GPU-based Iterative Cone Beam CT Reconstruction Using Tight Frame Regularization

Xun Jia¹², Bin Dong³, Yifei Lou⁴, and Steve B. Jiang¹²

¹Center for Advanced Radiotherapy Technologies, University of California San Diego, La Jolla, CA 92037-0843, USA
²Department of Radiation Oncology, University of California San Diego, La Jolla, CA 92037-0843, USA
³Department of Mathematics, University of California San Diego, La Jolla, CA 92093-0112, USA
⁴Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095-1555, USA

E-mail: sbjiang@ucsd.edu

X-ray imaging dose from serial cone-beam CT (CBCT) scans raises a clinical concern in most image guided radiation therapy procedures. It is the goal of this paper to develop a fast GPU-based algorithm to reconstruct high quality CBCT images from undersampled and noisy projection data so as to lower the imaging dose. For this purpose, we have developed an iterative tight frame (TF) based CBCT reconstruction algorithm. A condition that a real CBCT image has a sparse representation under a TF basis is imposed in the iteration process as regularization to the solution. To speed up the computation, a multi-grid method is employed. Our GPU implementation has achieved high computational efficiency and a CBCT image of resolution $512 \times 512 \times 70$ can be reconstructed in about $\sim 139$ sec. We have tested our algorithm on a digital NCAT phantom and a physical Catphan phantom. It is found that our TF-based algorithm leads to much higher CBCT quality than those obtained from a conventional FDK algorithm in the context of undersampling and low mAs levels. Higher spatial resolution has also been achieved comparing to a recently developed TV-based iterative CBCT reconstruction algorithm. We have also quantitatively analyzed the reconstructed CBCT image quality in terms of modulation-transfer-function and contrast-to-noise ratio under various scanning conditions. The results confirm the high CBCT image quality obtained from our TF algorithm. Moreover, our algorithm has also been validated in a real clinical context using a head-and-neck patient case.
1. Introduction

Cone Beam Computed Tomography (CBCT) is of central importance in cancer radiotherapy. It is particularly convenient for accurate patient setup in image guided radiation therapy (IGRT). Yet, the high imaging dose to healthy organs (a few cGy per scan) (Islam et al., 2006; Kan et al., 2008; Song et al., 2008) in CBCT scans is a clinical concern, especially when CBCT scan is performed before each fraction for the entire treatment course. The imaging dose in CBCT can be reduced by reducing the number of x-ray projections and lowering mAs levels (tube current and pulse duration). In these approaches, however, the consequent CBCT images reconstructed using conventional FDK algorithms (Feldkamp et al., 1984) are highly degraded due to insufficient and noisy projections. It is therefore desirable to develop new techniques to reconstruct high quality CBCT from undersampled and noisy projection data.

Recently, a burst of research in compressed sensing (Donoho and Tanner, 2005; Candes and Romberg, 2006; Candes et al., 2006; Candes and Tao, 2006; Donoho, 2006; Tsaig and Donoho, 2006) have demonstrated the feasibility of recovering signals from incomplete measurements through optimization methods in various mathematical situations. A number of techniques developed in this field have been introduced to the CT or CBCT reconstruction problems from undersampled data (Sidky et al., 2006; Sidky and Pan, 2008; Chen et al., 2008; Cho et al., 2009; Jia et al., 2010b) and have shown their tremendous power in solving such complicated problems. The key idea is that medical images can be sparsely approximated by certain linear transformation and penalizing the $\ell_1$-norm of the image in the transformed domain will enable us to recover the unknown image from highly undersampled data. Using the idea of compressed sensing and sparse approximation of images under transformations to perform CBCT reconstruction has indeed become one of the central topics in medical imaging. In particular, one of the image transformation techniques called tight-frame (TF) transform (Daubechies et al., 2003) has attracted a lot of attentions recently. These tight frames have the same structure as the traditional wavelets, except that they are redundant systems that generally provide sparser representations to piecewise smooth functions than traditional wavelets. The TF approach is found to be extremely effective and efficient in solving many image restoration problems (Cai et al., 2008; Cai et al., 2009b; Cai et al., 2009a; Cai and Shen, 2010). A short survey on the theory and applications of TF was given by Shen (2010) and a much more detailed survey was given by Dong and Shen (2010). CBCT reconstruction problem can be generally viewed as a 3-dimensional image restoration problem. In such a problem, it has been noted that the discontinuities of the reconstructed piecewise smooth image provide very important information, as they usually account for the boundaries between different objects in the volumetric image. In the TF approach, one tries to restore TF coefficients of the image, which usually correspond to important features, e.g. edges, as opposed to the image itself. This allows us to specifically focus on the reconstruction of the important information of the image, hence leading to high quality reconstruction results. It is for this reason that TF is superior to many other similar available algorithms, which cannot preserve edge information to some extent.
Besides its effectiveness, TF approach also has attractive numerical properties. First, recently invented numerical schemes specifically designed for the TF approach lead to a high convergence rate (Shen et al., 2009; Shen, 2010; Dong and Shen, 2010). Second, the numerical scheme only involves simple matrix-vector or vector operations, making it straightforward to implement the algorithm and parallelize it in a parallel computing structure. It is these numerical properties that lead to high computational efficiency in practice. Moreover, general purpose graphic processing units (GPUs) have offered us a promising prospect of increasing efficiencies of heavy duty tasks in radiotherapy, such as CBCT FDK reconstruction (Xu and Mueller, 2005; Li et al., 2007; Sharp et al., 2007; Xu and Mueller, 2007; Yan et al., 2008), deformable image registration (Sharp et al., 2007; Samant et al., 2008; Gu et al., 2009b), dose calculation (Jacques et al., 2008; Hissoiny et al., 2009; Gu et al., 2009a; Jia et al., 2010a), and treatment plan optimization (Men et al., 2009; Men et al., 2010). Taking advantages of the high computing power of the GPU, the computation efficiency of TF-based CBCT reconstruction is expected to be enhanced considerably.

We have developed a novel CBCT reconstruction algorithm based on TF and implemented it on GPU. This work, along with some validations, will be presented in this paper. Our experiments on a digital phantom, a physical phantom, and a real patient case demonstrate the possibility of reconstructing high quality CBCT images from extremely undersampled and noisy data. The associated high computational efficiency due to the good numerical property of the TF algorithm and our GPU implementation makes this approach practically very attractive. Our work, by introducing the novel TF algorithm to the CBCT reconstruction context for the first time, will potentially lead to the realization of low dose CBCT. The rest of this paper is organized as following. Section 2 will describe our method as well as implementation details. In section 3 we will provide the reconstruction results and necessary analysis on the reconstructed volumetric images. Finally, section 4 will conclude our paper.

2. Methods

2.1 Model and Algorithm

Let us consider a patient volumetric image represented by a function $f(x)$ with $x = (x, y, z) \in \mathbb{R}^3$. A projection operator $P^\theta$ maps $f(x)$ into another function on an x-ray imager plane along a projection angle $\theta$:

$$P^\theta[f](u) = \int_{0}^{L(u)} dl f(x_5 + nl),$$

where $x_5 = (x_5, y_5, z_5)$ is the coordinate of the x-ray source and $u = (u, v) \in \mathbb{R}^2$ is the coordinate of the projection point on the x-ray imager, $n = (n_1, n_2, n_3)$ being a unit vector along the projection direction. Fig. 1 illustrates the geometry. The upper integration limit $L(u)$ is the length of the x-ray line. Denote the observed projection image at the angle $\theta$ by $g^\theta(u)$. Mathematically speaking, a CBCT reconstruction
problem is formulated as to retrieve the volumetric image function \( f(\mathbf{x}) \) based on the observation of \( g^\theta(\mathbf{u}) \) at various angles given the projection mapping in equation (1).

Figure 1. The geometry of x-ray projection. The operator \( P^\theta \) maps \( f(\mathbf{x}) \) in \( \mathbb{R}^3 \) onto another function \( P^\theta[f](\mathbf{u}) \) in \( \mathbb{R}^2 \), the x-ray imager plane, along a projection angle \( \theta \). \( L(\mathbf{u}) \) is the length from \( \mathbf{x}_s \) to \( \mathbf{u}^* \) and \( l(\mathbf{x}) \) is that from \( \mathbf{x}_s \) to \( \mathbf{x} \). The source to imager distance is \( L_0 \).

The CBCT image reconstruction from few projections is an underdetermined problem. Because of insufficient measurements made at only a few x-ray projections, there are indeed infinitely many functions \( f \) satisfying the condition \( P^\theta[f](\mathbf{u}) = g^\theta(\mathbf{u}) \). Therefore, regularization based on some assumptions about the solution \( f \) has to be performed during the reconstruction process. These regularization-based CBCT reconstruction approaches usually result in solving challenging minimization problems. The most commonly used approach is an alternative iteration scheme, where, within each iteration step, conditions to be satisfied by the solution is imposed one after another. In our problem, there are three conditions that need to be satisfied by the solution, and three key operations will be performed in each iteration step accordingly. These conditions, as well as the operations ensuring them, will be described in the following.

First, the x-ray projection of the reconstructed volumetric image \( f(\mathbf{x}) \) should match the observation \( g^\theta(\mathbf{u}) \). This condition is commonly achieved by solving a linear system \( Pf = g \), where \( P \) is the matrix representation of the projection operator \( P^\theta \), and \( f \) and \( g \) are vectors corresponding to the solution \( f(\mathbf{x}) \) and the observation \( g^\theta(\mathbf{u}) \), respectively. Nonetheless, since this is a highly underdetermined problem, any numerical scheme tending to directly solve \( Pf = g \) is unstable. Instead, in this work we perform a minimization of an energy \( E[f] = \| Pf - g \|_2^2 \) by using a conjugate gradient least square (CGLS) algorithm. This algorithm is essentially an iterative algorithm, which generates a new solution \( f \) given an initial guess \( v \). We formally denote this process as \( f \leftarrow \text{CGLS}[v] \), and the details regarding the implementation of the CGLS algorithm will be discussed in section 2.2.2. The CGLS algorithm enables us to efficiently solve this minimization problem, and hence ensures the consistency between the reconstructed volumetric image \( f(\mathbf{x}) \) and the observation \( g^\theta(\mathbf{u}) \).

Second, we impose a regularization condition to the solution \( f(\mathbf{x}) \) that it has a sparse representation under a piecewise linear TF system \( X = \{ \psi_i(\mathbf{x}) \} \). Interested readers can
consult literatures (Daubechies et al., 2003; Han and Shen, 2009; Dong and Shen, 2010) for details. The solution \( f(x) \) can be decomposed by \( X \) into a set of coefficient as \( \alpha_i = \int dx \psi_i(x)f(x) \). In matrix notation, this decomposition can be simply denoted as \( \alpha = Df \), where \( \alpha \) is a vector consisting of the TF coefficients, while \( D \) is a matrix whose rows are the TF basis functions. Conversely, the function \( f(x) \) can be uniquely determined given a set of coefficients \( \{ \alpha_i \} \) by \( f(x) = \sum_i \psi_i(x)\alpha_i \), which can be denoted as \( f = D^\top \alpha \). It has been observed that many natural images has a very sparse representation under the TF system \( X \), i.e. there are only a small proportion of the elements within the coefficient vector \( \alpha \) that are significantly larger in magnitude than the rest of the elements. It is this property that can be utilized a priori as a regularization in our CBCT reconstruction problem. A common way of imposing this condition into the solution \( f \) is to throw away small TF coefficients. The deletion of these small coefficients not only sharpens edges but also removes noises in the reconstructed CBCT. As such, we first decompose \( f \) into the TF space, then soft-threshold the TF coefficients with a predetermined value \( \mu \), and finally reconstruct \( f \) based on the new coefficients. This process can be denoted as \( f \leftarrow D^\top \mathcal{J}_\mu Df \). Here the soft-threshold operation is defined as:

\[
\mathcal{J}_\mu \alpha = \begin{cases} 
\text{sgn}(\alpha)(|\alpha| - \mu) & \text{if } |\alpha| > \mu \\
0 & \text{if } |\alpha| < \mu
\end{cases}
\]  

(2)

where \( \text{sgn}(\cdot) \) is sign function defined as

\[
\text{sgn}(\alpha) = \begin{cases} 
1 & \text{if } \alpha > 0 \\
0 & \text{if } \alpha = 0 \\
-1 & \text{if } \alpha < 0
\end{cases}
\]  

(3)

It is understood that the soft-threshold operation \( \mathcal{J}_\mu \alpha \) is performed component-wise on the vector \( \alpha \).

Third, since the reconstructed CBCT image \( f(x) \) physically represents x-ray attenuation coefficient at a spatial point \( x \), its positivity has to be ensured during the reconstruction in order to obtain a physically correct solution. For this purpose, we also perform a correction step of the reconstructed image \( f(x) \) by setting its negative voxel values to be zero. Mathematically, this operation is denoted by \( f \leftarrow \mathcal{H}f \), where the operation \( \mathcal{H} \) stands for a voxel-wise truncation of the negative values in the CBCT image \( f \).

In considering all the components mentioned above, we summarize the reconstruction algorithm as in Algorithm A1:

**Algorithm A1:**

- **Initializ**e: \( f^{(0)} = 0 \).
- For \( k = 0, 1, \ldots \), do the following steps until convergence:
  1. Update: \( f^{(k+1)} = \text{CGLS}[f^{(k)}] \);
  2. Shrinkage: \( f^{(k+1)} \leftarrow D^\top \mathcal{J}_\mu Df^{(k+1)} \);
  3. Correct: \( f^{(k+1)} \leftarrow \mathcal{H}f^{(k+1)} \).
Note that there is only one tuning parameter $\mu$ in the algorithm. In practice, its value is carefully tuned so that the best image quality can be obtained. An example of how we choose this parameter is provided in Section 3.2.

Mathematically, the TF coefficients $Df^{(k)}$ generated by Algorithm A1 can be shown to converge to the following optimization problem

$$\arg\min_{\alpha} \frac{1}{2} \|PD^T\alpha - g\|_2^2 + \frac{1}{2} \| (1 - DD^T)\alpha \|_2^2 + \mu \| \alpha \|_1, \text{ s.t. } D^T \alpha \geq 0. \quad (4)$$

Optimization problem (4) is a rather successful model in solving image restoration problems (Shen, 2010; Dong and Shen, 2010). The proof of this link, however, is beyond the scope of this paper and can be found in Shen (2010) and Dong and Shen (2010). With a simple modification, the convergence rate of A1 can be considerably enhanced (Shen et al., 2009; Shen, 2010; Dong and Shen, 2010), leading to Algorithm A2 used in our reconstruction problem:

**Algorithm A2:**

Initialize: $f^{(0)} = f^{(-1)} = 0, \tau^{(0)} = \tau^{(-1)} = 1.0,$

For $k = 0, 1, \ldots$ do the following steps until convergence

1. Compute: $v^{(k)} \leftarrow f^{(k)} + \frac{\tau^{(k-1)}}{\tau^{(k)}} \left[ f^{(k)} - f^{(k-1)} \right]$;
2. Update: $f^{(k+1)} = \text{CGLS}[v^{(k)}]$;
3. Shrinkage: $f^{(k+1)} \leftarrow D^T D f^{(k+1)}$;
4. Correct: $f^{(k+1)} \leftarrow H f^{(k+1)}$;
5. Set: $\tau^{(k+1)} = \frac{1}{2} \left[ 1 + \sqrt{1 + 4 \tau^{(k)}^2} \right].$

2.2 Implementation

In this paper, the CBCT reconstruction problem is solved with the aforementioned algorithm A2 on an NVIDIA Tesla C1060 card. This GPU card has a total number of 240 processor cores (grouped into 30 multiprocessors with 8 cores each), each with a clock speed of 1.3 GHz. It is also equipped with 4 GB DDR3 memory, shared by all processor cores. Utilizing such a GPU card with a tremendous parallel computing ability can considerably elevate the computation efficiency. In this section, we describe some key components of our implementation.

2.2.1 GPU parallelization

In fact, a number of computationally intensive tasks involved in algorithm A2 share a common feature, i.e. applying a single operation to different part of data elements. For computation tasks of this type, it is straightforward to accomplish them in a data-parallel fashion, namely having all GPU threads running the same operation, one for a given subset of the data. Such a parallel manner is particularly suitable for the SIMD (single instruction multiple data) structure of a GPU and high computation efficiency can be therefore achieved.
Specifically, the following components in A2 fall in to this category: 1) We simply parallelize the component-wise soft-threshold in Step 3 and the voxel-wise positivity correction of the CBCT image in Step 4 with one GPU thread responsible for one TF coefficient or one voxel, respectively. 2) The decomposition of a CBCT image \( f \) into the TF basis, \( \alpha = Df \), is merely a inner product operation. Namely, each component \( \alpha_i \) of the \( \alpha \) vector is defined as \( \alpha_i = \int \mathbf{d}x \psi_i(x)f(x) \). This computation can be performed by having one GPU thread compute the inner product of \( f(x) \) with one basis function \( \psi_i(x) \). The task \( f = D^T\alpha \) can be also achieved in a similar manner. 3) A matrix vector multiplication of the form \( g = Pf \) is frequently used in the CGLS method. This operation corresponds to the forward x-ray projection of a volumetric image \( f(x) \) to the imager planes, also known as a digital reconstructed radiograph. In our implementation, it is performed in a parallel fashion, with each GPU thread responsible for the line integral of equation (1) along an x-ray line using Siddon’s ray-tracing algorithm (Siddon, 1985; Jacobs et al., 1998; Han et al., 1999).

2.2.2 CGLS method

Another key component in our implementation is the CGLS solution to the optimization problem \( \min_{x} \| Pf - g \|_2^2 \) in Step 2 of A2. In this step, a CGLS method is applied to efficiently find a solution \( f^{(k+1)} \) to this least square problem with an initial value of \( v^{(k)} \) in an iterative manner (Hestenes and Stiefel, 1952). Each iteration step of the CGLS algorithm includes a number of fundamental linear algebra operations. For those simple vector-vector operations and scalar-vector operations, we utilize CUBLAS package (NVIDIA, 2009) for high efficiency. In addition, there are two time-consuming operations requiring special attention in the algorithm, namely matrix-vector multiplications of the form \( g = Pf \) and \( f = P^Tg \), where \( P \) is the x-ray projection matrix. Though it is straightforward to accomplish \( g = Pf \) on GPU with the Siddon’s ray-tracing algorithm as described previously, it is quite cumbersome to carry out a computation of the form \( f = P^Tg \). It is estimated that the matrix \( P \), though being a sparse matrix, contains approximately \( 4 \times 10^9 \) non-zero elements for a typical clinical case studied in this paper, occupying about 16 GB memory space. Such a huge matrix \( P \) is too large to be stored in a GPU memory, not to mention computing its transpose. Therefore, a new algorithm for completing the task \( f = P^Tg \) has to be designed. Without thinking too much, one can compute \( f = P^Tg \) by still using the Siddon’s algorithm. Such an operation, however, is a backward one in that it maps a function \( g(\mathbf{u}) \) on the x-ray imager back to a volumetric image \( f(x) \) by updating its voxel values along all ray lines. If Siddon’s ray-tracing algorithm is still used in the GPU implementation with each thread responsible for updating voxels along a ray line, a memory conflict problem would take place due to the possibility of simultaneously updating a same voxel value by different GPU threads. When this conflict occurs, one thread will have to wait until another thread finishes updating. It is this fact that severely limits the maximal utilization of GPU’s massive parallel computing power.
To overcome this difficulty, we analytically compute the explicit form of the resulted volumetric image function $f(x)$ when the operator $P^T$ acts on a function $g(x)$ on the x-ray imager and obtained a close form expression

$$f(x) = [P^T g](x) = \frac{\Delta x \Delta y \Delta z}{\Delta u \Delta v} \sum_{\theta} \frac{l^3(u^*)}{L_0 l^2(x)} g^\theta(u^*).$$

(5)

Here $u^*$ is the coordinate for a point on imager where a ray line connecting the x-ray source at $x_s$ and the point at $x$ intersects with the imager. $L_0$ is the distance from the x-ray source $S$ to the imager, while $l(x)$ and $L(u^*)$ are the distance from $x_s$ to $x$ and from $x_s$ to $u^*$ on the imager, respectively. See Fig. 1 for the geometry. $\Delta u$ and $\Delta v$ are the pixel size when we discretize the imager during implementation and $\Delta x$, $\Delta y$, and $\Delta z$ are the size of a voxel. The derivation of Eq. (3) is briefly shown in the Appendix A. Eq. (3), in fact, indicates a very efficient way of performing $f = P^T g$ in a parallel fashion. To compute $f(x)$ at a given $x$, we simply take the function values of $g(u^*)$ at the coordinate $u^*$, multiply by proper prefactors, and finally sum over all projection angles $\theta$. In numerical computation, since we always evaluate $g(u)$ at a set of discrete coordinates and $u^*$ does not necessarily coincide with these discrete coordinates, a bilinear interpolation is performed to obtain $g^\theta(u^*)$. Now it is ready to perform the parallel computing with each GPU thread for a voxel at a given $x$ coordinate. Extremely high efficiency is expected given the vast parallelization ability of the GPU.

2.2.3 Multi grid method

Another technique we employed to increase computation efficiency is multi-grid method (Brandt, 2002). It has long been known that, the convergence rate of an iterative approach solving an optimization problem is usually worsened when a very fine grid size $\Delta x$, $\Delta y$, and $\Delta z$ is used. Moreover, fine grid also implies a large number of unknown variables, significantly increasing the size of the computation task. A well known multi-grid approach can be utilized to resolve these problems. Suppose we try to reconstruct a volumetric CBCT image $f(x)$ on a fine grid $\Omega_h$ of size $h$, we could start with solving the problem on a coarser grid $\Omega_{2h}$ of size $2h$ with the same iterative approach as in Algorithm A2. Upon convergence, we smoothly extend the solution $f_{2h}$ on $\Omega_{2h}$ to the fine grid $\Omega_h$ using, for example, linear interpolation, and use it as the initial guess of the solution on $\Omega_h$. Because of the decent quality of this initial guess, only a few iteration steps of Algorithm A2 are adequate to achieve the final solution on $\Omega_h$. This idea can be further used while seeking the solution $f_{2h}$ by going to an even coarser grid of size $4h$. In practice, we employed a 4-level multi-grid scheme, i.e. the reconstruction is sequentially achieved on grids $\Omega_{4h} \rightarrow \Omega_{2h} \rightarrow \Omega_{2h} \rightarrow \Omega_h$. A considerable efficiency gain is observed in our implementation.

3. Experimental Results
In this section, we present the CBCT reconstruction results on a NCAT phantom, a Catphan phantom, and a real patient at head-and-neck region. All of the reconstructed CBCT images are of a resolution $512 \times 512 \times 70$ voxels and the x-ray imager resolution is $512 \times 384$. A total number of 40 x-ray projections are used to perform the reconstruction. The reconstruction time in all cases studied is about ~139 sec. Comparing to those currently available similar iterative CBCT reconstruction algorithms, we have achieved extremely high efficiency owing to the advanced algorithm A2 and various techniques we employed.

3.1 NCAT phantom and Catphan phantom

We first test our reconstruction algorithm with a digital NURBS-based cardiac-torso (NCAT) phantom (Segars et al., 2001) and a physical Catphan phantom (The Phantom Laboratory, Inc., Salem, NY). For the NCAT phantom, it is generated at thorax region with a high level of anatomical realism (e.g., detailed bronchial trees). X-ray projections are numerically computed along various directions with Siddon's ray tracing algorithm (Siddon, 1985; Han et al., 1999; Jacobs et al., 1998). As for the Catphan phantom case, it is scanned using a Varian On-Board Imager system (Varian Medical Systems, Palo Alto, CA) under a full-fan mode in a range of $200^\circ$. In both cases, the x-ray source to axes distance is 100 cm and the source to detector distance is 150 cm.

Figure 2. One transverse slice of the reconstructed CBCT images for the digital NCAT phantom (top) and the physical Catphan phantom (bottom). (a) Reconstruction using the TF algorithm; (b) Zoom in of (a); (c) Reconstruction using the TV algorithm; (d) Zoom in of (c); (e) Reconstruction using FDK algorithm. Arrows indicate regions where TF algorithm is superior to the TV algorithm.

Transverse slices of the reconstructed CBCT images in both cases are shown in Fig. 2. For comparison, we have also presented here the same CBCT images reconstructed from a recent developed total variation (TV) based algorithm (Jia et al., 2010b) and the conventional FDK algorithm as well. Clearly, the images produced by the conventional FDK algorithm are full of streak artifacts due to the insufficient number of projections, making these results clinically unsatisfactory. In contrast, both the TV algorithm and the TF algorithm are able to reconstruct high quality CBCT images even under this extremely under-sampling circumstance. While comparing TV and TF methods, the latter
is found to be superior in that it leads to sharper CBCT images. In particular, by inspecting details of these images it is found that TF method is able to capture a lot more fine anatomical structures than is the TV algorithm. See, for example, the area indicated by the arrows. From real Catphan phantom studies, we can also conclude that the spatial resolution is enhanced by the TF approach.

3.2 Quantitative analysis

The Catphan phantom contains a layer consisting of a single point-like structure of a diameter 0.28 mm, see Fig. 3(a). This structure enables us to measure the in plane modulation transfer function (MTF) of the reconstructed CBCT images, which characterize the spatial resolution inside the transverse plane. For this purpose, we crop a square region of size $21 \times 21$ pixel$^2$ in this slice centering at this structure. After subtracting the background, we compute the point spread function. The MTF is obtained by first performing 2 dimensional fast Fourier transform and then averaging the amplitude along the angular direction.

![Figure 3.](image)

(a) A transverse slice of the Catphan phantom used to measure MTF. (b) A transverse slice of the Catphan phantom used to measure CNR.

First, at a constant mAs level of 1.0 mAs/projection, we compare the spatial resolution in the images reconstructed from three different methods, namely TF, TV, and FDK. The patch images used to compute MTF are shown in Fig. 4(a) and the measured MTF are plotted in Fig. 4(b). It is found that the TF method results in the best MTF curve among the three methods and therefore the highest spatial resolution on the reconstructed images. On the other hand, TV and FDK give similar MTF functions, though the TV method smooths out the background noise signals while sacrificing the spatial resolution. Second, for the TF method, we compare the results obtained at different mAs levels and the results are depicted in Fig. 4(c) and (d). As expected, the spatial resolution is deteriorated when low mAs level scan is used due to more and more noise component induced in the x-ray projections.

In order to quantitatively evaluate the contrast of the reconstructed CBCT images, we measure contrast-to-noise ratio (CNR). For a given region of interest (ROI), CNR is calculated as $CNR = 2|S - S_b|/(\sigma + \sigma_b)$, where $S$ and $S_b$ are the mean pixel values over
the ROI and in the background, respectively, and \( \sigma \) and \( \sigma_b \) are the standard deviation of the pixel values inside the ROI and in the background. Before computing the CNR, a key observation is that CNR is affected by the parameter \( \mu \) which controls to what extent we would like to regularize the solution via the TF term. In fact, a small amount \( \mu \) is not sufficient to regularize the solution, leading to a high noise level and hence a low CNR. In contrast, a large \( \mu \) tends to over-smooth the CBCT image and reduce the contrast between different structures. Therefore, there exists an optimal \( \mu \) level in the reconstruction. Take the case at 0.3 mAs/projection and 40 projections as an example, we perform CBCT reconstruction with different \( \mu \) values and compute the CNRs for the four ROIs indicated in Fig. 3(b). The results are shown in Fig. 5(a). Clearly, the best CNRs are achieved for \( \mu \sim 0.10 \). In our experiments, the optimal \( \mu \) is found to be dependent on the system parameters such as mAs levels, number of projections, reconstruction resolution etc. It is, however, not sensitive to the phantoms or patient cases. Therefore, in practice, one can construct a list of the optimal \( \mu \) values based on the system parameters in different scan/reconstruction contexts, and choose a proper value accordingly. This will enable us to avoid reestablishing the \( \mu \) value in each individual reconstruction case. The construction of such a list requires a comprehensive evaluation of the TF algorithm in a large range of scan/reconstruction situations and will be our future work. Throughout this paper, all the reconstruction cases are performed under the optimal \( \mu \) values except stated explicitly.

![Figure 4](image-url)

**Figure 4.** (a) Three patches used to measure MTF in CBCT images reconstructed from TF, TV, and FDK algorithms at 1.0 mAs/projection with 40 projections. (b) MTF measurements obtained from different methods. (c) Three patches used to measure MTF in CBCT images reconstructed from TF method at 1.0, 0.3, and 0.1 mAs/projections with 40 projections. (d) MTF measured at different mAs levels.

In Fig. 5(b), we plot the dependence of CNRs on mAs levels measured in those four ROIs in the CBCT images reconstructed using the TF method. The corresponding CNRs obtained from the conventional FDK algorithm are also shown in Fig. 5(c). As expected, a higher CNR can be achieved when a higher mAs level is used in the CBCT scan, and hence those curves in Fig. 5 (b) and (c) generally follow a monotonically increasing trend. Moreover, comparing Fig. 5 (b) and (c), our algorithm yields higher CNRs than the
FDK algorithms in all ROIs studied in all cases, indicating better image contrast. Note that we do not provide CNRs obtained from the TV method here. It is observed that the TV method usually leads to much higher CNR values. Nonetheless, this is caused by the vanishingly small denominator in the CNR definition in the TV case due to its tremendous ability to suppress the image noise. The contrast in the reconstructed images from the TV method is visually not different from those produced by the TF method.

![Graphs showing CNR as a function of mAs/projection and TV method contrast](image)

**Figure 5.** (a) $CNR$s at various ROIs as functions of the parameter $\mu$ at 0.3 mAs/projection and 40 projections. (b) $CNR$s computed at various ROIs as functions of mAs levels at 40 projections reconstructed using our TF algorithm. (c) $CNR$s computed at various ROIs as functions of mAs levels at 40 projections reconstructed using the FDK algorithm.

### 3.3 Patient case

![Patient scans](image)

**Figure 6.** Three transverse slices and one sagittal slice of a real head-and-neck patient CBCT reconstructed from TF method (top) and the conventional FDK algorithm (bottom) using 40 projections.

Finally, we validate our TF-based CBCT reconstruction algorithm on realistic head-and-neck anatomical geometry. A patient’s head-and-neck CBCT scan is taken using a Varian On-Board Imager system with 363 projections in 200 degrees and 0.4 mAs/projection. A subset of only 40 equally spaced projections is selected for the reconstruction. Fig. 6 shows the reconstruction results in this case from our algorithm as well as from the
conventional FDK algorithm. Due to the complicated geometry and high contrast between bony structures, dental filling, and soft tissues in this head-and-neck region, streak artifacts are extremely severe in the images obtained from FDK algorithm under this undersampling case. On the other hand, our algorithm successfully produces CBCT image with decent quality, where artifacts are hardly observed and high image contrast is maintained. In particular, when a metal dental filling exists in the patient, our algorithm can still capture it with high contrast, whereas the FDK algorithm causes a number of streaks in the CBCT image.

4. Conclusions

In this paper, we have developed a TF-based fast iterative algorithm for CBCT reconstruction. By iteratively applying three steps to impose three key conditions that the reconstructed CBCT should satisfy, we can reconstruct high quality CBCT images with undersampled and noisy projection data. In particular, the underline assumption that a real CBCT image has a sparse representation under a TF basis is found to be valid and robust in the reconstruction, leading to high quality results. Such an algorithm is mathematically equivalent to the so called balanced approach for TF-based image restoration. In practice, due to the GPU implementation, the multi-grid method, and various techniques we employed, high computational efficiency has been achieved.

We have tested our algorithm on a digital NCAT phantom and a physical Catphan phantom. It is found that our TF-based algorithm leads to much higher quality CBCT image than those obtained from a conventional FDK algorithm in the context of undersampling and low mAs scans. It is also superior to the recently developed TV-based algorithm in terms of spatial resolution. Quantitative analysis of the CBCT image quality has been performed with respect to the MTF and CNR under various scanning cases, and the results confirm the high CBCT image quality obtained from our TF algorithm. Moreover, reconstructions performed on a head-and-neck patient have presented very promising results in real clinical applications.

Acknowledgements

This work is supported in part by the University of California Lab Fees Research Program and by Varian Medical Systems. We would like to thank NVIDIA for providing GPU cards for this project.
Appendix
Derivation of equation (4)

Let \( f(.) : \mathbb{R}^3 \to \mathbb{R} \) and \( g(.) : \mathbb{R}^2 \to \mathbb{R} \) be two smooth enough functions in the CBCT image domain and in the x-ray projection image domain, respectively. The operator \( P^\theta \), being the adjoint operator of the x-ray projection operator \( P^\theta \), should satisfy the condition

\[
\langle f, P^\theta g \rangle = \langle P^\theta f, g \rangle, \tag{A1}
\]

where \( \langle \cdot, \cdot \rangle \) denotes the inner product. This condition can be explicitly expressed as

\[
\int \! dx \ f(x) P^\theta [g](x) = \int \! du \ P^\theta [f](u) g(u). \tag{A2}
\]

Now take the functional variation with respect to \( f(x) \) on both sides of equation (A2) and interchange the order of integral and variation on the right hand side. This yields

\[
P^\theta [g](x) = \frac{\delta}{\delta f(x)} \int \! du \ P^\theta [f](u) g(u) = \int \! du \ g(u) \frac{\delta}{\delta f(x)} P^\theta [f](u). \tag{A3}
\]

With help of a delta function we could rewrite equation (1) as

\[
P^\theta [f](u) = \int \! dldx \ f(x) \delta(x - x_0 - nl). \tag{A4}
\]

Now substituting (A4) into (A3), we obtain

\[
P^\theta [g](x) = \int \! dldu \ g(u) \delta(x - x_0 - nl) = \frac{L^2(u^*)}{L_0 l^2(x)} g(u^*), \tag{A5}
\]

where \( u^* \) is the coordinate of a point on imager, at which a ray line connecting the source \( x_0 \) and the point \( x \) intersects with the imager. \( L(u^*) \) is the length from \( x_0 \) to \( u^* \) and \( l(x) \) is that from \( x_0 \) to \( x \). The source to imager distance is \( L_0 \). Additionally, a summation over projection angles \( \theta \) is performed in equation (4) to account for all the x-ray projection images.

One caveat when implementing (A5) is that this equation is derived from condition (A1), where the inner product of two functions is defined in an integral sense. In the CGLS algorithm, both \( P \) and \( P^T \) are viewed as matrices. Therefore an inner product defined in the vector sense, \( \langle f, g \rangle = \sum_i f_i g_i \) for two vectors \( f \) and \( g \), should be understood in (A1). Changing the inner product from a function form to a vector form results in a numerical factor in Eq. (4), simply being the ratio of pixel size \( \Delta u \Delta v \) to the voxel size \( \Delta x \Delta y \Delta z \). We have tested the accuracy of such defined operator \( P^T \) in terms of satisfying condition expressed in Eq. (A1). Numerical experiments indicate that this condition is satisfied with numerical error less than 1%, which is found accurate enough for our iterative CBCT reconstruction purpose.
References


Cai J F and Shen Z W 2010 Framelet based deconvolution *Journal of Computational Mathematics* **28** 289-308


Candes E J, Romberg J and Tao T 2006 Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information *Ieee Transactions on Information Theory* **52** 489-509


Donoho D L 2006 Compressed sensing *Ieee Transactions on Information Theory* **52** 1289-306


Han B and Shen Z W 2009 Dual Wavelet Frames and Riesz Bases in Sobolev Spaces Constructive Approximation 29 369-406

Han G, Liang Z and You J 1999 A fast ray-tracing technique for TCT and ECT studies 1999 Nuclear Science Symposium and Medical Imaging Conference (Cat. No.99CH37019) 10.1109/NSSMIC.1999.842846


Hissoiny S, Ozell B and Després P 2009 Fast convolution-superposition dose calculation on graphics hardware Medical Physics 36 1998-2005


Jacobs F, Sundermann E, de Sutter B, Christiaens M and Lemahieu I 1998 A fast algorithm to calculate the exact radiological path through a pixel or voxel space Journal of Computing and Information Technology - CIT 6 89-94


Jia X, Lou Y, Li R, Song W Y and Jiang S B 2010b GPU-based Fast Cone Beam CT Reconstruction from Undersampled and Noisy Projection Data via Total Variation Medical Physics 37 1757-60


Li M, Yang H, Koizumi K and Kudo H 2007 Fast cone-beam CT reconstruction using CUDA architecture Medical Imaging Technology 243-50


NVIDIA 2009 CUDA CUBLAS Library 2.1


Sharp G C, Kandasamy N, Singh H and Folkert M 2007 GPU-based streaming architectures for fast cone-beam CT image reconstruction and demons deformable registration Physics in Medicine and Biology 52 5771-83
Siddon R L 1985 Fast calculation of the exact radiological path for a 3-dimensional CT array Medical Physics 12 252-5
Tsaig Y and Donoho D L 2006 Extensions of compressed sensing Signal Processing 86 549-71
Xu F and Mueller K 2005 Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware IEEE Transactions on Nuclear Science 52 654-63
Xu F and Mueller K 2007 Real-time 3D computed tomographic reconstruction using commodity graphics hardware Physics in Medicine and Biology 52 3405-19