Therefore, $\boldsymbol{\beta} = \boldsymbol{\alpha} \times \mathbf{n}$. In summary, the Katsevich formula can be following:

$$f(\mathbf{x}) = -\frac{1}{2\pi^2} \int_{\lambda^-}^{\lambda^+} d\lambda \ \frac{1}{\|\mathbf{x} - \mathbf{a}(\lambda)\|} g(\lambda, \mathbf{x}) \tag{2.50}$$

where

$$g(\lambda, \mathbf{x}) = \int_{-\pi}^{\pi} d\gamma \ \frac{1}{\sin\gamma} \ \frac{\partial}{\partial \nu} D_f(\mathbf{a}(\nu), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma))\Big|_{\nu=\lambda} \tag{2.51}$$

with

$$\boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma) = \cos\gamma \boldsymbol{\alpha}(\lambda, \mathbf{x}) + \sin\gamma \boldsymbol{\beta}(\lambda, \mathbf{x}) \tag{2.52}$$

and

$$\boldsymbol{\alpha}(\lambda, \mathbf{x}) = \frac{\mathbf{x} - \mathbf{a}(\lambda)}{\|\mathbf{x} - \mathbf{a}(\lambda)\|} \ , \quad \boldsymbol{\beta}(\lambda, \mathbf{x}) = \boldsymbol{\alpha}(\lambda, \mathbf{x}) \times \mathbf{n}(\lambda, \mathbf{x}) \ . \tag{2.53}$$

2.3 Backprojection filtration method

In this section, we review a Fourier domain separability of the DBP data that gives us insight in designing the new algorithm.

2.3.1 Fourier analysis of differentiated backprojection

The main part of this sub-section is not novel because it has already appeared in [6, 9, 10] to derive the BPF algorithms. Nevertheless, we repeat it here for self-inclusiveness.

Specifically, for a given object function $f(\mathbf{x})$, its d-dimensional inverse Fourier transform from the spectrum $\hat{f}(\boldsymbol{\omega})$ is given by

$$f(\mathbf{x}) = \frac{1}{(2\pi)^d} \int d\boldsymbol{\omega} \ \hat{f}(\boldsymbol{\omega}) e^{j\boldsymbol{\omega}\cdot\mathbf{x}} \tag{2.54}$$

where $d = 2, 3$ are for 2D and 3D tomography problems, respectively. In addition, if $\boldsymbol{\theta}$ denote the d-dimensional unit vector and $\mathbf{a} \in \mathbb{R}^d$ is the source location, then the x-ray transform D_f maps a function on $\mathbb{R}^d$ into the set of its line integrals

$$D_f(\mathbf{a}, \boldsymbol{\theta}) = \int_0^{\infty} dt \ f(\mathbf{a} + t\boldsymbol{\theta}) \ . \tag{2.55}$$

For a given object function $f(\mathbf{x})$ and a given source trajectory $\mathbf{a}(\lambda), \lambda \in [\lambda_{\min}, \lambda_{\max}]$, we now define the DBP:

$$g(\mathbf{x}) = -\int_{\lambda^-}^{\lambda^+} d\lambda \; \frac{1}{\|\mathbf{x}-\mathbf{a}(\lambda)\|} \; \frac{\partial}{\partial \mu} D_f(\mathbf{a}(\mu), \boldsymbol{\theta})\Big|_{\mu=\lambda} \tag{2.56}$$

where $[\lambda^-, \lambda^+] \subset [\lambda_{\min}, \lambda_{\max}]$ denotes the appropriate intervals from the source segments between $\lambda_{\min}$ and $\lambda_{\max}$, and $1/\|\mathbf{x}-\mathbf{a}(\lambda)\|$ denotes the distance weighting. Then, we have the following theorem holds.

Theorem 2.3.1. *Let the source trajectory $\mathbf{a}(\lambda), \lambda \in [\lambda^-, \lambda^+]$ have no discontinuities. Suppose, furthermore, $\mathbf{x}$ is on the connecting line of the two source positions $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$. Then, the DBP data in* (2.56) *can be represented as*

$$g(\mathbf{x}) = -\frac{1}{(2\pi)^d} \int d\boldsymbol{\omega} \; \hat{f}(\boldsymbol{\omega}) j \operatorname{sgn}(\boldsymbol{\omega} \cdot \boldsymbol{\alpha}(\lambda^-, \mathbf{x})) e^{j\boldsymbol{\omega}\cdot\mathbf{x}} \tag{2.57}$$

where

$$\boldsymbol{\alpha}(\lambda^-, \mathbf{x}) = \frac{\mathbf{x}-\mathbf{a}(\lambda^-)}{\|\mathbf{x}-\mathbf{a}(\lambda^-)\|} \; . \tag{2.58}$$

Proof. While it can also be derived from a result in [6, 9, 10], we provide a proof for completeness. The following scaling property for the delta function is useful:

$$\delta(\alpha t) = \frac{\delta(t)}{|\alpha|} \; . \tag{2.59}$$

In addition, let $\mathbf{a}(\lambda) \in \mathbb{R}^d$ denote a vector parameterized with respect to $\lambda \in [\lambda^-, \lambda^+]$. Then, for $\mathbf{x}, \boldsymbol{\omega} \in \mathbb{R}^d$, the delta function scaling property provides us

$$\delta(\boldsymbol{\omega} \cdot (\mathbf{x} - \mathbf{a}(\lambda))) = \sum_k \frac{\delta(\lambda - \lambda_k)}{|\boldsymbol{\omega} \cdot \dot{\mathbf{a}}(\lambda_k)|} \tag{2.60}$$

where $\{\lambda_k\}$ denotes the indices that satisfies

$$\{\lambda \in [\lambda^-, \lambda^+] : \boldsymbol{\omega} \cdot \mathbf{x} = \boldsymbol{\omega} \cdot \mathbf{a}(\lambda)\} \tag{2.61}$$

and $\dot{\mathbf{a}}(\lambda)$ denotes the derivative of $\mathbf{a}(\lambda)$ with respect to λ.

Now, we are ready to show the proof. Using the x-ray transform (2.55), we have

$$\begin{aligned}
g(\mathbf{x}) &= -\int_{\lambda^-}^{\lambda^+} d\lambda \; \frac{1}{\|\mathbf{x}-\mathbf{a}(\lambda)\|} \; \frac{\partial}{\partial \mu} D_f(\mathbf{a}(\mu), \boldsymbol{\theta})\Big|_{\mu=\lambda} && \text{(2.62a)} \\
&= -\int_{\lambda^-}^{\lambda^+} d\lambda \; \frac{1}{\|\mathbf{x}-\mathbf{a}(\lambda)\|} \int_{-\infty}^{\infty} dt \; \frac{1}{(2\pi)^d} \int d\boldsymbol{\omega} \; \frac{\partial}{\partial \mu} e^{j\boldsymbol{\omega}\cdot(\mathbf{a}(\mu)+t\boldsymbol{\theta})}\Big|_{\mu=\lambda} \hat{f}(\boldsymbol{\omega}) && \text{(2.62b)} \\
&= -\int_{\lambda^-}^{\lambda^+} d\lambda \; \int_{-\infty}^{\infty} dt \; \frac{1}{(2\pi)^d} \int d\boldsymbol{\omega} \; \frac{1}{\|\mathbf{x}-\mathbf{a}(\lambda)\|} j\dot{\mathbf{a}}(\lambda) \cdot \boldsymbol{\omega} e^{j\boldsymbol{\omega}\cdot(\mathbf{a}(\lambda)+t\boldsymbol{\theta})} \hat{f}(\boldsymbol{\omega}) && \text{(2.62c)} \\
&= -\int_{\lambda^-}^{\lambda^+} d\lambda \; \frac{1}{(2\pi)^d} \int d\boldsymbol{\omega} \; \frac{1}{\|\mathbf{x}-\mathbf{a}(\lambda)\|} j\dot{\mathbf{a}}(\lambda) \cdot \boldsymbol{\omega} e^{j\boldsymbol{\omega}\cdot\mathbf{a}(\lambda)} \hat{f}(\boldsymbol{\omega}) \int_{-\infty}^{\infty} dt \; e^{jt\boldsymbol{\omega}\cdot\boldsymbol{\theta}} && \text{(2.62d)} \\
&= -\int_{\lambda^-}^{\lambda^+} d\lambda \; \frac{1}{(2\pi)^d} \int d\boldsymbol{\omega} \; \frac{1}{\|\mathbf{x}-\mathbf{a}(\lambda)\|} j\dot{\mathbf{a}}(\lambda) \cdot \boldsymbol{\omega} e^{j\boldsymbol{\omega}\cdot\mathbf{a}(\lambda)} \hat{f}(\boldsymbol{\omega}) \delta(\boldsymbol{\omega}\cdot\boldsymbol{\theta}) \; . && \text{(2.62e)}
\end{aligned}$$

If we let

$$\boldsymbol{\theta} = \frac{\mathbf{x}-\mathbf{a}(\lambda)}{\|\mathbf{x}-\mathbf{a}(\lambda)\|} \; , \tag{2.63}$$

then

$$g(\mathbf{x}) = -\frac{1}{(2\pi)^d}\int d\boldsymbol{\omega}\ \hat{f}(\boldsymbol{\omega})\int_{\lambda^-}^{\lambda^+} d\lambda\ e^{j\boldsymbol{\omega}\cdot\mathbf{a}(\lambda)} j(\boldsymbol{\omega}\cdot\dot{\mathbf{a}}(\lambda))\delta(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{a}(\lambda))) \tag{2.64a}$$

$$= -\frac{1}{(2\pi)^d}\int d\boldsymbol{\omega}\ \hat{f}(\boldsymbol{\omega})\int_{\lambda^-}^{\lambda^+} d\lambda\ e^{j\boldsymbol{\omega}\cdot\mathbf{a}(\lambda)} j(\boldsymbol{\omega}\cdot\dot{\mathbf{a}}(\lambda))\sum_k \frac{\delta(\lambda-\lambda_k)}{|\boldsymbol{\omega}\cdot\dot{\mathbf{a}}(\lambda_k)|} \tag{2.64b}$$

where we use (2.60). Since $\{\lambda_k\}$ is the set of λ that satisfies $\boldsymbol{\omega}\cdot\mathbf{x} = \boldsymbol{\omega}\cdot\mathbf{a}(\lambda)$, we have

$$g(\mathbf{x}) = -\frac{1}{(2\pi)^d}\int d\boldsymbol{\omega}\ \hat{f}(\boldsymbol{\omega})e^{j\boldsymbol{\omega}\cdot\mathbf{x}} j\sum_k \operatorname{sgn}(\boldsymbol{\omega}\cdot\dot{\mathbf{a}}(\lambda_k))\ . \tag{2.65}$$

In addition, according to Pack and Noo [6], if the source trajectory does not have discontinuities, we have

$$\sum_k \operatorname{sgn}(\boldsymbol{\omega}\cdot\dot{\mathbf{a}}(\lambda_k)) = \frac{1}{2}[\operatorname{sgn}(\boldsymbol{\omega}\cdot\boldsymbol{\alpha}(\lambda^-,\mathbf{x})) - \operatorname{sgn}(\boldsymbol{\omega}\cdot\boldsymbol{\alpha}(\lambda^+,\mathbf{x}))] \tag{2.66}$$

where

$$\boldsymbol{\alpha}(\lambda^-,\mathbf{x}) = \frac{\mathbf{x}-\mathbf{a}(\lambda^-)}{\|\mathbf{x}-\mathbf{a}(\lambda^-)\|}\ , \quad \boldsymbol{\alpha}(\lambda^+,\mathbf{x}) = \frac{\mathbf{x}-\mathbf{a}(\lambda^+)}{\|\mathbf{x}-\mathbf{a}(\lambda^+)\|}\ . \tag{2.67}$$

This can be further simplified as follows if $\mathbf{a}(\lambda^+)$ and $\mathbf{a}(\lambda^-)$ are on the opposite sides relative to $\mathbf{x}$,

$$\sum_k \operatorname{sgn}(\boldsymbol{\omega}\cdot\dot{\mathbf{a}}(\lambda_k)) = \operatorname{sgn}(\boldsymbol{\omega}\cdot\boldsymbol{\alpha}(\lambda^-,\mathbf{x})) \tag{2.68}$$

since $\boldsymbol{\alpha}(\lambda^-,\mathbf{x}) = -\boldsymbol{\alpha}(\lambda^+,\mathbf{x})$. This concludes the proof. □

Note that (2.57) implies separability in the Fourier domain. The connecting line between the two sources position is often called a *chord line* [6, 9, 10]. If we choose a chord line which is parallel to $\mathbf{e} := \mathbf{e}(\lambda^-,\mathbf{x}) = \boldsymbol{\alpha}(\lambda^-,\mathbf{x})$ as a coordinate axis in the frequency domain, then we can find the $(d-1)$-dimensional orthonormal basis $E^\perp$ such that $V = [\mathbf{e}, E^\perp]$ consists of the basis for the local coordinate system at $\mathbf{a}(\lambda^-)$. Now, let $[\omega, \boldsymbol{\omega}_{E^\perp}]$ and $[x, \mathbf{x}_{E^\perp}]$ denote the corresponding new coordinates in frequency and spatial domain, respectively, with respect to the local coordinate system. Then, (2.57) can be further simplified as

$$g(\mathbf{x}) = -\frac{1}{(2\pi)^{(d-1)}}\int d\boldsymbol{\omega}_{E^\perp}\ e^{j\boldsymbol{\omega}_{E^\perp}\cdot\mathbf{x}_{E^\perp}}\frac{1}{2\pi}\int d\omega\ \hat{f}(\omega,\boldsymbol{\omega}_{E^\perp}) j\operatorname{sgn}(\omega)e^{j\omega x} \tag{2.69a}$$

$$= -\frac{1}{2\pi}\int d\omega\ f(\omega,\mathbf{x}_{E^\perp}) j\operatorname{sgn}(\omega)e^{j\omega x}\ , \tag{2.69b}$$

which becomes a separable 1D problem for each coordinate variable $\mathbf{x}_{E^\perp}$. Now, let

$$g_{\mathbf{x}_{E^\perp}}(x) := g(\mathbf{x}), \quad f_{\mathbf{x}_{E^\perp}}(x) := f(\mathbf{x})\ . \tag{2.70}$$

Since $-j\operatorname{sgn}(\omega)$ is the Fourier transform of $1/(\pi x)$, then we have

$$g_{\mathbf{x}_{E^\perp}}(x) = \frac{1}{\pi}\int dx'\ \frac{1}{x-x'} f_{\mathbf{x}_{E^\perp}}(x') := \mathcal{H}_{\mathcal{L}}\{f_{\mathbf{x}_{E^\perp}}(x)\} \tag{2.71}$$

where $\mathcal{H}_{\mathcal{L}}$ denotes the Hilbert transform along the chord line $\mathcal{L}$. Since $\mathcal{H}\mathcal{H} = -I$, we have

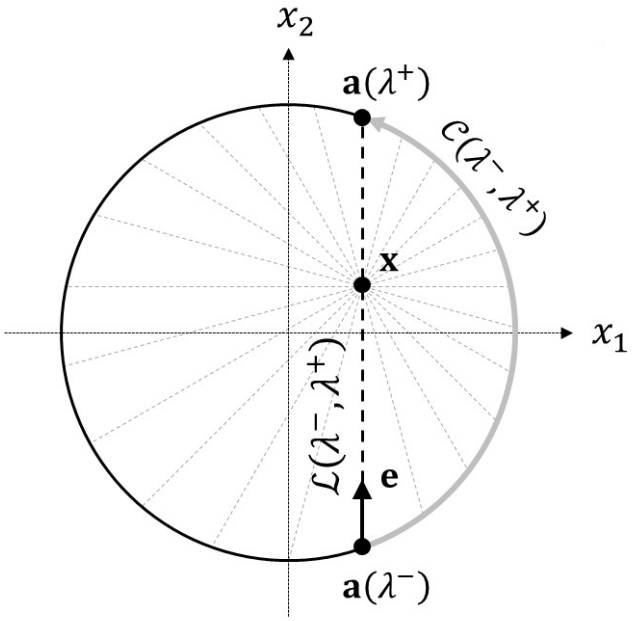
$$f_{\mathbf{x}_{E^\perp}}(x) = -\mathcal{H}_{\mathcal{L}}\{g_{\mathbf{x}_{E^\perp}}(x)\}\ , \tag{2.72}$$

which corresponds to the BPF method that recovers the object on each chord line by taking the inverse Hilbert transform of the DBP data [9, 10].

The followings are already well-known acquisition trajectories, where one or more sets of chord lines exist that cover the entire object domain.

2.3.2 Examples of scanning geometry

Circular fan-beam CT



KAIST

Figure 2.9: BPF reconstruction for circular fan-beam CT.

In 2D circular fan-beam CT, the chord line $\mathcal{L}$ that passes through $\mathbf{x}$ inside the circle is not unique; hence, we can choose infinitely many filtering lines shown in Figure 2.9. For example, we can use a set of filtering lines that are parallel to each other to simplify the implementation. If we make a vertical chord line go through $\mathbf{x}$, we need to find $\lambda^-(\mathbf{x})$ and $\lambda^+(\mathbf{x})$ for DBP. Similarly, we can use horizontal chord lines. Here, the segment of the chord line between $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$ is denoted as $\mathcal{L}(\lambda^-,\lambda^+)$, and the segment of the circle between $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$ is denoted as $\mathcal{C}(\lambda^-,\lambda^+)$. After backprojecting differentiated data from $\lambda^-(\mathbf{x})$ to $\lambda^+(\mathbf{x})$, the unknown signals can be reconstructed by (2.72) chord line by chord line.

Circle-plus-line cone-beam CT

We already defined the circle-plus-line source trajectory as (2.42) in section 2.2.2. In the circle-plus-line CT, the chord line $\mathcal{L}$ passing through a point $\mathbf{x}$ can be determined by $\mathbf{a}(\lambda^-)$ on the line trajectory part and $\mathbf{a}(\lambda^+)$ on the circle trajectory part. Then, the segment of the chord line between $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$ is denoted as $\mathcal{L}(\lambda^-,\lambda^+)$, and the segment of the trajectory between $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$ is denoted as $\mathcal{C}(\lambda^-,\lambda^+)$. The trajectory segment $\mathcal{C}(\lambda^-,\lambda^+)$ is indicated with gray color in Figure 2.10. Now, to reconstruct the point $\mathbf{x}$ on the chord line $\mathcal{L}(\lambda^-,\lambda^+)$, differentiated images of projections at source position from $\mathbf{a}(\lambda^-)$ to $\mathbf{a}(0)$ and from $\mathbf{a}(2\pi)$ to $\mathbf{a}(\lambda^+)$ are required, and the differentiation must be computed in this direction. After that, the differentiated projections are backprojected on the chord line. The BPF algorithm requires filtration along the chord line after backprojection, thus we backproject the differentiated images on the sampled points along the chord lines instead of the Cartesian grid. This way makes the filtration step simple, but the regridding process is required. The rebinned space can cover

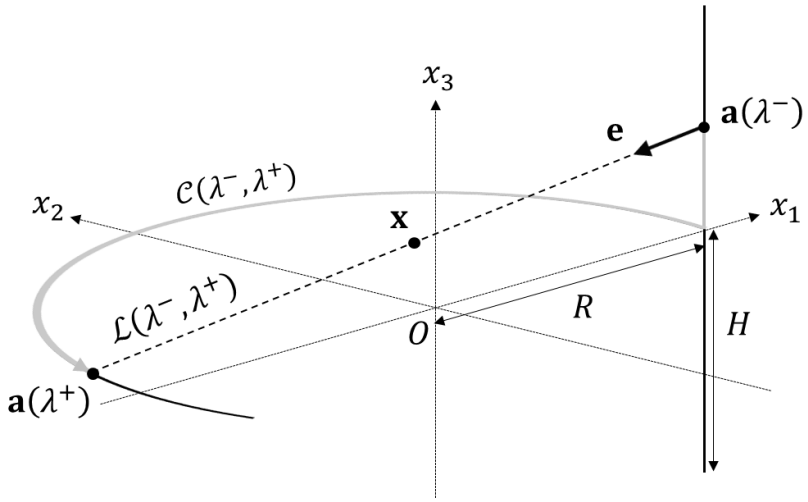


Figure 2.10: BPF reconstruction for circle-plus-line cone-beam CT.

all ROI varying $\mathbf{a}(\lambda^-)$ along the line trajectory and varying $\mathbf{a}(\lambda^+)$ along the circle trajectory. After the backprojection on the chord line, the object can be reconstructed by the inverse Hilbert transform as (2.72). The filtration result on chord lines is rebinned into regularly grided Cartesian coordinate space.

Helical cone-beam CT

KAIST

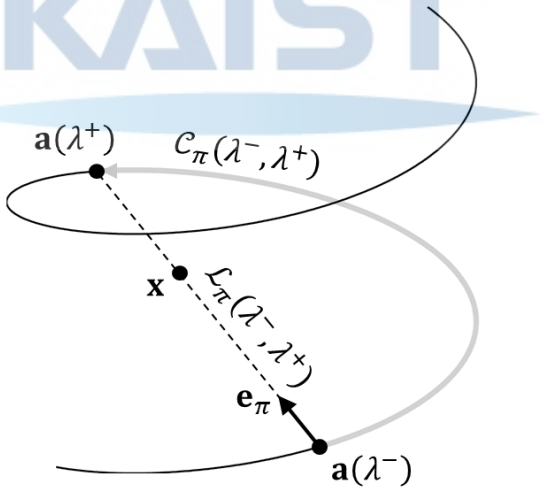


Figure 2.11: BPF reconstruction for helical cone-beam CT

As shown in Figure 2.11, a helical source trajectory is expressed as

$$\mathbf{a}(\lambda) = \left(R\cos\lambda, R\sin\lambda, \frac{H}{2\pi}\lambda \right)^T . \tag{2.73}$$

Here, λ is the rotation angle of the x-ray source which is restricted to $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, R is the distance from the source to the rotation axis, and H is the pitch of the helix. The pitch of the helix refers to the translation distance along the x_3-axis during one rotation.

In helical cone-beam CT, it is necessary to introduce PI line geometry. Specifically, if two points $\mathbf{a}(\lambda^-)$, $\mathbf{a}(\lambda^+)$ on helical trajectory are selected where $|\lambda^+ - \lambda^-| \leq 2\pi$, the line connecting these two

points is called a PI line, which is the particular name for a chord line in helical trajectory. In addition, the segment of the PI line between $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$ is denoted as $\mathcal{L}_\pi(\lambda^-, \lambda^+)$, and the segment of the helix between $\mathbf{a}(\lambda^-)$ and $\mathbf{a}(\lambda^+)$ is denoted as $\mathcal{C}_\pi(\lambda^-, \lambda^+)$ (Figure 2.11). This PI line of λ^- and λ^+ is unique for a given point $\mathbf{x} = (x_1, x_2, x_3)$ inside the helix, and the relationship can be explained through (2.47). Unlike the FBP reconstruction, we do not need to find the PI line for a given $\mathbf{x}$. Instead, the PI lines are made along the helix to cover the ROI [14], and the reconstruction points are sampled along each PI line. Then, the differentiated projections of the helix segment $\mathcal{C}_\pi(\lambda^-, \lambda^+)$ are backprojected on the PI line segment $\mathcal{L}_\pi(\lambda^-, \lambda^+)$. By taking the inverse Hilbert transform of the DBP signal along the PI line, the object can be reconstructed. Lastly, The reconstruction result on PI lines is rebinned into regularly grided Cartesian coordinate space [12].

2.4 Implementation

In this chapter, we explain how to implement the FBP and BPF algorithm with digital projection images.

2.4.1 Filtered backprojection type

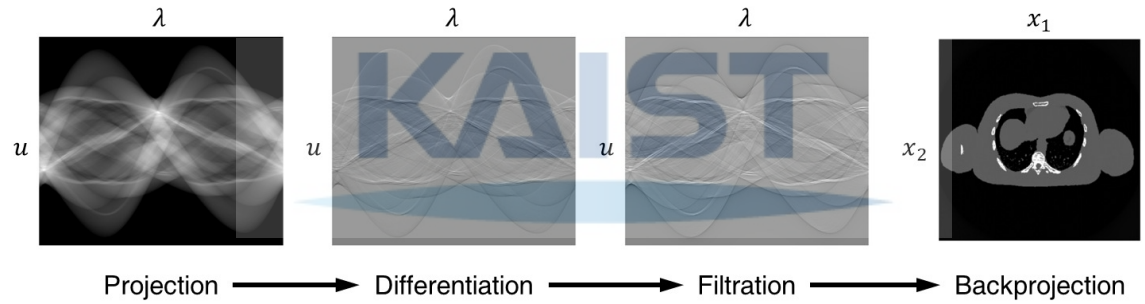


Figure 2.12: Total process of FBP reconstruction for fan-beam CT.

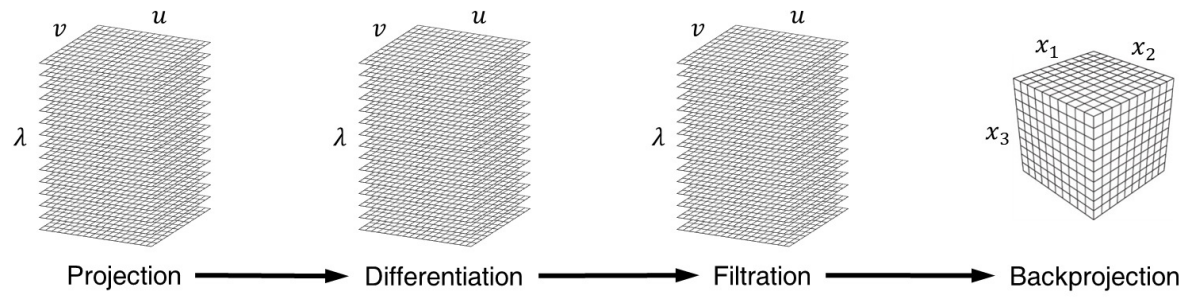


Figure 2.13: Total process of FBP reconstruction for cone-beam CT.

The total process of the FBP-type reconstruction is shown in Figure 2.12 and Figure 2.13 for fan-beam and cone-beam cases, respectively. In both cases, the first process is differentiation of projections with respect to the source trajectory. Then, differentiated signal is filtered with a particular filtering direction. After backprojecting the filtered signal on the object space, the object is finally reconstructed. Now, we explain more details of each process.

Differentiation

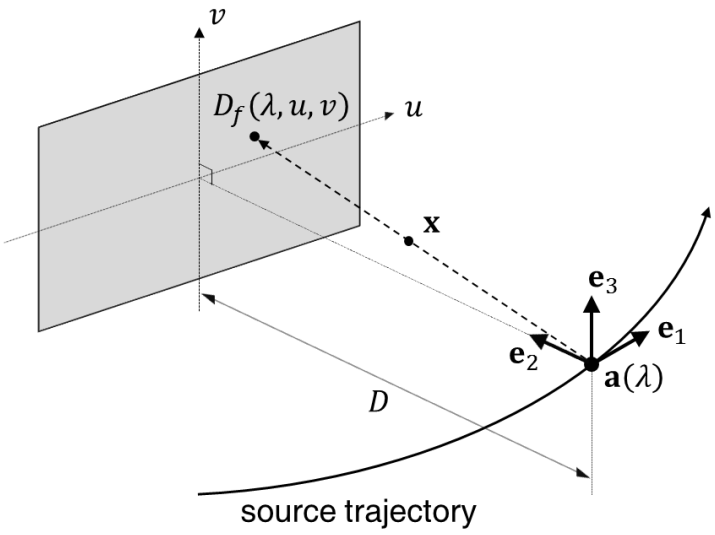


Figure 2.14: Cone-beam projection.

In the FBP method, the differentiation part in from (2.13) to (2.16) is

$$D'_f(\mathbf{a}(\lambda), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e})) = \frac{\partial}{\partial \mu} D_f(\mathbf{a}(\mu), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e}))\Big|_{\mu=\lambda} . \tag{2.74}$$

Here, the rotation angle parameter of the source location $\mathbf{a}$ and the projecting direction $\boldsymbol{\theta}$ are different such as μ and λ, and the differentiation is performed only with respect to μ. It means that the differentiation of the projection function $D_f(\mathbf{a}, \boldsymbol{\theta})$ is only with respect to $\mathbf{a}$ not $\boldsymbol{\theta}$. So, we do not need to consider the four parameters $(\lambda, \mathbf{x}, \gamma, \mathbf{e})$ of $\boldsymbol{\theta}$ during the differentiation procedure. Then, before explaining implementations, we need to change the vector parameters into the scalar parameters. More specifically, in (2.74), the two vector parameters $(\mathbf{a}(\mu), \boldsymbol{\theta})$ of the original projection function D_f are changed into three scalar parameters (μ, u, v) as Figure 2.14. To find relationship between these two parameter sets, we use a rotated coordinate system $\{\mathbf{e}_1(\mu), \mathbf{e}_2(\mu), \mathbf{e}_3(\mu)\}$. For example, in saddle or helical CT, the orthonormal basis vectors of the rotated coordinate system become

$$\mathbf{e}_1(\mu) = (-\sin\mu, \cos\mu, 0)^T , \tag{2.75a}$$

$$\mathbf{e}_2(\mu) = (-\cos\mu, -\sin\mu, 0)^T , \tag{2.75b}$$

$$\mathbf{e}_3(\mu) = (0, 0, 1)^T . \tag{2.75c}$$

With these vectors, (u, v) can be denoted as

$$u = \frac{D\boldsymbol{\theta} \cdot \mathbf{e}_1(\mu)}{\boldsymbol{\theta} \cdot \mathbf{e}_2(\mu)} , \tag{2.76a}$$

$$v = \frac{D\boldsymbol{\theta} \cdot \mathbf{e}_3(\mu)}{\boldsymbol{\theta} \cdot \mathbf{e}_2(\mu)} . \tag{2.76b}$$

For differentiation of D_f, we use the chain rule [40], which is

$$\frac{\partial}{\partial \mu} D_f(\mu, u, v)\Big|_{\mu=\lambda} = \frac{\partial D_f}{\partial \mu} + \frac{\partial D_f}{\partial u}\frac{\partial u}{\partial \mu} + \frac{\partial D_f}{\partial v}\frac{\partial v}{\partial \mu}\Big|_{\mu=\lambda} . \tag{2.77}$$

In (2.77), $^{\partial D_f}/_{\partial \mu}$,$^{\partial D_f}/_{\partial u}$,$^{\partial D_f}/_{\partial v}$ can be calculated directly from projection images, but $^{\partial u}/_{\partial \mu}$ and $^{\partial v}/_{\partial \mu}$ are necessary more derivation. In (2.76), u and v are partially differentiated with respect to μ, which yield

$$\frac{\partial u}{\partial \mu} = \frac{\partial}{\partial \mu}\frac{D\boldsymbol{\theta}\cdot\mathbf{e}_1(\mu)}{\boldsymbol{\theta}\cdot\mathbf{e}_2(\mu)} \ , \tag{2.78a}$$

$$\frac{\partial v}{\partial \mu} = \frac{\partial}{\partial \mu}\frac{D\boldsymbol{\theta}\cdot\mathbf{e}_3(\mu)}{\boldsymbol{\theta}\cdot\mathbf{e}_2(\mu)} \ . \tag{2.78b}$$

Since

$$\frac{\partial}{\partial \mu}\mathbf{e}_1(\mu) = \mathbf{e}_2(\mu) \ , \tag{2.79a}$$

$$\frac{\partial}{\partial \mu}\mathbf{e}_2(\mu) = -\mathbf{e}_1(\mu) \ , \tag{2.79b}$$

$$\frac{\partial}{\partial \mu}\mathbf{e}_3(\mu) = 0 \ , \tag{2.79c}$$

(2.78) will be

$$\frac{\partial u}{\partial \mu} = \frac{D(\boldsymbol{\theta}\cdot\mathbf{e}_2)^2 + D(\boldsymbol{\theta}\cdot\mathbf{e}_1)^2}{(\boldsymbol{\theta}\cdot\mathbf{e}_2)^2} = \frac{D^2+u^2}{D} \ , \tag{2.80a}$$

$$\frac{\partial v}{\partial \mu} = \frac{D(\boldsymbol{\theta}\cdot\mathbf{e}_3)(\boldsymbol{\theta}\cdot\mathbf{e}_1)}{(\boldsymbol{\theta}\cdot\mathbf{e}_2)} = \frac{uv}{D} \ . \tag{2.80b}$$

Consequently, (2.77) can be

$$\frac{\partial}{\partial \mu}D_f(\mu,u,v)\bigg|_{\mu=\lambda} = \frac{\partial D_f}{\partial \mu} + \frac{D^2+u^2}{D}\frac{\partial D_f}{\partial u} + \frac{uv}{D}\frac{\partial D_f}{\partial v}\bigg|_{\mu=\lambda} \ . \tag{2.81}$$

In the circle-plus-line CT, the differentiation of the circle trajectory part is the same with (2.81). However, the line trajectory part has no rotation, so the orthonormal basis vectors of the coordinate system become

$$\mathbf{e}_1(\mu) = (0,1,0)^T \ , \tag{2.82a}$$

$$\mathbf{e}_2(\mu) = (-1,0,0)^T \ , \tag{2.82b}$$

$$\mathbf{e}_3(\mu) = (0,0,1)^T \ . \tag{2.82c}$$

In result, the differentiation for the line trajectory will be

$$\frac{\partial}{\partial \mu}D_f(\mu,u,v)\bigg|_{\mu=\lambda} = \frac{\partial D_f}{\partial \mu}\bigg|_{\mu=\lambda} \ . \tag{2.83}$$

In the circular fan-beam CT, the difference is only the existance of the parameter v, so the differentiation can be implemented as

$$\frac{\partial}{\partial \mu}D_f(\mu,u)\bigg|_{\mu=\lambda} = \frac{\partial D_f}{\partial \mu} + \frac{D^2+u^2}{D}\frac{\partial D_f}{\partial u}\bigg|_{\mu=\lambda} \ . \tag{2.84}$$

The differentiation can be also implemented by the direct scheme to improve resolution and image quality [41].

KAIST

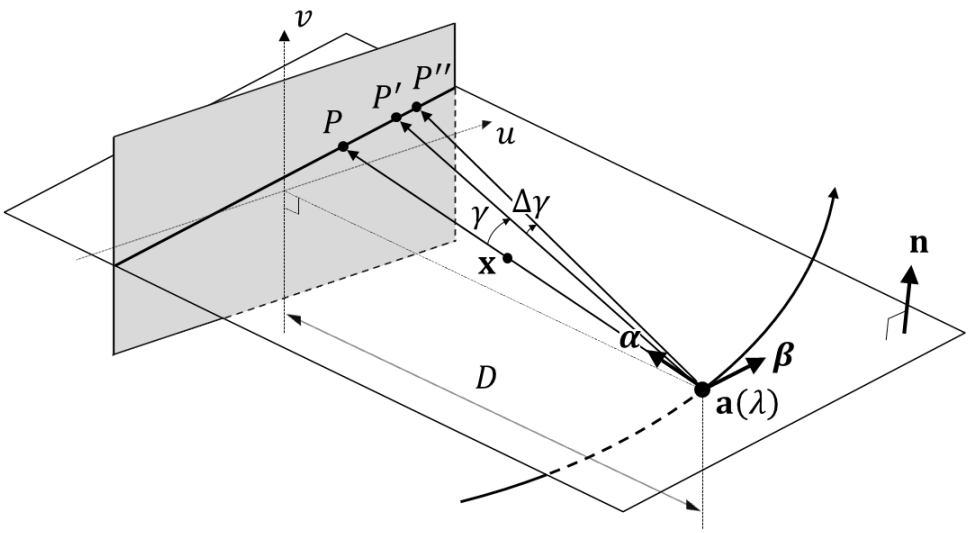


Figure 2.15: Filtering implementation of FBP algorithm.

Filtration

The second process in the FBP reconstruction is the filtration. This part is the most complicated in the FBP method. From (2.13) to (2.16), the filtration part is

$$g(\lambda, \mathbf{x}, \mathbf{e}) = \int_{-\pi}^{\pi} d\gamma \; \frac{1}{\sin\gamma} D'_f(\mathbf{a}(\lambda), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e})) \tag{2.85}$$

$$\boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e}) = \cos\gamma \boldsymbol{\alpha}(\lambda, \mathbf{x}) + \sin\gamma \boldsymbol{\beta}(\lambda, \mathbf{x}, \mathbf{e}) \tag{2.86}$$

$$\boldsymbol{\alpha}(\lambda, \mathbf{x}) = \frac{\mathbf{x} - \mathbf{a}(\lambda)}{\|\mathbf{x} - \mathbf{a}(\lambda)\|} \;, \quad \boldsymbol{\beta}(\lambda, \mathbf{x}, \mathbf{e}) = \frac{\mathbf{e} - \boldsymbol{\alpha}(\lambda, \mathbf{x})(\mathbf{e} \cdot \boldsymbol{\alpha}(\lambda, \mathbf{x}))}{\|\mathbf{e} - \boldsymbol{\alpha}(\lambda, \mathbf{x})(\mathbf{e} \cdot \boldsymbol{\alpha}(\lambda, \mathbf{x}))\|} \;. \tag{2.87}$$

For a given $\mathbf{e}$, the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are determined, and then we can define the filtering plane including the points $\mathbf{x}$, $\mathbf{a}(\lambda)$ and perpendicular to $\mathbf{n} = -\boldsymbol{\alpha} \times \boldsymbol{\beta}$ as Figure 2.15. This filtering plane and the detector plane make the intersection line, and we call this line a filtering line. Since $\boldsymbol{\theta} = \cos\gamma\boldsymbol{\alpha} + \sin\gamma\boldsymbol{\beta}$, the position of $D'_f(\mathbf{a}(\lambda), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma = 0, \mathbf{e}))$ is the P, and for $\gamma \in (-\pi, \pi)$, the position of $D'_f(\mathbf{a}(\lambda), \boldsymbol{\theta}(\lambda, \mathbf{x}, \gamma, \mathbf{e}))$ moves along the filtering line such as P'. Therefore, we can realize that the filtering integral of (2.85) is performed along this filtering line.

Now, for filtration, we need to find the equation of the filtering line with respect to the detector parameter (u, v). The filtering line is determined by the reconstruction point $\mathbf{x}$, so we need to find the family of the filtering lines for all $\mathbf{x}$. The family of the filtering lines is differed by the source trajectory and the filtering direction, but the fundamental idea to find the filtering line is almost same. First, the indexing parameter of the filtering line is chosen. Usually, when the source position is

$$\mathbf{a}(\lambda) = (a_1(\lambda), a_2(\lambda), a_3(\lambda))^T \;, \tag{2.88}$$

the $\bar{x}_3$ of the point $\bar{\mathbf{x}}$ along the x_3-axis, which is

$$\bar{\mathbf{x}} = (0, 0, a_3(\lambda) + \bar{x}_3)^T \;, \tag{2.89}$$

is used for the parameter of the filtering line. Then, the $\mathbf{e} := \boldsymbol{\alpha}(\lambda_Q^+, \bar{\mathbf{x}})$, $\boldsymbol{\alpha}(\lambda, \bar{\mathbf{x}})$, and $\boldsymbol{\beta}(\lambda, \bar{\mathbf{x}}, \mathbf{e})$ are determined, and we can find the filtering plane and the intersection line for the $\bar{x}_3$. In the circle-plus-line CT, the intersection line for the line trajectory part will be

$$v = -\frac{\bar{x}_3}{R} u + \frac{D}{R}\bar{x}_3 - \frac{(D-R)}{R}\frac{H}{\phi}\lambda \;, \tag{2.90}$$

and for the circle trajectory part will be

$$v = \frac{\bar{x}_3(\sin\lambda + 1)}{R\cos\lambda}u + \frac{D}{R}\bar{x}_3 \ . \tag{2.91}$$

In the saddle CT, the filtering direction is selected otherwise as Figure 2.5, and the intersection line will be the following two equations [39, 42]:

$$v = -\frac{\bar{x}_3\cos\lambda}{R\sin\lambda}u + \frac{D}{R}\bar{x}_3, \qquad \bar{x}_3 \geq 0 \ , \tag{2.92a}$$

$$v = \frac{\bar{x}_3\sin\lambda}{R\cos\lambda}u + \frac{D}{R}\bar{x}_3, \qquad \bar{x}_3 < 0 \ . \tag{2.92b}$$

In the helical CT, the object can be reconstructed by the general FBP method as (2.36), but also the Katsevich-type method as (2.50) is possible. In the Katsevich-type method, the filtering plane is defined by the parameter λ_1 (or λ_2) as (2.49), so it is chosen as the indexing parameter of the filtering line. (The λ_1 and λ_2 have the simple relationship such as $\lambda_1 = {}^{(\lambda\,+\,\lambda_2)}/_2$, so we can consider only one of them.) Next, the normal vector $\mathbf{n}(\lambda, \lambda_1)$ of the filtering plane is decided by λ_1 as (2.49), and then we can find the intersection line as following

$$v = \frac{H(\lambda_1 - \lambda)\cot(\lambda_1 - \lambda)}{2\pi R}u + \frac{DH}{2\pi R}(\lambda_1 - \lambda) \ . \tag{2.93}$$

For the intersection lines of each trajectory, the family of the filtering lines can be obtained by sampling $\bar{x}_3$ or $(\lambda_1 - \lambda)$.

Now, (2.85) is need to be deformed to change the parameter γ into (u, v) for implementation because the parameters of the detector image are (u, v). The following derivation was provided in [40]. In Figure 2.15, the projection of the $\mathbf{x}$ on the detector along the $\boldsymbol{\alpha}$ is denoted as P, and its local coordinate is (u, D, v) in the rotated coordinate system as Figure 2.14. Along the filtering line, we pick a point P' with the local coordinate (u', D, v'). If the angle between $\overrightarrow{OP}$ and $\overrightarrow{OP'}$ is γ, we have

$$\sin\gamma = \frac{|\overrightarrow{OP} \times \overrightarrow{OP'}|}{|\overrightarrow{OP}||\overrightarrow{OP'}|} = \frac{\sqrt{D^2(u'-u)^2 + D^2(v'-v)^2 + (u'v - uv')^2}}{|\overrightarrow{OP}||\overrightarrow{OP'}|} \ . \tag{2.94}$$

For arbitrary trajectories, the filtering line can be represented as

$$v = au + b \tag{2.95}$$

where a and b is constant. Then, $v' - v$ and $u'v - uv'$ will be

$$v' - v = a(u' - u) \tag{2.96a}$$

$$u'v - uv' = b(u' - u) \ . \tag{2.96b}$$

By (2.96), (2.94) can be simplified as

$$\sin\gamma = \frac{D(u'-u)\sqrt{1 + a^2 + b^2}}{|\overrightarrow{OP}||\overrightarrow{OP'}|} \ . \tag{2.97}$$

We pick another point P'' on the filtering line with local coordinate (u'', D, v''), and if the angle between $\overrightarrow{OP'}$ and $\overrightarrow{OP''}$ is $\Delta\gamma$, $\sin(\Delta\gamma)$ can be

$$\sin(\Delta\gamma) = \frac{D(u''-u')\sqrt{1 + a^2 + b^2}}{|\overrightarrow{OP'}||\overrightarrow{OP''}|} \ . \tag{2.98}$$

By (2.97) and (2.98), we have

$$\frac{\sin(\Delta\gamma)}{\sin\gamma} = \frac{(u''-u')\overrightarrow{OP}}{(u'-u)\overrightarrow{OP''}} \ . \tag{2.99}$$

Assume $\Delta\gamma$ is so small that $\Delta\gamma \approx \sin(\Delta\gamma)$, and finally we have

$$\frac{d\gamma}{\sin\gamma} = \frac{du'\overrightarrow{OP}}{(u'-u)\overrightarrow{OP'}} = \frac{du'\sqrt{D^2+u^2+v^2}}{(u'-u)\sqrt{D^2+u'^2+v'^2}} \ . \tag{2.100}$$

In result, (2.85) can be deformed as

$$g(\lambda,u,v) = \int_{-\infty}^{\infty} du' \ \frac{\sqrt{D^2+u^2+v^2}}{(u'-u)\sqrt{D^2+u'^2+v'^2}} D'_f(\lambda,u',v') \tag{2.101}$$

when the $\mathbf{e}$ is given. This equation can be implemented as following steps [39, 42, 43, 44]:

1. length-correction weighting.

$$g_1(\lambda,u,v) = \frac{D}{\sqrt{D^2+u^2+v^2}} D'_f(\lambda,u,v) \tag{2.102}$$

2. forward height rebinning. For all $\bar{x}_3$, compute

$$g_2(\lambda,u,\bar{x}_3) = g_1(\lambda,u,v(u,\bar{x}_3)) \ , \tag{2.103}$$

   where $v(u,\bar{x}_3)$ is determined by (2.90) and (2.91) for the circle-plus-line CT and by (2.92) for the saddle CT. In case of the helical CT, $\bar{x}_3$ is replaced with λ_1, and $v(u,\lambda_1)$ is determined by (2.93).

3. 1D Hilbert transform in u. Compute at each constant $\bar{x}_3$ or λ_1

$$g_3(\lambda,u,\bar{x}_3) = \int_{-\infty}^{\infty} du' \ \frac{1}{(u'-u)} g_2(\lambda,u',\bar{x}_3) \ . \tag{2.104}$$

4. backward height rebinning. Compute

$$g_4(\lambda,u,v) = g_3(\lambda,u,\bar{x}_3(u,v)) \ . \tag{2.105}$$

   Here, $\bar{x}_3(u,v)$ is

$$\bar{x}_3(u,v) = \frac{vR}{D-u} + \frac{\lambda H(D-R)}{\phi(D-u)} \tag{2.106}$$

   for the line trajectory part and

$$\bar{x}_3(u,v) = \frac{vR\cos\lambda}{u(\sin\lambda+1)+D\cos\lambda} \tag{2.107}$$

   for the circle trajectory part in the circle-plus-line CT. In the saddle CT, $\bar{x}_3(u,v)$ is

$$\bar{x}_3(u,v) = \frac{vR\sin\lambda}{D\sin\lambda - u\cos\lambda}, \qquad v \geq 0 \ , \tag{2.108a}$$

$$\bar{x}_3(u,v) = \frac{vR\cos\lambda}{D\cos\lambda - u\sin\lambda}, \qquad v < 0 \ . \tag{2.108b}$$

5. length-correction unweighting. Compute

$$g(\lambda,u,v) = \frac{\sqrt{D^2+u^2+v^2}}{D} g_4(\lambda,u,v) \ . \tag{2.109}$$

   Then, $g(\lambda,u,v)$ becomes the final filtering result.

Backprojection

The last process in the FBP-type reconstruction is the backprojection. In the reconstruction formula ((2.13) to (2.16)), the backprojection part is

$$\mathcal{K}(\mathbf{x},\mathbf{e},\lambda^-,\lambda^+) = -\frac{1}{2\pi^2}\int_{\lambda^-}^{\lambda^+} d\lambda\ \frac{1}{\|\mathbf{x}-\mathbf{a}(\lambda)\|} g(\lambda,\mathbf{x},\mathbf{e})\ . \tag{2.110}$$

For calculating backprojection signal, we use pixel driven approach. In (2.110), the interval of the backprojection integral is $[\lambda^-,\lambda^+]$, so we need to find the λ^- and λ^+ for the reconstruction point $\mathbf{x}$. Let $\mathbf{x} = (x_1,x_2,x_3)^T = (r\cos\xi, r\sin\xi, x_3)^T$. Then, using (2.47a) and (2.47b), we can derive the following two equations [40]:

$$t(r,\xi,\lambda^-) = \frac{R^2-r^2}{2R(R-r\cos(\xi-\lambda^-))}\ , \tag{2.111a}$$

$$\lambda^+(r,\xi,\lambda^-) = \lambda^- + 2\cos^{-1}\left(\frac{r\sin(\xi-\lambda^-)}{\sqrt{R^2+r^2-2Rr\cos(\xi-\lambda^-)}}\right)\ . \tag{2.111b}$$

If substituting (2.111) into (2.47c), we have

$$x_3(r,\xi,\lambda^-) = \frac{H}{2\pi}\Big(t(r,\xi,\lambda^-)\lambda^- + \big(1-t(r,\xi,\lambda^-)\big)\lambda^+(r,\xi,\lambda^-)\Big)\ . \tag{2.112}$$

Here, the x_3 and r,ξ are known, so (2.112) becomes the equation of only the λ^-. This equation can be solved by some numerical approaches to find the λ^-, and then t and λ^+ can be computed by (2.111). After backprojecting the filtered detector data for the interval $[\lambda^-,\lambda^+]$, the $f(\mathbf{x})$ can be finally reconstructed.

KAIST

2.4.2 Backprojection filtration type

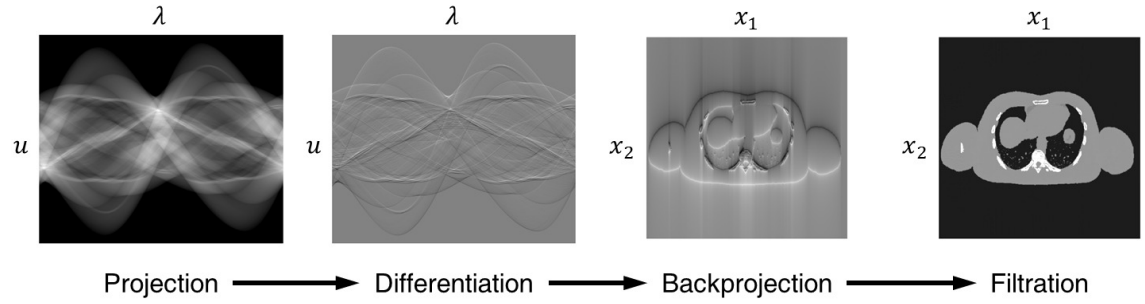


Figure 2.16: Total process of BPF reconstruction for fan-beam CT.

The total process of the BPF-type reconstruction is shown in Figure 2.16 and Figure 2.17 for fan-beam and cone-beam cases, respectively. In both cases, the first process is differentiation of projections with respect to the source trajectory. Then, differentiated signal is backprojected on the chord or PI line space. The reason why we use chord or PI lines instead of the Cartesian space is that the DBP signal will be filtered along the chord or PI line. All PI lines are reconstructed after filtration step, but the geometry of PI lines hardly provide intuition of 3D space [45]. So, we need to perform the regridding process lastly. In case of 2D fan-beam CT, the regridding process is not necessary. Now, we explain more details of each process.

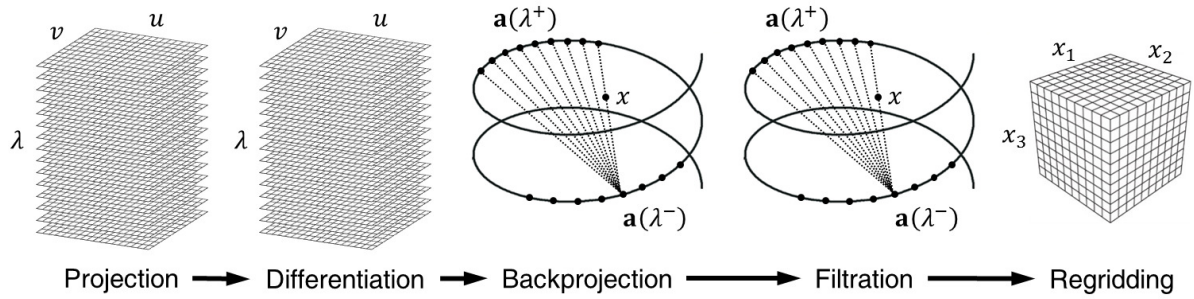


Figure 2.17: Total process of BPF reconstruction for cone-beam CT.

Differentiation

All differentiation process of the BPF-type reconstruction is same with the case of the FBP-type reconstruction.

Backprojection

Basically, backprojection in BPF method is similar to FBP method, and when calculating backprojection signal, we use pixel driven approach. The main difference of BPF from FBP method is using chord or PI line space in the filtration process. The chord or PI line space is represented with $(\lambda^-, \lambda^+, x)$.

In the circular fan-beam CT, we made vertical chord lines in Figure 2.9, and these chord lines are determined by x_1. In other words, to find the chord line $\mathcal{L}(\lambda^-, \lambda^+)$ for x_1, we need to find $\lambda^-(x_1)$ and $\lambda^+(x_1)$. This x_1 is sampled from $x_{1,min}$ to $x_{1,max}$ to cover the whole object. Then, along the chord line, the x is sampled to cover the whole object. Finally, the differentiated signals are backprojected multiplying the distance weight $1/\|\mathbf{x}-\mathbf{a}(\lambda)\|$ for a given point $(\lambda^-, \lambda^+, x)$. At this time, the differentiated signal is necessary only for $\mathcal{C}(\lambda^-, \lambda^+)$.

In the circle-plus-line cone-beam CT, the chord line is made with two source positions, one of which is on the line part and the other of which is on the circle part. First, we sample the λ^- to cover the whole object, and the increment of $\mathbf{a}(\lambda^-)$ along the x_3-axis should be shorter than the sampling distance along the x_3-axis in 3D Cartesian coordinate system. Secondly, λ^+ is sampled from $\lambda^+_{\min}$ to $\lambda^+_{\max}$ so that the set of chord lines can cover the whole object. Lastly, x is sampled to cover the whole object. As λ^-, grid of (λ^+, x) should be more narrow than horizontal plane. For backprojection of a given $\mathbf{x}$ on the chord line $\mathcal{L}(\lambda^-, \lambda^+)$, the differentiated images are required only for trajectory segment $\mathcal{C}_\pi(\lambda^-, \lambda^+)$.

In the helical cone-beam CT, to define the PI lines, firstly λ^- is sampled along helical trajectory. At this time, the increment of $\mathbf{a}(\lambda^-)$ along the x_3-axis should be shorter than the sampling distance along the x_3-axis in 3D Cartesian coordinate system. Secondly, λ^+ is sampled from $\lambda^+_{\min}$ to $\lambda^+_{\max}$ so that the set of PI lines can cover the whole object. Lastly, x is sampled to cover the whole object. As λ^-, grid of (λ^+, x) should be more narrow than horizontal plane. For backprojection of a given $\mathbf{x}$ on the PI line $\mathcal{L}_\pi(\lambda^-, \lambda^+)$, the differentiated images are required only for helix segment $\mathcal{C}_\pi(\lambda^-, \lambda^+)$.

Filtration

After backprojection, the DBP signal on each chord or PI line is filtered by taking the inverse Hilbert transform. At this time, the difference from the FBP method is the truncation of the signal. The original

Hilbert transform formula is

$$f(x) = \mathcal{H}g(x) = \frac{1}{\pi}\ \text{P.V.} \int_{-\infty}^{\infty} dx'\ \frac{1}{x - x'} g(x') \tag{2.113}$$

where P.V. denotes the Cauchy principal value. In this equation, the interval of the integral is infinite. In FBP method, if the projection signal is not truncated along the filtering line on the detector plane, the differentiated signal is not truncated either. Thus, when we take the Hilbert transform of the bounded signal $g(x)$ such as $g(x) = 0$ for $x \notin [x^-, x^+]$, the result $f(x)$ can be computed as

$$f(x) = \frac{1}{\pi}\ \text{P.V.} \int_{x^-}^{x^+} dx'\ \frac{1}{x - x'} g(x')\ . \tag{2.114}$$

In the BPF method, however, the DBP signal $g(x)$ is not bounded, and it is known only for the particular interval $[x^-, x^+]$ which covers the object. In this case, we need to use the truncated Hilbert transform formula:

$$f(x) = \frac{1}{\sqrt{(x^+ - x)(x - x^-)}} \left[\frac{1}{\pi}\ \text{P.V.} \int_{x^-}^{x^+} dx'\ \frac{1}{x - x'} g(x') \sqrt{(x^+ - x')(x' - x^-)} + c_f \right] \tag{2.115}$$

where $c_f = {}^1\!/_\pi \int f(x)\ dx$ [38]. For the inverse Hilbert transform, we only need to multiply -1 because $\mathcal{H}\mathcal{H} = -I$.

Regridding

In the BPF-type reconstruction, we use the chord or PI line space instead of the Cartesian space. In the 2D fan-beam CT, however, the chord line space is same with the Cartesian space, so the regridding process is not required. In the 3D cone-beam case, the regridding process is necessary to map the chord or PI line space into the Cartesian space. The regridding process is implemented by dividing the point value on the chord or PI line into the nearest eight voxels in the Cartesian coordinate space in proportion to the distance.

2.4.3 Programming by graphics processing unit

In this section, we will explain the parallelizing implementation only for the BPF-type reconstruction because both algorithms consist of the same process, and the difference is only the order. For various geometries, we choose only the helical cone-beam CT because the difference from the other geometries is only the way to define the chord or PI lines. After making the chord or PI lines, the fan-beam CT can be considered as the case with the lower dimension than the helical cone-beam CT and the other cone-beam CT can be considered as the same case in the parallelizing point of view.

Before explaining GPU programming, we define discrete indices of the cone-beam projection, back-projection, and object data as Figure 2.18. First, the cone-beam projection data are uniformly sampled with intervals $\Delta\lambda$, Δu, Δv for λ, u, v, so the discrete cone-beam projection images are denoted as

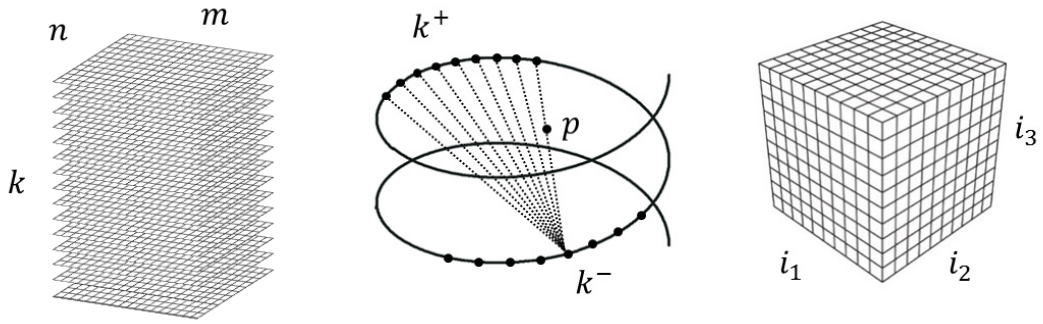
$$D_f[k, m, n] := D_f(\lambda_k, u_m, v_n), \quad 0 \le k < K,\ 0 \le m < M,\ 0 \le n < N \tag{2.116}$$

where

$$\lambda_k = \Delta\lambda \cdot k \tag{2.117a}$$

$$u_m = \Delta u \left(m - \frac{M-1}{2} \right) \tag{2.117b}$$

$$v_n = \Delta v \left(n - \frac{N-1}{2} \right)\ . \tag{2.117c}$$



(a) Cone-beam projection (b) PI lines space (b) 3D Cartesian space

Figure 2.18: Discrete indices for the cone-beam projection, PI line, and 3D Cartesian space.

Second, the backprojection data in the PI line space are also uniformly sampled with intervals $\Delta\lambda^-$, $\Delta\lambda^+$, Δx for λ^-, λ^+, x, and the data are denoted as

$$g[k^-, k^+, p] := g(\lambda^-_{k^-}, \lambda^+_{k^+}, x_p), \quad 0 \le p < \Pi,\ 0 \le k^- < K^-,\ 0 \le k^+ < K^+ \tag{2.118}$$

where

$$\lambda^-_{k^-} = \Delta\lambda^- \cdot k^- \tag{2.119a}$$

$$\lambda^+_{k^+} = \lambda^-_{k^-} + \pi + \Delta\lambda^+ \left(k^+ - \frac{K^+ - 1}{2}\right) \tag{2.119b}$$

$$x_p = \Delta x_p \left(p - \frac{\Pi - 1}{2}\right). \tag{2.119c}$$

Lastly, the object data in 3D Cartesian space are also uniformly sampled with intervals Δx_1. Δx_2, Δx_3 for x_1, x_2, x_3, and the data are denoted as

$$f[i_1, i_2, i_3] := f(x_{1,i_1}, x_{2,i_2}, x_{3,i_3}), \quad 0 \le i_1 < N_1,\ 0 \le i_2 < N_2,\ 0 \le i_3 < N_3 \tag{2.120}$$

where

$$x_{1,i_1} = \Delta x_1 \left(i_1 - \frac{N_1 - 1}{2}\right) \tag{2.121a}$$

$$x_{2,i_2} = \Delta x_2 \left(i_2 - \frac{N_2 - 1}{2}\right) \tag{2.121b}$$

$$x_{3,i_3} = \Delta x_3 \left(i_3 - \frac{N_3 - 1}{2}\right). \tag{2.121c}$$

From projection data $D_f[k, m, n]$, the differentiation $D'_f[k, m, n]$ can be approximately computed as (2.81). In (2.81), we can calculate u_m, v_n by (2.117), and $\frac{\partial D_f}{\partial \mu}, \frac{\partial D_f}{\partial u}, \frac{\partial D_f}{\partial v}$ can be approximated by

$$\frac{\partial D_f}{\partial \mu} \simeq \frac{D_f[k+1, m, n] - D_f[k-1, m, n]}{2\Delta\lambda} \tag{2.122a}$$

$$\frac{\partial D_f}{\partial u} \simeq \frac{D_f[k, m+1, n] - D_f[k, m-1, n]}{2\Delta u} \tag{2.122b}$$

$$\frac{\partial D_f}{\partial v} \simeq \frac{D_f[k, m, n+1] - D_f[k, m, n-1]}{2\Delta v}. \tag{2.122c}$$

For GPU programming of the backprojection, the backprojection for whole object needs all projection data, and their resolutions are $K \times N \times M$ and $K^- \times K^+ \times P$. but it could be impossible to

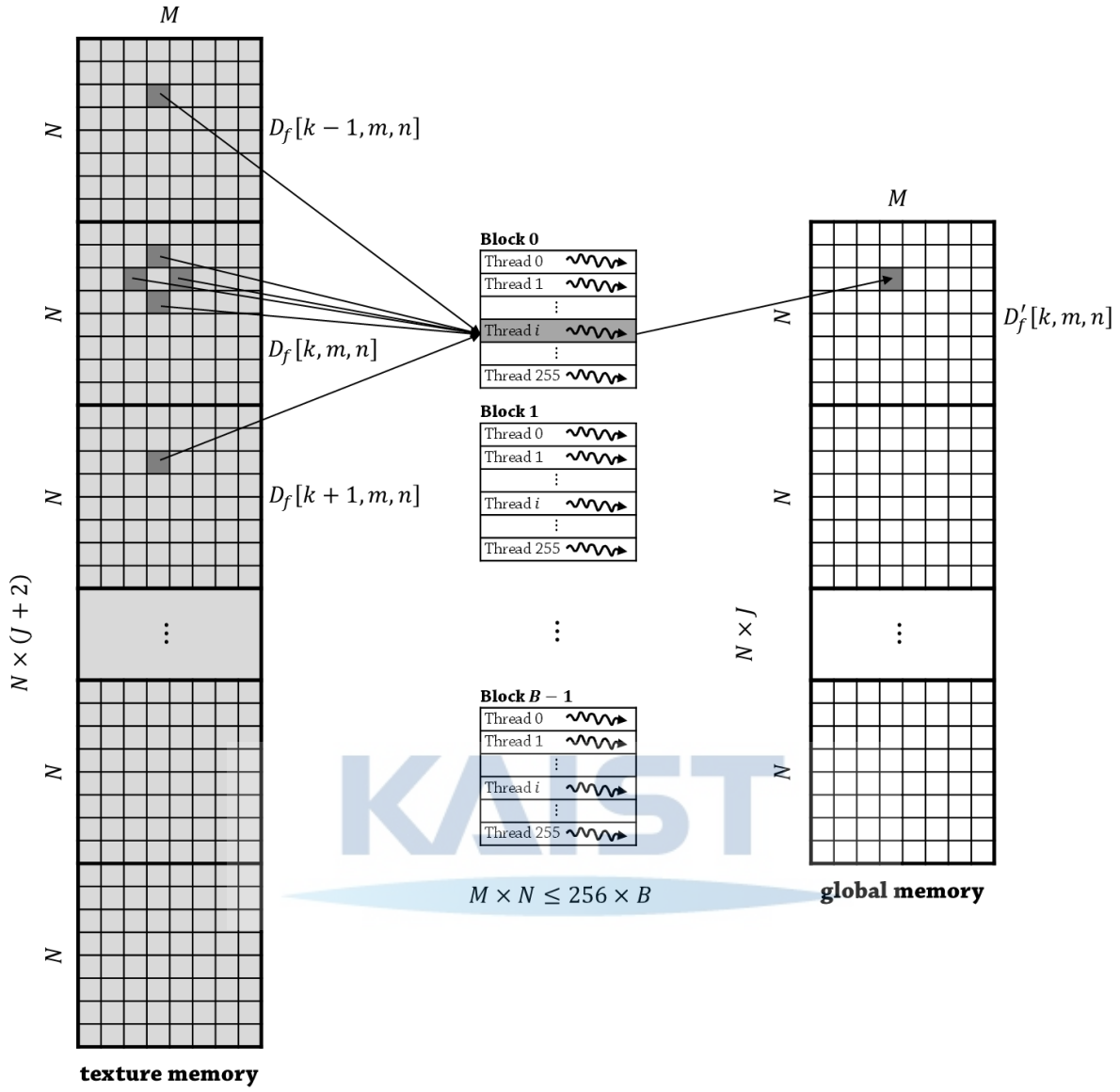


Figure 2.19: GPU parallelizing of differentiation. J projection images are loaded in GPU memory, and the number of J depends on a capacity of GPU. Each pixel of a differentiation image are parallelized, and for differentiation, the anterior and posterior images are necessary.

keep the all projection images and backprojection images in GPU memory at once. So, to avoid out of memory problem, we allocate GPU memory for whole backprojection images which has $K^- \times K^+ \times P$ size, and read only J projection images and two more projection images which is anterior and posterior of J projection images. These two anterior and posterior projection images are for calculating (2.122a). These J projection images are bound to texture memory for fast access to read, and we fetch them with integer indices because interpolation is not necessary by chain rule. Then, B blocks containing 256 threads are made, and B needs to satisfy $M \times N \leq 256 \times B$. Each thread reads projection value of $D_f[k+1,m,n]$, $D_f[k-1,m,n]$, $D_f[k,m+1,n]$, $D_f[k,m-1,n]$, $D_f[k,m,n+1]$, $D_f[k,m,n-1]$ as Figure 2.19 and computes the differentiation D'_f for each pixel $[k,m,n]$ by (2.122).

After we obtain differentiated projection $D'_f[k,m,n]$, the memory for differentiation images are bound to texture memory. Now, the B blocks containing 256 threads are made satisfying $K^- \times K^+ \leq$

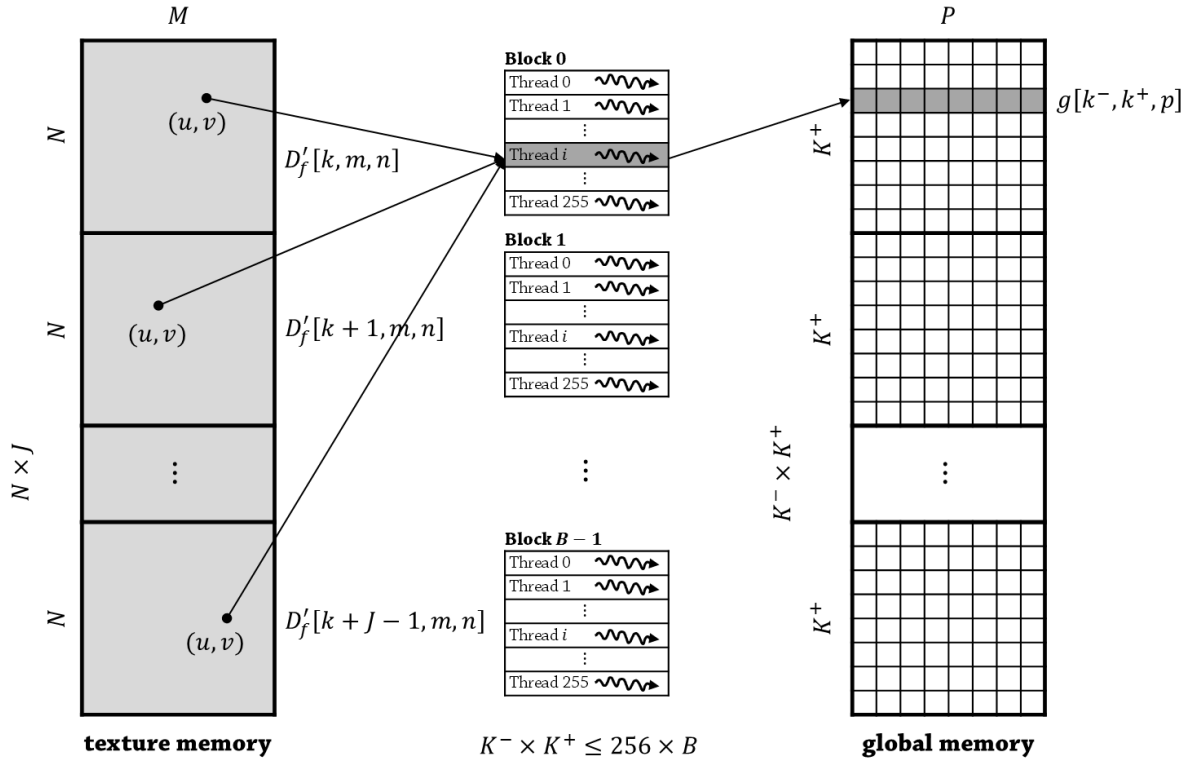


Figure 2.20: GPU parallelizing of backprojection. J differentiated images are loaded in texture memory, and it is accessed with real number. The each thread deals with a PI line, and its number is greater than or equal to the number of PI lines to cover all PI lines at once. Each differentiated images are judged if it is included in helix segment corresponding to the PI line, and if so, it is backprojected on all voxels of the PI line.

$256 \times B$, and each thread deals with a PI line. This is because in the same PI line, the backprojection range which is helix segment is same. For a source location $\mathbf{a}(\lambda_k)$ of differentiated projection, the thread judges if $\mathbf{a}(\lambda_k)$ is included in helix segment or not. If it is included, the thread reads differentiated projection $D'_f[k, m, n]$ for all points along the PI line with floating indices to interpolate $D'_f[k, m, n]$, and accumulates it on the PI line. To get complete DBP, we need to repeat this process J times.

Now, we apply the Hilbert transform along PI lines. By GPU programming, this process is parallelized for one backprojection image at once. The Hilbert transform is implemented by convolution for more accurate result than FFT, and Figure 2.21 show parallelization of the Hilbert transform. Each thread computes result of Hilbert transform by convolution of $g[k^-, k^+, p]$ of a PI line and Hilbert kernel which is in GPU global constant.

2.5 Simulation result

In this section, we show the result of the FBP reconstruction in the saddle cone-beam CT and the BPF reconstruction in the circle-plus-line CT. The implementations of the algorithms were evaluated using 3D Shepp-Logan phantom. The object size is $256 \times 256 \times 256$ voxels, and the voxel size is $0.5 \times 0.5 \times 0.5$ mm^3. For scanning system, the distance R form the source to the object is 400 mm, and the distance D from the source to the detector is 700 mm. The GPU implementations were programmed by

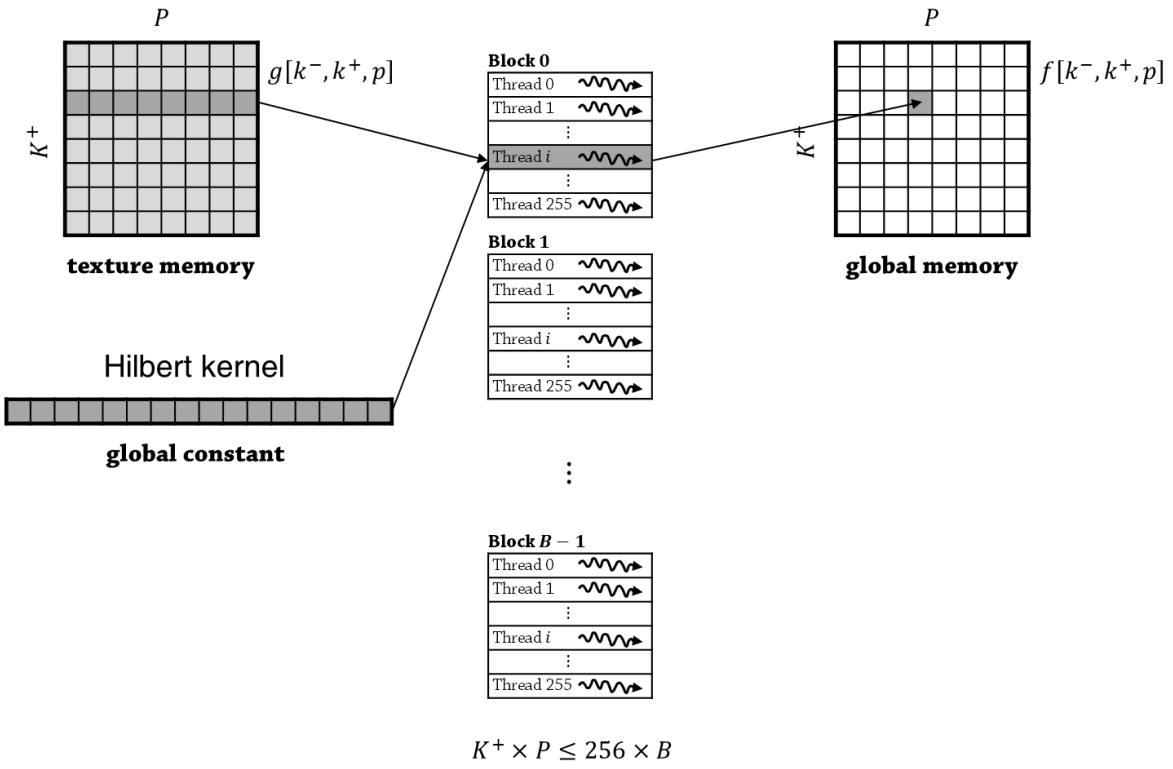


Figure 2.21: GPU parallelizing of Hilbert transform. The coefficients of Hilbert transform is initially saved as a constant array. A thread performs Hilbert transform of a PI line by convolution of Hilbert kernel and backprojection of the PI line.

using compute unified device architecture (CUDA) which is a parallel computing architecture developed by Nvidia, and the implementations were tested with TESLA C1060.

For the saddle CT simulation, H is 64 mm to cover all object. Because of this pitch, we use the 512×1024 pixel size detector to cover all projections of the object inside the detector in any source position, and the pixel size is 0.5×0.5 mm^2. We generated 900 cone-beam projections as λ increases uniformly from 0 to 2π, and Figure 2.22 shows the reconstruction result. The reconstruction image quality of the horizontal slice is clear, but in the vertical slices, there are some artifact lines in a diagonal direction. These artifacts come from the pitch of the saddle trajectory, so it can be reduced by denser scanning.

For the circle-plus-line CT simulation, the pitch of line trajectory was also decided to cover the object, which is about 153 mm, and the detector size was 512×512 pixels with 0.5×0.5 mm^2 pixel size. In the circle-plus-line CT, the detector does not need to cover all projections of the object when it was scanned in the line trajectory, thus it is smaller than the saddle CT. For simulation, 611 cone-beam projections in the line scanning part and 280 cone-beam projections in the circle scanning part were generated. As Figure 2.23, the reconstruction image quality of the horizontal slice is as almost same as saddle CT, but the vertical slices are clearer than saddle CT without artifacts. Even if the line profiles of the saddle and circle-plus-line CT are similar, the entire reconstruction image quality of the circle-plus-line CT is better than the saddle CT.

Table 2.1 shows the computational time. Even if the number of projection views of two simulations were similar to each other (900 views for the saddle CT and 890 views for the circle-plus-line CT), the

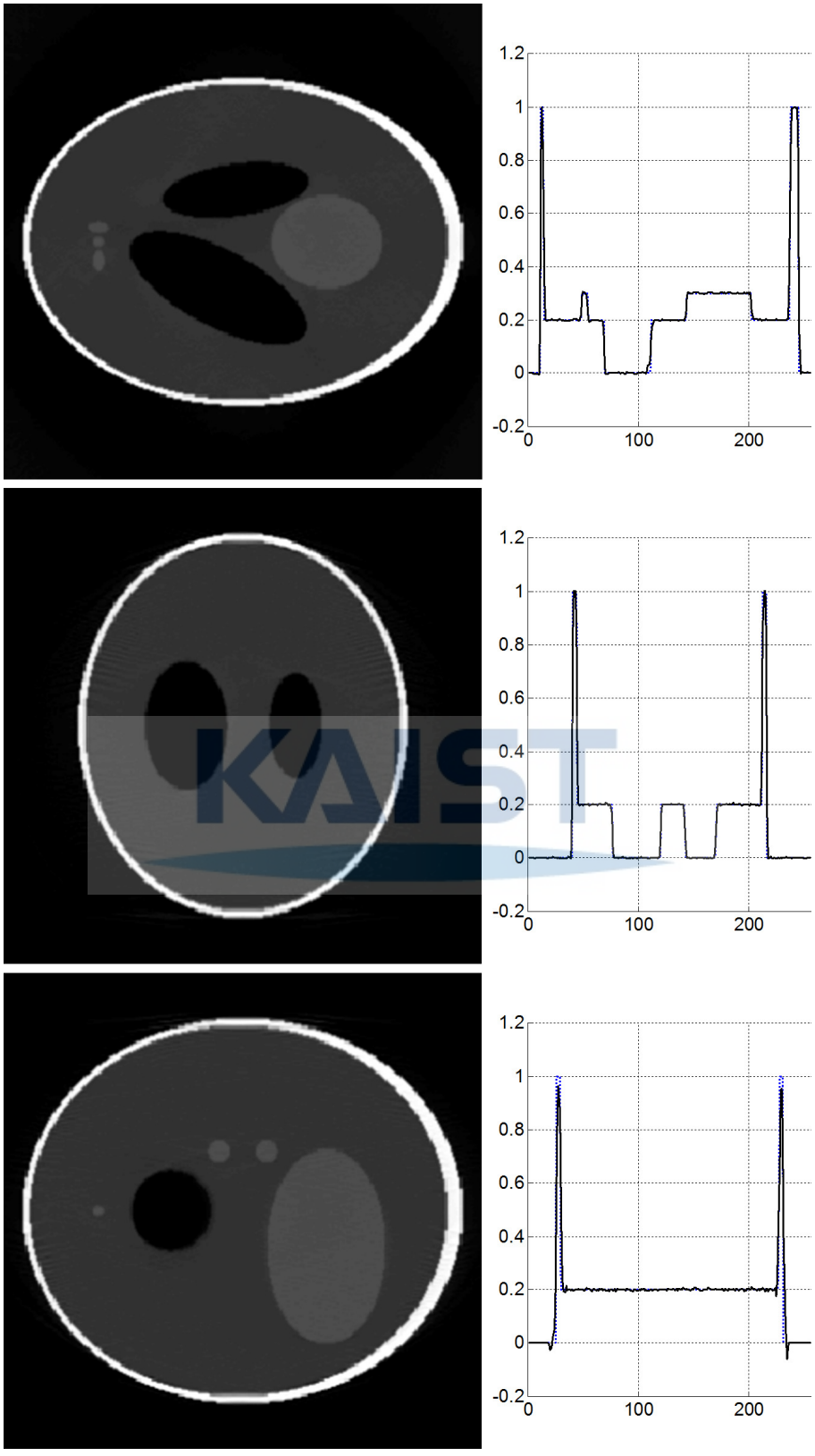


Figure 2.22: FBP reconstruction result of saddle CT. Transverse image and its row line profile (top), coronal image and its row line profile (middle), and sagittal image and its column line profile (bottom).

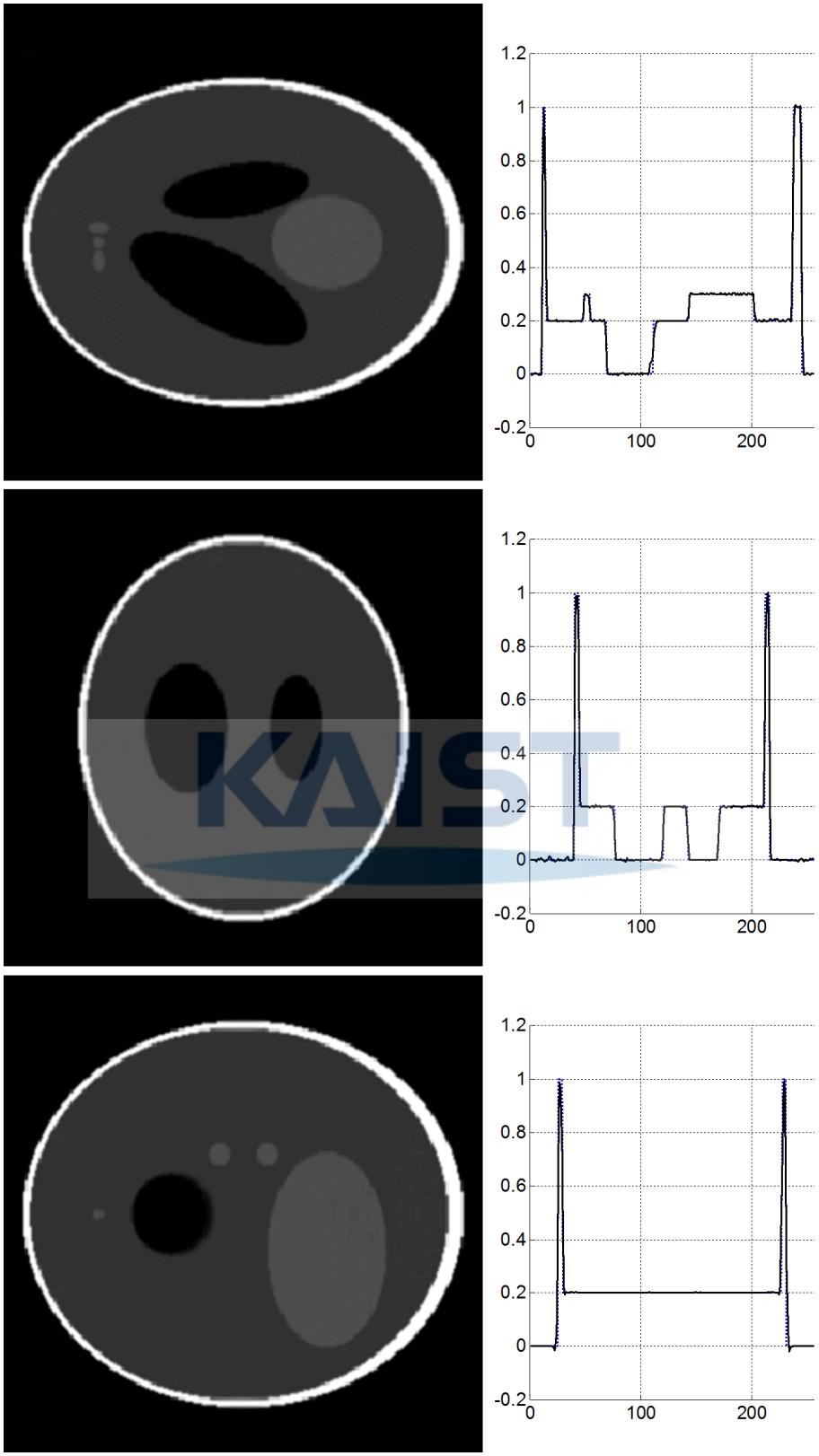


Figure 2.23: BPF reconstruction result of circle-plus-line CT. Transverse image and its row line profile (top), coronal image and its row line profile (middle), and sagittal image and its column line profile (bottom).

Table 2.1: Computational time (seconds) for analytic cone-beam reconstructions

	Saddle	Circle-plus-line
Derivative	12.948	3.415
Filtration	5.631	1.104
Backprojection	10.172	14.741
Regridding		5.271
total	28.751	24.531

saddle CT used the much bigger detector than the circle-plus-line CT, so the derivative process of the saddle CT takes more time than the circle-plus-line CT. In the saddle CT, the filtration requires the forward and backward rebinning process, so it takes longer than circle-plus-line CT. For the backprojection step the circle-plus-line CT takes longer because the chord length in the rebinned space is almost twice longer than the object width for the filtration. Finally, the total computational times are similar.

KAIST

Chapter 3. Parallel proximal algorithm for interior tomography with tiny *a priori* knowledge

With the increasing demands for low-dose CT to reduce radiation exposure on human bodies, we often encounter interior tomography problems, where only incomplete data are available due to the limited detector size or reduced illumination. Theoretical analysis has been extensively performed to show that the problem can be equivalently formulated as incomplete data extrapolation problem using DBP data [24, 25]. More specifically, after backprojecting the differentiated projection data on chord lines, a 2D or 3D tomography problem using fan-beam or cone-beam CT can be converted into a set of 1D problems, which makes the interior tomography problem much simpler extrapolation problem. Based on this formulation, general imaging geometries that allow perfect reconstruction have been identified [24]. For example, if the imaging FOV includes a background with air, then *a priori* knowledge of regions with air determines unique reconstruction. Another important example is that even though there exists no air interface in the FOV, if tiny regions within the FOV and boundary of the object are known *a priori*, an unique and stable solution can be computed [25]. Asides from the theoretical results, the authors in [24, 25] noted that the constraints due to *a priori* knowledge are convex and used POCS algorithm to obtain the solution by sequentially projecting on the constraint set. They demonstrated excellent reconstruction results using POCS. However, one of the main limitations is that the POCS method finds the optimized solution by sequentially projecting the current solution onto each convex set, so its convergence speed is slow.

To address this problem, we employ a parallel proximal algorithm (PPXA) that has been recently proposed to address multiple convex constraints using a proximal algorithm [27]. More specifically, PPXA calculates the multiple proximal projections and computes a weighted average using Douglas Rachford splitting algorithm [27]. As the interior tomography problem can be converted into a 1D optimization problem with multiple convex constraints, it is nicely fitted into PPXA framework. In particular, the convex constraints in the interior tomography problem are just a set of indicator functions, so the corresponding proximal mappings are very simple, which makes the computational complexity of PPXA minimal. This paper explains how to apply PPXA to the interior tomography with *a priori* knowledge, and show that PPXA converges faster than POCS method. Even though the current formulation is based on 2D formulation, extension to 3D is straightforward and will be reported later.

3.1 Interior tomography using *a priori* knowledge

As we now know that the 2D and 3D tomography problem can be solved using the 1D Hilbert transform on chord lines, the following research also focused on the 1D Hilbert transform relationship in interior tomography problems. With a slight abuse of notation, we denote $g(x) := g_{\mathbf{x}_{E^\perp}}(x)$ and $f(x) = f_{\mathbf{x}_{E^\perp}}(x)$. Then, we have the following 1D Hilbert transform relationship as

$$g(x) = \frac{1}{\pi} \text{ P.V.} \int_{-\infty}^{\infty} dx' \; \frac{1}{x - x'} f(x') = \mathcal{H} f(x) \; , \tag{3.1}$$

where P.V. denotes the Cauchy principal value. In this equation, $f(x)$ is unknown and $g(x)$ is measurement, so (3.2) can represent the system of the reconstruction problem. In reverse, the inversion of (3.1) is

$$f(x) = -\frac{1}{\pi}\ \text{P.V.}\int_{-\infty}^{\infty} dx'\ \frac{1}{x-x'}g(x') = -\mathcal{H}g(x)\ . \tag{3.2}$$

In (3.2), the DBP $g(x)$ is known only for the interval (x^-, x^+) of the FOV. If the FOV interval (x^-, x^+) covers the support of the object $f(x)$, the truncated version of (3.2) is possible such as

$$f(x) = \frac{1}{\sqrt{(x^+-x)(x-x^-)}}\left[-\frac{1}{\pi}\ \text{P.V.}\int_{x^-}^{x^+} dx'\ \frac{1}{x-x'}g(x')\sqrt{(x^+-x')(x'-x^-)} + c_f\right] \tag{3.3}$$

where $c_f = {}^1\!/_\pi \int f(x)\ dx$ [38]. For implementation, we can obtain the c_f value from the projection $D_f(\mathbf{a}, \boldsymbol{\theta})$.

If the DBP data $g(x)$ are available for the support of the object $f(x)$ such as Figure 1.1(a), the reconstruction can be simply done by performing (3.3). However, in case of interior tomography problem, the detector is truncated, as seen in Figure 1.1(b); thus the DBP data are only available within FOV. For the exact reconstruction for a given point x, the sufficient conditions need to be satisfied, implying that $g(x)$ along the $\mathcal{L}(\lambda^-, \lambda^+)$ should have no truncation of the support of $f(x)$. If the sufficient conditions are not satisfied, the analytic reconstruction is not available, which is known as the interior problem.

The main problem of such truncated Hilbert transform is the existence of the null space. More specifically, there exists nonzero $f_{\mathcal{N}}(x)$ such that

$$\mathcal{H}f_{\mathcal{N}}(x) = 0, \quad x \in (e^-, e^+) \tag{3.4}$$

where (e^-, e^+) denotes FOV. Indeed, $f_{\mathcal{N}}(x)$ can be expressed using an appropriate DBP data $\psi(x)$ outside of the ROI such that

$$f_{\mathcal{N}}(x) = -\frac{1}{\pi}\int_{\mathbb{R}\setminus(e^-,e^+)} dx'\ \frac{1}{x-x'}\psi(x')\ . \tag{3.5}$$

Note that the null space signal $f_{\mathcal{N}}(x)$ in (3.5) is differentiable in any order due to the removal of the origin in the integrand. Then, a required interior tomography formulation is to find an appropriate regularization term that suppresses the signal belonging to the null space of the truncated Hilbert transform by exploiting the infinite differentiability of $f_{\mathcal{N}}$.

In previous research, the authors in [25] solve this problem using tiny known subregions in FOV. Figure 3.1 illustrates the interior tomography using *a priori* knowledge. In this figure, the FOV is located completely inside of the object. Along the PI line, (b^-, b^+) refers to the support of the object, which is assumed to be approximately known, and (e^-, e^+) refers to FOV. In this case, FOV is the subset of the object support, $(e^-, e^+) \subset (b^-, b^+)$, so it cannot be recovered by (3.3). However, if we have some known subregions in (e^-, e^+) such as (d_1^-, d_1^+) and (d_2^-, d_2^+), the theory says that $f(x)$ can be determined uniquely.

In [25], using *a priori* knowledge, five convex constraints for $f(x) \in L^2(\mathbb{R})$ were identified:

$$C_1 = \{f(x) \in L^2(\mathbb{R}) : f(x) = 0, x \notin (b^-, b^+)\} \tag{3.6a}$$

$$C_2 = \{f(x) \in L^2(\mathbb{R}) : \mathcal{H}f(x) = g(x), x \in (e^-, e^+)\} \tag{3.6b}$$

$$C_3 = \{f(x) \in L^2(\mathbb{R}) : f(x) = \bar{f}(x), x \in (d_1^-, d_1^+) \cup (d_2^-, d_2^+)\} \tag{3.6c}$$

$$C_4 = \{f(x) \in L^2(\mathbb{R}) : \frac{1}{\pi}\int_{b^-}^{b^+} dx\ f(x) = c_f\} \tag{3.6d}$$

$$C_5 = \{f(x) \in L^2(\mathbb{R}) : f(x) \geq 0, x \in (b^-, b^+)\} \tag{3.6e}$$

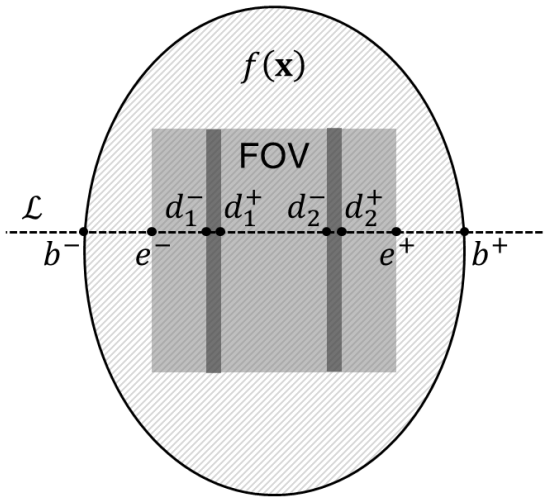


Figure 3.1: Interior problem with *a priori* knowledge.

where $\bar{f}(x)$ is true value of $f(x)$. The optimized solution of $\hat{f}(x)$ was solved by POCS. More specifically, if P_{C_j} is the orthogonal projector onto the convex set C_j for $j = 1, \ldots, 5$, the iteration of the POCS algorithm can be described as

$$\hat{f}_k(x) = P_{C_5} P_{C_4} P_{C_3} P_{C_2} P_{C_1} \hat{f}_{k-1}(x) \tag{3.7}$$

with some arbitrary initialization $\hat{f}_0(x)$. This algorithm converges to one element of the intersection of the constraints if the intersection is not empty. The simulation result prove this method works well. In general, however, it is difficult to know the intensity inside the object, so consequently it is necessary to improve the algorithm for reconstruction without inside known subregions.

3.2 Proximity operator

In this section, we show that the interior tomography problem can be addressed using a parallel proximal algorithm. If $\Gamma_0(\mathbb{R}^N)$ is the class of lower semicontinuous convex functions from $\mathbb{R}^N$ to $(-\infty, +\infty]$, the proximity operator of $\varphi \in \Gamma_0(\mathbb{R}^N)$ is defined as

$$\text{prox}_\varphi : \mathbb{R}^N \to \mathbb{R}^N : u \mapsto \underset{v \in \mathbb{R}^N}{\arg\min} \frac{1}{2}\|v - u\|^2 + \varphi(v), \tag{3.8}$$

The proximal mapping has been extensively used to derive new class of convex optimization algorithms [27]. However, in most cases, the number of convex penalty terms are limited to one. In our interior tomography problems, there exist multiple convex constraints, so we need a new algorithm that fit into the interior tomography problem [27].

3.3 Parallel proximal algorithm

PPXA is a new class of proximal mapping algorithms that address multiple convex constraints [46]

$$\text{Find } \hat{\xi} = \underset{\xi \in \mathbb{R}^N}{\arg\min} \sum_{j=1}^{J} \varphi_j(\xi) \tag{3.9}$$

where $\varphi_j \in \Gamma_0(\mathbb{R}^N)$ and $J \geq 3$. Algorithm 3.3.1 describes a PPXA implementation.

Algorithm 3.3.1: PPXA$((u_{j,0})_{1\leq j\leq J}, (\omega_j)_{1\leq j\leq J}, \lambda, \gamma, N)$

$\xi_0 = \sum_{j=1}^{J} \omega_j u_{j,0}$
for $k \leftarrow 0$ **to** $N-1$
do
- **for** $j \leftarrow 1$ **to** J
  - **do** $p_{j,k} = \text{prox}_{\gamma\varphi_j/\omega_j} u_{j,k} + a_{j,k}$
- $p_k = \sum_{j=1}^{J} \omega_j p_{j,k}$
- **for** $j \leftarrow 1$ **to** J
  - **do** $u_{j,k+1} = u_{j,k} + \lambda(2p_k - \xi_k - p_{j,k})$
- $\xi_{k+1} = \xi_k + \lambda_k(p_k - \xi_k)$

$\hat{\xi} = \xi_N$
return $(\hat{\xi})$

PPXA needs real constants $\gamma \in (0, +\infty)$ and $(\omega_j)_{1\leq j\leq J} \in (0,1]^J$ satisfying $\sum_{j=1}^{J} \omega_j = 1$. ω_j is the weight for the proximity operator of φ_j. $(u_{j,0})_{1\leq j\leq J} \in (\mathbb{R}^N)^J$ is the initialization. The relaxation parameter $\lambda_k \in (0,2)$ can be chosen differently at each step, and the data fidelity term $a_{j,k}$ is included when calculating a proximal vector $p_{j,k}$. The main feature of this algorithm is that at each iteration k, the proximal vectors $(p_{j,k})_{1\leq j\leq J}$ and the auxiliary vectors $(u_{j,k})_{1\leq j\leq J}$ can be computed simultaneously. This is the main difference from POCS method, and it makes the convergence of PPXA faster.

In our interior tomography problem, all the constraints can be formulated using a set of indicator functions. More specifically, the indicator function of convex C is

$$\iota_C(x) = \begin{cases} 0, & \text{if } x \in C \\ +\infty, & \text{if } x \notin C \end{cases}, \tag{3.10}$$

and its proximity operator is the projection onto C, P_C [27]. With this definition, we can make indicator functions for each constraint of C_1-C_4 as

$$\iota_{C_1}(\tilde{f}(x)) = \begin{cases} 0, & \text{if } (\mathcal{H}\tilde{f})(x) = g(x), x \in [e^-, e^+] \\ +\infty, & \text{otherwise} \end{cases}$$

$$\iota_{C_2}(\tilde{f}(x)) = \begin{cases} 0, & \text{if } \tilde{f}(x) = f(x), x \in [d_1^-, d_1^+] \cup [d_2^-, d_2^+] \\ +\infty, & \text{otherwise} \end{cases}$$

$$\iota_{C_3}(\tilde{f}(x)) = \begin{cases} 0, & \text{if } \tilde{f}(x) = 0, x \notin [b^-, b^+] \\ +\infty, & \text{otherwise} \end{cases}$$

$$\iota_{C_4}(\tilde{f}(x)) = \begin{cases} 0, & \text{if } \tilde{f}(x) \geq 0, x \in [b^-, b^+] \\ +\infty, & \text{otherwise} \end{cases}$$

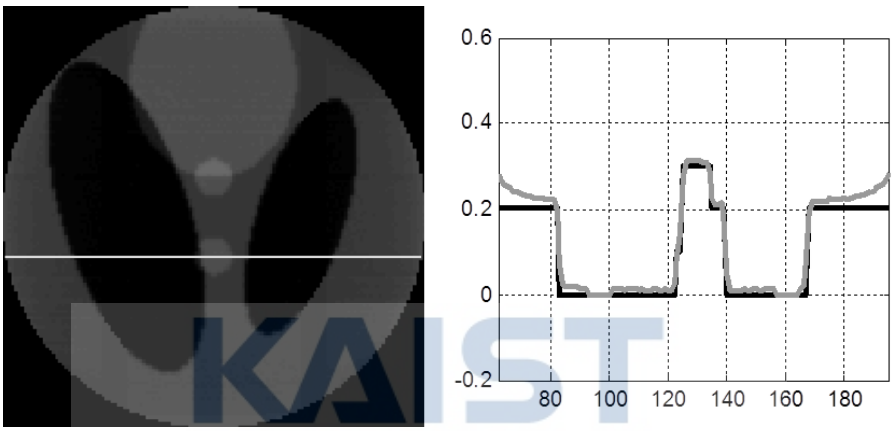
and for C_5, even though we do not make a indicator function, this constraint can be easily incorporated as constant C_f in computing the truncated Hilbert transform (3.3). Now, the minimization problem for the interior tomography with *a priori* knowledge can be formulated as

$$\text{Find } \hat{f}(x) = \arg \min_{\tilde{f}(x)\in L^2(\mathbb{R})} \sum_{j=1}^{4} \iota_{C_j}(\tilde{f}(x)), \tag{3.11}$$

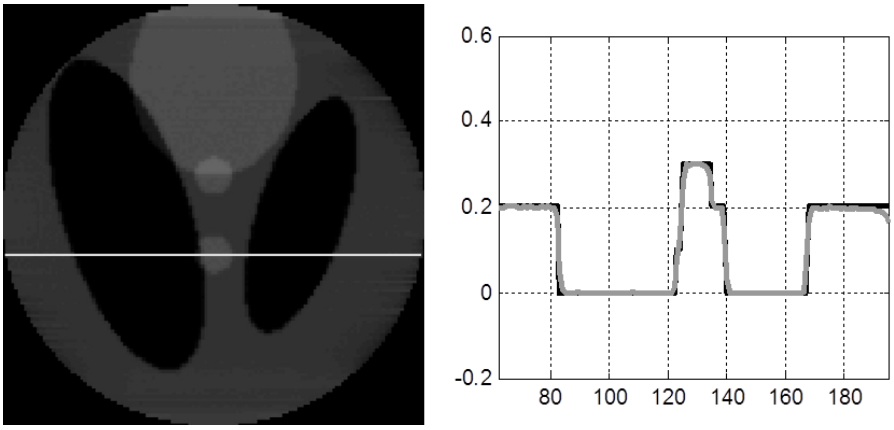
and PPXA can be applied to solve (3.11). As we know, the proximity operator of the indicator function of C is the projection onto C, so in Algorithm 3.3.1, $\text{prox}_{\gamma\varphi_j/\omega_j}$ will be replaced with P_{C_j}. Note that this algorithm can be converged under the following assumptions [46], which are trivially satisfied in our interior tomography problem.

1) $\lim_{\|\xi\|\to+\infty} \varphi_1(\xi) + \cdots + \varphi_J(\xi) = +\infty$.
2) $\cap_{j=1}^{J} \text{rint dom } \varphi_j \neq \varnothing$.
3) $(\forall j \in \{1, \ldots, J\}) \sum_{k\in\mathbb{N}} \lambda_k \|a_{j,k}\| < +\infty$.
4) $\sum_{k\in\mathbb{N}} \lambda_k(2 - \lambda_k) = +\infty$.

3.4 Simulation results



(a) POCS_{100} (b) POCS_{100}



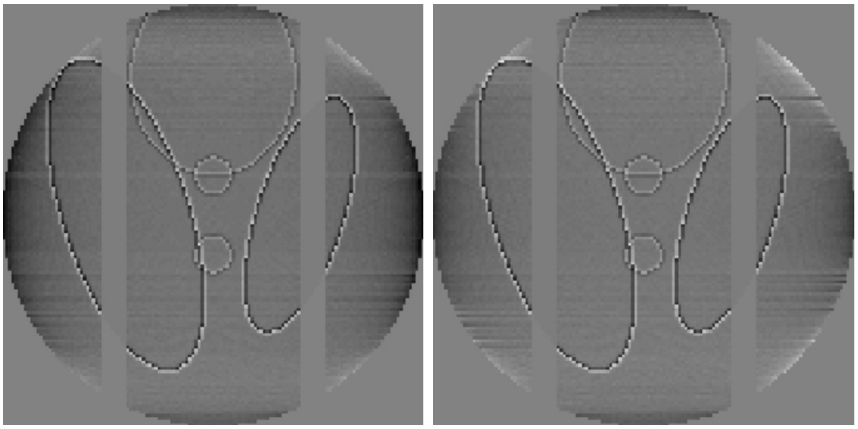
(c) PPXA_{100} (d) PPXA_{100}

Figure 3.2: Reconstruction images and line profiles after 100 iterations.

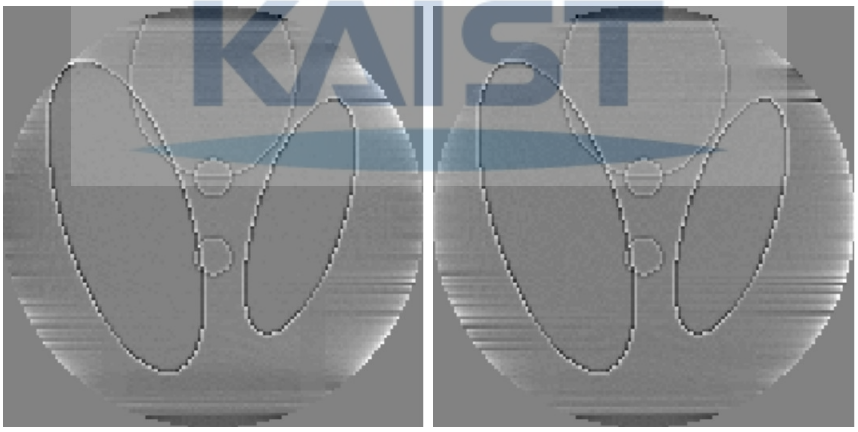
A simulation geometry for the interior problem of CT with *a priori* knowledge is described in Figure 3.1. The FOV is completely inside the object, and we have small known regions in FOV. By decomposing the 2D image into the 1D horizontal lines after applying DBP, we can cover all FOV with lines intersecting the two known regions except the small top and bottom areas, and a set of 1D problems corresponding to each line is reconstructed using PPXA. For simulation, we use 2D Shepp-Logan phantom, and the

size of the object is 256×256 pixels with 0.5×0.5 mm^2 size of the pixel. The 800 views of projections are sampled for 360°.

The reconstruction results of POCS and PPXA after 100 iterations are shown in Figure 3.2. The reconstruction images of both algorithms look similar, but in the line profile, the result of PPXA is more accurate. In the line profile of the POCS result, we can see the small known regions where the intensity values are accurate. Besides those regions, the intensities of the object still have differences from the ground truth. However, the line profile of the PPXA result is almost similar to the original phantom.



(a) POCS_{100} (b) POCS_{200}



(c) PPXA_{100} (d) PPXA_{200}

Figure 3.3: Error images after 100 or 200 iterations.

Figure 3.3 shows the difference images between reconstruction and the ground truth. All images are displayed in same display scale, and the intensity value same as the background represents no error. In POCS results, the two gray lines are apparent, which means the other areas are not accurate. The error after 200 iterations decreases, but is still significant. In contrast, in PPXA case, the error only after 100 iterations does not have distinguishable known areas because the areas near the known regions are very accurate, and the overall error after 100 iterations is much less than the error of POCS after 200 iterations. The convergence in term of mean square error (MSE) is shown in Figure 3.4. POCS method has overshoot point at the iteration number about 20, and PPXA has two big overshoots before about 50 iterations. Note that MSE value is not necessary to decrease monotonically since this is not what we minimize. We cannot plot the cost functions either since the cost functions are sum of indicator

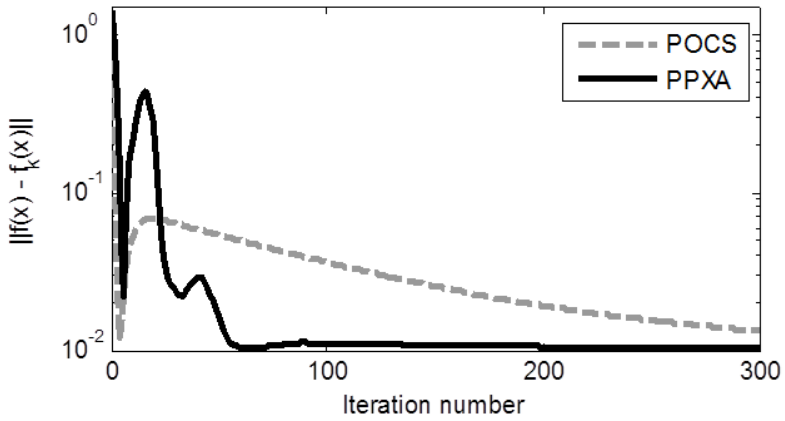


Figure 3.4: MSE convergence of POCS and PPXA.

functions that have infinite values until convergence. Therefore, the overshoots in MSE domain are normally expected. Even though there exist overshoots of MSE values during the initial few iterations, after the overshoots, both algorithms converge, but PPXA converges much faster than POCS. Using PPXA, after only 80 iterations, convergence has been achieved.

KAIST

Chapter 4. Interior tomography using 1D generalized total variation penalty and multiscale implementation by Bedrosian identity

4.1 Interior tomography using 1D generalized total variation

To overcome the limitation of the interior tomography using inside known subregions, our previous work [35] introduced a generalized 1D TV penalty minimization as a simplification of multidimensional TV in [31, 32, 33]. Specifically, we consider a regularization with respect to a Fourier multiplier operator L that is defined on $L_2(\mathbb{R})$ and satisfies two conditions. First, for any interval $E = (e^-, e^+) \subset \mathbb{R}$, we require L to map $C_c^\infty(E)$ to $C_c(E)$. Second, the null space of L, denoted as $\mathcal{N}_\mathrm{L}$, should consist of entire functions. An example of such an operator is a constant coefficient differential operator

$$\mathrm{L} := a_K \mathrm{D}^K + a_{K-1}\mathrm{D}^{K-1} + \ldots + a_1 \mathrm{D} + a_0 \tag{4.1}$$

where $K \geq 1$, D denotes the distributional derivative on $\mathbb{R}$, and each a_k is a real number. In this example, the finite-dimensional null space consists of linear combinations of exponential functions multiplied by polynomials. For an operator L and an interval $E \subset \mathbb{R}$, we formally define the *generalized TV seminorm*

$$\|f\|_{TV(\mathrm{L};E)} := \|\mathrm{L}f\|_{L_1(E)} \ , \tag{4.2}$$

which is valid when $\mathrm{L}f \in L_1(E)$. In order to ensure that this seminorm is valid for a larger class of functions, we use the dual formulation

$$\|f\|_{TV(\mathrm{L};E)} := \sup_{h \in C_h} \int_E \mathrm{L}f(x)h(x)\ dx \tag{4.3}$$

where $C_h = \{h \in C_c^\infty(E), \|h\|_{L_\infty} \leq 1\}$ and L is (4.1), and this seminorm can be also written as

$$\|f\|_{TV(\mathrm{L};E)} := \sup_{h \in C_h} \int_E f(x)\mathrm{L}^* h(x) dx \tag{4.4}$$

where the adjoint L^* is

$$\mathrm{L}^* = (-1)^K a_K \mathrm{D}^K + (-1)^{K-1} a_{K-1}\mathrm{D}^{K-1} + \ldots - a_1 \mathrm{D} + a_0 \ . \tag{4.5}$$

As in [31], our results are based on the fact that $f_\mathcal{N}(x)$ in (3.5) is infinitely smooth in E. Suppose, furthermore, the signal $f(x)$ to be reconstructed is a generalized L-spline, where L is a finite-order operator. It is this disparity between the infinitely smooth $f_\mathcal{N}(x)$ and the finitely smooth $f(x)$ that allows us to have perfect reconstruction.

Theorem 4.1.1. *Let $f_0(x)$ be a generalized* L*-spline such that*

$$\mathrm{L}f_0(x) = \sum_{n=1}^{N} a_n \delta(x - x_n) \tag{4.6}$$

on E. Then, the minimization problem

$$\arg\min_{f} \|f\|_{TV(\mathrm{L};E)} \quad \textit{subject to} \quad \mathcal{H}f_0(x) = \mathcal{H}f(x), \quad x \in E \tag{4.7}$$

has the unique solution equal to $f_0(x)$.

Proof. See our companion paper [35]. □

For reconstruction under generalized TV, our previous paper [35] used POCS algorithm with five constraint sets. Note that this optimization framework is based on continuous formulation in the function space; however, the algorithm can be implemented in a discrete setting for practical applications. Specifically, the constraint sets $\{C_j\}_{j=1}^5$ are given by

$$C_1 = \{f(x) \in L^2(\mathbb{R}) : f(x) = 0, x \notin (b^-, b^+)\} \tag{4.8a}$$

$$C_2 = \{f(x) \in L^2(\mathbb{R}) : \|f\|_{TV(\mathrm{L};E)} \le \tau\} \tag{4.8b}$$

$$C_3 = \{f(x) \in L^2(\mathbb{R}) : \mathcal{H}f(x) = g(x), x \in (e^-, e^+)\} \tag{4.8c}$$

$$C_4 = \{f(x) \in L^2(\mathbb{R}) : \frac{1}{\pi}\int_{b^-}^{b^+} dx\ f(x) = c_f\} \tag{4.8d}$$

$$C_5 = \{f(x) \in L^2(\mathbb{R}) : f(x) \ge 0, x \in (b^-, b^+)\}\ . \tag{4.8e}$$

where (b^-, b^+) denotes support of the object, and c_f can be obtained from $D_f(\mathbf{a}, \boldsymbol{\theta})$. The additional constraint beyond the necessity in proving uniqueness in theorem 4.1.1 is to make the algorithm converge faster. The optimized solution of $f(x)$ can be then solved by (3.7).

Here, the projections to the constraints C_1 and C_5 are quite straightforward, and we can just impose support constraint and the nonnegativity for the intermediate reconstruction. For the projection on the constraints C_3 and C_4, we can use the finite Hilbert transform formula such as (3.3). Now, the most complicated step is to project on the generalized TV ball. It turns out that the projection on C_2 from a point f_0 can be implemented as the following denoising step:

$$\hat{f} = \arg\min_{f \in C_f} \left\{ \|f - f_0\|_2^2 + 2\lambda \|f\|_{TV(\mathrm{L};E)} \right\} \tag{4.9}$$

for an appropriate Lagrangian parameter. In (4.9), $C_f = B_{l,u}$ is the bound constraint as

$$B_{l,u} = x : l \le x \le u\ . \tag{4.10}$$

Using the definition of the generalized TV seminorm in (4.4), the optimization problem is given by

$$\sup_{h \in C_h} \min_{f \in C_f} \left\{ \|f - f_0\|_2^2 + 2\lambda \langle f, \mathrm{L}^* h \rangle \right\}\ . \tag{4.11}$$

The inner minimization problem can be solved by

$$\begin{aligned} \frac{\partial}{\partial \hat{f}} \left[\|f - f_0\|_2^2 + 2\lambda \langle f, \mathrm{L}^* h \rangle \right] &= 0 \\ 2(\hat{f} - f_0) + 2\lambda \mathrm{L}^* h &= 0 \\ \hat{f} &= P_{C_f}(f_0 - \lambda \mathrm{L}^* h) \end{aligned} \tag{4.12}$$

where P_{C_f} is projection onto $C_f = B_{l,u}$, and P_{C_f} can be implemented as following (4.13).

$$P_{C_f}(f) = \begin{cases} l, & f < l \\ f, & l \le f \le u \\ u, & f > u \end{cases} \tag{4.13}$$

Before substituting $\hat{f}$ into (4.11), the cost function in (4.11) is reformulated as [47, 48]

$$\begin{aligned}\|f-f_0\|_2^2+2\lambda\langle f,\mathrm{L}^*h\rangle &= \int dx\ (f(x)-f_0(x))^2+2\lambda\int dx\ f(x)\mathrm{L}^*h(x)\\ &= \int dx\ \Big[(f(x))^2-2f(x)(f_0(x)-\lambda\mathrm{L}^*h(x))+(f_0(x)-\lambda\mathrm{L}^*h(x))^2\Big]\\ &\qquad -\int dx\ (f_0(x)-\lambda\mathrm{L}^*h(x))^2+\int dx\ (f_0(x))^2\\ &= \|f-(f_0-\lambda\mathrm{L}^*h)\|_2^2-\|f_0(x)-\lambda\mathrm{L}^*h(x)\|_2^2+\|f_0\|_2^2\ . \end{aligned}\tag{4.14}$$

Substituting $\hat{f}$ in (4.12) into (4.14) yields

$$\sup_{h\in C_h}\left\{\|P_{C_f}(f_0-\lambda\mathrm{L}^*h)-(f_0-\lambda\mathrm{L}^*h)\|_2^2-\|f_0(x)-\lambda\mathrm{L}^*h(x)\|_2^2\right\}\ ,\tag{4.15}$$

and the constant term $\|f_0\|_2^2$ is omitted. By multiplying -1, the supremization problem (4.15) is changed into a minimization problem as

$$\min_{h\in C_h}\left\{-\|P_{C_f}(f_0-\lambda\mathrm{L}^*h)-(f_0-\lambda\mathrm{L}^*h)\|_2^2+\|f_0(x)-\lambda\mathrm{L}^*h(x)\|_2^2\right\}\ ,\tag{4.16}$$

and considering the objective function of (4.16) as $q(h)$, the gradient of $q(h)$ is

$$\begin{aligned}\nabla q(h) &= \nabla\Big[-\|P_{C_f}(f_0-\lambda\mathrm{L}^*h)-(f_0-\lambda\mathrm{L}^*h)\|_2^2+\|f_0(x)-\lambda\mathrm{L}^*h(x)\|_2^2\Big]\\ &= 2\lambda\mathrm{L}\Big(P_{C_f}(f_0-\lambda\mathrm{L}^*h)-(f_0-\lambda\mathrm{L}^*h)\Big)+2\lambda\mathrm{L}(f_0-\lambda\mathrm{L}^*h)\\ &= 2\lambda\mathrm{L}P_{C_f}(f_0-\lambda\mathrm{L}^*h)\ .\end{aligned}\tag{4.17}$$

Then, the optimal solution $\hat{h}$ can be obtained by gradient projection method

$$h_k=P_{C_h}\left(h_{k-1}-\frac{2}{L(q)}\lambda\mathrm{L}P_{C_f}(f_0-\lambda\mathrm{L}^*h_{k-1})\right)\tag{4.18}$$

when P_{C_h} denotes the projection on the convex set C_h, and $L(q)$ denotes the Lipschitz constant of $\nabla q(h)$.

For given f_0, L (the upper bound of $L(q)$), and iteration number N, the algorithm of the TV denoising problem can be obtained by gradient projection (GP) as Algorithm 4.1.1. For superlinear convergence rate, we implement the fast gradient projection (FGP) algorithm based on Nestrov method as Algorithm 4.1.2 [49, 47, 48].

Algorithm 4.1.1: $\mathrm{GP}(f_0,\lambda,L,N)$

$h_0=0$
for $k\leftarrow 1$ **to** N
  do $h_k=P_{C_h}\left(h_{k-1}-\frac{2}{L}\lambda\mathrm{L}P_{C_f}(f_0-\lambda\mathrm{L}^*h'_{k-1})\right)$
$\hat{f}=f_0-\lambda\mathrm{L}^*h_N$
return $(\hat{f})$

KAIST

Algorithm 4.1.2: FGP(f_0, λ, L, N)

$h_0 = h'_1 = 0$
$t_1 = 1$
for $k \leftarrow 1$ **to** N
do $\begin{cases} h_k = P_{C_h}\left(h'_k - \frac{2}{L}\lambda \mathrm{L} P_{C_f}(f_0 - \lambda \mathrm{L}^* h_{k-1})\right) \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2} \\ h'_{k+1} = h_k + \left(\frac{t_k - 1}{k_{k+1}}\right)(h_k - h_{k-1}) \end{cases}$
$\hat{f} = f_0 - \lambda \mathrm{L}^* h_N$
return $(\hat{f})$

4.2 Multiscale implementation by Bedrosian identity

The interior tomography using a 1D formulation described so far can make the optimization problem much simpler than the interior tomography using high-dimensional TV[31, 32, 33]. This was the main implementation approach taken in our previous work [35]. However, the method was still computationally expensive for practical use.

To accelerate the reconstruction speed to a level for clinical use, in this section, we propose a multiscale decomposition method by exploiting the powerful Bedrosian identity in the Hilbert transform. This idea is novel and was not addressed in our previous work [35]. More specifically, we assume that the object $f(x)$ is decomposed as

$$f(x) = f_L(x) + f_H(x) \tag{4.19}$$

where f_L and f_H denote the appropriate low and high frequency components of f respectively, and f_L is well modeled as an L-spline for some Fourier multiplier operator L. Under this assumption, our multiscale interior tomography problem is formulated as follows:

$$\min_{f_L} \|f_L\|_{TV(\mathrm{L};E)} \quad \text{subject to} \quad g = \mathcal{H}\{f_L + f_H\} \ . \tag{4.20}$$

It is important to note that (4.20) is more realistic than (4.7) because images usually have textures or details that are often difficult to capture by a generalized TV. Moreover, even if the texture can be modeled as a higher order TV, the formulation in (4.20) still enables us to use a computationally more efficient lower order TV, as long as the remaining high frequency components can be reconstructed separately. In fact, such separate reconstruction of a high frequency component can be done accurately thanks to the Bedrosian theorem of the Hilbert transform.

Theorem 4.2.1 (Bedrosian [38])**.** *Let $w, f \in L_2(\mathbb{R})$ be low-pass and high-pass signals such that the Fourier transform of $w(x)$ vanishes for $|\omega| > \omega_0$, with $\omega_0 > 0$, and the Fourier transform of $f(x)$ vanishes for $|\omega| < \omega_0$. Then, we have*

$$\mathcal{H}\{w(x)f(x)\} = w(x)\mathcal{H}\{f(x)\} \ . \tag{4.21}$$

In order to appreciate the role of the Bedrosian theorem for the truncated Hilbert transform problem, note that the truncated DBP data can be represented as the product of the indicator function $w(x)$ that represents the truncation support and the untruncated DBP data $f(x)$ that correspond to the Hilbert

transform of the unknown signal. Then, (4.21) informs us that we can analytically recover the high frequency component f_H which does not overlap with the spectrum of $w(x)$.

More specifically, after reconstructing $f_L(x)$ by solving (4.20), the residual signal $g_H(x) = g(x) - \mathcal{H}f_L(x)$ can be obtained, which is equal to $\mathcal{H}f_H(x)$. Because we know $g_H(x)$ only in $E = (e^-, e^+)$, this can be considered as a multiplication of the rectangular window:

$$w(x) = \begin{cases} 1, \ x \in E \\ 0, \ otherwise \end{cases} \tag{4.22}$$

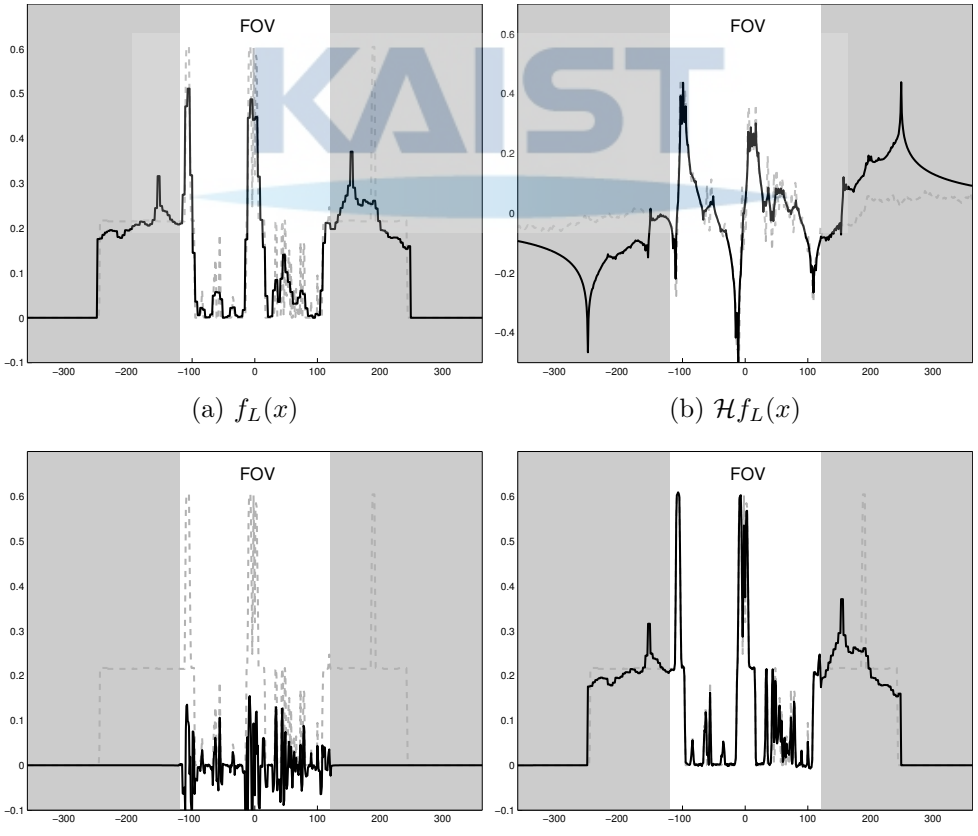
corresponding to the indicator function of the FOV whose spectral band is usually at low frequency ranges. If the spectral support of $g_H(x)$ does not overlap with that of $w(x)$, according to the Bedrosian theorem, we have

$$w(x)g_H(x) = w(x)\mathcal{H}f_H(x) = \mathcal{H}\{w(x)f_H(x)\} \ . \tag{4.23}$$

From (4.23), $f_H(x)$ can be analytically solved by

$$f_H(x) = \frac{-\mathcal{H}\{w(x)g_H(x)\}}{w(x)}, \quad x \in E \ , \tag{4.24}$$

which corresponds to applying the Hilbert transform to the truncated residual DBP data and dividing the result by the window.



(a) $f_L(x)$ (b) $\mathcal{H}f_L(x)$

(c) $f_H(x)$ (d) $I(x) = f_L(x) + f_H(x)$

Figure 4.1: Step-by-step reconstruction procedure along a PI line.

Figure 4.1 shows a step-by-step illustration of the multiscale decomposition method on a chord line. First, Figure 4.1(a) shows the low resolution signal $f_L(x)$ calculated by solving (4.20). Figure 4.1(b)

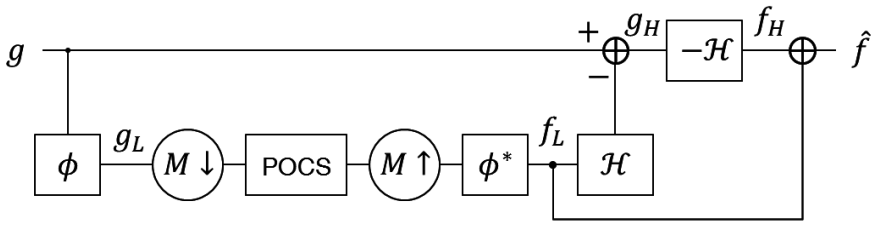


Figure 4.2: Flowchart of the proposed multiscale reconstruction approach for interior tomography problem.

shows the Hilbert transform of $f_L(x)$, whereas the gray dotted line is the original DBP data. Now, we compute the high-frequency component g_H of the DBP data as the residual signal between the original DBP data and $\mathcal{H}f_L(x)$. Then, by taking the inverse Hilbert transform of $g_H(x)$, we can obtain the high frequency signal $f_H(x)$ shown in Figure 4.1(c). By summing up the signals in Figures 4.1(a) and (c), the final full-resolution reconstruction is obtained in Figure 4.1(d).

Note that the regularized reconstruction method using the TV constraint in (4.20) needs to be solved using an iterative method, and we use the POCS algorithm. However, the POCS reconstruction method using the TV constraint often exhibits slow convergence in recovering high frequency signals. Interestingly, the multiscale decomposition from the Bedrosian identity also provides a remedy for this because an iterative approach can be used to recover only the low frequency component $f_L(x)$, and the iterative recovery of f_L can be further accelerated by downsampling.

More specifically, for a given ROI window $w(x)$ with the bandwidth of ω_0, the measured DBP data are low-pass filtered with the bandwidth ω_0 to obtain g_L (see Figure 4.2). Because the signal content of g_L is now low-passed, g_L can be downsampled without aliasing at an appropriate downsampling factor. Then, POCS is applied to obtain a reconstruction in a downsampled domain, which is then upsampled to the original resolution to obtain the low frequency reconstruction f_L. The downsampling and upsampling procedure can be easily implemented by wavelet decomposition and reconstruction. Specifically, we should use the matched wavelets to the generalized TV operator L in (4.1). For example, if we use the first order TV operator (which is the case in our numerical experiments), the matched wavelet is the Haar wavelet. Note that the computationally expensive POCS operation is implemented in a downsampled domain, which is computationally much more efficient and provides faster convergence.

4.3 Implementation

In this chapter, we explain how to implement the proposed algorithm with digital projection images. The proposed method was verified using numerical experiments. We implemented the proposed multiscale interior tomography algorithm for 2D fan-beam CT, 3D helical cone-beam CT. To demonstrate that the algorithm is very practical for clinical applications, all the numerical experiments were performed using typical clinical CT geometry parameter settings.

The process of the proposed interior tomography algorithm is same with the BPF-type cone-beam reconstruction except the filtration step in Figure 2.16 and Figure 2.17. First, the all projection images are differentiated with respect to the source trajectory. Then, we backproject the differentiated images on chord or PI lines of (λ^-, λ^+) covering the ROI with enough resolution. In filtration step, we do not have enough DBP signal to cover the whole object so that it will be implemented with multiscale interior

tomography using 1D generalized TV. After all chord or PI lines are reconstructed, the regridding process is performed to get the final reconstruction in the Cartesian coordinate space.

4.3.1 Differentiated backprojection

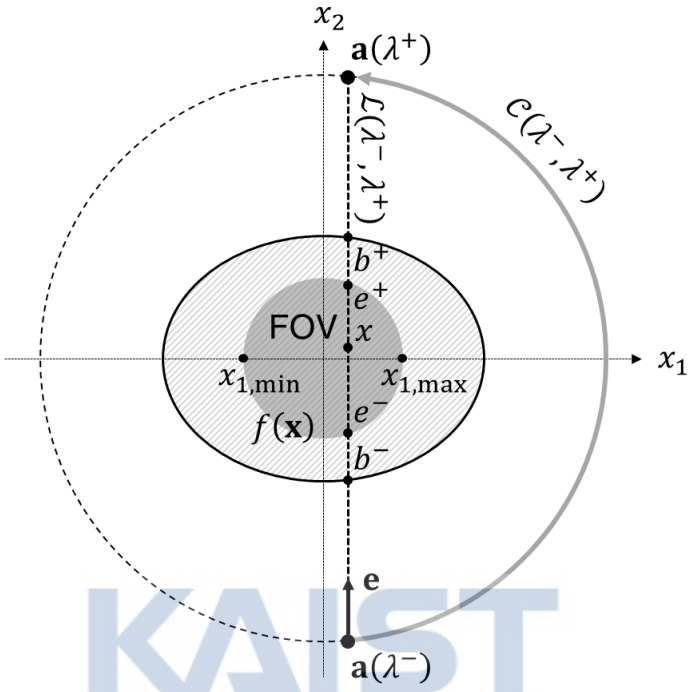


Figure 4.3: The implementation of the circular fan-beam interior tomography.

In the circular fan-beam CT, we made vertical chord lines in section 2.3.2, and these chord lines are determined by x_1. For interior tomography, this x_1 is sampled from $x_{1,min}$ to $x_{1,max}$ to cover only the FOV not the whole object in Figure 4.3. This is one of the advantage of the interior tomography using the Hilbert transform relationship and DBP signal. The chord line $\mathcal{L}(\lambda^-, \lambda^+)$ is determined, then lastly, x is sampled to cover the whole object. For a given point x on the chord line $\mathcal{L}(\lambda^-, \lambda^+)$, the differentiated signals are backprojected multiplying the distance weight $1/\|\mathbf{x} - \mathbf{a}(\lambda)\|$, and at this time, the differentiated signal is necessary only for the trajectory segment $\mathcal{C}(\lambda^-, \lambda^+)$.

In the helical cone-beam CT, to define the PI lines, firstly λ^- is sampled along helical trajectory. At this time, the increment of $\mathbf{a}(\lambda^-)$ along the x_3-axis should be shorter than the sampling distance along the x_3-axis in 3D Cartesian coordinate system. Secondly, λ^+ is sampled from $\lambda^+_{\min}$ to $\lambda^+_{\max}$ so that the set of PI lines can cover only the FOV not whole object as Figure 4.4. Lastly, x is sampled to cover the whole object. As λ^-, grid of (λ^+, x) should be more narrow than horizontal plane. For backprojection of a given $\mathbf{x}$ on PI line $\mathcal{L}_\pi(\lambda^-, \lambda^+)$, the differentiated images are required only for helix segment $\mathcal{C}_\pi(\lambda^-, \lambda^+)$ as the circular trajectory. In both case of the circular fan-beam and helical cone-beam CT, even if x cover the whole object, the backprojection signal is necessary only for (e^-, e^+) because the backprojection outside (e^-, e^+) cannot be computed correctly.

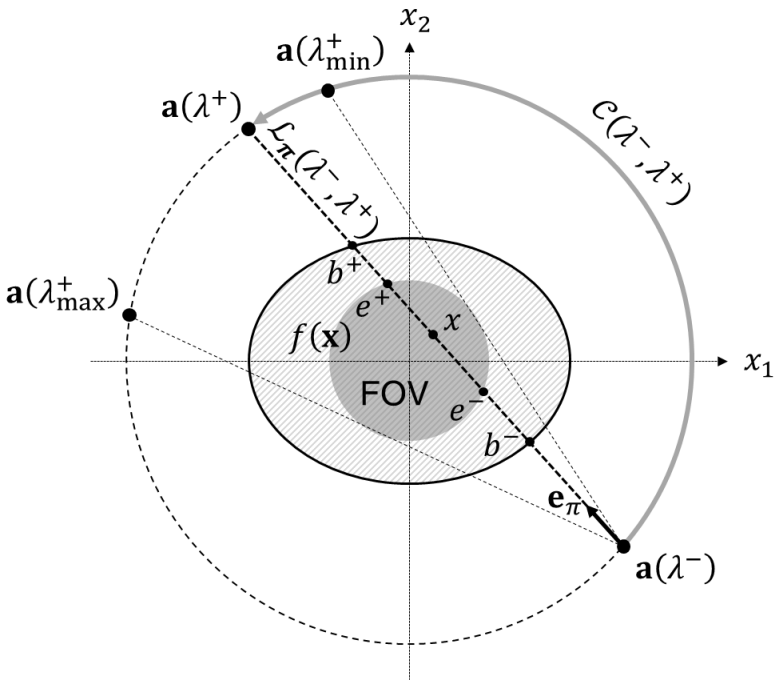


Figure 4.4: The implementation of the helical cone-beam interior tomography.

4.3.2 Multiscale decomposition

To recover the low frequency signal, we first take downsampling of the original DBP by wavelet decomposition and taking the lowest frequency band. For wavelet decomposition, zero-padding could be required to make the size of DBP signal power of two. From the downsampled DBP signal, the low frequency part of signal is reconstructed using POCS and FGP algorithm, and the reconstructed signal is then upsampled to the original resolution signals. This procedure takes much less time than iterative reconstruction of the original full-resolution DBP signal.

After low frequency reconstruction, the reconstructed low frequency signal is upsampled using wavelet reconstruction. The upsampled low resolution image is then transformed using Hilbert transform to extract the high frequency residual DBP signals. Then, the resulting high frequency residual signal is applied to an Hilbert transform to obtain high frequency signal. Finally by adding the reconstructions of low and high frequency, we can get the interior tomography with the original resolution (see figure 4.2).

4.4 Simulation results

In the POCS algorithm, we need the boundary of an object support, but the exact boundary is hardly known in real scanning cases. Thus, for a realistic simulation, we assumed that only the maximum width and height of the body (W_{body}, H_{body}) in the transverse planes are known, and then the approximate boundary is defined as an ellipse with a bigger width and height than that of W_{body} and H_{body}. The width and height of the ellipse are indicated as W_{ellipse} and H_{ellipse} in

$$W_{\text{ellipse}} = W_{\text{body}} + L_{\text{extension}} \tag{4.25a}$$

$$H_{\text{ellipse}} = H_{\text{body}} + L_{\text{extension}} \tag{4.25b}$$

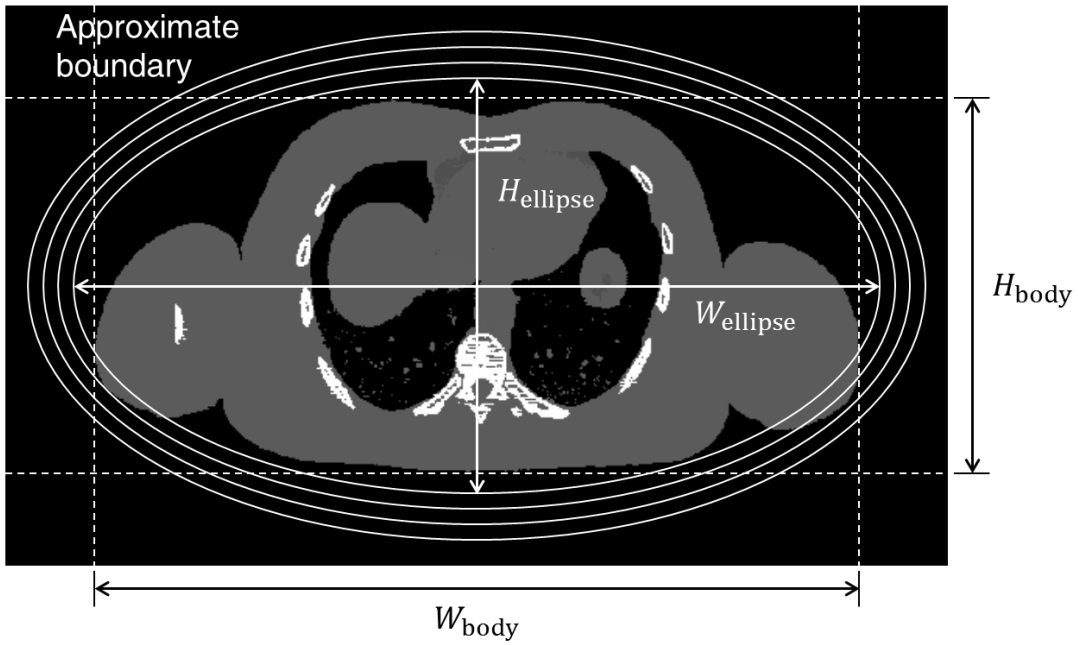


Figure 4.5: Approximate ellipse boundary.

with the positive real number $L_{\text{extension}}$. As in Figure 4.5, we can define ellipses with various $L_{\text{extension}}$. In some cases, the ellipse could not cover the whole support such as the arms. In the reconstruction simulation of a circular fan-beam and helical cone-beam CT, we used the ellipse boundary when $L_{\text{extension}}$ was 60 mm.

4.4.1 Circular fan-beam CT

The first reconstruction result was obtained from a 2D circular fan-beam projection, in which the projection data at each view were truncated. For the simulation, we used a 2D inner organ phantom, as shown in Figure 4.5, which is 512×512 in size with 1×1 mm^2 size pixel. The number of detectors in the array was 380 with a 1 mm pitch, and the number of views was 1200. The distance from the source to the rotation axis was 800 mm, and the distance from the source to the detector was 1400 mm. The radius of the ROI was about 107 mm which cannot cover the whole object. For the multiscale implementation, we used the first order operator L in (4.1), which corresponds to the conventional TV penalty. The iteration number of the POCS was 100. For Algorithm 4.1.2, the l and u in (4.10) were 0 and 0.060442, respectively, for the inner organ phantom, and the λ was 0.001, and N was 20. For the multiscale implementation, the depth of Haar wavelet decomposition was 2.

In Figure 4.6, the first image in (a) is the original ground truth. Figure 4.6(b)-(c) show the low and high frequency reconstruction images, respectively, and the final reconstruction is shown in Figure 4.6(d), which is nearly identical to the original image.

4.4.2 Helical cone-beam CT

The second experiment is from a 3D cone-beam projection with a helical scanning trajectory, where the projections at each view extend only to the shadow of a cylindrical region completely interior to the subject being scanned. This situation is important for low-dose cardiac CT imaging using a clinical

(a) Ground truth (b) f_L

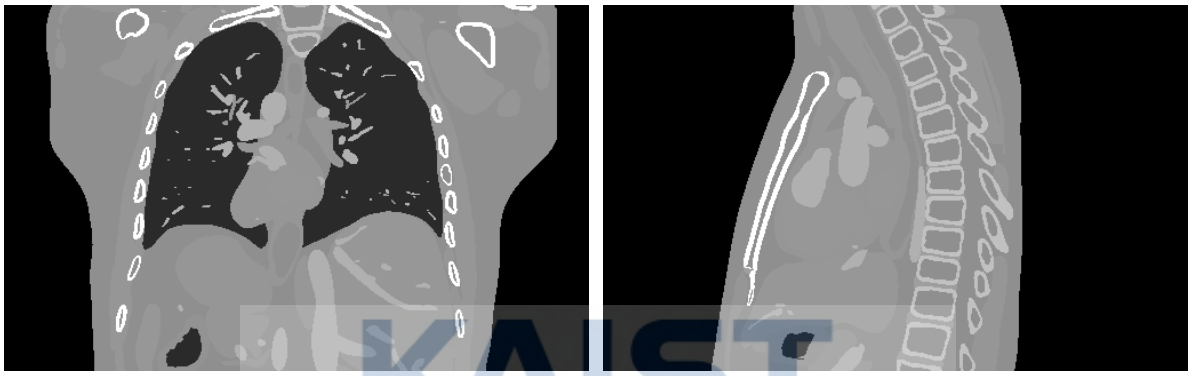
(c) f_H (d) $f_L + f_H$

Figure 4.6: 2D fan-beam reconstruction. For the simulation, the K of the L is 1, and the iteration number of the POCS is 100. The l and u are 0 and 0.060442, respectively, and the λ is 0.001, and N is 20. The depth of the multiscale implementation is 2. (a) Ground truth, (b) low frequency signal reconstruction, (c) high frequency signal reconstruction, and (d) the final reconstruction result.

multislice helical CT.

For helical cone-beam simulation, we used two kinds of 3D phantoms: XCAT and inner organ phantoms. For comparison, we implemented two types of analytic exact cone-beam reconstruction methods: the Katsevich-type FBP [2] and BPF algorithm, and the fully iterative reconstruction method using the 3D TV regularizer. The first result is for the XCAT phantom, and its resolution was $512 \times 512 \times 320$ voxels with a voxel size $1 \times 1 \times 1$ mm^3. Figure 4.7 shows the transverse, coronal, and sagittal planes of the phantom. The distances from the source to the rotation axis and from the source to the detector were the same as in the circular fan-beam simulation. The detector resolution was 400×109 pixels with

Transverse



Coronal Sagittal

KAIST

Figure 4.7: 3D XCAT phantom.

a pitch of 1×1 mm^2; thus the radius of the ROI was about 127 mm. For the helical scanning trajectory, the number of rotations was 3, the helical pitch H was 100 mm, and the number of views was 800 per rotation. The object was reconstructed in the PI line space first, and the resolution of the PI line space to cover the ROI was $727 \times 245 \times 339$. After reconstructing all the PI lines, the final reconstruction of about 236 slices was obtained by the regridding process. For the simulation parameters, the K of the L, the iteration number of the POCS, λ, N, and the depth were all the same as the fan-beam simulation, and the l and u were 0 and 0.030888, respectively.

Figure 4.8 shows the reconstruction result in the ROI by FBP, BPF, 3D TV, and the proposed method. From the first to third columns, the first row is for the ideal phantom, and the second to fourth rows are for the reconstructed images by FBP, BPF, 3D TV, and the proposed method, respectively. As shown in the images of Figure 4.8, it is clearly observed that the FBP images are much darker compared to the original phantom, and the BPF images are still dark, while the images by our proposed method show almost the same intensity as the original. Even when comparing the result of the 3D TV method, the proposed method is competitive. This can also be seen in the cut view, which is the last column of Figure 4.8. Moreover, the root-mean-square error (RMSE) and normalized root-mean-square error (NRMSE) values indicated on the images provide clear quantitative evidence.

The second result is for the noisy simulation of the 3D inner organ phantom. Figure 4.9 shows the transverse, coronal, and sagittal planes of the phantom. To make the noisy projection data, we assumed the detector measurements have a Poisson distribution, and the signal-to-noise ratio (SNR)

was set to about 30dB in the intensity domain. The helical CT geometry was the same as before, and the hyperparameters for the proposed interior tomography algorithm were the same as those for the 2D experiments.

As shown in Figure 4.10, the FBP and BPF methods still show dark images, and the 3D TV method and the proposed method recover the images correctly. However, there is a difference between the 3D TV and the proposed algorithms. Compared to the reconstruction without truncation, the recovered textures of the proposed algorithm are very similar, which indicates that the algorithm did not provide unnatural smoothings. On the other hand, the 3D TV algorithm reduces not only the noise of the image but also the details. The noise reduction is necessary for some cases, but it can also remove important structures of the image. Our proposed algorithm conserved the details of the image thanks to the analytic reconstruction of the high frequency details.

	Proposed		
TR	56.1%	60.9%	65.8%
DBP	14.1	13.7	13.3
Downsampling	0.7	0.6	0.5
f_L reconstruction	85.4	170.4	335.1
Upsampling	0.3	0.3	0.2
f_H reconstruction	9.1	8.9	8.6
Regridding	2.7	2.5	2.2
Total (236 slices)	112.3	196.4	359.9
Speed (slices/second)	2.102	1.202	0.656

Table 4.1: Computational time (seconds) by truncation rate for helical cone-beam CT when depth is 2.

	3D TV	Proposed		
Depth		0	1	2
DBP		13.7	13.8	13.7
Downsampling		0	0.4	0.6
f_L reconstruction		1418.5	558.5	170.4
Upsampling		0	0.1	0.3
f_H reconstruction		8.5	8.4	8.9
Regridding		2.6	2.5	2.5
Total (236 slices)	3141.2	1443.3	583.7	196.4
Speed (slices/second)	0.075	0.164	0.404	1.202

Table 4.2: Computational time (seconds) by depth for helical cone-beam CT when truncation rate is 60.9%.

Tables 4.1 and 4.2 show the computational time of each process step in the case of the 3D inner organ phantom simulation. In Table 4.1, time was measured by increasing the truncation rate (TR) with

System	Specification
CPU	Intel Core i7-4700 Haswell Quad-Core 3.40GHz
RAM	Samsung 16GB kit (4 × 4GB) DDR3 PC3-128000, 1600MHz
GPU	GeForce GTX TITAN
OS	Windows 7 Enterprise with CUDA 5.5

Table 4.3: Whole workstation environment for simulating the proposed method.

a depth of 2, and the TR corresponds to

$$\mathrm{TR} = \frac{l - l_T}{l} \times 100\% \tag{4.26}$$

where l and l_T denote the length of the original and truncated detector, respectively. For reference, the result in Figure 4.10 is for the case in which the truncation rate was 60.9%. All processes were implemented by a GPU to accelerate the reconstruction speed, and a whole workstation environment is presented in Table 4.3. Table 4.1 shows more computational burden with a more severe truncation due to the increase in the iteration number for convergence. Because the time for the low frequency reconstruction is the longest compared to all the others, the total amount of time mainly consists of the low frequency part.

On the other hand, Table 4.2 shows a reduced computational time with more multiresolution decomposition. Table 4.2 also shows the computational time of the 3D TV method. In Table 4.2, the computational time of the 3D TV method is quite long because the process for the 3D helical projection and backprojection is repeated. However, the computation time of the proposed method is an order of magnitude faster than that of the 3D TV method. Moreover, by increasing the depth, the time for the low frequency reconstruction rapidly decreases to a practical level. Table 4.2 confirms that the computational time is quite reasonable (1.2 slices/second) even for clinical multislice helical CT applications. This is because of the multiscale decomposition which leads to a dramatic reduction in the reconstruction time.

4.5 Discussion

4.5.1 Comparison with extrapolation method

To demonstrate the limitation of the conventional method using sinogram extrapolation [50, 51], we compared the conventional method and our proposed method by increasing the $L_{\text{extension}}$ in (4.25). In our POCS implementation, we used information from the approximate ellipse boundary, and thus in the extrapolation method, we also used the same approximate ellipse boundary to determine the extension length of the sinogram. The extrapolation method is sensitive to the extension length; therefore, by changing the size of the ellipse, the accuracy can be also changed. For extrapolation, we chose the square root function $\sqrt{a\xi^2 + b\xi + c}$ which was used in [51], and from the extrapolated sinogram, the object was reconstructed by FBP and BPF. Figure 4.11 shows the error images of the 3D inner organ phantom by extrapolated FBP, extrapolated BPF, and the proposed method when $L_{\text{extension}}$ was 140 mm, and all images are displayed with the same gray-scale map. Figures 4.11(a)-(c) have a similar high frequency error, but the low frequency error appears only in Figures 4.11(a)-(b). Figure 4.12 represents the NRMSE of the reconstructed result by extrapolated FBP, extrapolated BPF, and the proposed method as the

$L_{\text{extension}}$ increases. When the $L_{\text{extension}}$ was 60 mm, the errors by all methods were similar, but as the $L_{\text{extension}}$ increased, the extrapolated methods had much bigger errors than that of the proposed method. As shown in Figures 4.11 and 4.12, our proposed method is more stable than the extrapolation methods.

4.5.2 Downsampling Factor

Previously, we mentioned that if the low frequency signal $f_L(x)$ is band-limited with the bandwidth less than ω_0, then the corresponding DBP data $g_L = -\mathcal{H}f_L$ should have the same bandwidth. However, one of the main technical difficulties herein is that for the truncated DBP signal, the converse is not true. More specifically, if we apply the low-pass filter to $w(x)g(x)$ to obtain $g_L(x)$ whose bandwidth is ω_0, the recovered signal $f_L(x)$ from the truncated Hilbert transform under 1D TV penalty does not guarantee recovery of all frequency content of $f(x)$ up to ω_0. This is because our reconstruction method is nonlinear due to the generalized TV penalty to deal with the data truncation. Consequently, the residual signal $g(x) - (-\mathcal{H}f_L(x))$ should still have lower frequency content below ω_0. This indicates that the bandwidth ω_0 should be determined very conservatively compared to the linear recovery case. Here, we will explain this issue in detail.

Suppose that the DBP measurement is not truncated. In this case, the low frequency content f_L can be recovered using the Hilbert transform of g_L, and it is clear that the spectrum of the residual signal $g_H = g(x) - (-\mathcal{H}f_L(x))$ vanishes below ω_0. Therefore, from the window $w(x)$ for the truncated DBP, we can obtain its spectral bandwidth of the window using the standard way of calculating the bandwidth (for example, full-width-half-maximum, or -10dB, -20dB bandwidth, etc). Then, g_L is obtained with low-pass filtering of $g(x)$ with the filter with bandwidth of ω_0 and downsampling to π/ω_0 without aliasing; as the signal is not truncated, the low frequency signal f_L with spectral bandwidth ω_0 can be obtained from the downsampled DBP data using the Hilbert transform and upsampling.

However, if the DBP is truncated, $g_H = g(x) - (-\mathcal{H}f_L(x))$ does contain frequencies below ω_0. Therefore, ω_0 should be set much more conservatively compared to the usual definition of the spectral bandwidth. This implies that the downsampling factor for f_L recovery can be reduced accordingly. Furthermore, to suppress the spectral overlap between the window $w(x)$ and $g_H(x)$, we can use a higher order window with a faster decay rate. Accordingly, we use a third order spline window, as shown in Figure 4.13. The advantage of using a third order spline rather than first or second order spline is that the decay of its spectrum is faster. As a rule of thumb, we found that -65dB bandwidth is sufficient enough for our purpose of determining the cut-off frequency ω_0.

Figures 4.14(a)-(b) show reconstruction results with a large and small FOV, respectively; and Figures 4.14(c)-(d) are the spectrum of $w(x)$ and $g_H(x)$ for the large and small FOVs. Since the size of the window varies with size of the FOV, so the bandwidth of the window spectrum varies with the FOV, too. More specifically, a larger $w(x)$ has a smaller bandwidth. Accordingly, if we set the cut-off frequency ω_0 as -65dB bandwidth, the cut-off frequency ω_0 of a large FOV is smaller than small FOV as Figures 4.14(c)-(d). In other words, in case of large FOV, we can perform more aggressive downsampling of the DBP signal for low frequency reconstruction. In Figure 4.14(c), the cut-off frequency is about $B/16$, which means the maximum downsampling rate will be 8, and in Figure 4.14(d), the cut-off frequency is larger than $B/4$, which means the maximum downsampling rate will be 4. Accordingly, $f(x)$ can be reconstructed well with a downsampling rate up to 8 in Figure 4.14(a), but in Figure 4.14(b), the downsampling rate only till 4 can produce correct reconstruction.

Figure 4.15 shows the reconstruction result in the helical cone-beam interior tomography with the downsampling rates 0 to 4. When the downsampling rate is 0, there is no downsampling and upsam-

pling, but the multiscale implementation is still working, which means that the high frequency part is reconstructed by the one-step Hilbert transform. As mentioned before, there is a limit of the downsampling rate for each case, so if we increase the downsampling rate, the object cannot be reconstructed correctly. In Figure 4.15, when the downsampling rate till 4, the image qualities of the reconstructions are similar, but the RMSE and NRMSE increase as the downsampling rate increases. This is because the convergence speed is slow when the image has the high resolution. If we make the iterative number increase, these RMSE and NRMSE will decrease. The reconstruction result of the downsampling rate 8 and 16 have streak artifacts because the high frequency part cannot be recovered by the Bedrosian equality.

KAIST

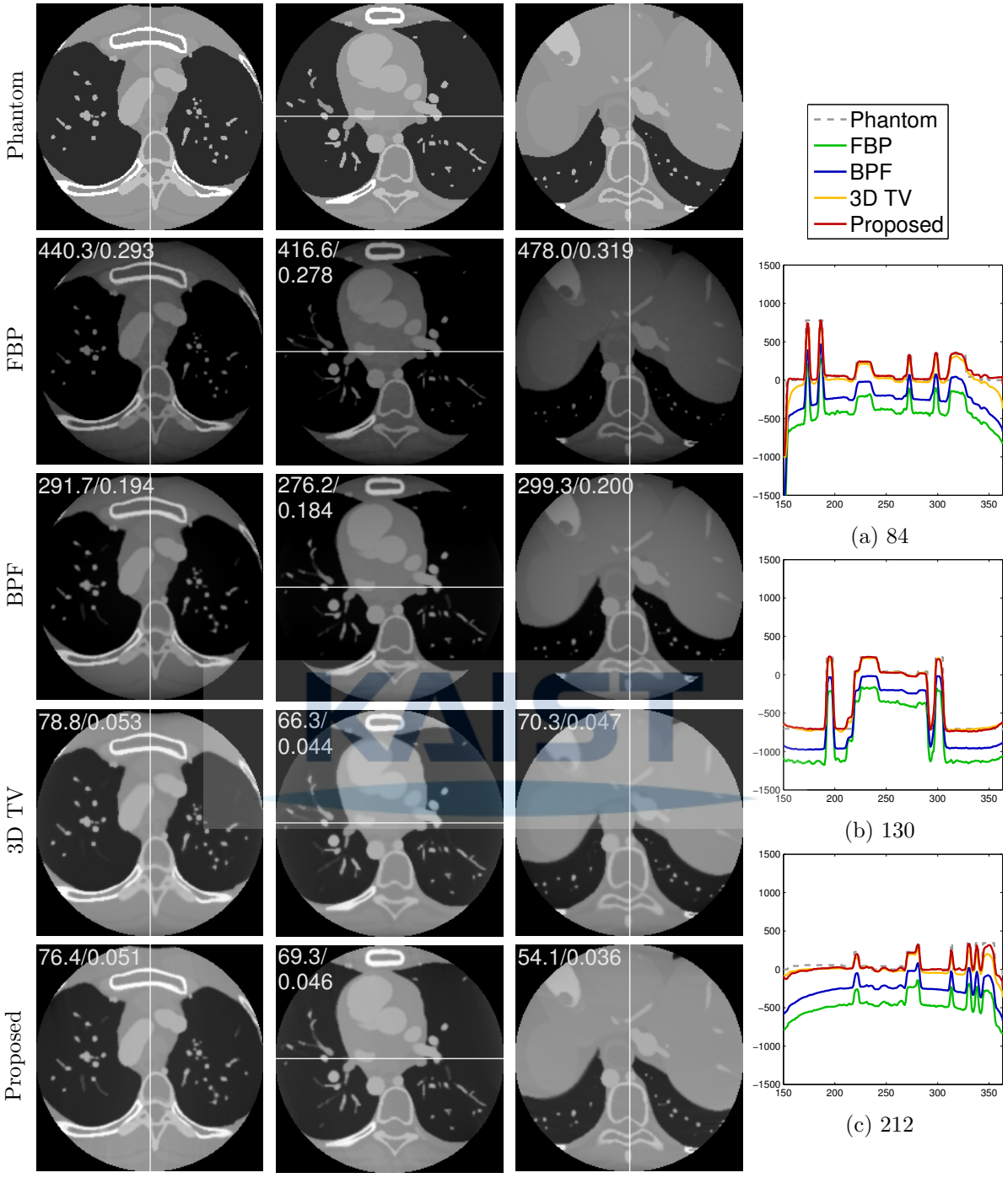


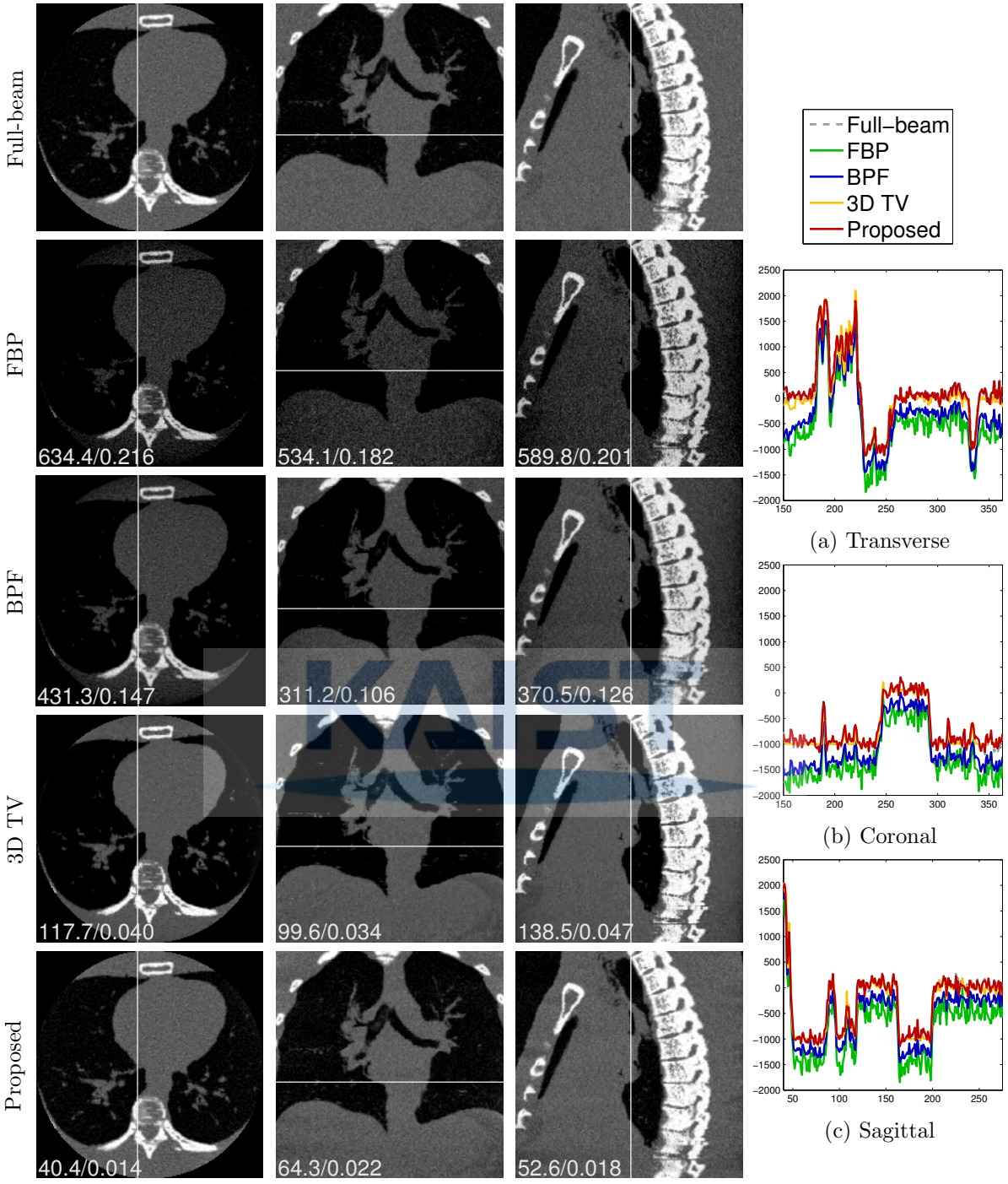
Figure 4.8: Reconstructed images by the truncated helical cone-beam CT of the 3D XCAT phantom. For the simulation, the K of the L is 1, and the iteration number of the POCS is 100. The l and u are 0 and 0.0309, respectively, and the λ is 0.001, and N is 20. The depth of the multiscale implementation is 2. From the first to third columns, each column shows the slice image when the slice number is (a) 84, (b) 130, and (c) 212, and each row is for the ideal phantom, FBP, BPF, 3D TV, and the proposed method. The last column shows the cut views indicated by the white line on the images. The number written at the corner of the reconstructed image shows the RMSE and NRMSE in the Hounsfield unit (HU) scale.

Transverse

Coronal Sagittal

Figure 4.9: 3D inner organ phantom.

KAIST



(a) Transverse (b) Coronal (c) Sagittal

(a) Transverse

(b) Coronal

(c) Sagittal

Figure 4.10: Reconstructed images by the truncated helical cone-beam CT of the 3D inner organ phantom with noise. For the simulation, the K of the L is 1, and the iteration number of the POCS is 100. The l and u are 0 and 0.0604, respectively, and the λ is 0.001, and N is 20. The depth of the multiscale implementation is 2. From the first to third columns, each column shows the same (a) transverse, (b) coronal, and (c) sagittal image, and each row is for full-beam BPF, FBP, BPF, 3D TV, and the proposed method. The last column shows the cut views indicated by the white line on the images. The number written at the corner of the reconstructed image shows the RMSE and NRMSE in the HU scale.

(a) Extrapolated FBP (b) Extrapolated BPF (c) Proposed

Figure 4.11: Error images from the extrapolated projection by FBP, BPF, and the proposed method when $L_{\text{extension}} = 140$.

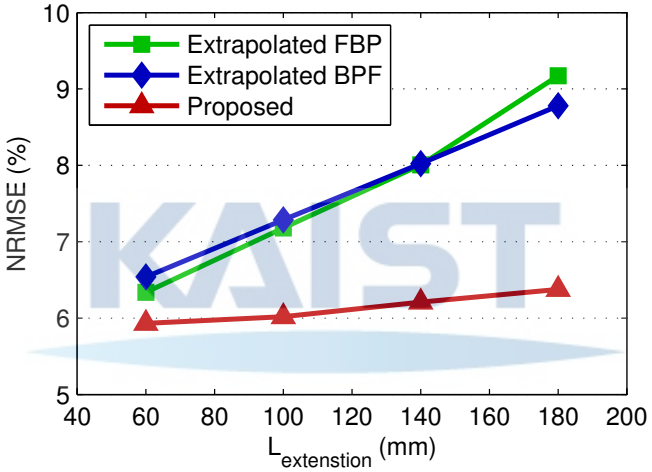
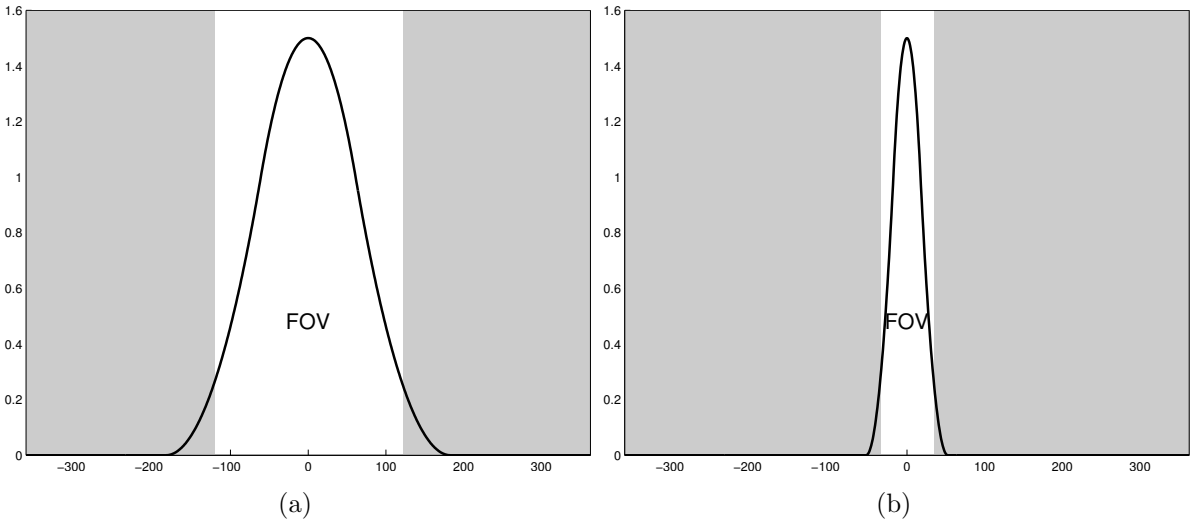


Figure 4.12: NRMSE value changes with respect to $L_{\text{extension}}$.



(a) (b)

Figure 4.13: Window for interior tomography $w(x)$. (a) is for the large FOV case, and (b) is for the small FOV case.

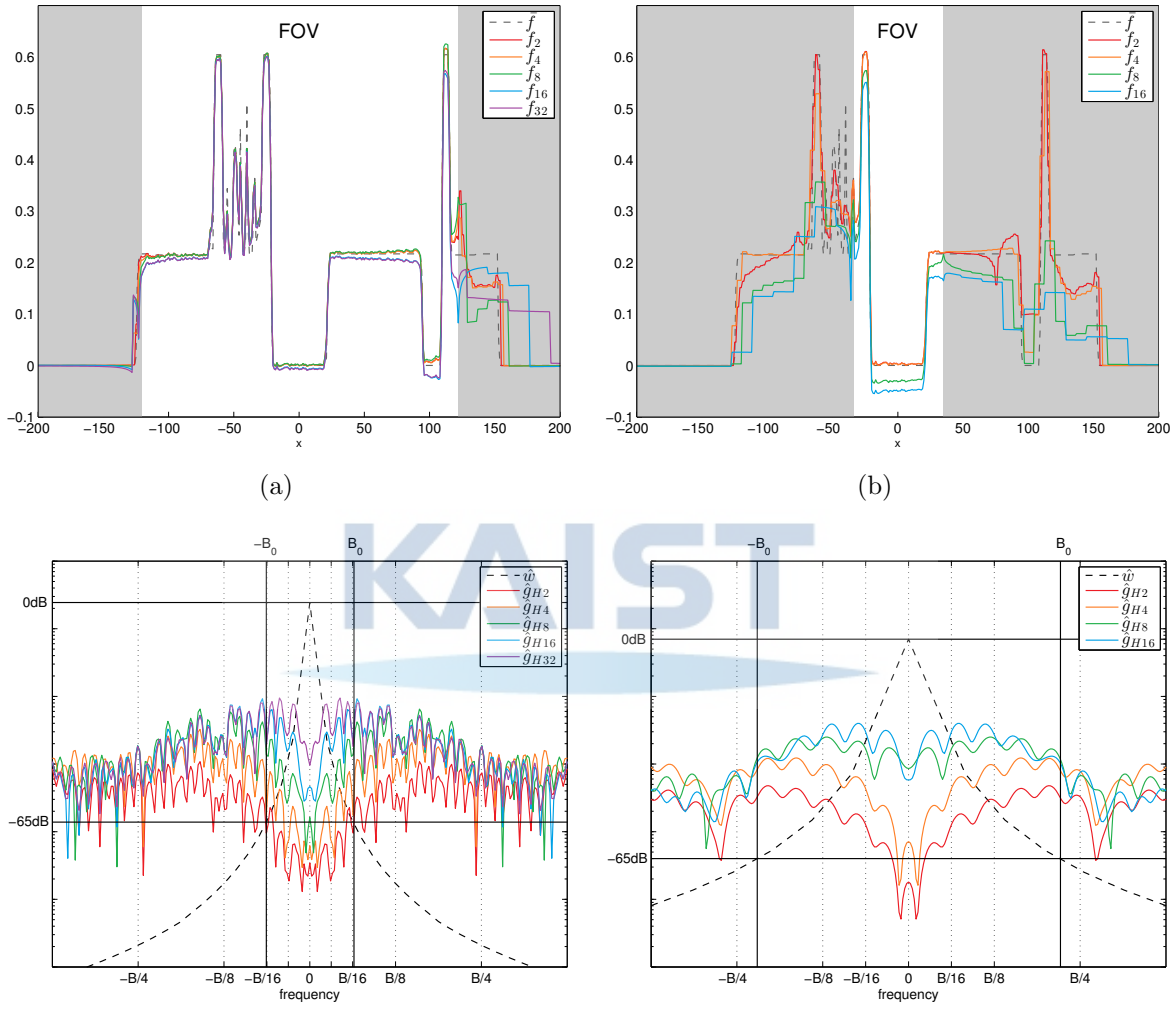


Figure 4.14: Interior tomography reconstruction results with various downsampling rate for two kinds of window size. Reconstructed $f(x)$ from (a) a large FOV and (b) a smaller FOV. The spectrum $\hat{w}(\omega)$ and $\hat{g}_H(\omega)$ for (c) a large FOV and (d) a smaller FOV. The subscript numbers after f and $\hat{g}_H$ represent the downsampling rate. In (c)-(d), B refers to the Nyquist sampling rate of the original DBP data.

-

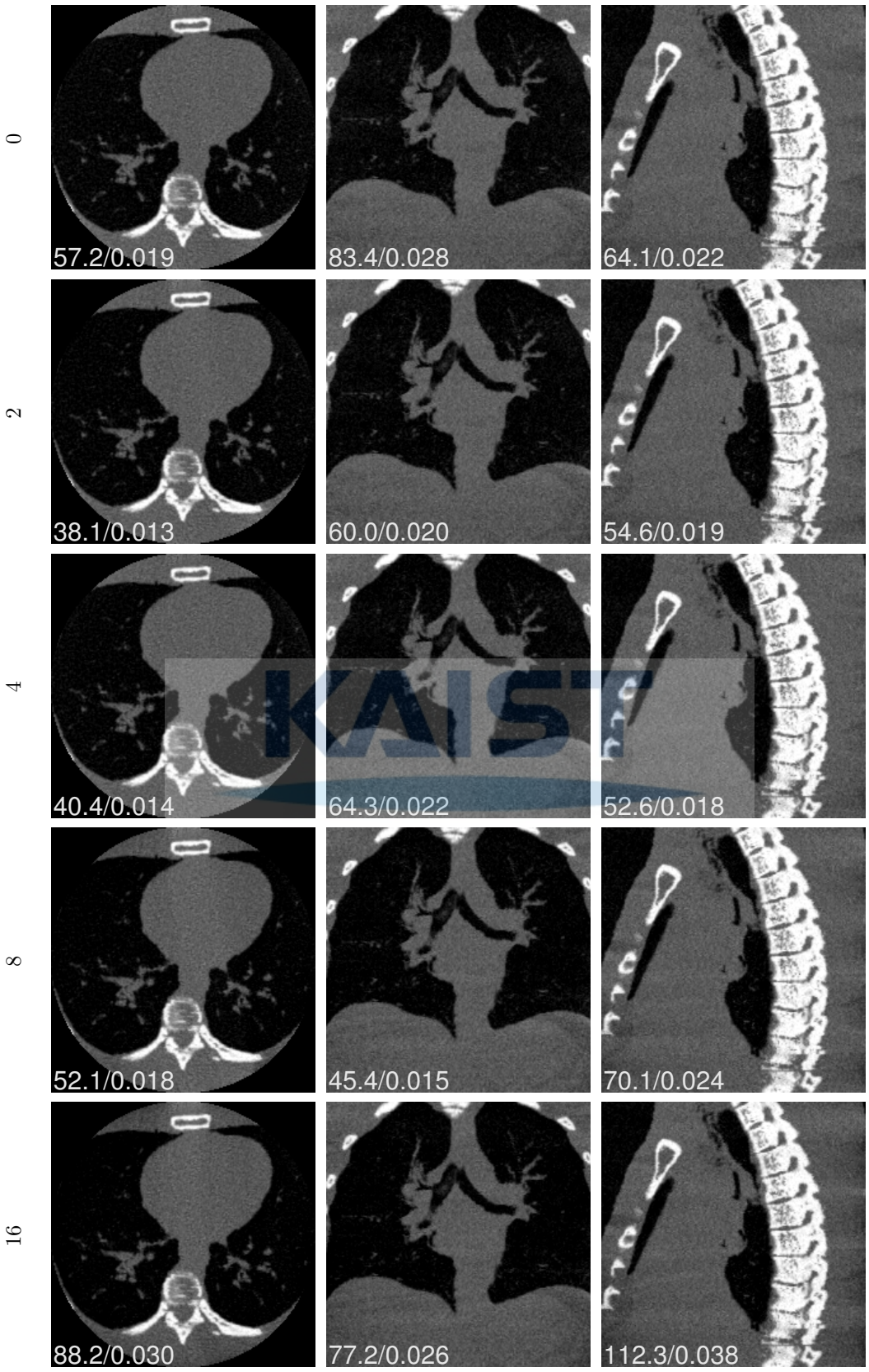


Figure 4.15: Interior tomography reconstruction result in the noisy helical cone-beam CT with varying the downsampling rate. The left most numbers indicate the downsampling rate, and for each downsampling rate, transverse, coronal, and sagittal images of the reconstruction are shown. The number written at the corner of the reconstructed image shows the RMSE and NRMSE in the HU scale.

Chapter 5. Conclusion

In this research, the exact cone-beam reconstructions with the FBP and BPF algorithms of various geometries were studied and implemented by GPU. From GPU implementation point of view, the FBP algorithm can be considered simpler than the BPF algorithms because it has no regridding process, but its filtration on the detector needs the forward and backward rebinning process. So, the FBP algorithm takes longer in filtering process. In the reconstruction results with two kinds of trajectories with the similar number of views, the line profiles looked similar, but the entire quality of circle-plus-line CT was clearer without artifacts. Moreover, the saddle CT requires the much bigger detector area than circle-plus-line CT to cover the object. Thus, the BPF reconstruction of circle-plus-line CT is more economical in practice than the FBP reconstruction of the saddle CT.

Especially the BPF-type algorithm inspired the new interior tomography algorithm. Using the Hilbert transform relationship between the DBP and the object, interior tomography problem can be converted to a set of 1D inverse problems. In particular, the interior tomography problem with tiny known areas and support of the object can be converted into an optimization problem with multiple convex constraints. Even though originally proposed POCS method for this problem provided excellent reconstruction, the convergence rate was shown slow. Hence, we employed PPXA algorithm to overcome the limitation of POCS. Using numerical experiments, we demonstrated that PPXA provides much faster convergence and improves reconstruction result in the local tomography problem compared to POCS method. despite the acceleration by the PPXA, knowing the subregions before reconstruction is impractical.

The existing iterative method of interior tomography based on 2D or 3D TV penalties does not need the known subregions, but it is computationally too expensive to be used clinically because it requires multiple projections and backprojections. To overcome the disadvantage of the existing approaches, in our previous work, we proposed a 1D generalized TV seminorm penalty that is more relaxed but still sufficient to guarantee perfect recovery with significantly reduced computational complexity.

To further accelerate the algorithm, this paper proposed a multiscale interior tomography by exploiting the Bedrosian theorem. Thanks to the Bedrosian theorem, high frequency components of an object can be reconstructed analytically by the Hilbert transform; thus, only for the low frequency reconstruction, computationally expensive iterative reconstruction was performed at a very coarse resolution, which significantly reduced the computational complexity. Moreover, our TV modeling is only for the low frequency signal, which is more practical in capturing high frequency details that cannot be modeled by the generalized TV.

The results of the numerical experiments showed that the proposed algorithm can produce a high quality reconstruction for both 2D fan-beam CT, 3D helical cone-beam CT, and 3D circular cone-beam CT, and its computational time can be significantly reduced down to a practical level. Moreover, the spectral blending approach significantly reduced cone-beam and missing frequency artifacts with minimal increases in the computational complexity.

KAIST

References

[1] Alexander Katsevich, "Analysis of an exact inversion algorithm for spiral cone-beam CT," *Physics in Medicine and Biology*, vol. 47, no. 15, pp. 2583, 2002.

[2] A. Katsevich, "Theoretically exact filtered backprojection-type inversion algorithm for spiral CT," *SIAM Journal on Applied Mathematics*, vol. 62, no. 6, pp. 2012–2026, 2002.

[3] Alexander Katsevich, "A general scheme for constructing inversion algorithms for cone beam CT," *International Journal of Mathematics and Mathematical Sciences*, vol. 2003, no. 21, pp. 1305–1321, 2003.

[4] A. Katsevich, "Image reconstruction for the circle and line trajectory," *Physics in Medicine and Biology*, vol. 49, no. 22, pp. 5059, 2004.

[5] Alexander Katsevich, "An improved exact filtered backprojection algorithm for spiral computed tomography," *Advances in Applied Mathematics*, vol. 32, no. 4, pp. 681–697, 2004.

[6] Jed D. Pack and Frédéric Noo, "Cone-beam reconstruction using 1D filtering along the projection of M-lines," *Inverse Problems*, vol. 21, no. 3, pp. 1105, 2005.

[7] Jed D. Pack, Frédéric Noo, and Hiroyuki Kudo, "Investigation of saddle trajectories for cardiac CT imaging in cone-beam geometry," *Physics in Medicine and Biology*, vol. 49, no. 11, pp. 2317, 2004.

[8] Guang-Hong Chen, "An alternative derivation of katsevich's cone-beam reconstruction formula," *Medical physics*, vol. 30, no. 12, pp. 3217–3226, 2003.

[9] Yu Zou and Xiaochuan Pan, "Exact image reconstruction on PI-lines from minimum data in helical cone-beam CT," *Physics in Medicine and Biology*, vol. 49, no. 6, pp. 941, 2004.

[10] Yu Zou and Xiaochuan Pan, "Image reconstruction on PI-lines by use of filtered backprojection in helical cone-beam CT," *Physics in Medicine and Biology*, vol. 49, no. 12, pp. 2717, 2004.

[11] Yu Zou, Xiaochuan Pan, and Emil Y. Sidky, "Theory and algorithms for image reconstruction on chords and within regions of interest," *J. Opt. Soc. Am. A*, vol. 22, no. 11, pp. 2372–2384, 2005.

[12] Hengyong Yu, Yangbo Ye, Shiying Zhao, and Ge Wang, "A backprojection-filtration algorithm for nonstandard spiral cone-beam ct with an n-pi-window," *Physics in Medicine and Biology*, vol. 50, no. 9, pp. 2099, 2005.

[13] Lifeng Yu, Dan Xia, Yu Zou, Emil Sidky, and Xiaochuan Pan, "A rebinning-type backprojection-filtration algorithm for image reconstruction in helical cone-beam ct," in *Nuclear Science Symposium Conference Record, 2006. IEEE*. IEEE, 2006, vol. 5, pp. 2869–2872.

[14] Lifeng Yu, Dan Xia, Yu Zou, Emil Y. Sidky, Junguo Bian, and Xiaochuan Pan, "A rebinned backprojection-filtration algorithm for image reconstruction in helical cone-beam CT," *Physics in medicine and biology*, vol. 52, no. 18, pp. 5497, 2007.

KAIST

[15] Seungryong Cho, Charles A Pelizzari, and Xiaochuan Pan, "Phase angle optimization for a single-turn helical cone-beam ct with a π-line-based bpf algorithm," in *World Congress on Medical Physics and Biomedical Engineering 2006*. Springer, 2007, pp. 1746–1749.

[16] Seungryong Cho, Dan Xia, Charles A Pelizzari, and Xiaochuan Pan, "Exact reconstruction of volumetric images in reverse helical cone-beam ct," *Medical physics*, vol. 35, no. 7, pp. 3030–3040, 2008.

[17] Wenyuan Bi, Zhiqiang Chen, Li Zhang, and Yuxiang Xing, "Accelerate helical cone-beam ct with graphics hardware," in *Medical Imaging*. International Society for Optics and Photonics, 2008, pp. 69132T–69132T.

[18] Jiang Hsieh, E. Chao, J. Thibault, Brian Grekowicz, Amy Horst, Scott McOlash, and T. J. Myers, "Algorithm to extend reconstruction field-of-view," in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*. IEEE, 2004, pp. 1404–1407.

[19] Rolf Clackdoyle, Frédéric Noo, Junyu Guo, John Roberts, et al., "Quantitative reconstruction from truncated projections in classical tomography," *IEEE Transactions on Nuclear Science*, vol. 51, no. 5, pp. 2570–2578, 2004.

[20] Frédéric Noo, Rolf Clackdoyle, and Jed D. Pack, "A two-step Hilbert transform method for 2D image reconstruction," *Physics in Medicine and Biology*, vol. 49, no. 17, pp. 3903, 2004.

[21] B. Zhang and G. L. Zeng, "Two-dimensional iterative region-of-interest (ROI) reconstruction from truncated projection data," *Medical physics*, vol. 34, no. 3, pp. 935–944, 2007.

[22] Yu Zou, Xiaochuan Pan, and Emil Y. Sidky, "Image reconstruction in regions-of-interest from truncated projections in a reduced fan-beam scan," *Physics in Medicine and Biology*, vol. 50, no. 1, pp. 13, 2005.

[23] Lifeng Yu, Yu Zou, Emil Y. Sidky, Charles A. Pelizzari, Peter Munro, and Xiaochuan Pan, "Region of interest reconstruction from truncated data in circular cone-beam CT," *IEEE Transactions on Medical Imaging*, vol. 25, no. 7, pp. 869–881, 2006.

[24] Michel Defrise, Frédéric Noo, Rolf Clackdoyle, and Hiroyuki Kudo, "Truncated Hilbert transform and image reconstruction from limited tomographic data," *Inverse Problems*, vol. 22, no. 3, pp. 1037, 2006.

[25] M. Courdurier, F. Noo, M. Defrise, and H. Kudo, "Solving the interior problem of computed tomography using a priori knowledge," *Inverse Problems*, vol. 24, no. 6, pp. 065001, 2008.

[26] Hiroyuki Kudo, Matias Courdurier, Frédéric Noo, and Michel Defrise, "Tiny a priori knowledge solves the interior problem in computed tomography," *Physics in medicine and biology*, vol. 53, no. 9, pp. 2207, 2008.

[27] Patrick L Combettes and Jean-Christophe Pesquet, "Proximal splitting methods in signal processing," *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212, 2011.

KAIST

[28] Xinzhe Jin, Alexander Katsevich, Haoyong Yu, Guibin Wang, Luoqing Li, and Zhe Chen, "Interior tomography with continuous singular value decomposition," *IEEE Transactions on Medical Imaging*, vol. 31, no. 11, pp. 2108–2119, 2012.

[29] Alexander Katsevich and A. Tovbis, "Finite hilbert transform with incomplete data: null-space and singular values," *Inverse Problems*, vol. 28, no. 10, pp. 105006, 2012.

[30] E. Katsevich, A. Katsevich, and G. Wang, "Stability of the interior problem with polynomial attenuation in the region of interest," *Inverse problems*, vol. 28, no. 6, pp. 065022, 2012.

[31] Hengyong Yu and Ge Wang, "Compressed sensing based interior tomography," *Physics in Medicine and Biology*, vol. 54, no. 9, pp. 2791, 2009.

[32] Jiansheng Yang, Hengyong Yu, Ming Jiang, and Ge Wang, "High-order total variation minimization for interior tomography," *Inverse Problems*, vol. 26, no. 3, pp. 035013, 2010.

[33] Weimin Han, Hengyong Yu, and Ge Wang, "A general total variation minimization theorem for compressed sensing based interior tomography," *Journal of Biomedical Imaging*, vol. 2009, pp. 21, 2009.

[34] Ludwig Ritschl, Frank Bergner, Christof Fleischmann, and Marc Kachelrieß, "Improved total variation-based CT image reconstruction applied to clinical data," *Physics in medicine and biology*, vol. 56, no. 6, pp. 1545, 2011.

[35] John Paul Ward, Minji Lee, Jong Chul Ye, and Michael Unser, "Interior tomography using 1D generalized total variation – part I: mathematical foundation," *SIAM Journal on Imaging Sciences*, vol. 8, no. 1, pp. 226–247, 2015.

[36] G. Shechter, Th. Köhler, A. Altman, and R. Proksa, "The frequency split method for helical cone-beam reconstruction," *Medical physics*, vol. 31, no. 8, pp. 2230–2236, 2004.

[37] Esther Meyer, Rainer Raupach, Michael Lell, Bernhard Schmidt, and Marc Kachelrieß, "Frequency split metal artifact reduction (FSMAR) in computed tomography," *Medical physics*, vol. 39, no. 4, pp. 1904–1916, 2012.

[38] Frederick W. King, *Hilbert transforms*, vol. 2, Cambridge University Press Cambridge, UK, 2009.

[39] H. Yang, M. Li, K. Koizumi, and H. Kudo, "Exact cone beam reconstruction for a saddle trajectory," *Physics in Medicine and Biology*, vol. 51, no. 5, pp. 1157–1172, 2006.

[40] Hengyong Yu and Ge Wang, "Studies on implementation of the Katsevich algorithm for spiral cone-beam CT," *Journal of X-ray Science and Technology*, vol. 12, no. 2, pp. 97–116, 2004.

[41] Frédéric Noo, Stefan Hoppe, Frank Dennerlein, Günter Lauritsch, and Joachim Hornegger, "A new scheme for view-dependent data differentiation in fan-beam and cone-beam computed tomography," *Physics in medicine and biology*, vol. 52, no. 17, pp. 5393, 2007.

[42] Haiquan Yang, Meihua Li, Kazuhito Koizumi, and Hiroyuki Kudo, "View-independent reconstruction algorithms for cone beam ct with general saddle trajectory," *Physics in medicine and biology*, vol. 51, no. 15, pp. 3865, 2006.

KAIST

[43] Stefan Hoppe, Joachim Hornegger, Günter Lauritsch, Frank Dennerlein, and Frédéric Noo, "Truncation correction for oblique filtering lines," *Medical physics*, vol. 35, no. 12, pp. 5910–5920, 2008.

[44] Frédéric Noo, Jed Pack, and Dominic Heuscher, "Exact helical reconstruction using native cone-beam geometries," *Physics in Medicine and Biology*, vol. 48, no. 23, pp. 3787, 2003.

[45] Yangbo Ye, Shiying Zhao, Hengyong Yu, and Ge Wang, "A general exact reconstruction for cone-beam ct via backprojection-filtration," *Medical Imaging, IEEE Transactions on*, vol. 24, no. 9, pp. 1190–1198, 2005.

[46] Nelly Pustelnik, Caroline Chaux, and Jean-Christophe Pesquet, "Parallel proximal algorithm for image restoration using hybrid regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2450–2462, 2011.

[47] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[48] Amir Beck and Marc Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009.

[49] Yurii Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," in *Soviet Mathematics Doklady*, 1983, vol. 27, pp. 372–376.

[50] Paul S Cho, Anthony D Rudd, and Roger H Johnson, "Cone-beam ct from width-truncated projections," *Computerized Medical Imaging and Graphics*, vol. 20, no. 1, pp. 49–57, 1996.

[51] Katia Sourbelle, Marc Kachelrieß, and Willi A. Kalender, "Reconstruction from truncated projections in CT using adaptive detruncation," *European Radiology*, vol. 15, no. 5, pp. 1008–1014, 2005.

KAIST

Summary

Interior tomography using 1D generalized total variation penalty and multiscale implementation by Bedrosian identity

의료 영상 진단 기기로 많이 사용되고 있는 엑스선 컴퓨터 단층촬영(CT: Computed Tomography)은 다중채널 CT의 보급으로 3차원의 영상을 단시간에 얻을 수 있게 되었다. 다중채널 CT 복원은 2차원 복원을 근사화 한 방법으로 이루어 지고 있는 경우가 대부분인데 이는 구현이 쉬운 장점이 있으나 다중채널이 증가하면 증가할 수록 z축 방향으로 왜곡 현상이 심해지게 된다. 이러한 단점을 극복하기 위해 다양한 궤적으로 촬영한 다중채널 CT의 정확한 복원 방법이 개발되었는데 크게 선 필터처리 후 역투영(FBP: Filtered Backprojection)과 선 역투형 후 필터처리(BPF: Backprojection Filtration) 방식으로 구분할 수 있다. 본 연구에서는 이 두 가지 알고리즘의 다양한 궤적에 대한 구현 방법을 설명하고 그래픽 처리 장치(GPU: graphic processing unit)를 이용한 병렬화 구현하였다. 선 필터처리 후 역투영 방식과 역투영 후 필터처리 방식 중 특히 선 역투영 후 필터처리 방식은 최근 방사선 조사량을 줄이기 위한 내부 단층 촬영 기술 개발에 큰 영향을 미치고 있다. 내부 단층 촬영 기술은 검출기의 크기가 모든 각도에 대해 인체를 포함하지 않고도 인체의 내부를 복원하는 기술을 말한다. 선 역투영 후 필터처리 방식은 기존의 2차원 혹은 3차원의 복원 문제를 여러 개의 1차원 문제들로 축소시킬 수 있다는 장점이 있는데 이러한 방법을 기반으로 아주 작은 영역이라도 복원하려는 관심 영역(ROI: Region of Interest)안의 일부분의 밝기 값을 알고 있다면 이를 이용해 나머지 영역을 유일하게 복원할 수 있다는 것이 증명 되었다. 하지만 아주 작은 영역이라도 내부의 밝기 값을 미리 안다는 것은 특수한 경우에만 가능하기 때문에 비실용적이라는 단점이 있다. 이를 극복하기 위해 2, 3차원의 총변동(TV: Total Variation)을 사용하여 복원이 가능한데 이는 정확한 결과를 얻는데는 성공하였지만 고차원의 TV를 적용하기 위해 많은 제한과 복잡도가 들어가는 단점이 있다. 우리는 이를 극복하기 위해 1차원의 일반화된 TV를 이용한 내부 단층촬영 기술을 제안하였다. 제안된 방법은 여러가 경우에 대해 보다 유연하고 간단한 구현을 가능하게 하였다. 여기서 더 나아가 우리는 실제 병원에서 사용할 수 있을 정도의 복원 속도를 가능하게 하기 위해 다중해상도 복원법을 제안하였다. 이는 선 역투영 후 필터처리 복원에서 필터처리에 해당하는 힐버트 변환이 고주파 성분에 대해서는 신호의 절단이 변환 결과에 영향을 미치지 않는다는 이론에서 기반되었다. 따라서 복원 시, 고주파 성분은 내부 촬영값을 이용한 힐버트 변환으로 복원하고 저주파 성분은 1차원의 일반화된 TV를 이용하여 반복적으로 복원하되 다운 샘플링을 통해 그 시간을 단축 시킬 수 있다. 최종 적으로 이 두 가지 신호를 합치면 최종 복원이 이루어 진다. 제안된 알고리즘은 컴퓨터 시뮬레이션을 통해 1차원의 일반화된 TV를 이용한 복원이 정확한 복원이 가능하다는 것과 해상도를 낮춘 신호에 대해서만 반복적으로 복원을 하는 것이 복원 시간을 현저히 낮출 수 있다는 것을 증명하는데 성공하였다.