AN EFFICIENT FRAMEWORK FOR HIGH-QUALITY LOW-DOSE CT RECONSTRUCTION AND REFERENCE-BASED IMAGE RESTORATION

A Dissertation Presented

by

Wei Xu

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

December 2012

Stony Brook University

The Graduate School

Wei Xu

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Klaus Mueller – Dissertation Advisor
Professor, Department of Computer Science

Dimitris Samaras - Chairperson of Defense
Associate Professor, Department of Computer Science

Jerome Zhengrong Liang – Committee Member
Professor, Department of Radiology

Donald Harrington, Outside Committee Member
Professor and M.D., Stony Brook Hospital

This dissertation is accepted by the Graduate School

Charles Taber
Interim Dean of the Graduate School

Abstract of the Dissertation

An Efficient Framework for High-Quality Low-Dose CT Reconstruction and Reference-based Image Restoration

by

Wei Xu

Doctor of Philosophy
in
Computer Science

Stony Brook University

2012

CT imaging procedures have been shown to considerably increase the medical radiation dose to patients, giving rise to cancer. As a result, low dose imaging modalities have gained much attention recently. However, this renders traditional CT reconstruction processes no longer sufficient. Iterative reconstruction algorithms using numerical optimization paradigms are better suited, but they suffer from (1) expensive computation, (2) problems with the selection of optimal parameters to simultaneously optimize speed and (3) poor reconstruction quality.

To cope with these problems, we have made several contributions to low-dose CT. First, we have devised a GPU-accelerated ordered subset iterative CT reconstruction algorithm (OS-SIRT) with regularization and effective parameter learning. We generalized two algebraic algorithms (SIRT, SART) to an ordered subset scheme which balanced the speed of computation and the rate of convergence of these algorithms. Second, we mapped the computation to GPUs, achieving remarkable performance gains. Third, for high-quality reconstruction, we introduced four filters for denoising and streak artifact reduction, i.e., bilateral, trilateral, non-local means (NLM) and optimal adaptive NLM, all of which are popular in computer vision. We have used these filters within an interleaved CT reconstruction regularization pipeline and found that they compare favorably with the traditionally used TVM algorithm. Fourth, to overcome the difficulties with optimal parameter tuning within our algorithm and for any parameter-dependent applications, we devised two parameter-learning approaches – exhaustive benchmark testing and multi-objective optimization – and allow user interaction via an interactive parameter space visualization tool. We then generalized our framework to Electron Tomography. Our fifth contribution is a scheme that broadens the low-dose image restoration capability of traditional NLM filtering to also include high-dose reference images. We developed two variants. The first variant uses a prior scan of the same patient when available. The second variant generalizes this concept to a database of images of other patients to learn the reference images. Our experiments show that this scheme has vast potential to restore the quality of low-dose CT imagery.

To my beloved husband Jiansong, my parents Xiangyang and Hongyun, and my son Jacques Zhuo Chen.

Contents

List of Figures

List of Tables

Publications

Publications relevant to the topic of this dissertation:

1. W. Xu, S. Ha and K. Mueller, “Database-Assisted Low-dose CT Image Restoration,” *Medical Physics*, 2012 (tentatively accepted).
2. W. Xu and K. Mueller, “Efficient Low-Dose CT Artifact Mitigation Using An Artifact-Matched Prior Scan,” *Medical Physics*, vol. 39 (8), pp. 4748-4760, 2012.
3. W. Xu, S. Ha and K. Mueller, “Database-Assisted Low-dose CT Image Restoration,” *The Second International Conference on Image Formation in X-Ray Computed Tomography*, Salt Lake City, Utah, June, 2012 (Oral).
4. W. Xu and K. Mueller, “A Reference Image Database Approach for NLM Filter-Regularized CT Reconstruction,” *11th Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, Potsdam, Germany, pp.116-119, July 2011 (Oral).
5. Z. Zheng, W. Xu and K. Mueller, “Performance Tuning for CUDA-Accelerated Neighborhood Denoising Filters,” *11th Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine conjunct High Performance Image Reconstruction Workshop*, Potsdam, Germany, pp.52-55, July 2011 (Oral).
6. W. Xu and K. Mueller, “Using GPUs to Learn Effective Parameter Settings for GPU-Accelerated Iterative CT Reconstruction Algorithms,” *GPU Computing Gems Emerald Edition*, Chapter 43, pp.693-708, January 26, 2011.
7. W. Xu and K. Mueller, “Evaluating Popular Non-Linear Image Processing Filters for their Use in Regularized Iterative CT,” *IEEE Medical Imaging Conference*, Knoxville, TN, October, 2010.

8. W. Xu, F. Xu, M. Jones, B. Keszthelyi, J. Sedat, D. Agard and K. Mueller , "High-Performance Iterative Electron Tomography Reconstruction with Long-Object Compensation using Graphics Processing Units (GPUs)," *Journal of Structural Biology*, 171(2):142-153, 2010.
9. W. Xu and K. Mueller, "Parameter space visualizer: an interactive parameter selection interface for iterative CT reconstruction algorithms," *SPIE on Medical Imaging*, San Diego, California, Vol. 7625, 76251Q, February 2010.
10. F. Xu, W. Xu, M. Jones, B. Keszthelyi, J. Sedat, D. Agard and K. Mueller, "On the Efficiency of Iterative Ordered Subset Reconstruction Algorithms for Acceleration on GPUs," *Computer Methods and Programs in Biomedicine*, 98(3):261-270, 2010.
11. W. Xu and K. Mueller, "Learning Effective Parameter Settings for Iterative CT Reconstruction Algorithms," *10th Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pp.251-254, Beijing, China, September 2009 (Oral).
12. W. Xu and K. Mueller, "A Performance-Driven Study of Regularization Methods for GPU-Accelerated Iterative CT," *10th Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine conjunct High Performance Image Reconstruction Workshop*, pp.20-23, Beijing, China, September 2009 (Oral, Best paper award).
13. W. Xu and K. Mueller, "Accelerating regularized iterative CT reconstruction on commodity graphics hardware (GPU)," *IEEE International Symposium on Biomedical Imaging*, pp.1287-1290, Boston, MA, June 2009 (Oral).

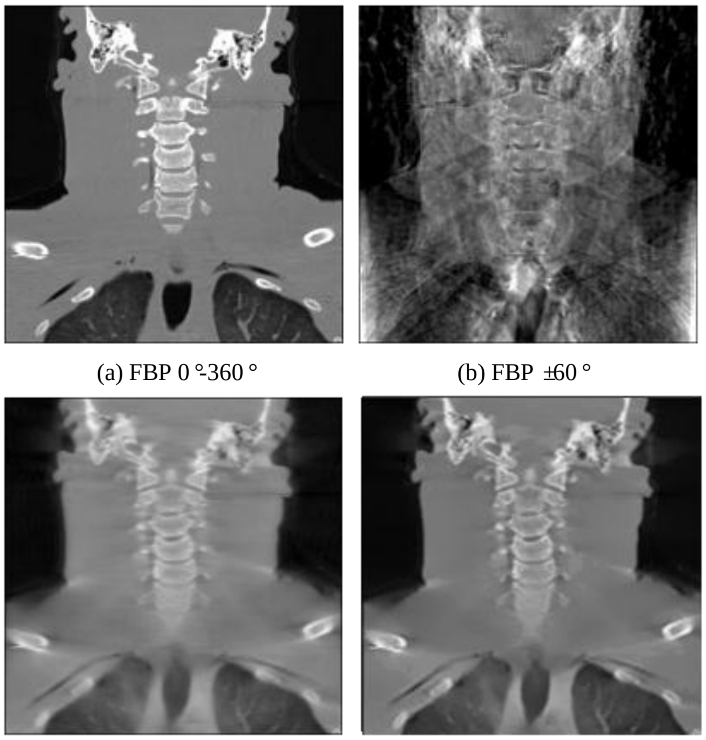
Chapter 1

Introduction

1.1 Challenges in Computed Tomography (CT)

It has been forty whole years since the public announcement of the first commercial CT scanner in 1972 [3]. CT is a medical imaging process utilizes two-dimensional X-ray images taken around the object to reconstruct the volumetric representations of its structures in the form of cross-sectional images. CT has been used widely in various medical disciplines to diagnose diseases for instance in head, lung, heart and abdomen. In these past years, both the scanner and the reconstruction algorithms have been extremely developed, which substantially lessen the scanning time, improve the quality of the produced object volume and the efficiency of the reconstruction process. The revolutionary progress of the techniques makes CT a very convenient means in the hospital for the purpose of diagnosis and therapy.

Recent studies in the US show that from 1993 to 2006 the number of CT imaging procedures has increased at an annual rate of over 10%, leading to a considerable increase in patient radiation dose [77]. Although CT only amounts to about 15% of the total number of radiological imaging procedures, it contributes to over 50% of the medical radiation dose to the US population [77]. This has rather serious consequences for example, one study suggests that 0.4% of all recent cancer deaths can be attributed to CT radiation dose incurred between 1991 to 1996 [7]. It is due to this growing dose awareness that low-dose CT imaging has been gaining considerable momentum recently. Essentially, the strategies available to lower CT patient dose are three-fold: (1) reduce the number of CT exams per procedure or health condition, (2) reduce the number of X-ray projections (views) taken per exam, and (3) reduce the amount of X-ray energy (kV, mA) expended per projection. More specifically, while a conventional CT scan typically has a dose of around 30 mGy, a 25-view low-energy flat panel cone-beam scan would reduce the dose by 95%, to around 1.5 mGy.


(a) FBP 0°-360° (b) FBP ±60°

(c) iterative ±60° (d) regularized iterative

Figure 1.1: Reconstructions of the visible human neck area, coronal cuts. (a) FBP - 360 projections, 0˚-360˚, (b) FBP - 60 projections, ±60˚, (c) iterative - 60 projections, ±60˚, (d) regularized iterative - 60 projections, ±60˚.

For the wellbeing of the patients, using less dose in the imaging procedure is preferable. However, the frugal use of X-ray radiation has its drawbacks. It starves the well-established fast and high quality analytical reconstruction schemes (such as filtered backprojection (FBP)) from the massive X-ray projection data they require, reducing their effectiveness. The realities of low-dose CT are noisy and sparse X-ray projections, which subsequently lead to significant noise and streak artifacts in the reconstructions, obliterating the structures of interest. As in Figure 1.1(b), the clear artifact is inevitable when reconstructed by FBP. Evidence exists that these adverse conditions are bound to usher in a new era in medical CT reconstruction, one in which iterative reconstruction algorithms (such as simultaneous algebraic reconstruction technique (SART)) will likely greatly gain in popularity. Iterative schemes replace the closed form mathematics of analytical CT reconstruction by numerical optimization. Then, via numerical optimization one can effectively exploit the fact that medical images offer a great deal of prior knowledge about the objects to be reconstructed, lending great prospects to offset the lack of abundant raw data in low-dose CT. For example, a much better reconstruction by SART is shown in Figure 1.1(c). Despite this promise, it is a sobering fact that numerical optimization can be excruciatingly expensive to compute. A number of iterations have to be performed to approach the convergence of the optimization, while in each iteration a series of forward projections and backprojections are incorporated.

Beside the speed issue, there are other associated difficulties which are left unsolved. With less data available we need to somehow substitute/complement the missing information. Although it is clear that CT reconstruction is always an ill-posed problem, no matter how much data is obtained [81][47], there is a great deal of anatomical knowledge available to guide the reconstruction to the most probable solution, yet not excluding the detection of unexpected pathologies. By devising a framework that does not reconstruct objects completely uninformed from scratch, but incorporates this knowledge as priors into the reconstruction procedure (referred as regularization or restoration as shown in Figure 1.1(d)) we are bound to gain a good amount of independence from (now) redundant data and the radiation that they cause.

Another issue is to find optimal parameter settings for iterative reconstruction algorithms with regularization, so that given arbitrary input datasets we could simultaneously optimize speed and reconstruction quality. Due to the fact that very few literatures work on this arena, the parameter selection is rather difficult without prior knowledge about the dataset and usually ended as an educational guess.

1.2 Contributions

In order to cope with these problems, we have made several contributions to low-dose CT:

First, we have devised a GPU-accelerated ordered subset iterative CT reconstruction algorithm (OS-SIRT) with regularization and effective parameter learning. We generalized two algebraic algorithms Simultaneous Iterative Reconstruction Technique (SIRT) and simultaneous algebraic reconstruction technique (SART) by using ordered subsets to balance the speed of computation and the rate of convergence of these algorithms.

Second, we mapped the computation to the architecture of modern commodity graphics hardware (GPUs) so that our iterative algorithms became more efficient and in fact achieved remarkable performance gains. Besides, the parameter settings were studied in order to obtain smallest reconstruction time.

Third, to achieve good quality reconstruction comparable to the one using total variation minimization as regularization operator, and meanwhile to overcome its time consuming iterative nature, we carefully proposed bilateral filter (BLF) to OS-SIRT for the purpose of performance-driven. It was shown to have similar (or even superior) reconstruction results in most cases while spending much less time.

Fourth, to gauge the reconstruction quality, beside bilateral we introduced to CT three other nonlinear neighborhood filters for denoising and the reduction of streak artifacts, i.e. trilateral (TLF), non-local means (NLM) and adaptive non-local means (ANLM), all of which are popular in computer vision. We have used these filters within an interleaved CT reconstruction regularization pipeline and found that the group of NLM filters compare favorably for both time and quality with the traditionally used total variation minimization (TVM) algorithm.

Fifth, to overcome the difficulties associated with optimal parameter tuning within our regularized iterative algorithm and more generally for any parameter-dependent applications, we devised two parameter-learning approaches – the exhaustive benchmark testing and multi-objective optimization. A parameter space visualization tool was also provided to enable interactive parameter learning for various datasets.

Sixth, our framework is very general and we demonstrated it by ways of a GPU-accelerated OS-SIRT framework for Electron Tomography (ET) as a specific example of low-dose reconstruction under difficult data acquisition settings. To conquer the vignetting effect, we devised a limited detector/long object compensation scheme.

Seventh, for more extreme low-dose conditions when too much information is missing, we devised a scheme that broadens the low-dose image restoration capability of traditional NLM filtering to also include high-dose reference images to recover the salient high-quality features back. We developed two variants. The first variant uses a prior scan of the same patient when available. A two-step restoration framework was proposed including the offline reference preparation step and the online denoising step. The second variant generalizes this concept to a database of images of other patients to learn the reference images as an extra step to the framework of the first variant. A complete image database retrieval framework was presented to match the target image. Our experiments show that this scheme has vast potential to mitigate the artifacts and restore the quality of low-dose CT imagery.

1.3 Thesis Outline

To make it more clear, the overall framework is illustrated in Figure 1.2. In the center is the patient CT image which is being operated on by two separate corrective image update processes run in iterative alternation: (1) the data-driven iterative image reconstruction and (2) the knowledge driven iterative image regularization / restoration. Both are controlled by parameters configured by a computerized observer-driven parameter optimizer. All of these processes are implemented on high-performance commodity parallel computing hardware (GPU) to ensure real-time or near real-time image feedback to the clinician. Our work was focused on the data-driven correction, knowledge-driven correction and parameter optimizer. Then, to demonstrate its generalization, we applied our framework to Electron Tomography as a specific example of low-dose CT reconstruction. Finally, we extended the image restoration capability of NLM to also include the reference image either from prior scan of the same patient, or from the database of other patients.

The remainder of the dissertation is organized as follows. In chapter 2, we briefly review the background and previous work of CT reconstruction algorithms, GPU acceleration techniques and the low-dose related works. In chapter 3, we present the GPU-accelerated iterative ordered subset reconstruction algorithms as the data-driven correction part. For the knowledge-driven correction part, we present performance-driven regularization methods for iterative CT using bilateral filter in chapter 4 and a more comprehensive study of nonlinear neighborhood filters as regularization methods in chapter 5. For the part of parameter optimizer, we describe how to learn effective parameter settings for regularized iterative CT reconstruction using both exhaustive benchmark study and multi-objective optimization in chapter 6 as well as a parameter space visualizer. In chapter 7, the application to electron tomography with long object compensation is introduced. In chapter 8, the reference based CT image restoration framework using the prior scan of the same patient is presented. In chapter 9, we extend the framework to a more general database based one where the high-dose references are from other patients. Finally, we conclude the dissertation and outline the future research work in chapter 10.

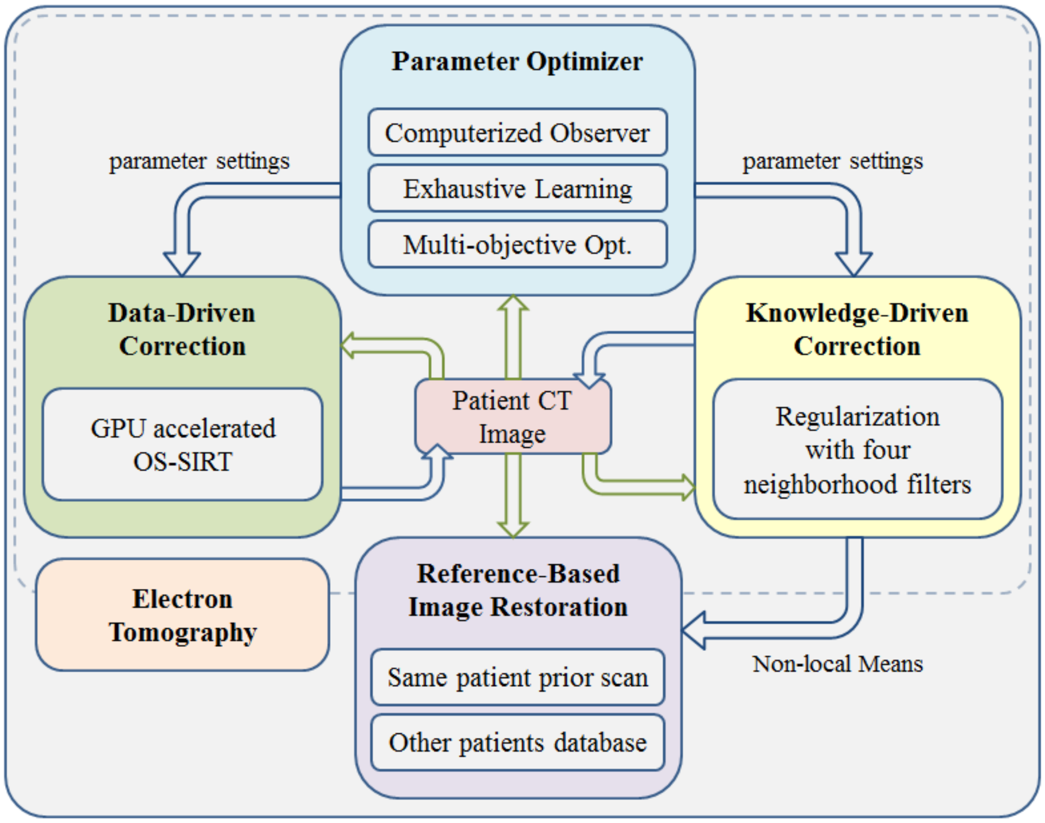


Figure 1.2: The overall framework of the dissertation.

Chapter 2

Background

2.1 CT Reconstruction Algorithms

The CT reconstruction methods could be broadly categorized to two classes: analytical algorithms and iterative algorithms. For the analytical algorithms, they are based on two solid mathematical principles - the Radon Transform and Central Slice Theorem [60]. The most popular methods are Filtered backprojection (FBP) [93] for 2D case and Feldkamp filtered backprojection (FDK) [40] for 3D case. When the number of the acquired X-ray images around the object (called projections) is sufficient, they can generate the exact or quasi-exact reconstructed results. The simplicity of these methods guarantees efficient computation which makes them popular in many clinical applications.

For the other category, the iterative methods model the imaging process as a linear algebra problem therefore the reconstruction procedure becomes solving an equation system [60]. To seek the solution (reconstructed volume), a more favorable approach is to perform a numerical optimization process. Started from an initial guess, the estimated volume is updated iteratively until the convergence is reached when the difference between the acquired measurement (projections) and the simulated measurement (projections from the estimated volume) is minimized. The iterative methods can be further categorized into projection onto convex sets (POCS) algorithms such as SART [2], SIRT [45], and POCS [132] and statistical algorithms such as EM [103], OS-EM [50], and MAP [65]). Due to its iterative nature, the computation burden of these algorithms is relatively high. However, they perform much better when low-dose conditions are met. Besides, it is more convenient to incorporate prior knowledge into the reconstruction procedure compared to analytical methods by adding another constraint to the optimization function. Considering these benefits, we chose iterative approach to code with low-dose CT problems. In chapter 3, a detailed algorithmic description of the devised framework has been shown.

2.2 GPU Acceleration Technique

The rapid growth in speed and capabilities of programmable commodity graphics hardware boards (GPUs) has propelled high performance computing to the desktop, spawning applications far beyond those used in interactive computer games. High-end graphics boards, such as the NVIDIA GeForce GTX 680, featuring 3,090 G Flops and more, are available for less than $500, and their performance is consistently growing at a triple of Moore's law that governs the growth of CPUs. Speedups of 1-2 orders of magnitude have been reported by many researchers when mapping CPU-based algorithms onto the GPU, in a wide variety of domains [145], including medical imaging [1][95][117][114]. These impressive gains originate in the highly parallel SIMD (*Same Instruction Multiple Data*) architecture of the GPU and its high-bandwidth memory access. For example, the NIVIDIA 8800 GTX has 128 such SIMD pipelines while the most recent NVIDIA card, the GTX 680, has 1536 processors (*cores*).

It is important to note, however, that the high speedup rates facilitated by GPUs do not come easy. They require one to carefully map the target algorithm from the single-threaded programming model of the CPU to the multi-threaded SIMD programming model of the GPU where each such thread is dedicated to computing one element of the (final or intermediate) result vector. Here, special attention must be paid to keep all of these pipelines busy. While there are 100s of SIMD processors on the GPU, many more threads need to be created to hide data fetch latencies. It is important to avoid both thread congestion (too many threads waiting for execution) and thread starvation (not enough threads available to hide latencies). These conditions are in addition to avoiding possible contingencies in local registers and caches that will limit the overall number of threads permitted to run simultaneously. For example, in [117], it was shown that a careful mapping of Feldkamp's filtered backprojection algorithm to the GPU yielded a $20\times$ speedup over an optimized CPU implementation, enabling cone-beam reconstructions of 512^3 volumes from 360 projections at a rate of 52 projections/s, greatly exceeding the data production rates of modern flat-panel X-ray scanners that have become popular in fully-3D medical imaging.

The great performance of GPUs comes from their highly parallel architecture, and the vast potential of these boards for general high performance computing has given rise to the recent trend of General Purpose Computing on GPUs (GPGPU)

[82]. In the past, GPU-programming was only possible via graphics APIs, such as CG, GLSL and HDSL, which required programmers to have some background in computer graphics. In order to make the hardware more accessible to non-graphics programmers, a C-like parallel computing programming interface called CUDA (Compute Unified Device Architecture) has recently been introduced by GPU manufacturer NVIDIA. A similar but more general API called OpenCL has also become available. We have used CG, GLSL and CUDA for implementation in our works. In this dissertation, we describe our framework with the concept of parallel computing while ignoring the details of implementation with specific GPU programming languages.

2.3 Low-dose CT

Motivated by the need to minimize the radiation exposed to patients, a growing number of research efforts have been dedicated to the topic of low-dose CT. Lowering the radiation dose can be achieved either by reducing the number of X-rays, their energy, or both. However, a direct effect of these dose reduction efforts are CT images with strong noise artifacts, streaks and reduced feature detail – all of which impede image readability in diagnostic tasks. To overcome these problems one can either apply iterative reconstruction schemes with the goal of optimizing the reconstruction given the limited data [52][101] or one can try to reduce the artifacts in the image domain via a suitable image restoration method [55].

For the latter option, neighborhood filters, in particular the Non-Local Means (NLM) filter [4], have shown great promise for the restoration of degraded low-dose CT imagery [71][122]. Originally devised for general image de-noising tasks, NLM is essentially an extended Gaussian filter. It updates a given pixel by looking for pixels with statistically similar local neighborhoods in the image and then Gaussian-weighs their contributions by the degree of similarity. The extent of the search is specified by a search window, while the size of the neighborhood used for similarity matching is called a neighborhood patch. A more recent trend in CT reconstruction has been to extend the search window beyond the image subject to restoration. Schemes have been devised that utilize a prior scan of the same patient to search for high-quality updates [55][75][128][135]. While this produces excellent results, it does require a prior high-dose scan of the patient which may not be available.

Therefore, using collections of images to reduce noise is explored. There are in fact two rather disjoint schools of thought, and both aim to cope with the extremely large space of possible image detail. The first approach first constructs a large-scale database of possible detail at some level and then uses a sophisticated matching strategy to retrieve the detail of interest from this database

[48]. The other approach is based on sparse coding. It first constructs a dictionary of representative base patterns which must then be optimally combined for reproducing the desired detail of interest [32]. While the first approach is a top-down search, the second is bottom-up. Both strategies can be justified by theories on how humans perform visual search, which likely is a conjunction of both [51].

Chapter 3

GPU-Accelerated Ordered Subset Iterative CT Reconstruction Algorithms (OS-SIRT)

Expectation Maximization (EM) and the Simultaneous Iterative Reconstruction Technique (SIRT) are two iterative computed tomography reconstruction algorithms often used when the data contain a high amount of statistical noise, have been acquired from a limited angular range, or have a limited number of views. A popular mechanism to increase the rate of convergence of these types of algorithms has been to perform the correctional updates within subsets of the projection data. This has given rise to the method of Ordered Subsets EM (OS-EM) and the Simultaneous Algebraic Reconstruction Technique (SART). Commodity graphics hardware (GPUs) has shown great promise to combat the high computational demands incurred by iterative reconstruction algorithms. However, we find that the special architecture and programming model of GPUs add extra constraints on the real-time performance of ordered subsets algorithms, counteracting the speedup benefits of smaller subsets observed on CPUs. This gives rise to new relationships governing the optimal number of subsets as well as relaxation factor settings for obtaining the smallest wall-clock time for reconstruction – a factor that is likely application-dependent. Finally, while we restrict our study to the behavior of algebraic reconstruction algorithms, similar trends will be observed with EM as well.

3.1 Introduction

The rapid growth in speed and capabilities of programmable commodity graphics hardware boards (GPUs) has announced the generation of General Purpose GPU (GPGPU) showing its wide application for general computation tasks. It is important to note, however, that the high speedup rates facilitated by GPUs do not come easy. They require one to carefully map the target algorithm from the single-threaded programming model of the CPU to the multi-threaded SIMD programming model of the GPU.

The compute-intensive nature of iterative reconstruction algorithms motivated their acceleration via commodity graphics hardware early on, first using graphics workstations [80], and later GPU boards [115]. We now refine these works by analyzing the acceleration of iterative reconstruction algorithms more closely in terms of the underlying GPU programming model and architecture. Iterative algorithms are different from analytical algorithms in that they require frequent synchronization which interrupts the stream of data, requires context switches, and limits the number of threads available for thread management. Iterative algorithms, such as Expectation Maximization (EM) [103] or the Simultaneous Iterative Reconstruction Technique (SIRT) [45] consist of three phases, executed in an iterative fashion: (1) projection of the object estimate, (2) correction factor computation (the updates), and (3) backprojection of the object estimate updates. Each phase requires a separate pass. Flexibility comes from the concept of ordered subsets, which have been originally devised mostly because they accelerated convergence. The projection data is divided into groups, the subsets, and the data within each of these groups undergo each of the three phases simultaneously. Here, it was found that the larger the number of subsets (that is, the smaller the groups) the faster is typically the convergence, but adversely also the higher the noise since there is more potential for over-correction. In EM, the method of Ordered Subsets (OS-EM) has become widely popular. OS-EM conceptually allows for any number of subsets, but the limit with respect to noise has been noted already in the original work by Hudson and Larkin [50]. For the algebraic scheme, embodied by SIRT, the Simultaneous Algebraic Reconstruction Technique (SART) [2] is also an OS scheme, but with each set only consisting of a single projection. In SART, the over-correction noise is kept low by scaling the updates by a relaxation factor $\lambda<1$. Block-iterative schemes for algebraic techniques have been proposed as well [11]. In fact, the original ART [46] is the algorithm with the smallest subset size possible: a single data point (that is, ray or projection pixel).

It is well known that SART converges much faster than SIRT, and a well-chosen λ can overcome the problems with streak artifacts and reconstruction noise, allowing it produce good reconstruction results [1]. On the CPU, faster rate of convergence is directly related to faster time performance. But, as we previously demonstrated [118], when it comes to acceleration on a streaming architecture such as the GPU, SART is not the fastest algorithm in terms of time performance. In fact, the time performance is inversely related to the number of subsets, making SIRT the faster scheme. This is due to the overhead incurred by the frequent context switching when repeatedly moving through the three iterative phases: projection, correction, and backprojection. In the same work, we also demonstrated that specific subset sizes can optimize both reconstruction quality and performance. However, the influence of the relaxation factor λ had not been considered in these experiments.

In this chapter we first introduce the fundamental module of our framework – OS-SIRT algorithm and then complete the study of [118]. We also investigate the role of λ on GPU reconstruction speed performance, in relation to subset size. Here we find that an optimal choice of λ can have a great impact. Despite the fact that SART is slower than SIRT per iteration, an optimized λ–setting can reduce the number of required iterations for convergence to a greater extent than the extra per-iteration cost incurred by GPU data streaming and context switching. This reinstates SART and the lower subset versions of SIRT as the fastest iterative CT reconstruction schemes also on GPUs.

We shall note, however, that the optimal setting is likely application dependent, which is not unique to the GPU setting alone. Here we make the reasonable assumption that a certain application will always incur similar types of data and thus an optimal parameter setting, once found, will likely be close to optimal for all data within that application setting. In that sense, our aim is not to provide optimal subset and relaxation factor settings for all types of data, but rather to raise awareness to this phenomenon and offer an explanation.

This chapter is structured as follows. First, in Section 3.2, we discuss iterative algorithms in the context of ordered subsets, present a generalization of SIRT to OS-SIRT, and describe their acceleration on the GPU. Then, in Section 3.3, we study the impacts of both subset size and relaxation factor on GPU reconstruction performance and present the results of our studies. Finally, Section 3.4 ends with conclusions.

3.2 Accelerating Iterative Reconstructions on GPUs

In the following discussion, we have only considered algebraic reconstruction algorithms (SART, SIRT), but our arguments and conclusions readily extend to expectation maximization (EM) algorithms as well since they are very similar with respect to their mapping to GPUs [115].

3.2.1 Iterative Algebraic Reconstruction: Decomposition into Subsets

Most iterative CT techniques use a projection operator to model the underlying image generation process at a certain viewing configuration (angle) φ. The result of this projection simulation is then compared to the acquired image obtained at the same viewing configuration. If scattering or diffraction effects are ignored, the modeling consists of tracing a straight ray r_i from each image element (pixel) and summing the contributions of the (reconstruction) volume elements (voxels) v_j. Here, the weight factor w_{ij} determines the contribution of a v_j to r_i and is given by the interpolation kernel used for sampling the volume. The projection operator is given as:

$$r_i = \sum_{j=1}^{N} v_j \cdot w_{ij} \qquad i = 1, 2, \ldots, M \tag{3.1}$$

where M and N are the number of rays (one per pixel) and voxels, respectively. Since GPUs are heavily optimized for computing and less for memory bandwidth, computing the w_{ij} on the fly, via bilinear interpolation, is by far more efficient than storing the weights in memory. The correction update for projection-based algebraic methods is computed with the following equation:

$$v_j^{(k+1)} = v_j^{(k)} + \lambda \frac{\sum_{p_i \in OS_i} \frac{p_i - r_i}{\sum_{i=1}^{N} w_{il}} w_{ij}}{\sum_{i=1}^{N} w_{ij}} \qquad r_i = \sum_{l=1}^{N} w_{il} \cdot v_l^{(k)} \tag{3.2}$$

We have written this equation as a generalization of the original SART and SIRT equations to support any number of subsets. Here, the p_i are the pixels in the M/S acquired images that form a specific (ordered) subset OS_s where $1 \leq s \leq S$ and S is the number of subsets. The factor λ is the relaxation factor, as mentioned above, which will be subject to optimization. The factor k is the iteration count, where k is incremented each time all M projections have been processed. In essence, all voxels v_j on the path of a ray r_i are updated (corrected) by the difference of the projection ray r_i and the acquired pixel p_i, where this correction factor is first normalized by the sum of weights encountered by the (back-projection) ray r_i. Since a number of back-projection rays will update a given v_j, these corrections need also be normalized by the sum of (correction) weights. For SIRT, these normalization weights are trivial.

3.2.2 GPU-Accelerated Reconstruction: Threads and Passes

The NVIDIA 8800 GTX board has 128 generalized SIMD processors. Up to very recently, the only way to interface with GPU hardware was via a suitable graphics API, such as OpenGL or DirectX, and using CG [FK03] (or GLSL or HLSL) for coding *shader* programs to be loaded and run on the SIMD *fragment processors*. With the introduction of a new API, CUDA (Compute Unified Device Architecture) [143], the GPU can now directly be perceived as a multi-processor. With CUDA, the CG fragments become the CUDA (SIMD) *computing threads* and the shader programs become the computing kernels. With the CUDA specifications, much more information about the overall GPU architecture is now available, which helps programmers to fine-tune thread and memory management to optimize performance, viewing GPUs as the multi-processor architecture it really is. Reflecting this GPGPU trend, new GPU platforms have now become available that do not even have graphics display capabilities, such as the NVIDIA Tesla board. Although we used GLSL shaders to obtain the results presented in this chapter, similar symptoms also occur in CUDA where synchronization operations have to be formally called to finish executions of all threads within a thread block to resume the pipeline. After all, the underlying hardware and its architecture remain the same, just the API is different.

A number of papers [115][117] have described in great detail how projection and backprojection operations (phases 1 and 3) can be efficiently performed on the GPU. Since the subject is mainly the impact of the iterative update schedule on the management of computing threads, we shall express all operations in that context, neglecting the API used for implementation (CG or CUDA). In this work, the 3D object estimate is stored as a stack of slices (2D arrays). For projection a thread is spawned for each target pixel, interpolating the slices according to the projective viewing transform. For backprojection a thread is spawned for each target voxel and projection image, interpolating the projection images according to the (same) projective viewing transform. The computation of the correction factors and the normalizations are simple vector operations.

The GPU memory model differentiates itself significantly from its CPU counterpart, posing greater restrictions on memory access operations in order to reduce latency and increase bandwidth. Here not only registers and local memory are reduced or even completely eliminated – the global memory also allows only read instructions during the computation. Further, the write operator can be executed only at the end of a computation, when the thread (or fragment) is released from the pipeline, to be blended with the target. Therefore, in general-purpose computing using GPUs, computations are triggered by initializing a “pass”. A pass includes setting up the computation region and attaching a kernel program to simultaneously apply specific operations on every thread generated. The data is then streamed into the pipeline, where the modification can be done only at the end of the pass. Cycles and loops within a program can be implemented either inside the kernel or by running multiple passes. The former is generally faster since evoking a rendering pass and storing intermediate results in memory are costly, but there exists a register count limit in the current hardware which prevents unconstrained use of loops in the kernel.

3.2.3 Ordered Subsets: GPU-Accelerated Reconstruction from Projections

Having described the relevant elements of the underlying GPU hardware we are now ready to describe their impact in the context of subsets. Equation 3.2 above described the generalization of algebraic reconstruction into an *OS* configuration. What is left to define is how the subsets OS_s are composed and how λ is chosen for given number of subsets *S*. As specified above, OS_s is the set of projections contained in each subset, to be used in a pair of simultaneous forward projections and simultaneous backward projections. In our application, each subset has the same number of projections, that is $|OS_s|=|OS|$, which is typical. Thus, the total number of projections is $M = |OS| \cdot S$. The traditional way of filling a certain subset OS_s is to select projections whose indices m ($1 \le m \le M$) satisfy $m \bmod S = s$. This is what has been adopted in OS-EM. In contrast, we use a randomized approach to fill the subsets, which we find yields better results than the regular subset population approach. For this, we simply generate a projection index list in random order and sequentially divide this list into *S* subsets.

In [118] we proposed the following linear equation for the relaxation factor λ to be used for an arbitrary *S*, setting λ =1 for SIRT and $\lambda = \lambda_{SART} = 0.1$ for SART:

$$\lambda = (\lambda_{SART} - 1)(\frac{S-1}{N-1}) + 1 \qquad 1 \le S \le N \tag{3.3}$$

This scheme sought to balance the smoothing effect achieved by the application of a larger set of simultaneous projections (employed in SIRT) with that obtained by a lower relaxation factor (when a smaller set of projections is applied with SART or SIRT with smaller subsets). That is, the lesser projections in a subset, the lower the λ.

In our current work, we collected results for all possible integer-subsets, each for a representative set of λ-levels, to produce a more accurate $\lambda(S)$ function than the linear function in (3). In the following, we shall call the first scheme the *linear λ-selection scheme,* and the second method the *optimal λ-selection scheme*.

In general, the number of subsets has the following impact on GPU-performance per iteration. In the projection phase, each (pixel) thread computes its entry point, exit point, and ray direction vector (or it looks these up from a texture) and interpolates the slices in SIMD lock step. So, when the size of the subset increases (the OS_s projections), more threads are being spawned. In the backprojection, each voxel (thread) computes its mapping to all OS_s projections and interpolates its updates. Having a greater number of projections in the set makes the kernel program longer and increases its efficiency. This is in addition to the reduction of the number of context switches as OS_s increases. The above are the main reasons for the lower speed of single iterations with SART or large-*OS* SIRT, over single iterations with basic SIRT or low-*OS* SIRT. The results we obtained will show, however, that these extra costs incurred by SART and large-*OS* SIRT are more than compensated by the faster convergence rates they can achieve, given proper λ settings.

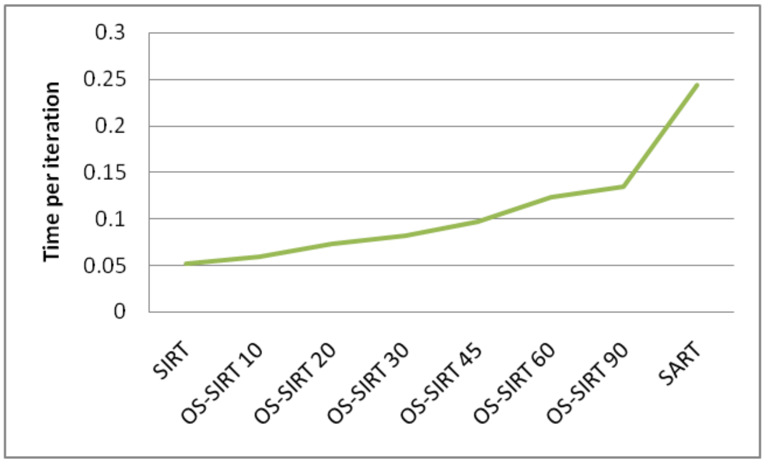
3.3 Experiments and Results

All our experiments were conducted on an NVIDIA 8800GTX GPU, programmed with GLSL. For the first set of experiments we used the 2D Barbara test image to evaluate the performance of the different reconstruction schemes described above. We used this image, popular in the image processing literature, since it has several coherent regions with high-frequency detail, which present a well observable test for the fidelity of a given reconstruction scheme. The target 2D image is obtained by cropping the original image to an area of 256×256 pixels resolution. We obtained 180 projections at uniform angular spacing of [-90˚, +90˚] in a parallel projection viewing geometry. We also simulated a limited-angle scenario, where iterative algorithms are often employed. Here, we produced 140 projections in the interval [-70˚, +70˚]. All reconstructions used linear interpolation.

We will use the cross-correlation coefficient (CC) as the metric to measure the degree of similarity between the original image and its reconstruction:

$$CC = \frac{\sum_j (r_j - \mu_r)(o_j - \mu_o)}{\sqrt{\sum_j (r_j - \mu_r)^2 \sum_j (o_j - \mu_o)^2}} \tag{3.4}$$

where j counts the number of image pixels, r_j and o_j are pixels in the reconstruction and original image, respectively, and the μ factors are their mean values.



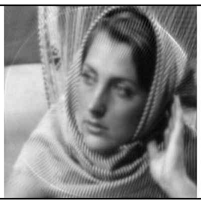
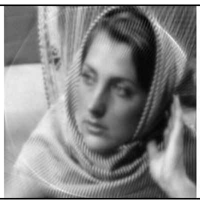
Original	SIRT	OS SIRT 10
# iterations	87	47
		
OS SIRT 20	OS SIRT 60	SART
32	15	6

Figure 3.1: (upper) Time per iteration (in seconds) as a function of number of subsets; (lower) Reconstructions obtained with the linear λ–selection schedule for various subset sizes for a fixed CC=0.95 and 180 projections in an angular range of 180˚. We observe that OS SIRT with 10 subsets of 18 projections each reaches this fixed CC value 12% faster than SIRT and 50% faster than SART.

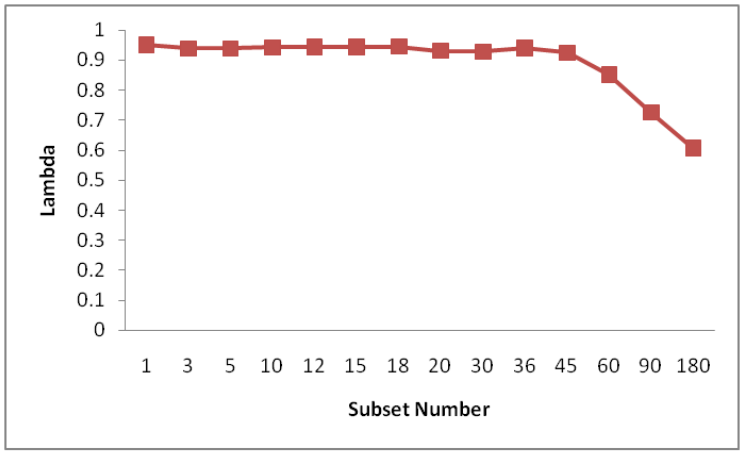
We shall now explore if there is an optimum in terms of the number of subsets. Such an optimal subset size could then be used to generate the best reconstruction in the smallest amount of (wall-clock) time. First, we evaluated the time per iteration for each configuration. Figure 3.1(upper) plots the time per iteration for each subset configuration, for the full-angle case of 180 projections. We observe a roughly linear relationship between the number of subsets and the time per iteration, with SIRT requiring the smallest and SART the longest time (about 5 times more than SIRT which is significant). This was to be expected given the arguments provided above.

However, in practice we are not interested in time per iteration but in time for convergence. Next, we test the two strategies presented (the linear and the optimal λ-selection schemes) to choose the relaxation factor λ for each possible integer-subset configuration.

Figure 3.1(lower) shows the reconstruction results obtained with the linear λ-selection schemes, for a fixed CC (comparing reconstruction with the original) which means that all reconstructed images are nearly identical to each other (in terms of statistical error). We observe that the smaller the number of subsets, the greater the number of iterations that are required to reach the set convergence threshold. We measured that with the linear λ-selection scheme SART on the GPU takes nearly twice as long as SIRT and using 10 subsets (5 for the limited angle case with fewer projections) achieves the best timing performance compared to the other subset configurations. For a CPU implementation, where the wall clock time is directly related to the number of iterations, SIRT would be roughly 87/6=14 times slower than SART. However, due to the mentioned overhead involved in the GPU-based framework, different wall-clock times are produced with a GPU implementation and the ratio is not that severe.

We next evaluate the optimal λ-selection scheme. Figure 3.2(upper) shows the plot of the optimal λ found for each subset configuration for a CC=0.95. We observe that the relationship is far from linear, with λ being relatively stable at around 0.95 until OS=45 and then falling off to around 0.6 for SART. Thus, the linear curve underestimates λ most of the time, leading to unnecessarily small corrective updates. Figure 3.2(lower) shows the corresponding reconstruction results for the Barbara dataset. We see that SART now is the fastest algorithm, roughly 20 times faster than SIRT.

Figures 3.3 (a) and (b) compare the reconstruction quality vs. wall clock time for both λ-schedules, respectively. We clearly observe that SIRT behaves very similarly, due to the similar λ-factor chosen, while the other subsets converge much more expediently for the optimal selection scheme, with SART reaching convergence after just one iteration, closely followed by OS-SIRT 60.



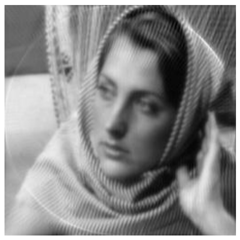
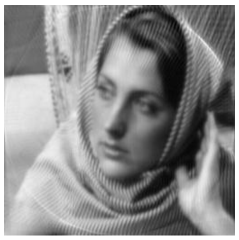
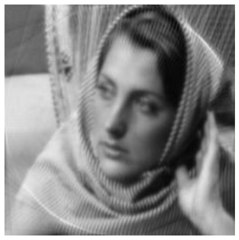
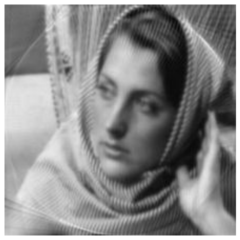
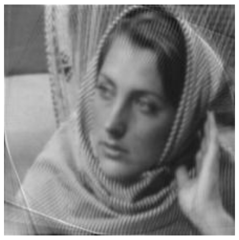
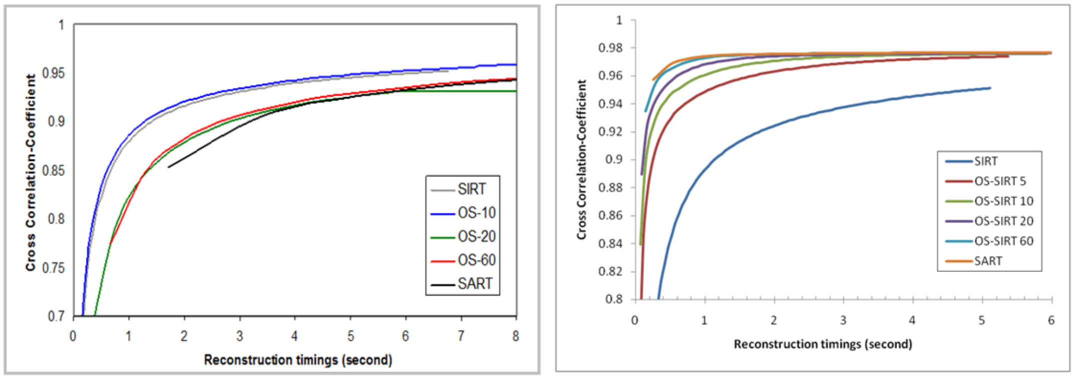
		
Original	SIRT	OS-SIRT 10
# Iterations	95	10
		
OS-SIRT 20	OS-SIRT 60	SART
5	2	1

Figure 3.2: (upper) Optimal relaxation factor λ as a function of number of subsets for the optimal λ–selection scheme; (lower) Reconstructions with the optimized λ–selection scheme obtained with various subset sizes for a fixed CC=0.95 and 180 projections in an angular range of 180˚. We observe that now the number of subsets is directly related to wall clock computation time, with SART being the fastest.



(a) (b)

Figure 3.3: (a) CC vs. wall clock time for the linear λ-selection scheme. We observe that OS-SIRT achieves the reconstruction in the smallest amount of time, within our GPU-accelerated framework; (b) CC vs. wall clock time for the optimized λ-selection scheme. We observe that now OS-SART achieves the reconstruction it in the smallest amount of time, with a clear ordering from small subsets to larger ones.

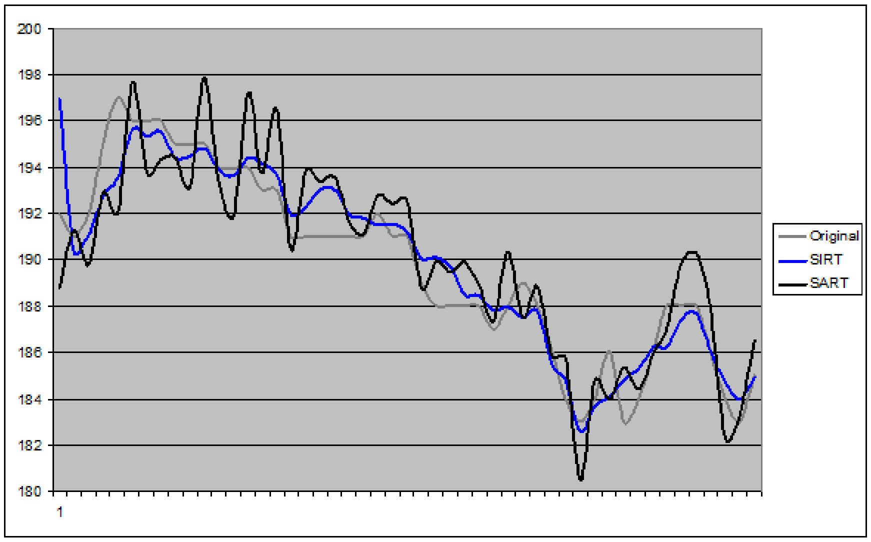


Figure 3.4: Line profiles of the image background for SART, SIRT, and the original Barbara image.

The tendency of SART to produce reconstructions noisier than the original and that of SIRT to produce reconstructions smoother than the original is also demonstrated in Figure 3.4, where we show the renditions of a line profile across another area of the Barbara image (only for the original image and reconstructions with SART and SIRT).

We also studied a medical dataset, a slice of a baby head of size 256^2. To assess the error we used the R-factor:

$$R = \frac{\sum_j (|SO_j| - |SC_j|)}{\sum_j |SO_j|} \tag{3.5}$$

where the *SO* and *SC* are the acquired and simulated sinograms, respectively. The R-factor is more practical for real reconstruction scenarios because it measures convergence based on the acquired data and not the (typically unavailable) object. All reconstructions were stopped once an R-factor of 0.007 was reached.

Figure 3.5 shows the results obtained with the linear λ−selection framework. We observe that OS-SIRT with 10 subsets of 18 projections each reaches the preset R-factor 26% faster than SIRT and 86% faster than SART.

Figure 3.6 shows the reconstruction results of the baby head obtained with the optimized λ−selection scheme (shown above the table) from 180 projections in an angular range of 180˚. We see that OS-SIRT with 20 subsets is fastest in this case. We also note that the time is about 7 times faster than with the linear selection scheme since the optimal λ-factor is higher. Finally, Figure 3.7 presents reconstructions of the baby head from a limited set of 64 projections in an angular range of 180˚ again using the optimized λ−selection scheme We observe that OS-SIRT with 16 subsets is the fastest in this case. The reconstruction time is close to a mere 10th of a second which demonstrates the capability of GPUs to also facilitate iterative reconstructions in interactive modes (10 frames/s).

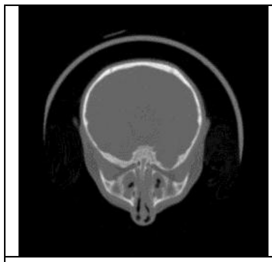
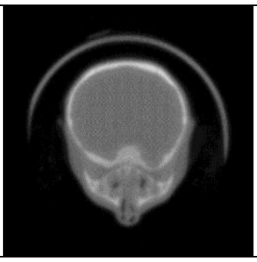
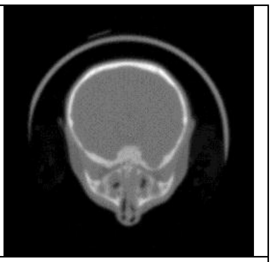
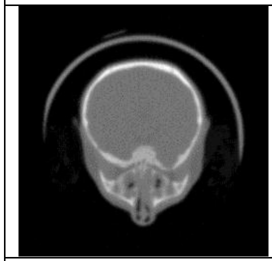
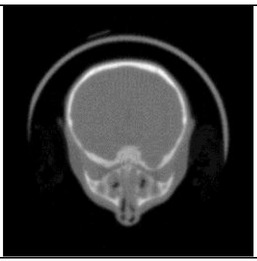
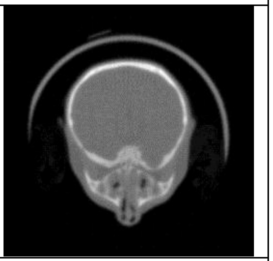
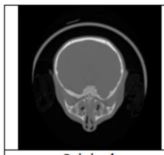
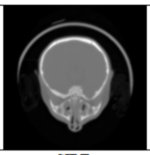
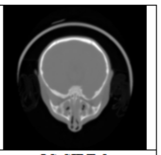
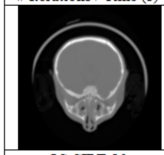
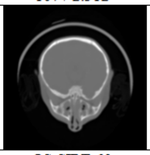
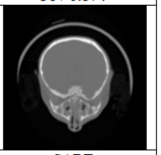
		
Original	SIRT	OS-SIRT 3
# Iterations	100	85
		
OS-SIRT 10	OS-SIRT 60	SART
54	16	6

Figure 3.5: Reconstructed baby head using high-quality simulated projection data of the volume labeled 'Original'. Results were obtained with the linear λ–selection scheme with various subset sizes for a fixed R-factor = 0.007 and 180 projections in an angular range of 180˚. OS-SIRT with 10 subsets of 18 projections each reaches this set R-factor value 26% faster than SIRT and 86% faster than SART.

		
Original	SIRT	OS-SIRT 3
# Iterations / Time (s)	107 / 2.312	36 / 0.874
		
OS-SIRT 20	OS-SIRT 60	SART
6 / 0.203	6 / 0.328	6 / 0.734

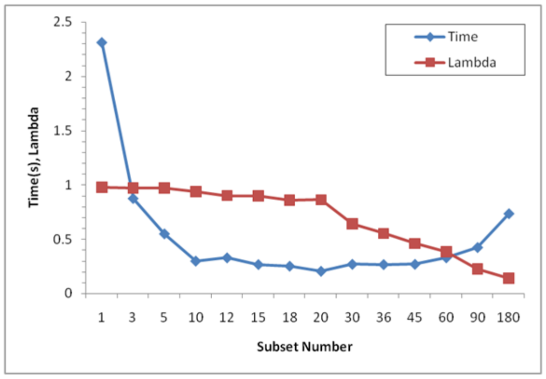


Figure 3.6: Reconstructed one slice of baby head using high-quality simulated projection data of the volume labeled 'Original'. Results were obtained with the optimized λ–selection scheme with various subset sizes for a fixed R-factor = 0.007 and 180 projections in an angular range of 180˚. OS-SIRT with 20 subsets of 9 projections each is the fastest in this case. It reaches this set R-factor value 91% faster than SIRT and 72% faster than SART.

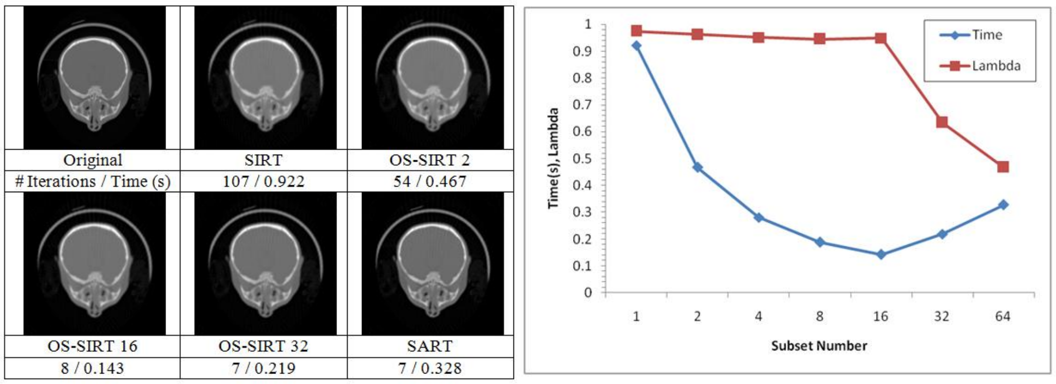


Figure 3.7: Reconstructed one slice of baby head using high-quality simulated projection data of the volume labeled ‘Original’. Results were obtained with the optimized λ–selection scheme with various subset sizes for a fixed R-factor = 0.007 and 64 projections in an angular range of 180˚. OS-SIRT with 16 subsets of 4 projections each is the fastest in this case. It reaches this set R-factor value 84% faster than SIRT and 56% faster than SART.

3.4 Conclusions

We have shown that iterative reconstruction methods used in medical imaging, such as EM or SIRT, have special properties when it comes to their acceleration on GPUs. While splitting the data used within each iterative update into a larger number of smaller subsets has long been known to offer greater convergence and computation speed on the CPU, it can be vastly slower on the GPU. This is a direct consequence of the thread fill rate in the projection and backprojection phase. Larger subsets spawn a greater number of threads, which keeps the pipelines busier and also reduces the latencies incurred by a greater number of passes and context switches. This is different from the behavior on CPUs where this decomposition is less relevant in terms of computation overhead per iteration.

We have also shown that these effects can be mitigated by optimizing the relaxation factor λ, restoring the theoretical advantage of data decompositions into smaller ordered subsets. In this chapter we were mainly concerned with demonstrating that the poor per-iteration GPU performance for iterative CT with small subsets (in the limit SART) can be compensated for by choosing an appropriate λ-factor for the reconstruction. This leads to fast convergence of the small subset methods and thus provides a good amortization of the high per-iteration cost.

The question now remains how one chooses such an optimal λ-value in a practical setting. While this was not the topic of this chapter, preliminary results have shown that the optimal number of subsets seems to vary depending on the domain application, the general reconstruction scenario, and also the level of noise present in the data. Thus, in order to identify the optimal subset number, as well as λ, for a new application setting and noise level, to be used later for repeated reconstructions within these scenarios, one may simply run a series of experiments with different numbers of subsets and λ-settings, and then use the setting combination with the shortest wall clock time required for the desired reconstruction quality. In fact, such strategies are typical for GPU-accelerated general-purpose computing applications (GPGPU). For example, the GPU bench was designed to run a vast benchmark suite [146] to determine the capabilities of the tested hardware. In chapter 6, we will present the study of parameter setting related to the noise level.

We also believe that our findings with SIRT readily extend to EM since the two methods have, as far as the computations are concerned, similar operations and overhead. Finally, although we have used GLSL shaders to obtain the results presented, similar symptoms also occur in CUDA where synchronization operations have to be formally called to finish executions of all threads within a thread block to resume the pipeline. In general, the underlying hardware, its architecture, and the overall thread management remain the same – just the API is different, enabling a tighter control over the threads and also memory. As future work, the reported effects to a more detailed extent in CUDA could be studied, to determine if a shift in the performance-optimal subset configuration occurs. But in fact, this is likely to happen for every new generation of the hardware.

Chapter 4

Bilateral Regularized OS-SIRT

Iterative reconstruction algorithms can produce better reconstructions from few views and in the presence of significant noise than analytical algorithms. In this chapter we move forward to seek for even improved quality by incorporating the regularization into the reconstruction. We focus on the particularities associated with the GPU acceleration of it. Specifically, we not only focus on reconstruction speed but also on reconstruction quality which reveals a number of important interaction effects and trade-offs. To obtain this insight, we use exhaustive benchmark tests to determine the optimal settings of the various parameters associated with the algorithm, here OS-SIRT. The same mindset we also apply in the selection of the most GPU-amenable regularization mechanism, where we compare the traditionally used TVM filter with the less frequently used bilateral filter, which we find to be a viable and cost-effective means for regularization.

4.1 Introduction

Iterative reconstruction methods have gathered significant interest in recent years since they can cope well with limited projection sets and noisy data. These scenarios occur most often in low-dose CT, where one seeks to either limit the dose per projection, or the number of projections overall, or both. Low dose CT has been a response to growing concern about the high radiation dose delivered to a patient in multi-slice X-ray CT, but the noise associated with reduced radiation dose decreases SNR and the few-view scenario can lead to prominent streak artifacts in the reconstruction. Both can obliterate the features of interest and generally make the CT image hard to read. While exact or approximate exact CT reconstruction methods do not work well under these conditions, iterative methods can produce acceptable results. These methods, however, suffer from high computational effort, which has prevented a deployment in routine clinical applications so far as these computational demands cannot be met by reasonable CPU-based platforms.

High-performance graphics chips (GPUs) are poised to provide a breakthrough in this problem as we introduced in last chapter. In this chapter we specifically address the acceleration of iterative optimization algorithms for the purpose of low-dose CT with reduced sets of noisy projections. Our framework alternates projection-space prediction-correction with object-space regularization. The former ensures adherence of the solution to the data, while the latter seeks to drive the former to a more plausible solution. Our prominent aim is to make this procedure amenable to GPU-acceleration.

4.2 Related Work

We chose the algebraic reconstruction method OS-SIRT as the predictor-corrector method. In this scheme SIRT and SART form two extremes, with SIRT having just one and SART having M subsets (M being the number of projections). We showed that while on the CPU there is little difference in the running time per iteration, on the GPU an iteration with SART is typically the slowest, due to the many projection-backprojection context switches which disturb parallelism and data flow. This has significant implications for the overall reconstruction wall clock time, where SART, in the noise-free case, is no longer the fastest method (which it is on the CPU). This effect has also been observed by other authors [57], but there the focus was solely on reconstruction speed. In contrast, we have found, in the present work, that once reconstruction quality is considered as well, these relationships are altered and SART becomes more competitive again. In addition to this insight, we also address the issue of noise, and revisit GPU OS-SIRT under these new circumstances.

For few-view, limited-angle, and noisy projection scenarios, the application of regularization operators between reconstruction iterations seeks to tune the final or intermediate results to some *a-priori* model. A simple regularization scheme is to enforce positivity. In [97], the method of total variation (TV) was proposed for additional regularization (in conjunction with POCS reconstruction). TV minimization (TVM) has the effect of flattening the density profile in local neighborhoods and thus is well suited for noise and streak artifact reduction. Based on the assumption of a relatively sparse gradient object, the method has been shown to work quite well under a variety imperfect imaging situations, yet this assumption may not be realistic in general. In computer vision, two prominent TV models are frequently used, that is, the ROF model and the TV-L^1 model [86].

A number of variational algorithms have been designed as a minimizer of the energy functional of the models. They are mainly based on solving the associated Euler Lagrange differential equation with optimization techniques. These methods are well suited for the removal of noise and other unwanted fine scale details while preserving edges. However, in the context of high performance computing, due to its iterative procedure TVM is quite time-consuming, even when accelerated on GPUs.

4.3 Methodology

4.3.1 Bilateral Filter

We aim to devise a method that is not iterative but has the same goals as TVM, that is, the reduction of local variations (noise, streaks) while preserving coherent local features. The bilateral filter [107] is such a method. It combines a range filter with a domain filter, giving rise to a non-linear filter designed for edge-preserving smoothing. When based on the Gaussian function, two parameters are required, σ_r and σ_d, to control the weight of each filter. We then compare this filter with a TVM method [10] to explore its performance under different scenarios.

The bilateral filter non-linearly averages similar and nearby pixels values. To achieve effective and efficient computation, the averaging only occurs inside a fixed window area. It consists of two filter components, the domain filter and the range filter:

$$h(x) = \frac{\sum_{\varepsilon \in W} f(\varepsilon) c(\varepsilon, x) s(f(\varepsilon), f(x))}{\sum_{\varepsilon \in W} c(\varepsilon, x) s(f(\varepsilon), f(x))} \tag{4.1}$$

Here, W is the window centered at x, ε and x represent the spatial variables, f is the input image, and c and s are the measured closeness and pixel value similarity, respectively. The geometric closeness function acts as the domain filter controlling the contribution according to spatial distance, while the pixel value similarity function acts as a range filter generating very low weights for dissimilar pixel values. Normalization forces the sum of pixel weights to 1. In our work, we model the closeness and similarity functions as Gaussians:

$$c(\varepsilon, x) = e^{-\frac{\|\varepsilon - x\|^2}{2 \cdot \sigma_d^2}} \qquad s(\varepsilon, x) = e^{-\frac{(f(\varepsilon) - f(x))^2}{2 \cdot \sigma_r^2}} \tag{4.2}$$

where σ_r and σ_d control the amount of smoothing.

The implementation of GPU-accelerated bilateral filtering is as follows. The rendering target is a texture of the size of the reconstructed image, with image texture and other parameters (size of image, σ_r, σ_d, etc.) passed into the GPU. We avoid the expensive evaluation of the exponential function by pre-computing both closeness and similarity functions and storing them into two 1-D lookup textures. We implemented bilateral filtering both in 2D and 3D.

4.3.2 Total Variation Minimization (TVM)

We also implemented a TVM algorithm [10] to compare it with our bilateral filter framework. The TVM solution is obtained by minimizing the following energy functional:

$$\min_u \left\{ \frac{\|u(x) - f(x)\|^2}{2\lambda} + \sum\nolimits_{x\in\Omega} |\nabla u(x)| \right\} \qquad (4.3)$$

where *Ω* is the image domain, *x* is the spatial variable, *f* is the input image, *u* is the sought-after solution and *λ* is a parameter controlling the level of smoothing. The TV of *u* is:

$$\sum\nolimits_\Omega |\nabla u| = \sum\nolimits_\Omega \sqrt{(\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2} \qquad (4.4)$$

In this equation, *x* and *y* are the horizontal and vertical coordinates, respectively. The minimization is transformed to its dual formulation, and a semi-implicit gradient descent algorithm is used to compute the nonlinear projection off. The solution *u* is then obtained after convergence, with *τ* set to some value constraint:

$$u = f - \lambda \operatorname{div} p^n \qquad p^{n+1} = \frac{p^n + \tau\nabla(\operatorname{div} p^n - f/\lambda)}{1 + \tau|\nabla(\operatorname{div} p^n - f/\lambda)|} \qquad (4.5)$$

Here, div is the divergence. In practice, when $\tau \le 1/4$ the algorithm converges.

4.3.3 Regularized OS-SIRT

In our new regularized OS-SIRT, bilateral filtering is applied after each iteration (after backprojecting all subsets). This removes artifacts at the very beginning when the errors are just generated and thus steers the reconstruction towards more plausible and favorable solution regions. Since the target texture (to be filtered) is already in GPU memory, this operation does not require any expensive texture upload/download operations between the CPU and GPU.

4.4 Results

Our experiments were conducted on an NVIDIA GTX 280 GPU, programmed with GLSL and an Intel Core 2 Quad CPU @ 2.66GHz and 2.67GHz. We group our results into two sections: (1) the OS-SIRT results showing the relationship between noise levels and parameters settings, and (2) the performance of our GPU-accelerated bilateral filter using both Cg and CUDA and the reconstruction results using bilateral filter and total variation minimization.

4.4.1 OS-SIRT with Noisy Data

We used the 2D Baby Head test image (size 256^2) to evaluate the performance of the different reconstruction schemes. We obtained 180 projections at uniform angular spacing of [-90˚, +90˚] in a parallel projection viewing geometry. We then added different levels of Gaussian noise to the projection data to obtain SNRs of 15, 10, 5, and 1. Figure 4.1 presents the best reconstruction results (using the correlation coefficient CC between original and reconstructed image), for each SNR, at the smallest wall-clock time.

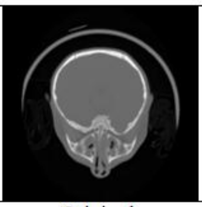
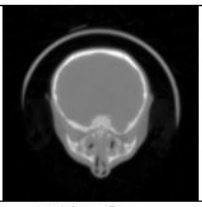
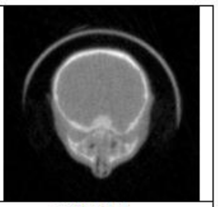
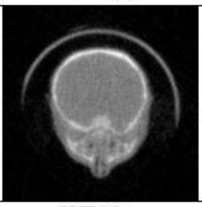
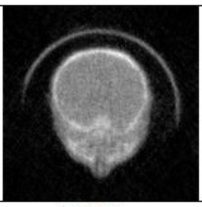
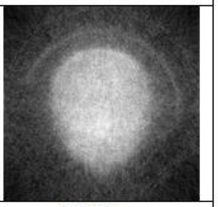
		
Original	Noise-free	SNR15
CC	0.99	0.98
Time (s)	0.045	0.035
		
SNR10	SNR5	SNR1
0.98	0.96	0.82
0.035	0.027	0.012

Figure 4.1: Reconstructions obtained with different SNR levels for the Baby head test image.

The optimal parameter settings we learned through exhaustive benchmark tests greatly depend on the particular imaging situation at hand, such as SNR, total number of projections and their angular range, the imaged object, scanner, etc. Figure 4.2 presents results on the influence of SNR. The plot gives quantifying hints on how to pick the best-performing number of subsets and the associated λ (to obtain the best possible quality within the smallest time), for each expected SNR level. For example, we observe that low SNR requires a low number of subsets to gain more smoothing in the reconstruction process. As for the relaxation factor λ, it is related to both subset number and noise level. For each noise level, the curve of λ is approximately piece-wise linear with a turning point at some subset number. For example, the λ values for SNR 10 are 1 from subset number of 1 to 60, then decreasing until hitting the lowest value of 0.4 at subset number of 180. This is a strong departure from the linear model used on [118] – a higher λ will lead to faster convergence and confirmed by our exhaustive benchmark tests we know it also leads to more accurate results.

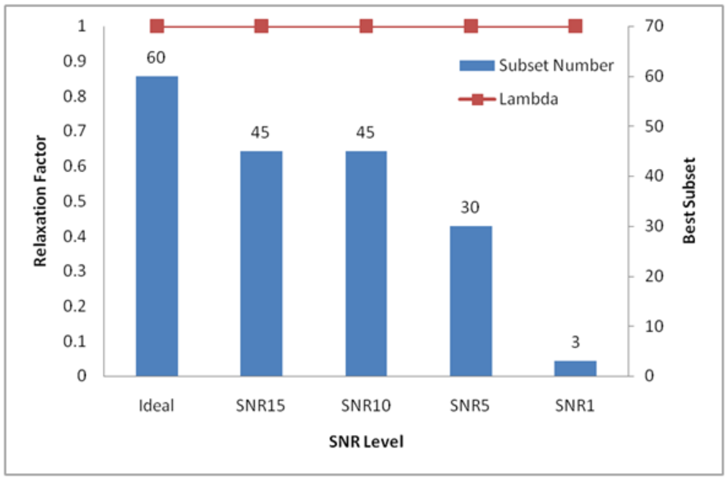


Figure 4.2: Best performaing (both in terms of time and image quality) subset number and relaxation factor as a function of imaging condition, here SNR.

Test size	Window size	CPU time (s)	GPU (Cg) time (s)	GPU (CUDA) time (s)
256x256	11x11	0.622	0.007	0.005
	31x31	4.891	0.013	0.011
	61x61	18.626	0.037	0.033
	91x91	39.031	0.069	0.066
512x512	11x11	2.652	0.011	0.007
	31x31	19.998	0.038	0.032
	61x61	74.319	0.119	0.112
	91x91	164.065	0.253	0.241
1024x1024	11x11	10.811	0.033	0.017
	31x31	84.618	0.133	0.098
	61x61	> 300	0.452	0.368
	91x91	> 300	0.983	0.823
256x256x256	3x3x3	46.831	0.492	N/A
	7x7x7	592.969	1.535	N/A
	11x11x11	> 600	4.823	N/A

Table 4.1: Wall clock time (in s) of GPU vs. CPU bilateral filter.

4.4.2 Bilateral Filter Regularized OS-SIRT

We tested the speed of both 2D and 3D bilateral filters with different sizes of images and windows on both CPU and GPU (using Cg). Table 4.1 shows that speedups of more than two orders of magnitude can be obtained by using the GPU. For 2D images, we also implemented a CUDA version of our scheme.

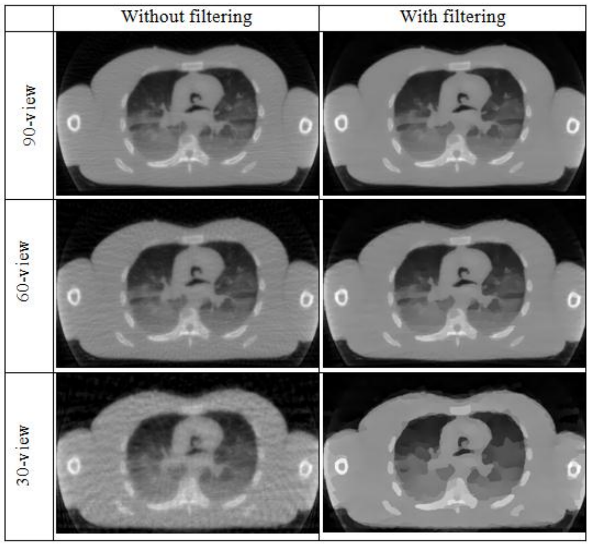


Figure 4.3: Comparison of bilateral filter for the noise-free, few-view case.

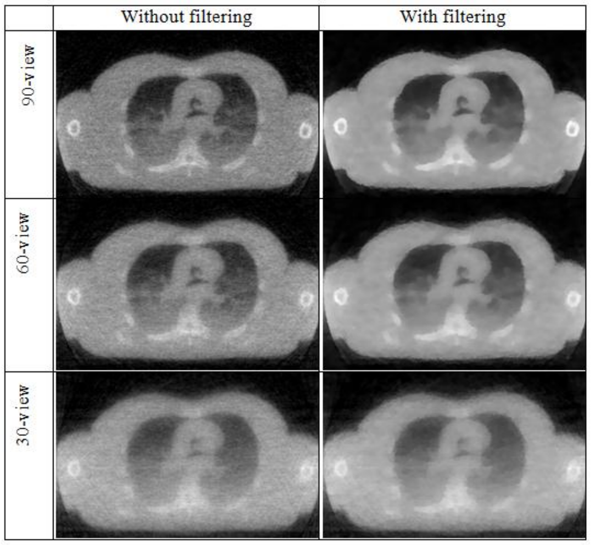


Figure 4.4: Comparison of bilateral filter for the noise (SNR=10), few-view case.

To gauge the performance of the regularized reconstruction for both the few-view and the noise (SNR=10) scenario, we used the NIH Visible Human dataset at 512^3 resolution. We ran SART with 8 iterations for the noise-free few-view case. The filter window size was fixed to 11. Figure 4.3 shows one slice of the reconstructed volume with and without filtering, respectively, for reconstructions from 90, 60, and 30 views. We notice that SART is already well suited for the few-view reconstruction. For the regularized reconstructions we tested a number of combinations of representative σ_r and σ_d and selected the best results. In particular the 30-view reconstruction shows prominent streaking artifacts, which can be avoided by intermediate bilateral filter regularization.

Figure 4.4 shows the results for the noisy few-view case, after 5 iterations. Like in the noise-less case we observe that the salient features are well preserved in both size and shape. Finally, Table 4.2 lists the GPU-accelerated reconstruction time required for one SART iteration, for the Visible Human dataset at 512^3 resolution for both 180 and 30 projections. The 1-ch time uses only the R-channel of the GPU hardware, while the 4-ch time uses all 4 (RGBA) channels in parallel. Using 4 channels yields a 2.5-fold speedup, while regularization with bilateral filtering (BF) adds only a moderate time overhead. Note here we choose SART which is the slowest algorithm to show the lower bound of the time performance of the OS-SIRT family.

#proj	1-ch	1-ch w/ BF	4-ch	4-ch w/ BF
180	91.81	94.96	34.79	34.94
30	21.94	25.69	9.21	10.12

Table 4.2: Time for one GPU-acceleration SART iteration (512^3 volume). The 1-ch and 4-ch accelerate the reconstruction with 1 (R) or 4 (RGBA) color channels, respectively. A NVIDIA GTX 280 GPU was used.

4.4.3 Bilateral Filter vs. Total Variation Minimization

We tested the same slice with identical settings for both the few-view (30 projections) and the noisy few-view (30 projections and SNR=10) case and show the results in Figure 4.5.

We observe that for the noise-free case, bilateral filtering achieves similar results than TVM (maybe even slightly better). However, TVM works better for the noisy case. This is not surprising since for TVM the energy functional imposes a constraint over the image, while bilateral filtering just averages the neighboring values which cannot eliminate all noise for higher noise levels. Nevertheless, both successfully preserve salient features and remove noise and streaking artifacts.

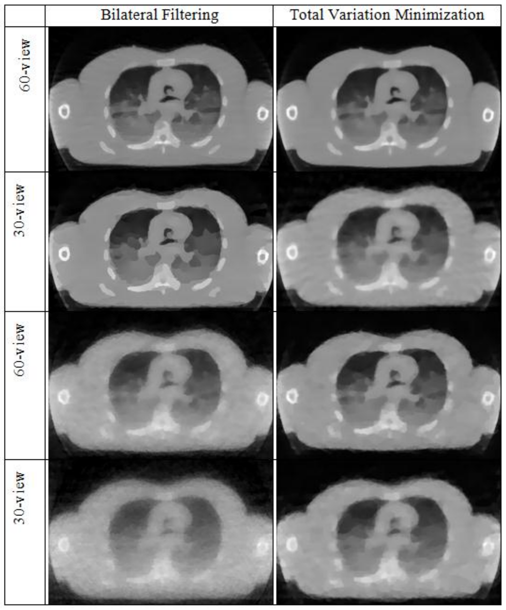


Figure 4.5: Bilateral filtering vs. TVM: (first two rows): the noise-free few-view case; (last two rows): the noisy (SNR=10) few-view case.

From the perspective of high performance computing, the bilateral filter is a better choice. Table 4.1 shows that the computation time is less than 1s. Although a GPU-accelerated version of TVM exists [86], once the parameter λ grows larger, which is needed for very noisy data, the computation time (usually >> 1s) is still far greater than with the bilateral filter.

4.5 Conclusions

We have demonstrated that careful parameter-tuning taking into account reconstruction quality results in better speed performance. This is particularly true for ordered subsets approaches in the presence of adverse data scenarios, such as noise and sparse views. We also demonstrated that bilateral filtering represents a viable option for regularization compared with Total Variation Minimization (TVM), with the added advantage that it accelerates very well on GPUs.

Chapter 5

Nonlinear Neighborhood Filters for Use in Iterative CT

As we introduced in last chapter, one way to cope with the associated streak and noise artifacts for low-dose CT is to interleave a regularization objective with the iterative reconstruction framework. This time we want to be more aggressive toward the bank of the reconstruction quality. In this chapter, we investigate a number of non-linear neighborhood filters popular in the image processing literature for their suitability in iterative CT application. We first introduce and review edge-preserving denoising with both iterative regularization methods and nonlinear neighborhood filtering methods including some relative features as well as their application to iterative CT reconstruction algorithms with regularization. Then we specify the compared filters with their usage in our specific GPU accelerated reconstruction algorithm. The comparison results of all filters for both quality and time are presented under different data configurations.

5.1 Introduction

5.1.1 Iterative Regularization Methods for Denoising

For image denoising, most algorithms are based on this image model:

$$f(x,y) = u(x,y) + n(x,y) \tag{5.1}$$

where *u* is the true but unknown image function, *f* is the observed image, *n* is the unknown additive noise function which is usually assumed to have zero-mean, independent and random values and *x* and *y* are spatial coordinates. To recover *u* from the observation *f*, regularization methods are mostly used. These methods define a specially designed energy functional composed of a fidelity term *F(u, f)* evaluating the distance between *u* and *f* functions and a regularization term *R(u)* encoding the prior information or constraints into the solution:

$$\hat{u} = \arg\min_{u} \left\{ F(u, f) + R(u) \right\} \tag{5.2}$$

By applying energy functional minimization which usually includes a gradient descent process, we can iteratively obtain the solution $\hat{u}$.

Related partial differential equations (PDEs) based methods such as anisotropic diffusion [69], variational methods such as total variation minimization [10][92][88], weighted least squares [64], robust estimation technique [136] are well-known examples of this iterative regularization group. They were shown to emerge from a solid theory of statistical estimators and regularization theory [31] to demonstrate their validity and reliability to do the edge-preserving smoothing.

5.1.2 Neighborhood Filtering Methods for Denoising

To achieve better computational performance and local manipulation, the neighborhood filters are proposed in the domain of computer vision and image processing for a variety of applications including image denoising, visual appearance-preserving contrast reduction, tone mapping and multi-scale image decomposition [85]. These methods are non-iterative and based on a pixel-wise operation over a small neighborhood which eventually split the input image to a denoised piecewise smooth base layer with preserved strong edges as estimation of *u* and a detailed layer with noise or unwanted details as estimation of *n*. For instance, bilateral filter is a spatial-range domain joint filter falling into this category which appeals more attention recently due to its simplicity, effectivity and efficiency [104].

Though with the non-iterative nature, a brute force implementation of bilateral filter still takes a few minutes for megapixel images in CPU. Therefore, many papers have been dedicated to handle the speed issue by exploiting different levels of approximation. Pham and Vliet [89] proposed separable bilateral filter approximating 2D bilateral filter with two 1D bilateral filters which is significantly faster but introduces new axis-aligned streaks. Weiss [109] replaced the spatial weight with a square box and used local histogram to compute range weight due to the highly neighborhood overlapping of adjacent pixels. But three iterations are enforced to remove band artifacts close to edges. Paris and Durand [84] transform bilateral filter to a convolution operation by introducing homogeneous intensity in a higher dimensional space. A down sampling is performed to speed up before convolution followed by up sampling. The running time was reduced to seconds for megapixel images.

Beside speed issue, improving the denoising quality of bilateral filter also has a potential. To meliorate the piecewise constant effect, trilateral filter [22] was introduced so that the filter window is tilted according to the local slope and the window size is self-adaptive. Extra data structure has to be created to pre-compute the filtering parameters. A linear regression correction [5] was devised to reduce the commonly appeared staircase effect in neighborhood filters. By using a pair of images, cross or joint bilateral filter [33] was introduced to smooth first image while preserving the edges of the second image. These algorithms improve the quality but inevitably increase the computation complexity.

To compare the similarity of two pixels purely by the difference of their pixel intensities is not stable when the image has a low signal-to-noise ratio. Another line of extensional work belongs to the patch based technique traditionally used widely for texture synthesis [34][59] and image inpainting [19]. It is based on the assumption that there is a high degree of redundancy of any natural image. So any small window (the patch) has many similar windows close by in the same image [4]. Non-local means (NLM) [4] was introduced with the idea of patch comparison incorporated with the neighborhood filter. The similarity of two pixels is determined by how much their patches (here 7×7) match. From the perspective of high dimensional space, the patch is actually a feature vector with 49 dimensions. Stemming from the assumption that patch vectors exist on a lower-dimensional manifold rather than the full space, Tasdizen performed the patch comparison in a lower-dimensional subspace projected from the patch vector space by principal component analysis to bring more accuracy and computational performance [105].

5.1.3 Multi-scale, Adaptivity and GPU Acceleration

For all the methods mentioned in last two subsections, there are several other aspects which are commonly used to improve the filtering performance on speed and quality: the multi-scale scheme, adaptivity and GPU acceleration.

Multi-scale decomposition is mostly for feature extraction and detail enhancement while very few for denoising. Filters could perform as a better decomposer [38][58] to enhance edges or work for denoising after decomposition through wavelet transform [23], creating Laplacian pyramid [73] or a sequence of dyadic scales by varying regularization parameters [17]. However, these methods are either sensitive to noise for the finest scale or have little improvement to catch up the time cost.

The level of smoothness is dependent on the regularization parameters for all these methods. A single scale for the whole image is seldom optimal. Therefore, local adaptivity is helpful to boost the regularizing performance. There are many related papers with adaptivity for anisotropic diffusion [74][44] and total variation minimization [93], bilateral filter with local statistics [137] and with local phase characteristics [110] and non-local means with three window sizes [108] and with optimal weight and window size [54].

As for the speed issue, by mapping the computation to the modern GPU architecture is an effective approach to acceleration. Iterative regularization like total variation minimization was effectively accelerated [88] but still needed more than one second for megapixel smoothing with large number of iterations. Chen et al. [18] improved the performance of the bilateral grid by parallelizing the algorithm on modern GPUs. The running time for megapixel image is reduced to only a few milliseconds. Actually due to the pixel-wise operating nature i.e. the computation of every pixel could theoretically be processed in parallel, all neighborhood filters could be extremely matched for GPU acceleration. We will demonstrate this improvement in our work by implementing the employed filters in CUDA.

5.1.4 Iterative Reconstruction Algorithms with Regularization

For CT reconstruction, iterative methods are preferable when the recovery problem becomes ill-posed, for example, when the data are noisy, few-view or limited in angle. Under this circumstance, in order to further remove the noise or streak artifacts appeared in the reconstruction using incomplete or imperfect projections, additional regularization is usually incorporated between iterations. Traditionally the iterative regularization methods are widely applied. Sidky and Pan [97] developed a POCS algorithm in conjunction with minimizing the total variation of reconstructed image in each iteration for divergent-beam CT. Later, Chen et al [21] concluded this as a utilization of compressed sensing image reconstruction method. The sparsifying transform is the L_1 norm of the discrete gradient image i.e. the total variation of the reconstructed image.

Inspired by the success of nonlinear neighborhood filters for noise reduction in image processing scenario, we apply these filters here instead as the regularization mechanisms in our iterative CT reconstruction. Compared to iterative regularization, neighborhood filters seem to lack a well-established theory behind. Fortunately, Elad proved that the bilateral filter also emerges from the Bayesian approach as a single iteration of some well-known iterative algorithm [31]. This proof theoretically connected the neighborhood filters to the classical approaches.

The whole CT reconstruction is GPU-accelerated, which is a nice platform to accommodate any GPU-accelerated filtering.

5.2 Methodology

For nonlinear neighborhood filters, the updated pixel value at pixel x is determined by the weighted sum of the pixel values *f(x+t)* at pixel *(x+t)* inside its surrounding window area W_x with x as spatial variable and t as spatial offset:

$$NNF(x, f, W_x, \alpha) = \frac{\sum_{t \in W_x} \alpha(x, t, f(x), f(x+t)) f(x+t)}{\sum_{t \in W_x} \alpha(x, t, f(x), f(x+t))} \tag{5.3}$$

where α is the weight coefficient which is usually a nonlinear composite function. The normalization forces the sum of pixel weights to 1. The window area called the neighborhood could be various sizes for pixels at different positions. We use a square shape in this work. But a circle is another option for rotation invariance. To compute the weights of the neighborhood, a distance metric is used to measure the similarity between the current pixel and the central pixel.

Focused on the quality aspect, we compared three nonlinear neighborhood filters bilateral filter (as introduced before), trilateral filter and non-local means together with a selected total variation minimization method in [122]. This first tryout revealed the superiority of non-local means on the reconstruction quality. As a complete study, non-local means with optimal spatial adaptation [53] is added as another method in this section. Both the reconstruction quality and speed are compared. Two incorporated schemes of these filters are implemented to the reconstruction process: interleaved- and end- regularization. In the following sections, we show each filter in the form of their different α functions.

5.2.1 Bilateral Filter

The bilateral filter (BLF) [104] as described in last chapter averages similar and nearby pixels values inside a fixed size neighborhood. Its weight coefficient is the multiplication of a spatial distance weight c_d and a range distance weight s_r:

$$\begin{gathered} \alpha_{BF}(x, t, f) = c_d(t) s_r(f(x), f(x+t)) \\ c_d(t) = \exp(-\|t\|^2 / (2d^2)) \\ s_r(f(x), f(x+t)) = \exp(-(f(x) - f(x+t))^2 / (2r^2)) \end{gathered} \tag{5.4}$$

where d and r control the amount of smoothing. On the one hand, the function c_d performs as a domain filter to ensure the averaging with the spatial closeness so that the far away pixels have no effects. On the other hand, the function s_r

performs as a range filter to ensure averaging with the range similarity so that the sharp edges could be well preserved.

5.2.2 Trilateral Filter

In order to accommodate the preservation of gradients, which the BLF does not support, the trilateral filter (TLF) was introduced [22]. Essentially, the TLF is a tilted BLF according to the smoothed gradient which performs inside an automatically generated adaptive neighborhood. The introduction of the local gradients makes TLF a piecewise linear approximation while traditional BLF is only a piecewise constant approximation. Moreover, the adaptive neighborhood excluding the outliers enables a sharp bound.

The implementation of TLF includes three parts: generate the tilting vector (the smoothed local gradient), approximate the adaptive neighborhood and then apply the filter. Firstly, the discrete gradient vector for each pixel is computed by forward differencing. The tilting vector $G_\theta(x)$ is acquired by applying the BLF to the gradient vector image with fixed neighborhood size:

$$G_\theta(x) = NNF(x, \nabla f, W, \alpha_{BF}(x, t, \nabla f)) \tag{5.5}$$

Then the tilted plane $P(x,t)$ through $f(x)$ and the new range differences $f_\Delta(x,x+t)$ from the original value to the corresponding position on tilted plane are generated:

$$\begin{aligned} P(x,t) &= f(x) + G_\theta(x)\cdot t \\ f_\Delta(x, x+t) &= f(x+t) - P(x,t) \end{aligned} \tag{5.6}$$

Finally, another BLF to smooth the range differences is applied:

$$TLF(x) = f(x) + NNF(x, f_\Delta, f_\theta, \alpha_{BF}(x, t, f_\Delta)) \tag{5.7}$$

where f_θ is the adaptive neighborhood inside which the gradient magnitudes are similar. The TLF itself includes 7 internal parameters which could mostly be generated by following an automatically running routine when given one parameter. We implemented all TLF stages including parameter computation on the GPU by applying several passes.

5.2.3 Non-local Means

In a word, NLM replaces the pixel with a mean of the pixels inside the fixed neighborhood W whose patch look similar to the patch of the central pixel:

$$NLM(x) = NNF(x, f, W, \alpha_{NLM}) \tag{5.8}$$

To measure the patch matching, the Gaussian weighted difference of pixel values at corresponding positions of two patches is accumulated. The Gaussian kernel G_a is of zero-mean and standard deviation a. Therefore the weight coefficient is as below:

$$\alpha_{NLM}(x,t,f) = \exp(-(\textstyle\sum_{p\in P} G_a(t)\left|f(x+p) - f(x+t+p)\right|^2)/h^2) \qquad (5.9)$$

where P is the fixed-size patch and h acts as a filtering parameter when increased, the weights to dissimilar pixels are increased to allow for more smoothing. In contrast to BLF, NLM removes the spatial smoothing form but increases the dimension of the range filter. The comparison of the similarity of two patches is actually to compute the distance of two vectors in high dimensional space with the Gaussian kernel G_a to average the contribution from each dimension. This modification brings more accuracy to the smoothing but also costs more time.

5.2.4 Adaptive Non-local Means

As an extension to NLM, we employ a novel approach with optimal spatial adaptation [53]. We call it adaptive NLM (ANLM) in the following parts. ANLM is an extension to NLM by introducing spatial adaptivity which includes two aspects: first, the size of the neighborhood is variable; second, the weight coefficient inside each neighborhood is adaptive. To implement the adaptivity, the method runs for a few iterations. In each iteration n, the mean $u_n(x)$ and standard deviation $v_n(x)$ at position x of a neighborhood are calculated with current weight coefficients α_n:

$$\begin{aligned} u_n(x) &= \textstyle\sum_{t\in W_{x,n}} \alpha_n(x,t,u_{n-1}) f(x+t) \\ v_n^2(x) &= \sigma^2 \textstyle\sum_{t\in W_{x,n}} \alpha_n(x,t,u_{n-1})^2 \end{aligned} \qquad (5.10)$$

where $W_{x,n}$ is the current neighborhood size and σ is the robustly estimated initial standard deviation from input image. They control the growth of the neighborhood size and as input to compute the weights in the next iteration:

$$\begin{aligned} \alpha_n(x,t,u_n) &= \frac{\exp(-dist(u_{n-1}(x), u_{n-1}(x+t))/h^2)}{\sum_{t\in W_{x,n}} \exp(-dist(u_{n-1}(x), u_{n-1}(x+t))/h^2)} \\ dist &= \textstyle\sum_{p\in P} (u_{n-1}(x+p) - u_{n-1}(x+t+p))^2 \left(\frac{1}{2v_{n-1}^2(x+p)} + \frac{1}{2v_{n-1}^2(x+t+p)}\right)) \end{aligned} \qquad (5.11)$$

As in (5.11), the computation of the weights is similar to NLM except using a modified distance function normalized by the current variances. Although this method includes several iterations, as in (5.10) it performs only with the initial image in the averaging procedure as a local M-smoother [53]. The adaptivity is optimal since the method enables to approximately minimize the point-wise L_2 risk of the estimation.

There are five parameters included in this method: the initial noise variance σ^2, the patch size P, two control parameters h and ρ and the iteration number N_Δ. Among them, σ^2 could be automatically generated through robust estimation; P was suggested an appropriate value for most cases after experiments; and the rests were shown to have nearly unchanged effects by varying in a suitable range. So the authors concluded their method as nearly parameter-free.

5.2.5 Total-Variation Minimization

We selected the same TVM implementation algorithm [10] as in last chapter (and rewrite here for clarity) to compare it with other filters. The solution *u* is the argument that minimizes the following energy functional:

$$\min_u \{\|u(x) - f(x)\|^2 / (2\lambda) + \textstyle\sum_{x\in\Omega} |\nabla u(x)|\} \tag{5.12}$$

where *Ω* is the image domain, *x* is the spatial variable, *f* is the input image and
is a smoothing parameter. The Total Variation (TV) of *u* is:

$$\textstyle\sum_\Omega |\nabla u| = \sum_\Omega \sqrt{(\partial u/\partial x)^2 + (\partial u/\partial y)^2} \tag{5.13}$$

where *x* and *y* are the horizontal and vertical coordinates, respectively. The minimization is transformed according to its dual formulation, and a semi-implicit gradient descent algorithm is used to compute the nonlinear projection of *f*. The solution *u* is then obtained until convergence:

$$\begin{gathered} u = f - \lambda \operatorname{div} p^n \\ p^{n+1} = \frac{p^n + \tau\nabla(\operatorname{div} p^n - f/\lambda)}{1 + \tau\nabla(\operatorname{div} p^n - f/\lambda)} \end{gathered} \tag{5.14}$$

Here, *div* is the divergence. In practice, when $\tau \le 1/4$ the algorithm converges.

5.2.6 Parameters Setting and Adjustment

All the neighborhood filters mentioned above are based on the exponential function which when with a bigger denominator parameter contributes a flatter and wider weighting function increasing the overall smoothness in the corresponding range or spatial domain. In Table 5.1, we list all the parameters of every filter. The regularization parameters directly control the level of smoothness while the size related parameters and others are internal parameters.

We fix the size related parameters *W* and *P* to 11×11 and 7×7 respectively for every filter which are well performed for the test data for a fair comparison. The choice of regularization parameter is usually data dependent and variable to the noise level and type of the input image. Therefore, we employ the same exhaustive learning strategy as introduced in [119] for all filters. Although for ANLM, even the regularization parameters were shown in the paper to be able to estimate from data or pre-define, practically it is intensively sensitive to the noise level so that the estimated value is not suitable which still demands a learning process. The choice of other parameters follows the similar way that is mostly from the suggested setting or from the automatic estimation process in the corresponding papers but needs adjustment sometimes.

Filter	**Regularization parameters**	**Size related parameters**	**Other parameters**
BLF	d, r	W	N/A
TLF	$\sigma_{c\theta}$	W	$\sigma_{s\theta}, \sigma_c, \sigma_s, \beta, R, f_\theta$
NLM	h	W, P	a
ANLM	σ, h	P	ρ, N_Δ
TVM	λ	N/A	τ

Table 5.1: Parameter table for all filters: the regularization parameters directly determine the level of smoothness, size related ones are fixed after experiments and others are internal parameters.

The reason for the adjustment is rooted from the inconsistency to the assumption that the noise *n(x, y)* as in (5.1) is independent, randomly distributed noise with zero-mean and an unknown variance. When applying the method to the data reconstructed from few views or noisy projections, this assumption mostly does not confirm. To illustrate, the first column in Figure 5.1 shows a uniform object with Gaussian noise added and below its frequency transform (we call it the reference case). The next two columns show reconstructions (spatial and frequency domain) of the same uniform object from (i) 30 noise-free projections and (ii) 180 noisy projections (the same Gaussian noise as in the reference case was used). As one would expect, for the few-view reconstruction both image and spectrum are quite different from the reference case. Further, while the differences with the reconstruction from noisy data are a bit more subtle, one can still clearly observe that the noise is grainier and has a higher amplitude range than the reference case. The frequency spectrum reflects this finding, showing a low-passing of the spectrum, in fact, even a boosting of the lower frequencies. Therefore, we uniformly scale up the estimated parameters such as σ and h for ANLM and $\sigma_{c\theta}$ and $\sigma_{s\theta}$ for TLF to allow more smoothness. This operation successfully improves the reconstruction quality.

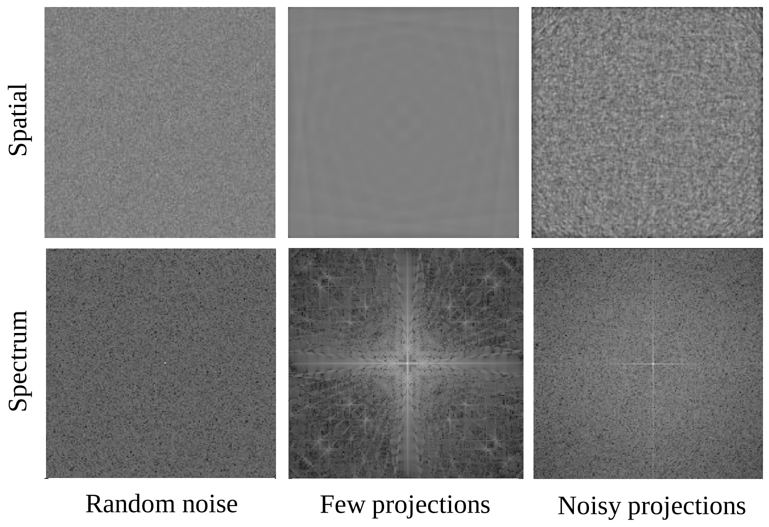


Figure 5.1: Comparing true random noise (left) with the streak and noise artifacts resulting from reconstructing a uniform square object from limited or noisy projections (right). We observe significantly different appearances in the spatial domain and strong biases in the frequency domain.

5.2.7 Regularized OS-SIRT in GPU

Just like the CT reconstruction all regularization computations are also fully GPU-accelerated, and so do not require any expensive texture upload/download operations between the CPU and GPU. The technical details of the GPU-accelerated OS-SIRT were introduced in [125]. Here we focus on the performance of the filters in GPU.

5.2.7.1 Using Lookup Textures and Shared Memory

We found that better results can be obtained when the filters are applied in 2D within volume slices only (as opposed to 3D operations). The regularized images appear sharper since the 3D operations tend to over-blur the results. While this could be prevented by choosing different settings for the various parameters, we still did not observe any benefits in our experiments. Therefore we chose to perform all filtering operations in 2D which also secured higher computational performance. For all filters, if the range of the input to the expensive exponential function is known beforehand, we can pre-compute the possible values and store them into a 1D lookup texture. In some specific application when the window size is small and fixed, it could be further accelerated by using loop unrolling to expand the loops for convolutions. With a CUDA implementation, to access the on-chip shared memory is 200 times faster than to access off-chip global memory which ends up as the bottle neck of GPU computing. For the neighborhood filters, two adjacent pixels share most of their neighborhoods which makes the data re-use occur very often. Therefore, by pre-fetching the data into the shared memory together with some manual cache control, no cache-miss optimization resulting in better performance can be guaranteed. We call this optimized implementation (OPT) while the one without using shared memory non-optimized implementation (NOPT). We will show the performance difference in Section 5.3.

5.2.7.2 Interleaved OS-SIRT vs. End OS-SIRT

There are two modes to perform regularization: interleaved mode and end mode. Interleaved mode is the same used in [97][120] whose filtering is applied for each iteration as a final step after backprojecting all subsets. The object will be filtered for the number of times as many as the number of iterations. On the contrary, end mode regularizes the object only once after last iteration. The former mode removes artifacts at their early onset when the errors are just generated and thus steers the reconstruction towards more plausible and favorable solution regions, while the end mode consumes less time. We compared the results of both modes.

5.3 Results

The NVIDIA GTX 480 GPU was interfaced with an Intel Core 2 Quad CPU @ 2.66GHz host processor. For testing, we employed the NIH Visible Human's torso (size 256^3) which has a prominent spine structure with different bone sizes and small structures, a brain dataset (NIH Visible Human brain, size 256^3) which also has some finer structures, and the Catphan phantom (size 512^2) to enable a comparison of the detail resolution with a gold standard. We used a high-quality X-ray simulator to obtain various projection sets for the torso, the brain and the Catphan phantom. In section 5.3.1, we first test the power of various filters as a denoising operator and their sensitivity to the settings of the parameters. We compare the interleaved mode and the end mode for incorporating the regularization filter in section 5.3.2. Then from 5.3.3 to 5.3.5, the performance of filters as the regularization to the reconstructions is compared with interleaved mode for three different datasets. Finally, the time performance of the filters is shown in section 5.3.6.

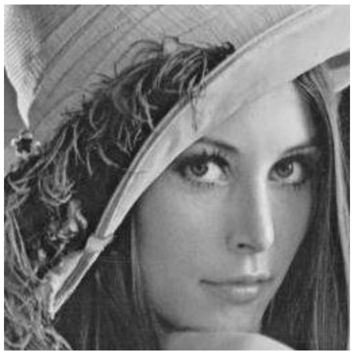
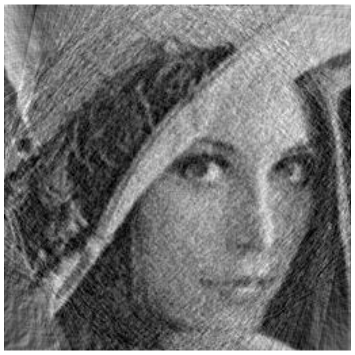
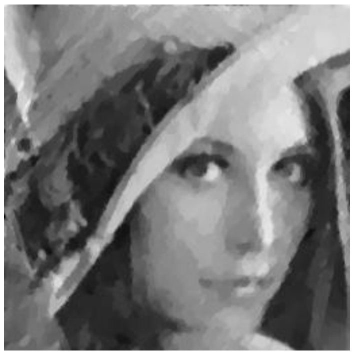
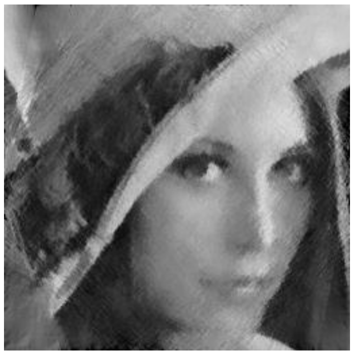
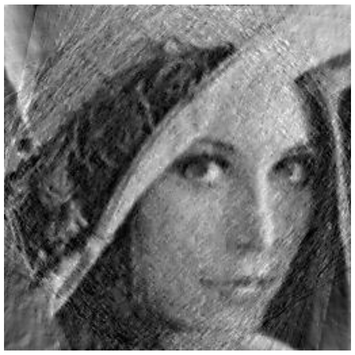
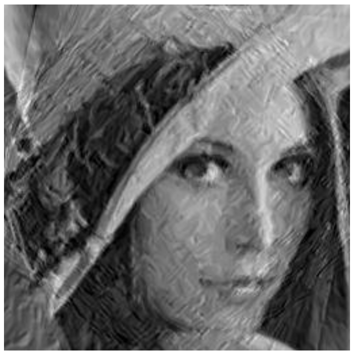
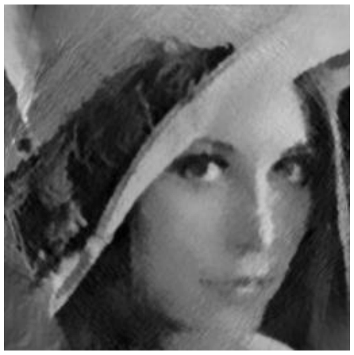
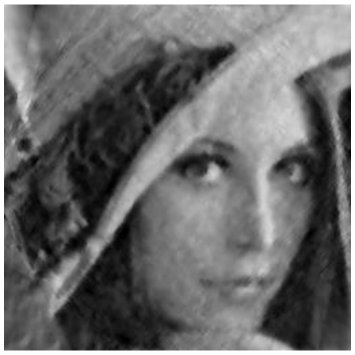
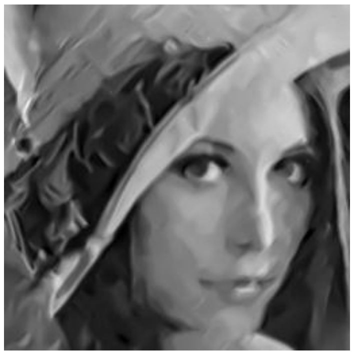
		
Original	w/o filtering	TVM
Parameters	N/A	λ=15
E-CC	0.48	0.70
		
BLF	TLF (default)	ANLM (default)
σ_d=5, σ_r=41	$\sigma_{d\theta}$=σ_d=2,$\sigma_{r\theta}$=σ_r=R=25.4	σ_0=8.73, h=113.5, ρ=3, N_Δ=4
0.68	0.53	0.62
		
NLM	TLF (optimized)	ANLM (optimized)
h=33	$\sigma_{d\theta}$=$\sigma_{d\theta}$=inf,σ_d=2,σ_r=57.5, R=20	2.75σ_0, h=113.5, ρ=3, N_Δ=4
0.73	0.69	0.75

Figure 5.2: Regularization quality obtained with the 5 presented regularization filters (BLF, TVM, TLF, NLM, ANLM) assessed by a perceptual quality metric, E-CC and using the Lena image (Original) with added Gaussian noise SNR 10 (w/o filtering).

5.3.1 Pure Filtering Effects

Our first experiment employs the popular Lena image (size 256^2) to explore the various regularization schemes we presented and also their sensitivity to the settings of their parameters. For this, we added to the original image severe noise and streak artifacts similar to those shown in the few and noisy projection columns of Figure 5.1. The regularization parameters were set according to the highest E-CC metric scores they achieved with our automated parameter learning framework. The results of our study are presented in Figure 5.2. A first observation we make is that both the TLF and ANLM filters can greatly benefit from parameter optimization, and we also observe that the (optimized) ANLM filter preserves salient edges and features better than the NLM filter, which is to be expected. Another interesting result is that both filters are also superior to TVM which generates a somehow blocky result, widening some of the structures. Further, TVM also exhibits more blurring (see, for example, the feathers on the hat). On the other hand, the TLF also reduces the noise but still contains a similar noise distribution. The BLF and NLM filters both show better noise suppression but they also remove small details. Hence, the ANLM appears to be the only filter able to successfully remove the severe artifacts along the lips while still keeping a smooth appearance and preserving small details as well. Finally, judging the filters by their E-CC scores, we get the following ranking: ANLM (optimized) > NLM > TVM > TLF (optimized) > BLF > ANLM (default) > TLF (default).

Due to the somewhat mediocre performance of the TLF filter in our application scenario, its many parameters requiring extensive (although automated) tuning, and its relatively high computational overhead, we did not consider it further in our more advanced studies with real medical and phantom datasets. On the other hand, although the BLF filter also did not perform overly well (at least when compared to the NLM and ANLM filters), we maintained it due to its great simplicity and low computational effort. In the following we will refer to the optimized ANLM filter simply as ANLM filter.

5.3.2 Interleaved Mode vs. End Mode

We used the NIH Visible Human's Human Head test image (size 256^2) to compare the performance of interleaved mode and end mode. We obtained 20 projections at uniform angular spacing of [-90^o, $+90^o$] in a parallel projection viewing geometry. OS-SIRT with 10 subsets was used as the reconstruction algorithm. We chose NLM filter to do the regularization. In Figure 5.3(a) from top row to bottom listed the reconstruction results without regularization, with end-NLM and with interleaved-NLM respectively. We showed the results changing gradually from left to right with iteration increasing from 1 to 100.

Apparently, interleaved mode generates much better result than end mode. For the following sections, we always chose interleaved mode but tested different filters.

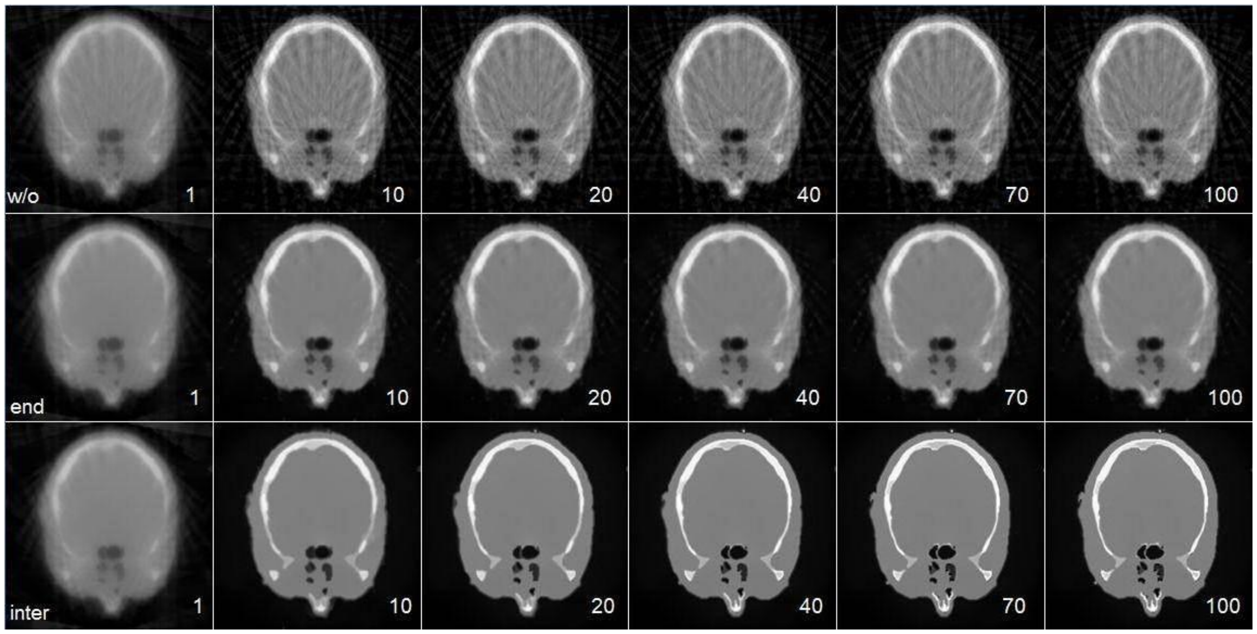
5.3.3 Qualitative and Quantitative Filter Comparison: Torso Dataset

We simulated 180 uniformly distributed projections over a half-circle trajectory. For the few-view case we selected every 9th projection from the set, yielding a total of 20 projections. Then, for each of the 4 regularization schemes (BLF, TVM, NLM and ANLM), we interleaved regularization with OS-SIRT (10 subsets) and ran this pipeline for a total of 200 iterations. For the second series of experiments, we added significant Gaussian noise (SNR=10) to all 180 projections and ran the same pipeline again, but this time for only 20 iterations since this yields about the same number of updates than the few-view case (the introduction of this much noise typically also causes the reconstruction procedure to diverge when the noisy projections are not pre-filtered).

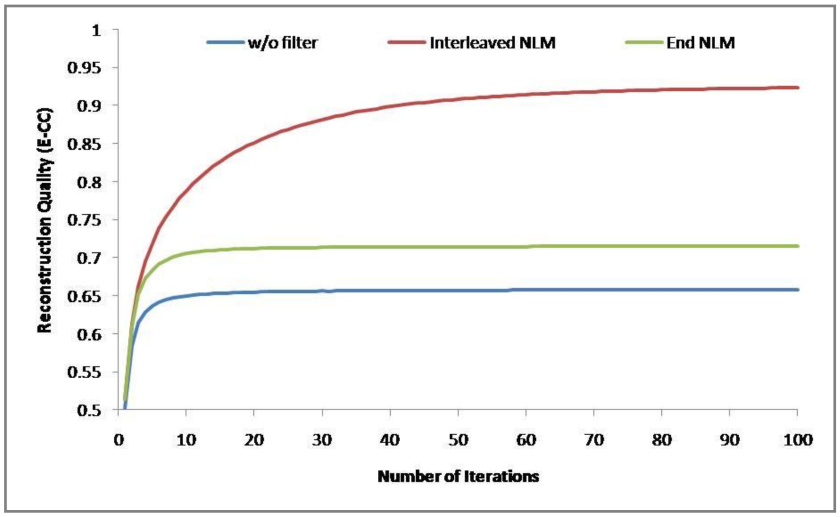
The results of these two experiments are shown in Figure 5.4 along with the corresponding parameter settings and the best E-CC metric scores they could achieve. We provide zoomed results for two critical regions, spine and lung. The left-most full-body reconstructions were obtained without regularization. A first observation we make is that streaks seem to be easier to remove than heavy noise – the E-CC obtained with regularization is roughly 12-15% higher for the former for all regularization schemes. We also readily observe that all filters are able to reduce streaks and noise, recovering some structural parts which can be hardly seen in the non-filtered result. In the following we focus our detailed discussion on the spine – similar observations can also be made for the lung.

The BLF and TVM perform quite similarly, but while the BLF keeps sharper edges and provides better streak removal than TVM, it also gives the image a more binary look signified by abrupt changes along adjacent varying-intensity areas. TVM, on the other hand, has smoother transitions here, and it also seems to perform better with noise. However, neither of the filters is able to recover more subtle features.

The NLM-based methods (NLM and ANLM) successfully master the problems encountered with BLF and TVM. Both NLM and ANLM recover the gaps separating the individual vertebrae. The shape and structure of the vertebrae is also better described, delineating the bony shell around the vertebrae body well. However, for the noisy projections case, the ANLM filter is the only one to do so.



(a)



(b)

Figure 5.3: (a) Regularization scheduling (20 projections, OS-SIRT 10 subsets, NLM filter with h=25.0). The number below each image indicates the iteration step. Row 1: No regularization at all; Row 2: Regularization only post-reconstruction, after the indicated iteration step; Row 3: Regularization interleaved with every iteration step. (b) Reconstruction quality of the three regularization schedules (Figure 5.3(a)), measured by a perceptual quality metric, E-CC.

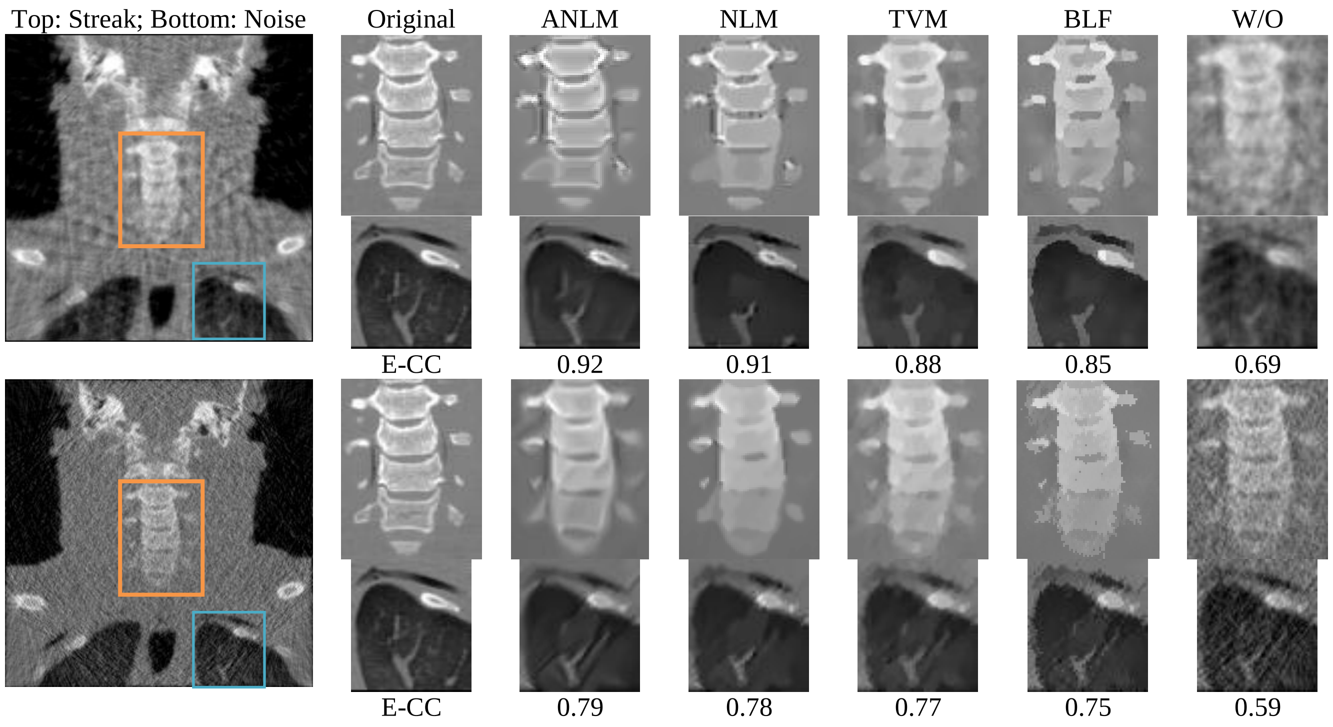


Figure 5.4: Torso dataset, reconstructed with the interleaved regularization pipeline both for the few-view (top, 20 projections) and noisy (bottom, 180 projections, SNR 10) scenarios. Zoomed results for two critical regions are shown, indicated by the orange (spine) and blue (lung) boxes in the left-most reconstructions obtained without regularization. The reconstructions appear ordered according to their E-CC scores.

5.3.4 Qualitative Filter Comparison: Brain Dataset

We used the same conditions than for the torso dataset (20 projections for the few-view case, 180 projections with SNR 10 Gaussian noise added for the noisy case) and the same regularized construction strategy (200 iterations with interleaved OS-SIRT 10 for the view-few case, 20 iterations of OS-SIRT 10 for the noisy projection case).

Figure 5.5 shows the results obtained for the few-view case. We observe that in terms of sharpness and detail preservation ANLM and NLM have similar outcomes, but that the ANLM better preserves the small structures pointed by the arrow in the Original image. We further observe that the BLF produces slightly sharper and detailed images than TVM, but not quite as good as the NLM filter.

Figure 5.6 shows the results for the noise case. Here the differences of NLM and ANLM are not as profound as for the streak case. However, similar qualitative differences can be observed for the BLF and TVM.

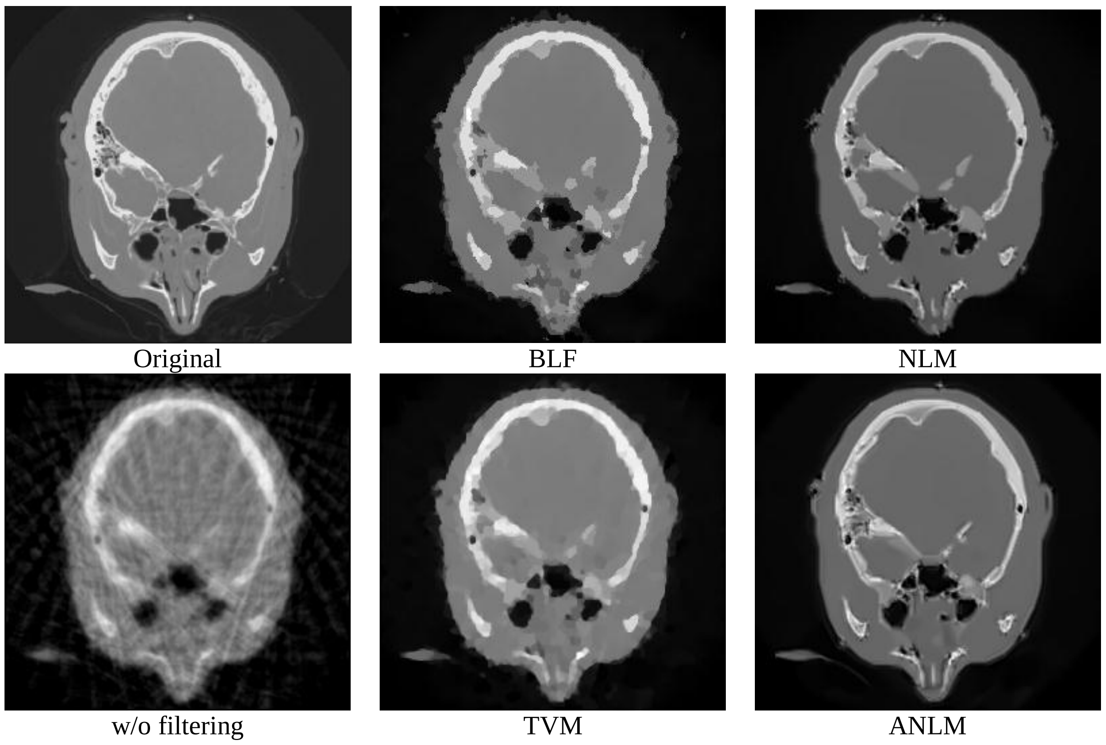


Figure 5.5: Brain dataset, reconstructed with the interleaved regularization pipeline for the few-view scenario (20 uniformly distributed projections, 200 iterations with OS-SIRT 10).

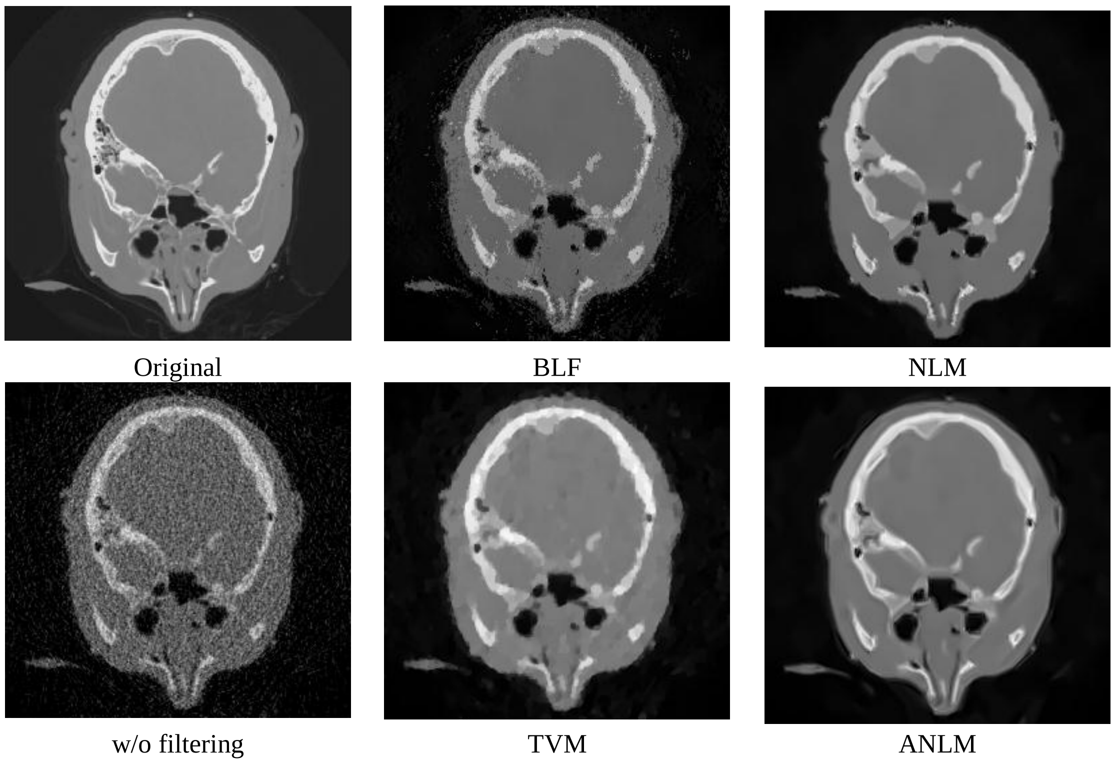


Figure 5.6: Brain dataset, reconstructed with the interleaved regularization pipeline from noisy data (SNR 10, 180 equi-angular projections, 10 iterations with OS-SIRT 10).

5.3.5 Qualitative and Quantitative Filter Comparison: Catphan Dataset

The Catphan dataset is a 2D image with 512^2 resolution. We used 45 projections for the few-view case and 180 projections with added SNR 25 Gaussian noise for the noise case. For the regularized construction strategy, we used 200 iterations with interleaved OS-SIRT 5 for the view-few case and 20 iterations of OS-SIRT 5 for the noisy projection case.

Figure 5.7 presents results we obtained for the few-view case, and Figure 5.8 presents enlargements of two specific areas: the upper contrast features and some of the phantom periphery (indicated by the orange and blue boxes, respectively). We observe that only the NLM and ANLM filters can resolve the 5th contrast feature from the right. The BLF filter and TVM are only able to distinctively resolve the 4th feature from the right. These qualitative differences are also quantified by the higher E-CC scores of the (A)NLM filters. Further, on the periphery, both BLF and TVM still exhibit some of the streak artifacts, while the

NLM shows some banding. Only the ANLM can fully restore the original appearance.

Figure 5.9 presents results we obtained for the noise case, and Figure 5.10 presents enlargements of two specific areas. We again observe that the noise case is more difficult to mitigate than the streak case with each result suppressing different levels of noise while somehow blurring the details. Just like before, ANLM performs best.

5.3.6 Regularized Reconstruction Time Performance

5.3.6.1 Time Performance of Filters

Table 5.2 lists the run times to filter images of 3 different sizes (256^2, 512^2, and 1024^2) on the GPU. We have noted before that performing filtering in 3D did not yield any improvements so restricting our experiments to 2D is well justified. In the table we list both the timings for the non-optimized (NOPT) and the optimized (OPT) GPU implementation (see Section 5.2.7.1). Our optimizations achieve a speedup of about 1.2 for the BLF, about 4 for the NLM filter, and about 3.2 for the ANLM filter. We already reported in [120] speedups of 2 orders of magnitude over a corresponding CPU implementation.

While we did not implement TVM on the GPU (all our results were generated with a CPU version), reference GPU implementations for TVM exist, such as the one by Pock et al. [86]. They used a NVIDIA 8800 GTX for their experiments and we report their timings in Table 5.2 as well. Further, in order to make these timings comparable to the ones reported here we also extrapolated them to the GTX 480 using commonly reported speedup numbers. Here we may add, however, that once the parameter λ grows larger, which is needed for the rather noisy data we have used here, the computation time tends to increase significantly over those listed here.

Overall we find that there is about an order of magnitude difference in the run times for each of the filters: BLF, NLM, and ANLM, with BLF being the fastest. TLF requires about twice the time of BLF [22] (not including the parameter estimation procedure), but since our results indicated that it was not practical for regularized CT reconstruction, we did not include its performance into Table 5.2. The TVM requires about the same time as NLM.

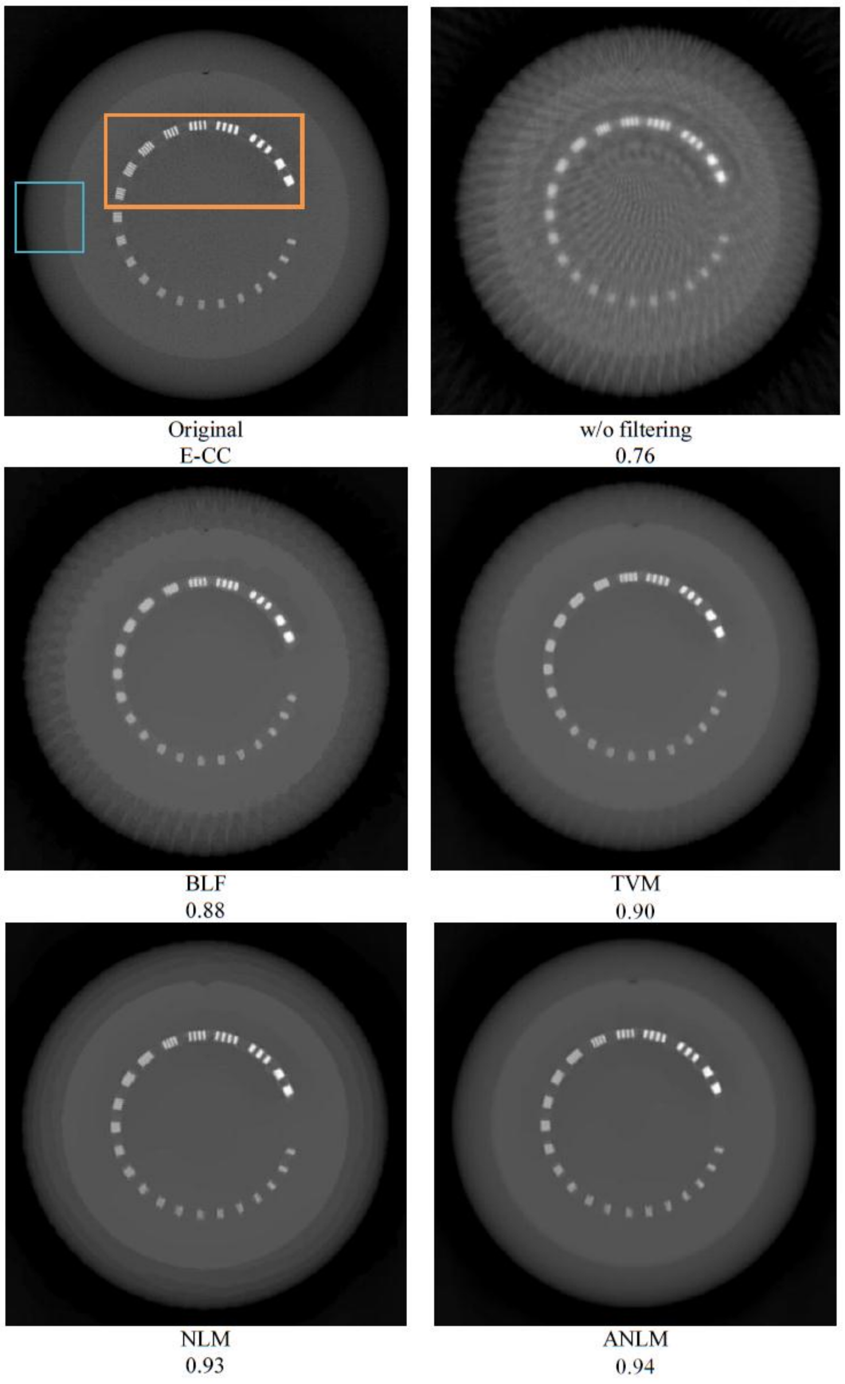


Figure 5.7: Catphan dataset, reconstructed with the interleaved regularization pipeline for the few-view scenario (45 uniformly distributed projections, 200 iterations with OS-SIRT 5).

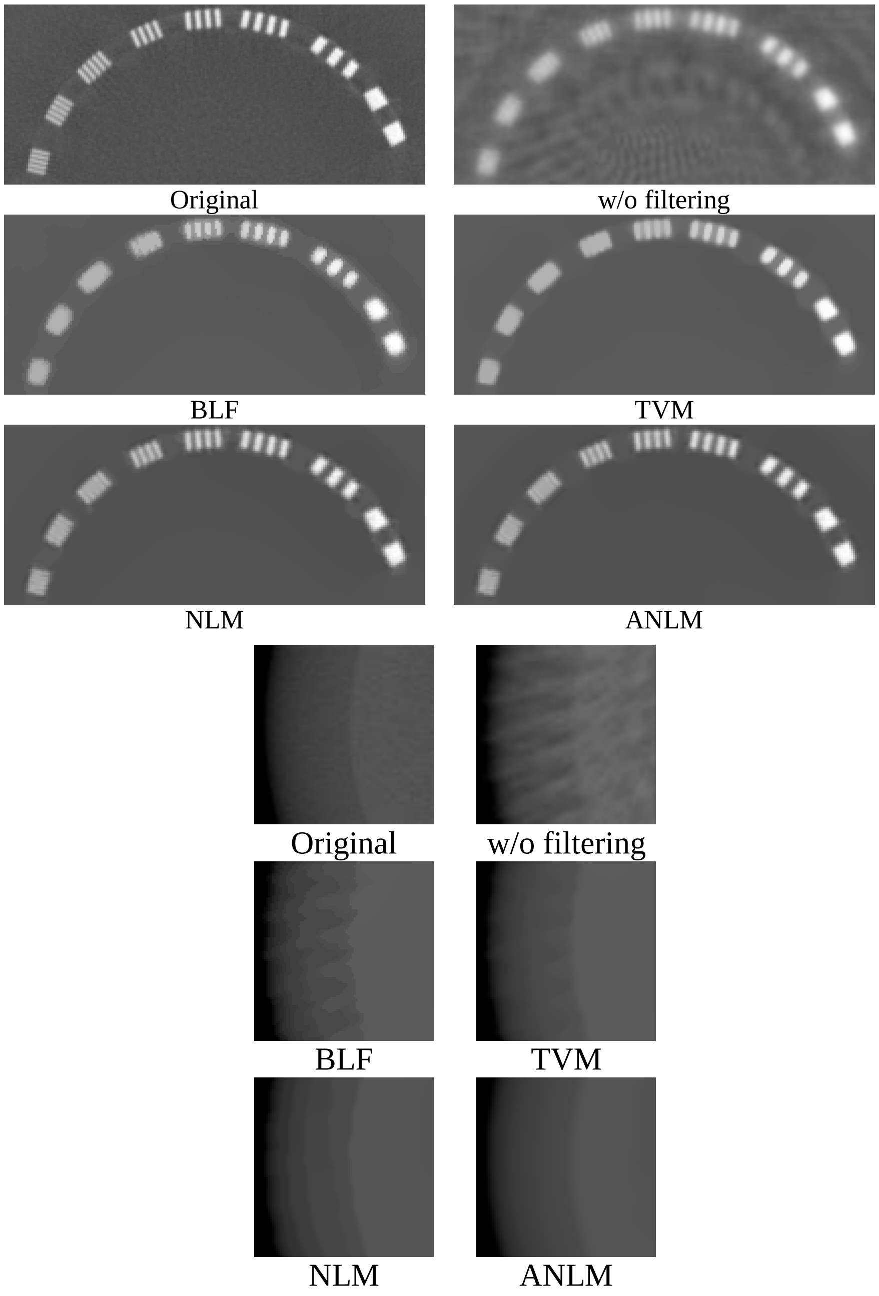


Figure 5.8: Zoomed detail views for Figure 5.7. Top rows: enlarged area of the upper contrast features (orange box); Bottom rows: enlarged periphery (blue box).

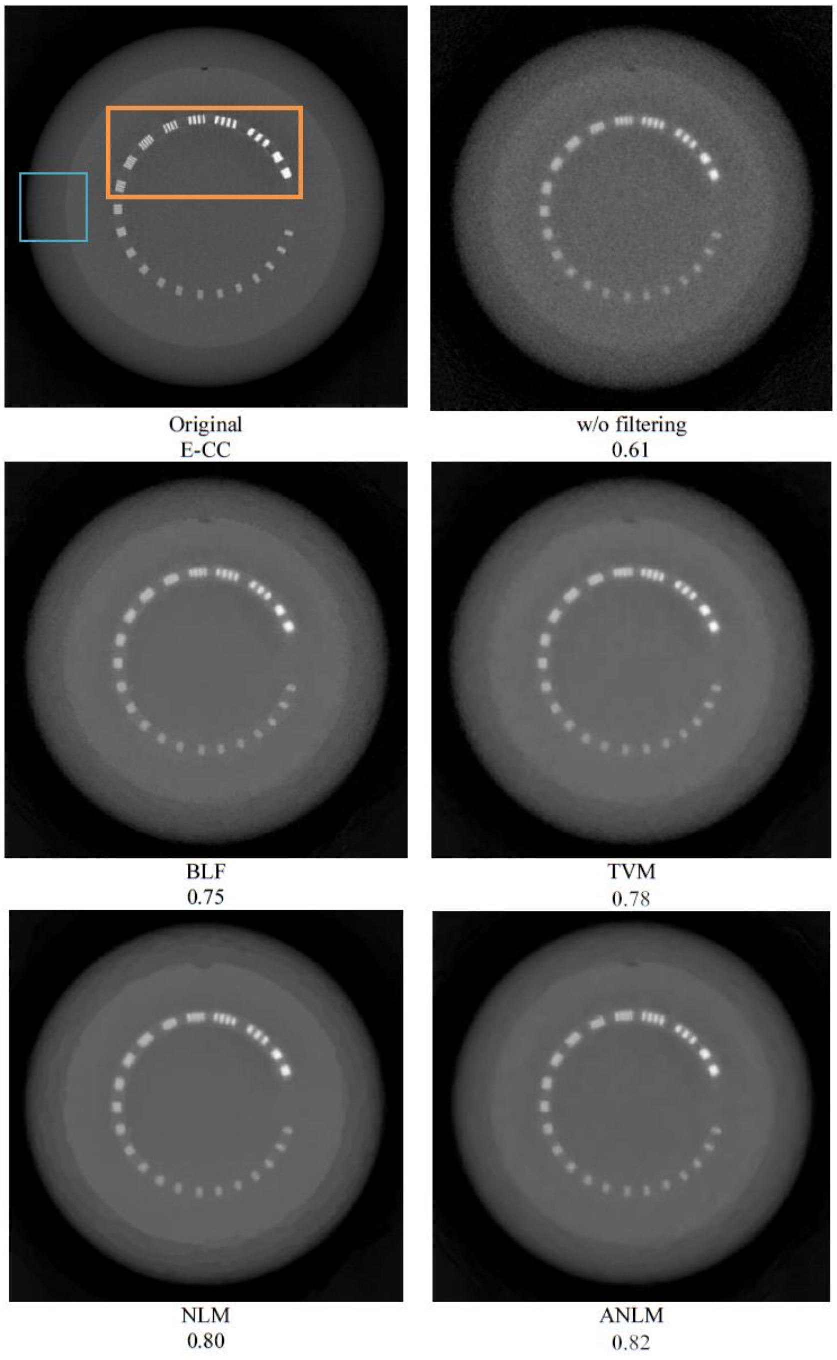


Figure 5.9: Catphan dataset, reconstructed with the interleaved regularization pipeline from noisy data (SNR 25, 180 equi-angular projections, 20 iterations with OS-SIRT 5).

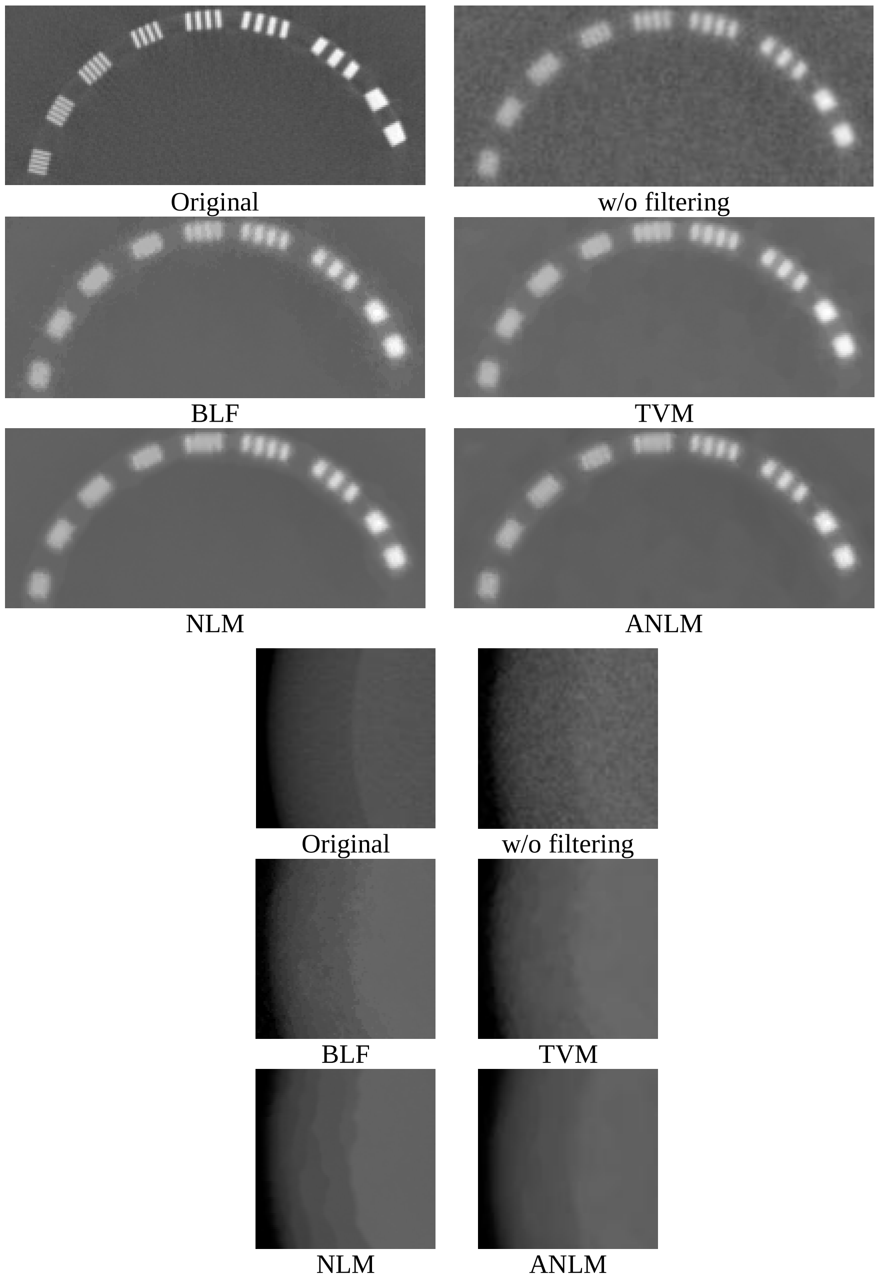


Figure 5.10: Zoomed detail views for Figure 5.9. Top rows: enlarged area of the upper contrast features (orange box); Bottom rows: enlarged periphery (blue box).

5.3.6.2 Time Performance of Regularized Reconstruction

Finally, we attempt to put these timings into the context of an entire iterative reconstruction pipeline. One iteration step with OS-SIRT 10 for 60 projections and a 512^3 volume takes about 10s on a 480 GTX. Performing BLF filtering for such a volume takes 1.76ms ·512=0.9s. This would add about 10% overhead to the CT reconstruction time which is tolerable. Less attractive from this point of view is TVM which produces about the same quality as BLF but is roughly 10 times slower (or more, see our comments above), making its overhead 100%. Further, the time required for NLM filtering is roughly the same as for TVM, but according to our experiments it produces substantially better results, so it appears that NLM is preferable. The ANLM filter, on the other hand, takes 117ms ·512=59s which is about 6 times the reconstruction time. It takes longer than the NLM since it is really a 4-step NLM plus some parameter estimation operations. For each ANLM iteration step, the window size doubles from 3×3 to 17×17 while the patch size is fixed to 7×7. However, given its excellent results this overhead may be well justifiable in some cases. Nevertheless, the NLM filter appears to provide the best cost-benefit ratio.

Test Size	**BLF**			**NLM**			**ANLM**			**TVM (**from [86])	
	NOPT	**OPT**	**Ratio**	**NOPT**	**OPT**	**Ratio**	**NOPT**	**OPT**	**Ratio**	**8800 GTX**	**GTX 480**
256^2	0.65	**0.53**	1.23	51.09	**12.70**	4.02	142.32	**43.57**	3.27	17.50	6.74
512^2	2.15	**1.76**	1.22	182.49	**42.06**	4.34	374.8	**117.24**	3.20	59.60	22.95
1024^2	8.08	**6.54**	1.24	699.23	**161.25**	4.34	2072.67	**597.91**	3.47	504.10	194.15

Table 5.2: Wall clock time (in ms) of the GPU-accelerated BLF, NLM, and ANLM filters, both optimized (OPT) and non-optimized (NOPT) for different image sizes. Ratio indicates the speedup NOPT/OPT. The GPU was an NVIDIA GTX 480 and the neighborhood size was 17×17. We also provide reference timings for the TVM filter, where we extrapolated the timings given by [86] for the older NVIDIA 8800 GTX GPU to the state-of-the-art GTX 480, using commonly reported speedup numbers. However, once the parameter λ grows larger, which is needed for the rather noisy data we have used here, the TVM computation times tend to increase significantly over those listed here.

5.4 Conclusions

We have explored the use of local nonlinear neighborhood filtering as a non-iterative alternative to the popular TVM method for regularized CT reconstruction. Our results indicate that these types of filters are indeed advantageous to TVM, meeting and exceeding its capabilities, in particular in terms of computational overhead. The latter could originate from the fact that the noise and streak artifacts in low-dose CT reconstruction are inherently local processes and so the global optimization of TVM is unnecessarily complex and consequently computationally wasteful.

As a possible future research direction one may try – as an alternative to adapting the ANLM filter's search window – to rather adapt the smoothing factor *h*, as a function of the local variance (see Coupe at al. [24]). Further, the search for suitable patch candidates can be accelerated as well (see Mahmoudi and Sapiro [76]) by only involving patches with similar means, variances, and gradients which can be effectively identified by pre-computed maps. Lastly, one could also make the patch matching process itself more efficient, using dimension reduction or hashing for better patch indexing.

As observed in the results, all of the regularization schemes we tested, including TVM, seemed to yield better results for the few-view case than for the noise-case. While we believe that these artifacts were of similar severity in our experiments, further research is advocated to substantiate this observation. Also, we did not explore if more expensive filters, such as the ANLM, would allow reconstructions to be completed with fewer iteration steps and so become more competitive overall (we ran all reconstructions for the same fixed number of iteration steps). Our parameter learning framework can be used here to optimize the number of iterations used for a given filter and data scenario.

Chapter 6

Learning Parameter Settings

Iterative reconstruction methods typically offer a set of parameters that allow some control over the convergence process, both in terms of quality and speed as we presented in previous chapters. Examples include relaxation factor, number of subsets, regularization coefficients, and the like. The interactions among these parameters and within the various data conditions can be complex, and thus effective combinations can be difficult to identify, leaving their choice often to educated guesses. As a contribution to this rarely researched area, we devise a data-driven learning approach and a multi-objective optimization approach to match given data configurations with their most effective reconstruction parameter configurations. We overcome the computational challenges associated with such a data-intensive approach by using commodity high-performance computing hardware (GPUs), which themselves have interacting parameters as well. A new perceptual image quality metric is presented. Finally a small toolbox called parameter space visualizer is included to interactively assist parameter selection for any parameter-related applications.

6.1 Introduction

An effective way to limit the overall radiation dose a patient is subjected to in a CT scan is to reduce the number of projections and also the radiation per projection. Both, however, increase noise in the CT reconstructions, compromising contrast as well as resolution. Iterative algorithms have been shown to excel in these adverse settings, but due to the absence of an exact solution, careful parameter tuning is typically required to converge to a solution close to the exact. Examples for such parameters include relaxation factor, number of subsets, and regularization coefficients. Their choice is often made ad-hoc based on some prior experience, yet typically not endorsed by a certified level of confidence. An added difficulty is that parameters often interact in their effects on reconstruction speed and outcome. Thus it can be a non-trivial task to derive the most suitable combination for a given data scenario. There are two main strategies by which one may arrive at effective parameter settings: optimization and data-driven learning. Optimization is similar to the reconstruction process itself, seeking to find the optimal solution (here the parameter configuration) constrained by some objective function. However, optimization can be vulnerable to local minima and it also lacks in some sense the capability to adapt to new data scenarios. Learning, on the other hand, aims to determine a process model (described by parameters) from a set of collected observations. In our application, these observations are reconstructions obtained with parameterizations of a given iterative reconstruction algorithm, where the quality of the reconstructions then drives the parameterization.

Clearly, the more observations we can provide and the greater their diversity, the more accurate our model is set to be. An important factor in this context is the quality metric. Since we aim to provide reconstructions to be examined by human observers, we require a quality metric that is perceptually based. Further, since we strive for a large number of observations, this perceptual metric needs to be computer-based and efficient to compute. In the following, we first describe our reconstruction parameter-learning framework and then present a metric fitting these requirements.

6.2 Methodology

In this work, we explain two approaches to learn the parameter settings for the GPU-accelerated OS-SIRT with bilateral filter regularization as presented in chapter 3. OS-SIRT has three parameters, the relaxation factor λ, the number of subsets S and the number of iterations. The bilateral filter has two more parameters σ_r and σ_d controlling the amount of smoothing (the search window size is fixed here). Our approaches are in fact very general to be able to work for any parameter learning process. In order to facilitate the computation of a ground truth-based figure of merit, we only used projection data acquired via simulation from known objects. We then added Gaussian noise at different SNR levels.

6.2.1 Approach 1: Exhaustive Benchmark Test

Using the simulated projection data, we compute a representative set of reconstructions, sampling the parameter space in a comprehensive manner. We then evaluate these reconstructions with a perceptual quality metric, as discussed below. Adaptive sampling can be used to drive the data collection into more "interesting" parameter regions (those that produce more diverse reconstruction results in terms of the quality metrics). Having acquired these observations, we label them according to certain criteria, such as "quality, given a certain wall-clock time limit" or "reconstruction speed, given a certain quality threshold". The observations with the higher marks, according to some grouping, subsequently receive higher weights in determining the reconstruction algorithm parameters. Currently, we either use the max-function or a fast-decaying Gaussian function to produce this weighting.

6.2.2 Approach 2: Multi-Objective Optimization

For OS-SIRT, the best quality and minimum time are the two objectives which are conflicting in nature with one another. Essentially, there is no single solution to satisfy every objective. Finding the optimal solutions is to make good compromises or trade-offs of the objectives. Therefore this parameter learning problem is actually a multi-objective optimization (MOO) [12][56]. Generally speaking, there are two classes of approaches to this problem: combining objective functions to a single composite function or determine a non-dominated optimal solution set (a so-called Pareto optimal set). The kernel of the approach 1 described above belongs to the former class of multi-objective optimization problems. The two objectives are weighted together to form a single objective function by selecting different labeling criteria. However, when the user does not have a specific quality or time demand but is curious about the level of potential progress toward each objective when the reconstruction continues, a non-dominated Pareto optimal set is more helpful and reasonable.

To solve the MOO problems, genetic algorithms (GA) are well suited. Genetic algorithms inspired by the evolutionary theory about the origin of species have been developed for decades. They mimic the species evolutionary process in nature that strong species have higher opportunity to pass down their genes to the next generations and at the same time random changes may occur to bring additional advantages. By natural selection, the weak genes will be eliminated eventually. GA operates in a similar way by evolving from a population with individuals representing initial solutions with operators such as selection, crossover and mutation to create descending generations. The fitness function is used to evaluate an individual which determines its probability of survival for the next generation. For MOO several fitness functions are used. Numerous GA methods have been described, such as Weighted-Based Genetic Algorithm (WBGA), Vector Evaluated Genetic Algorithm (VEGA), Multi-Objective Genetic

Algorithm (MOGA) and Non-dominated Sorting Genetic Algorithm (NSGA) [12]. Approach 1 is a customized application of the combination of fitness functions of WBGA to accommodate our special requirements. VEGA is limited due to the "middling" problem while MOGA is highly dependent on an appropriate selection of the sharing factor. NSGA progresses more slowly than MOGA and is shown to be computationally inefficient.

In our work, we have selected NSGA-II [26], which is the improved version of NSGA as the method to find the Pareto optimal set. This method uses elitism and a crowded comparison operator to keep both the inheritance from well-performing chromosomes and the diversity of the found optimal solutions while keeping the computational efficiency. We execute the learning process in this way: after generating the initial population, the reconstruction algorithm OS-SIRT and the parameter searching algorithm NSGA-II are run alternatively until the stopping criterion has been satisfied (for instance after a sufficient number of generations). This scheme falls into the category of a master-slave model of parallel genetic algorithms [106]. Within each generation, OS-SIRT is executed as a parallel fitness values evaluator while NSGA-II combines these values to find the current solution set with a user defined set size. Time is recorded as one objective and reconstruction quality is evaluated by a perceptual metric E-CC introduced below. To cast the optimization into a minimization problem, the quality metric is modified to its distance to unity. The process stops until the solutions in the current set are non-dominated to each other. For the selection, crossover and mutation operators, we use binary tournament selection, real coded blend crossover and a non-uniform mutation. While the parameters we found may not be substantially better for a specific user demand, they provide a pool of candidates that are trade-offs of time and quality. More importantly, they are obtained considerably faster since fewer reconstructions need to be computed. With Approach 1 we need to perform every combination of the representative parameters while with Approach 2, the computational complexity is the number of generations multiplying twice the size (user defined) of Pareto optimal solutions which is usually much smaller.

6.2.3 Image Quality Metrics

In CT reconstruction, the commonly used metrics for gauging reconstruction quality are mostly statistical, such as the cross-correlation coefficient (CC), mean absolute error (MAE), root mean square (RMS) error, and R-factor. However, the assessment of image quality should include both objective and subjective metrics [8]. Objective metrics, such as blurriness and contrast, measure the physical and geometric properties of the image and their effect on human perception. Subjective methods, rooted in psychophysics, more formally introduce observer perceptual metrics to gauge overall image quality. For example, Zhou et al. presented a comprehensive study of image comparison metrics [139], some described in the spatial and the frequency domain and some formulated in terms of the human visual system – the perception-based metrics – such as the visual differences predictor (VDP). While this (and many other) studies strongly suggest that perception-motivated metrics are superior to statistical ones due to the fact that human vision is highly sensitive to structural information [131], most of these involve heavy computation and thus are not desirable for the large data quantities we anticipate to process in our learning framework.

We first design a new group of metrics. Since the structural information could be well captured in the gradient domain, by ways of an edge-filtered image calculated via a Sobel Filter operator as we used in this study, every traditional metric can be easily transformed to an "edge sensitive" metric. We have labeled this group of metrics by prefix "E-" as in Figure 6.2. For example, the E-CC metric stands for the CC of two edge images. Further, shifting effects caused by the CT reconstruction backprojection step can be alleviated by Gaussian-blurring the reconstruction image before edge-filtering. We have labeled this group of metrics with prefix "BE-".

Another method to gauge structural information is Structural Similarity (SSIM) which combines luminance, contrast and structure [WB04]. Given two signal images *x* and *y*, the definition of SSIM index is defined as:

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{6.1}$$

where α, β and γ are parameters adjusting relative importance. The terms $l(x,y)$, $c(x,y)$ and $s(s,y)$ are the luminance, contrast and structure comparison functions, respectively. These functions are computed from local image statistics [131].

6.3 Results

All computations used an NVIDIA GTX 280 GPU / Intel 2 Quad CPU 2.66GHz. We group our results into five sections: (1) the metrics comparison showing which metrics give consistent scores, (2) the OS-SIRT results showing the relationship between noise levels and parameters settings, (3) the OS-SIRT results for few-projection parameters settings, and (4) the parameter settings with our GPU-accelerated bilateral filter, (5) the tryout results of MOO for noise levels parameter settings.

6.3.1 Comparing the Metrics

Our first (somewhat informal) study examined various image quality metrics and their suitability to replace a human judge/observer. We note that our emphasis in the onset was to select metrics that are fairly easy to evaluate, in order to keep the assessment of the expected large data volume manageable. We use a baby head scan (size 256^2) to demonstrate our results (see Figure 6.1). All metrics are in the range [0, 1] and thus allow direct numerical comparison. We compared the metrics at two levels: 0.99 (row 1 of Figure 6.1) and 0.82 (row 2). We observe that the scores of E-CC and SSIM are quite faithful to the scores a human observer might give (0.99 for a near-perfect image and 0.82 for a "B+ similarity"). On the other hand, CC is clearly bound to over-score the images. It assigned the left bottom image a score of 0.82 even though the object structure is barely discernable, and it assigned the top left image a near-ideal score even though the structures are still quite blurry.

For a more comprehensive study of all metrics discussed in Section II, we reconstructed the popular "Barbara" test image (size 256^2), varying the number of projections and the level of noise (quantified by SNR). This image contains very high frequency detail and is thus quite sensitive to small errors. For all reconstructions we stopped iterations when the CC reached a value of 0.89. We chose this relatively low CC value so all SNR levels could reach it – better reconstructions are possible. Figure 6.2 presents the images reconstructed from 180 and 140 noise-free projections (NF 180 and NF 140), respectively, and from 180 projections with Gaussian noise added (SNR 10 and SNR 5). We clearly see that these images do not look the same from a perceptual point of view, although the CC metric has the same outcome. Informally, an observer would likely rank these images in the order NF140, NF180, SNR10, SNR 5, with NF140 being the best (small detail is better visible there, although there are some slight yet tolerable artifacts). From the table in Figure 6.2 we observe that only the edge-based metrics (E-MAE, E-NRMS, and E-CC) as well as SSIM can reproduce this ranking (note that the ordering for CC is the opposite than for RMS and MAE since they have reverse maxima, and also note that we scaled some metrics by the given multiplicative factor to better visualize the contrast of the bars). The other metrics have either a wrong ranking or could not distinguish some images at all. In the remainder of the work, we use E-CC since it better tolerates global density shifts and is faster to compute (than SSIM).

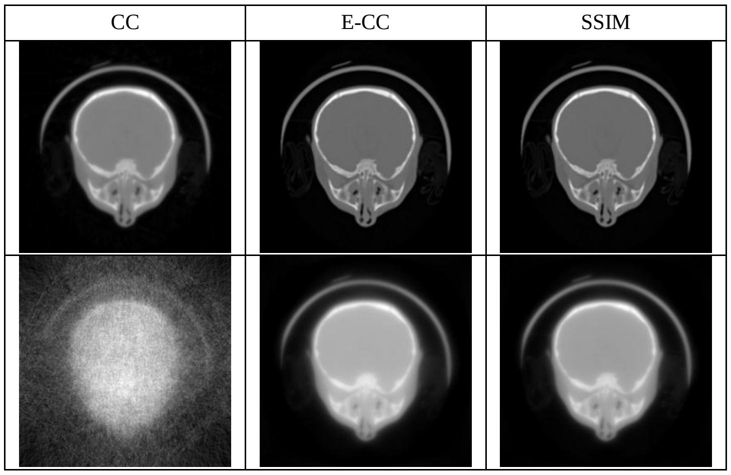


Figure 6.1: Image metric comparison I: the value for each metric is fixed to 0.99 in the first row and to 0.82 in the second row.

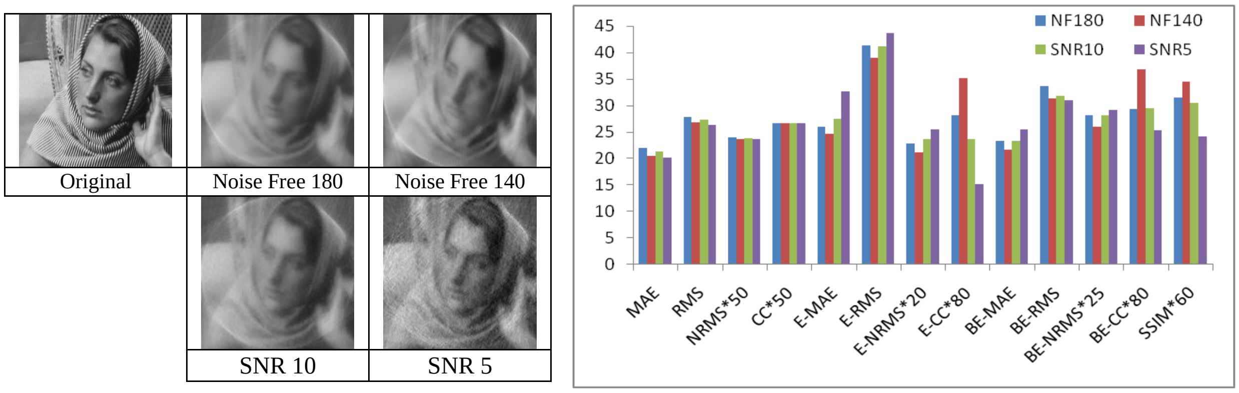


Figure 6.2: Image metric comparison II: four reconstructed images that have the same CC (0.89) but score differently with other metrics (see plot).

6.3.2 Learning Parameters with Exhaustive Benchmark Testing

With a suitable quality metric in place we are now ready to learn the most effective parameters for the iterative OS-SIRT reconstruction algorithm we used to evaluate our approach. We first simulated, from the baby head CT scan, 180 projections at uniform angular spacing of [-90˚, +90˚] in a parallel projection viewing geometry. We then added different levels of Gaussian noise to the projection data to obtain SNRs of 15, 10, 5, and 1. The first column of Figure 6.3

presents the best reconstruction results (using the E-CC between original and reconstructed image), for each SNR, in terms of the "reconstruction speed, given a certain quality threshold" criterion. In other words, the images shown are the reconstructions that could be obtained at the shortest wall clock time given a certain minimal E-CC constraint. This constraint varies for each projection dataset (low SNR cannot reach high E-CC levels), and this is also part of the process model.

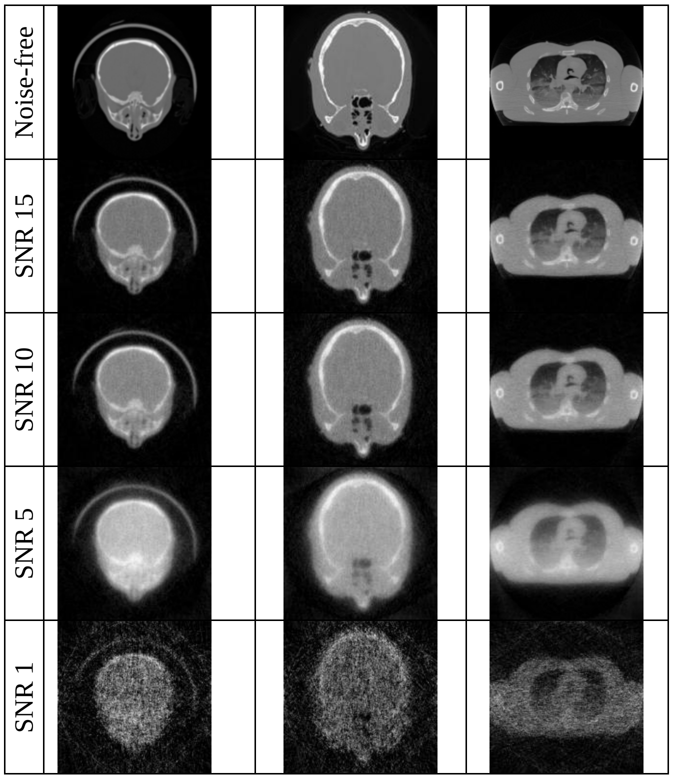


Figure 6.3: Results for all SNR levels of three datasets using the same parameter settings, those found most effective for the baby head dataset: columns from left to right - baby head, visible human head, and visible human lung, rows from top to bottom - noise-free, SNR 15, 10, 5 and 1.

Figure 6.4 summarizes the various parameters obtained for the various data scenarios mentioned above. The "Best Subset" and "Best Lambda" values denote the parameter settings that promise to give the best results, in terms of the given quality metric and label criterion. The "Lowest Lambda" and "Turning Point" values describe the shape of the λ–curve as a function of the number of subsets. The λ-factor is always close to 1 for small subsets and then linearly (as an approximation) falls off at the “Turning Point” to value “Lowest Lambda” when each subset only consists of one projection (which is SART) [118].

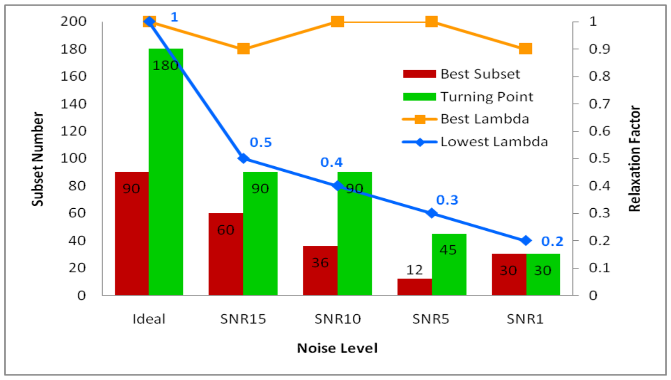


Figure 6.4: Optimal parameter settings using E-CC for the baby head dataset: subset number and relaxation factor as a function of imaging condition (SNR) and the turning point and lowest lambda for each SNR level.

The summary plot of Figure 6.4 helps practitioners to pick the best-performing number of subsets and the associated λ (to obtain the best possible quality within a given time) for a given expected SNR level. For example, we observe that low SNR requires a low number of subsets, while less noisy data can use a higher number of subsets. This trend is well confirmed by prior studies and field experience and thus validates the correctness of our general approach.

We then explored if the knowledge we learned translates to other similar data and reconstruction scenarios. Column 2 and 3 of Figure 6.5 show the results obtained when applying the optimal settings learned from the baby head to reconstructions of the Visible Human head (size 256^2) and Visible Human lung (512^2), from similar projection data. We observe that the results are quite consistent with those obtained with the baby head, which is promising. As future work, we plan to compare the settings with those learned directly from these two candidate datasets.

Next, we investigated the few-view reconstruction scenario. Here, we “learned” that SART consistently gave the best results. Figure 6.5 shows the best reconstructions of the baby head with S=180 to 20 in the top two rows. The third row shows the results of lung dataset using the same parameters than used in the second row. They are quite similar which confirms the generalization of the learned parameters.

Finally, we sought to learn the λ–parameters for the bilateral filter used for regularization. Figure 6.6 presents results. We found that the bilateral filter helped to reduce the number of iterations especially for the smaller number of subsets. Since the bilateral filter is less expensive than an iteration it pays off to use it.

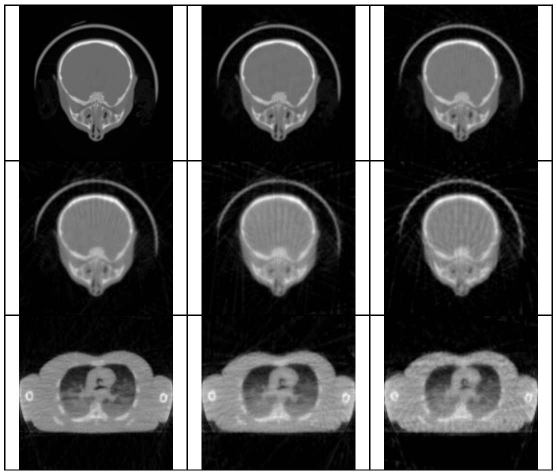
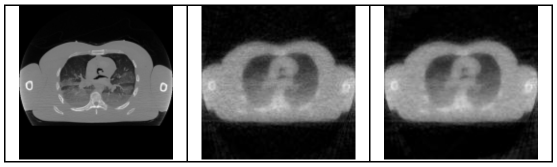


Figure 6.5: Results for Figure 6.6. (Top) The best results (using the time criterion) for the lung dataset with SNR 10 and 30 projections (E-CC=0.6). (left to right): original image, reconstruction without and with bilateral filter. (Bottom) Comparing the number of iterations required with / without bilateral-filter regularization.



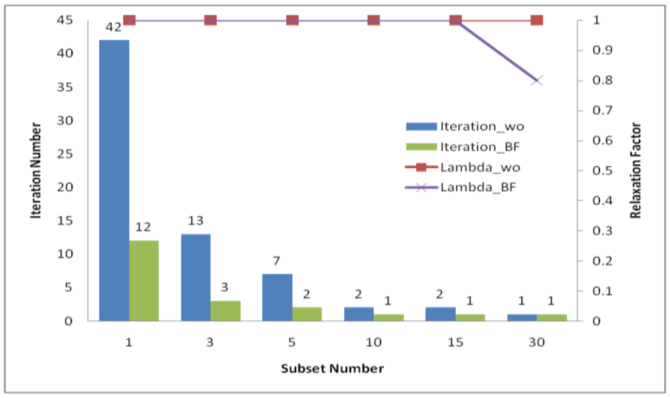


Figure 6.6: (Top) The best results (using the time criterion) for the lung dataset with SNR 10 and 30 projections (E-CC=0.6). (Left to right): original image, reconstruction without and with bilateral filter. (Bottom) Comparing the number of iterations required with / without bilateral-filter regularization.

6.3.3 Learning Parameters Using Multi-Objective Optimization

To test the performance of the second approach, the Visible Human head (size 256^2) with different noise levels (ideal, SNR 10 and SNR 5) is used. We set the size of any Pareto optimal set to 20 and obtained the corresponding sets for parameters (number of iterations, number of subsets and relaxation factor λ) after adequate generations (here until any solution is non-dominated by others in the same set). The results for different noise levels are shown in Figure 6.7. For each noise level, the gradual evolutions of the fitness values after selected number of generations are plotted. We observed that when the generation develops, the fitness values are closer to the axes which means closer to its optimal value. The solutions pool is listed in Table 6.1. It demonstrates that when the noise level increases the reconstruction is more difficult while at the same time the needed numbers of iterations and subsets are getting smaller, which is confirmed by Approach 1 as well.

	Ideal			SNR 10			SNR 5		
Index	#Iter	#Subset	λ	#Iter	#Subset	λ	#Iter	#Subset	λ
1	55	180	0.84	2	30	0.755	2	90	0.162
2	76	90	0.997	2	30	0.82	2	90	0.167
3	68	90	0.92	2	20	0.982	2	15	0.884
4	56	90	0.87	2	20	0.892	2	15	0.909
5	53	90	0.87	2	18	0.938	2	12	0.961
6	43	90	0.952	2	15	0.991	2	15	0.975
7	48	60	0.787	2	15	0.862	2	12	0.99
8	50	45	0.902	2	12	0.978	2	12	0.966
9	47	45	0.974	2	10	0.966	2	10	0.995
10	38	60	0.811	2	10	0.896	2	10	0.993
11	16	90	0.965	2	9	0.732	2	10	0.883
12	19	60	0.923	2	6	0.998	2	6	0.998
13	20	45	0.969	2	6	0.943	2	6	0.995
14	19	30	0.964	2	6	0.894	2	6	0.998
15	8	60	0.83	2	6	0.877	2	5	0.984
16	17	15	0.997	2	3	0.913	2	4	0.923
17	2	60	0.875	2	2	0.973	2	3	0.998
18	3	10	0.459	2	2	0.764	2	3	0.993
19	2	30	0.033	2	2	0.739	2	2	0.999
20	2	4	0.175	2	2	0.288	2	2	0.972

Table 6.1: Pareto optimal set learned by Approach 2.

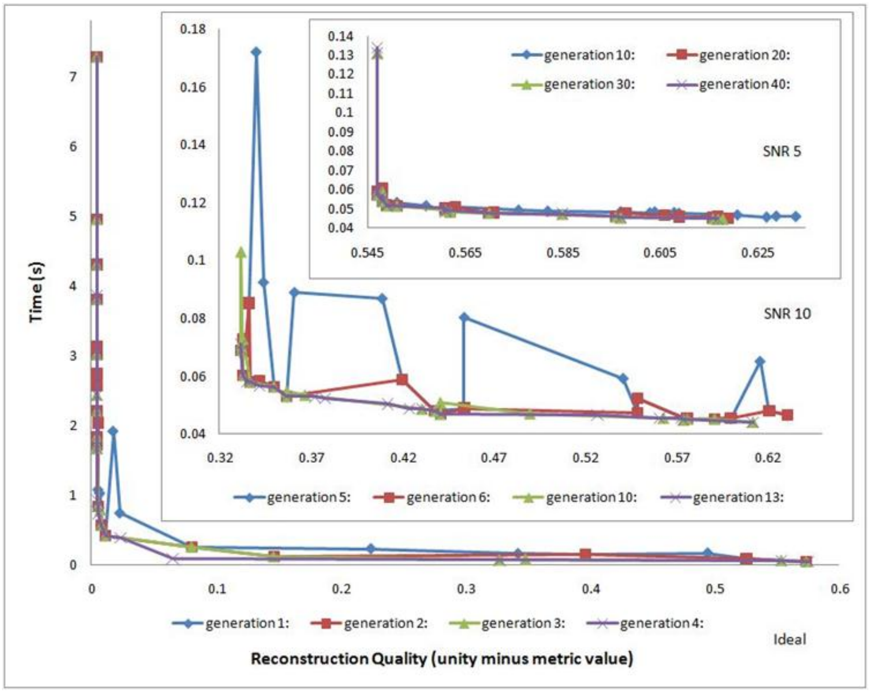


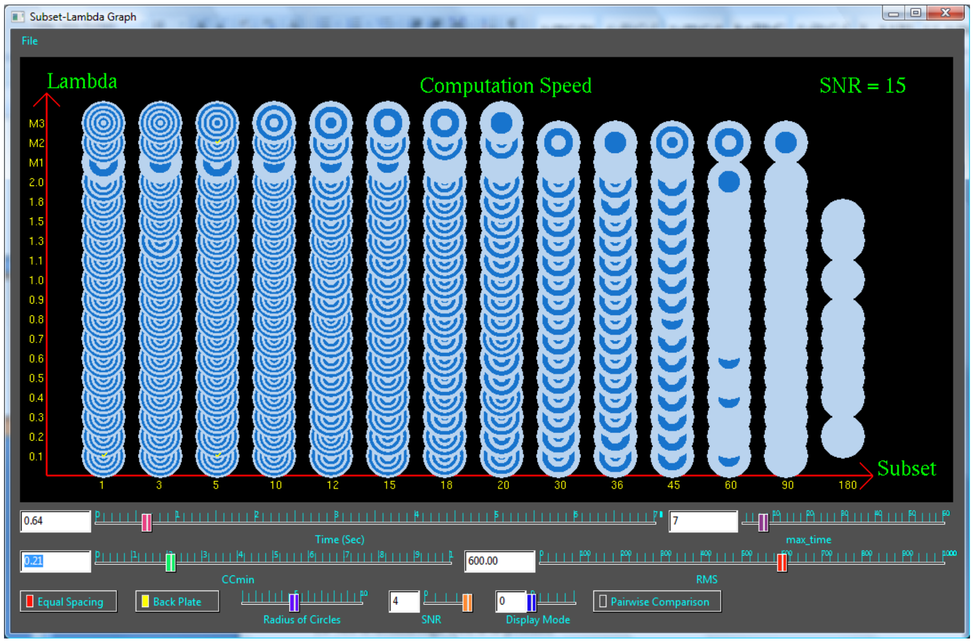
Figure 6.7: Fitness values of solutions set evolutions for different SNR.

6.4 Parameter Space Visualizer

We also devise an effective parameter space navigation interface as a toolbox allowing users to interactively assist parameter selection for iterative CT reconstruction algorithms (here for OS-SIRT). This toolbox is especially useful when users have no prior knowledge about the suitable parameter settings for the given dataset. After performing offline computation, our toolbox gathers the performance of all combinations of parameter settings and could interactively display the results based on users' online demands. In general, it is based on a 2D scatter plot with six display modes to show different features of the reconstruction results based on the user preferences. It also enables a dynamic visualization by gradual parameter alteration for illustrating the rate of impact of a given parameter constellation. Finally, we note the generality of our approach, which could be applied to assist any parameter selection related systems.

There are seven parameters to represent one reconstruction: SNR-level, relaxation factor λ, number of subsets S, number of iterations, time, CC and RMS. With the given projections, we selected a representative set of integer-subsets, λ and SNR-levels respectively. For each combination of <S, λ , SNR-level> (called *grid points*), <time, CC, RMS> (called *tuples*) were recorded every five iterations.

With this pool of tuples, we use our parameter space visualizer to visualize the parameters with user control.



Time (Sec): 0.64
max_time: 7
CCmin: 0.21
RMS: 600.00
SNR: 4
Display Mode: 0

Figure 6.8: Parameter Space Visualizer in Computation Speed Mode.

The interface of the parameter space visualizer is composed of a main window and a control panel as shown in Figure 6.8. The main window is a 2D scatter plot of number of subsets (x-coordinate) and relaxation factor (y-coordinate) with different sizes and colors of circles to show the features of tuples. The size and color of the circle represents the number of tuples satisfying current parameter settings controlled by users in the control panel. In the control panel, users could change the values of the time, CC and RMS bars to set boundary values to sift tuples. SNR bar is used to switch visualizations among different SNR-levels. There are six “display modes” showing various features of the tuples: Computation Speed Mode, Percentage Mode, Absolute Number Mode, Minimum Time Mode, Maximum CC Mode and Starting CC Mode. Users could choose to better plot the results by modifying the status/values of “equal spacing”, “back plate” and “radius of circles” at the bottom of the control panel. “Pairwise comparison” is another control for Absolute Number Mode. The interface assists users to quickly find the preferred parameter settings under the input constraints. It provides a dynamic view of subtle parameter changes by moving the control bars so that users could be aware of the impact of some specific parameter. For details and other features, please go to [123].

6.5 Conclusions

We demonstrated an intelligent framework that has the potential to automate the parameter selection for CT reconstruction tasks. Iterative algorithms likely benefit the most from this scheme, since they tend to have a variety of parameters to adapt the optimizer to the present data conditions and reconstruction goals. Such a system can be helpful to practitioners that do not have the expertise to tune these parameters by hand. Finally a parameter space visualizer is provided to assist users select parameters interactively.

As future directions, for approach 1 the framework could be refined such that it can recognize “signatures” directly from the projection data, combining them with other information about scanner and object, and use this information to index the parameter knowledge base. For approach 2, the limitation is that the genetic algorithm itself includes a few parameters to control selection, crossover and mutation operators. A more careful study exploring suitable settings as well as the various choices of operators would better tune the solutions.

Chapter 7

High-Performance Iterative Electron Tomography Reconstruction

Iterative reconstruction algorithms pose tremendous computational challenges for 3D Electron Tomography (ET). Similar to X-ray Computed Tomography (CT), graphics processing units (GPUs) offer an affordable platform to meet these demands. In this chapter, we outline a CT reconstruction approach for ET that is optimized for the special demands and application setting of ET. It exploits the fact that ET is typically cast as a parallel-beam configuration, which allows the design of an efficient data management scheme, using a holistic sinogram-based representation. Our method produces speedups of about an order of magnitude over a previously proposed GPU-based ET implementation, on similar hardware, and completes an iterative 3D reconstruction of practical problem size within minutes. We also describe a novel GPU-amenable approach that effectively compensates for reconstruction errors resulting from the TEM data acquisition on (long) samples which extend the width of the parallel TEM beam. We show that the vignetting artifacts typically arising at the periphery of non-compensated ET reconstructions are completely eliminated when our method is employed.

7.1 Introduction

Electron Tomography (ET) (see for example [37] or [66]) uniquely enables the 3D study of complex cellular structures, such as the cytoskeleton, organelles, viruses and chromosomes. It recovers the specimen's 3D structure via computerized tomographic (CT) reconstruction from a set of 2D projections obtained with Transmission Electron Microscopy (TEM) at different tilt angles. ET can be accomplished using exact analytical methods (weighted back-projection WBP [91] and more recently electron lambda-tomography [90]) or via iterative schemes, such as the Simultaneous Algebraic Reconstruction Technique (SART) [2], the Simultaneous Iterative Reconstruction Technique (SIRT) [45], and others. The dominant use of the analytical methods is most likely due to their computational simplicity and consequently fast reconstruction speed. Iterative methods, however, have the advantage that additional constraints can be easily and intuitively incorporated into the reconstruction procedure. This, for example, can be exploited to better compensate for noise [100] and to perform alignment corrections [30][43][61] during the iterative updates. Additional challenges are imposed by the fact that the projection sinogram is vastly undersampled, both in terms of angular resolution (due to dose constraints) and in terms of angular range (due to limited sample access). These types of scenarios can be handled quite well using iterative reconstruction approaches [1].

Thus, iterative approaches have great potential for ET. However, as data collection strategies [138] and electron detectors improve, the push has been to reconstruct larger and larger volumes ($2048^2 \times 512$ pixels and beyond). Although the benefits are significant, the major obstacle preventing the widespread use of iterative methods in ET so far has been the immense computational overhead associated with these, leading to reconstruction times on the order of hours to days for practical data scenarios. As in many other scientific disciplines, the typical solution to meet these high computational demands has been the use of supercomputers and large computer clusters [36][39][141], but such hardware is expensive and can also be difficult to use and gain access to. Fortunately, the recently emerging graphics processing units (GPUs) offer an attractive alternative platform, both in terms of price and performance. GPUs are available at a price of less than $500 at any computer outlet and, driven by the ever-growing needs and tremendous market capital of computer entertainment, their performance has been increasing at triple the rate of Moore's law, which governs the growth of CPU processors. For example, the recent NVIDIA GPU board GTX 280 has a peak performance of nearly one Trillion floating point operations per second (1 TFlop), which is 1-2 orders of magnitude greater than that of a state-of-the-art CPU.

The great performance of GPUs comes from their highly parallel architecture, and the vast potential of these boards for general high performance computing has given rise to the recent trend of General Purpose Computing on GPUs (GPGPU) [82]. In the past, GPU-programming was only possible via graphics APIs, such as CG, GLSL and HDSL, which required programmers to have some background in computer graphics. In order to make the hardware more accessible to non-graphics programmers, a C-like parallel computing programming interface called CUDA (Compute Unified Device Architecture) has recently been introduced by GPU manufacturer NVIDIA. A similar but more general API called OpenCL has also become available. We have used GLSL for our implementation.

The high potential of GPUs for accelerating Computed Tomography (CT) has been recognized for quite some time in the field of X-ray CT [14][15][54][116][95][112][117][125][114], and more recently also for ET [28][27][63][102][98]. The majority of GPU algorithms developed for X-ray CT have focused on 3D reconstruction from data acquired in perspective (cone- and fan-beam) viewing geometries, using flat-panel X-ray detectors in conjunction with X-ray point sources. This poses certain constraints on how computations can be managed (pipelined) given the highly parallel SIMD (Single Instruction Multiple Data) architecture of GPUs. However, data acquisition in ET is typically posed within a parallel-beam configuration, and this allows for additional degrees of freedom in the implementation, which are not available in the cone- and fan-beam configurations. Our approach exploits these opportunities to derive a novel high-performance GPU-accelerated iterative ET reconstruction framework.

The GPU method proposed by Castano-Diez et al. can be viewed as a first step towards achieving high-performance ET. Our framework is a substantial advance of their method, speeding up their calculations by an order of magnitude. Such speedups are especially significant when it comes to 3D reconstructions, which are the ultimate goal of ET. The method of Castano-Diez at al. enables only 2D iterative reconstructions to be accomplished at reasonable speeds (where *reasonable* is defined here as being on the order of minutes). However, reconstructions at the same resolution, but in 3D, still take hours to compute. Our framework, on the other hand, obtains these 3D reconstructions in an order of minutes, on comparable hardware. Finally, the latest generation of GPU hardware enables further considerable speed increases, which may be valued as another demonstration of the immense potential GPUs have for iterative ET.

As a mechanism to express a compromise between SART and SIRT, we presented OS-SIRT algorithm in previous chapters. A similar compromise has been introduced as OS-SART by [113]. The rational was similar to that of [50] who devised OS-EM, an ordered subsets algorithm for the Expectation Maximization (EM) algorithm [103]. In OS-EM, the best subset size is one that most optimally balances the noise compensation offered by larger subsets (many projections in one subset) and the smaller number of iterations required for convergence offered by smaller subsets (many corrective updates within one iteration). However, for our OS-SIRT the focus was to provide a mechanism by which one can balance GPU *runtime performance* (which is convergence as measured in wall-clock time) with noise cancelation (for better reconstruction quality). For the work here we have applied this framework to CT reconstruction from TEM data and show that OS-SIRT also provides a favorable algorithm here. We note that the study of OS-SIRT and its optimization for TEM data is not the focus of this work – this is subject of future work. Rather, in the current work we have aimed to provide more insight into GPU-accelerated computing for ET reconstruction.

Finally, an important issue in CT is the "long object" reconstruction problem. It arises in spiral CT when the goal is to reconstruct a region-of-interest (ROI) bounded by two trans-axial slices, using a set of axially truncated cone-beam projections corresponding to a spiral segment long enough to cover the ROI, but not long enough to cover the whole axial extent of the object [29]. Essentially, in this situation some rays used for ROI reconstruction also traverse object regions not within the ROI, and these rays are sometimes called "contaminated" rays. This problem is similar to the "local tomography" problem in ET. While for ET the data acquisition trajectory is orthogonal to the one in spiral CT, ray contamination occurs whenever the object contains material not covered by every projection (that is, only a sub-region of the object is exposed to electrons in a local view). These areas are then incompletely reconstructed in the iterative procedure, which is evidenced by vignetting - a brightness fall-off in the peripheral regions of the reconstructed object. We derive a method, within our iterative framework, which effectively compensates for this effect, correcting the contaminated rays for the missing information.

Our work is structured as follows. Section 7.2 presents relevant background both on reconstruction algorithms and on GPU hardware. Section 7.3 describes our various contributions, that is, the advanced GPU acceleration framework, the extension to the OS-SIRT mechanism, and the long-object compensation method. Section 7.4 presents results and Section 7.5 ends with conclusions.

7.2 Background

Before detailing the contributions of this work, we first give a brief overview over the implemented – and then accelerated and extended – reconstruction algorithms and the relevant intricacies of GPU hardware. We have only considered algebraic reconstruction algorithms, but the hardware acceleration generalizes readily to expectation maximization (EM) type procedures (for more information, see [115]).

7.2.1 Iterative Algebraic Reconstruction: Theory and Practice

As we have shown in section 3.2.1, most iterative CT techniques use a projection operator to model the underlying image generation process at a certain viewing configuration (angle) φ. The result of this projection simulation is then compared to the acquired image obtained at the same viewing configuration. If scattering or diffraction effects are ignored, the modeling consists of tracing a straight ray r_i from each image element (pixel) and summing the contributions of the volume elements (voxels) v_j. Here, the basis function w_{ij} determines the contribution of a v_j to r_i. One can cast the choice of the weighting factors w_{ij} as an interpolation problem, where the rays traverse a field of basis functions (kernels) w_{ij}, each centered at a voxel v_j [79]. The most efficient basis functions for GPUs are the nearest-neighbor and linear interpolation functions, in conjunction with point sampling. We have shown in earlier work [116] that for 3D iterative reconstruction (with SART) this type of sampling is sufficient. There we showed that the error function was similar to the one obtained with Siddon's line and area/volume integration schemes which assume lower-quality nearest-neighbor kernels (but integrate them). A similar observation was also made for SIRT [6].

Once the number of rays (pixels) and voxels are sufficiently large, it becomes infeasible to store the w_{ij} as a pre-computed array. In fact, since GPUs are heavily optimized for computing and less for memory bandwidth (which is consequence of general semi-conductor technology), computing these w_{ij} on the fly is by far more efficient. This is even more so, since linear interpolation up to three dimensions and up to 32-bit floating-point precision is implemented on GPUs in special extra-fast ASIC circuitry. As it turns out, on the latest GPU cards there is almost no difference in performance for nearest-neighbor and bi-linear interpolation. This is fortunate, since iterative algorithms are very sensitive to the accuracy of the projector and thus bi-linear interpolation is a requirement for high-quality reconstructions (see Section 7.4). This sensitivity comes from the need for accurate correction factors to be used for the iterative updates. Here, linear interpolation strikes a good balance between aliasing and smoothing. The correction update for projection-based algebraic methods is computed in equation (3.2).

7.2.2 Graphics Hardware: Architecture and Programming Model

GPUs have their origin as dedicated graphics processors. In graphics, high visual detail (when geometric detail is not needed) can be generated by simply mapping an image of sufficient resolution onto a large polygon. This requires two units. First, one needs a high-performance parallel polygon rasterizer, since each such polygon potentially affects many screen pixels onto which the image must be mapped. Second, with each rasterized pixel giving rise to a *fragment* one also requires a high-performance parallel fragment processor, able to process the oncoming front of fragments efficiently. Traditionally, each such fragment computation was just an interpolation of the mapped image at the coordinates attached to the fragment by the rasterizer. But now these computations can be much more involved, such as generating sophisticated lighting effects, driving complex physical simulations, or, in a GPGPU application, the execution of an arbitrary program (called *shader*). The key to a successful GPGPU implementation is that these computations can be cast as parallel operations, executed in lock-step (SIMD = Single Instruction Multiple Data) on the parallel fragment processors. This puts certain constraints on both data flow and programming model, which often requires a creative re-organization of the existing CPU program to enable optimal utilization of all parallel GPU resources.

The NVIDIA G70 chip (which forms the GPU in the 7800 GTX and in the Quadro FX 4500) has 24 SIMD fragment processors (and 8 vertex processors), 512MB of DDR memory, and 165 GFlops performance. On the other hand, the recent GTX 280 has 240 generalized processors, 1GB of DDR memory, and nearly 1TFlops performance (for further detail on these chips and boards the reader is referred to the corresponding Wikipedia pages). As mentioned, up to recently, the only way to interface with GPU hardware was via a graphics API, such as OpenGL or DirectX, and using CG, GLSL, or HDSL for coding the shader programs to be loaded and run on the fragment processors. With CUDA, the GPU can now directly be perceived as a multi-processor, and a suitable programming interface is available for this model where fragments become the CUDA (SIMD) *computing threads* and the shader programs become the *computing kernels*, which can be launched by a single instruction.

Textures are the data structures utilized most frequently in GPU graphics programming (and they can also be used with CUDA). Textures are essentially 1D-3D arrays supporting data types of various precisions ranging from 8-bit fixed point to 32-bit floating point. Among those, 2D floating point textures are most suitable for general purpose computing due to their complete support of all data formats and interpolation schemes. Conceived within a graphics context, textures are designed to contain up to four channels (RGBA) of data, where RGB is (red, green, blue) color and the A (alpha) channel carries the opacity (1-transparency) value. This feature provides additional data parallelism, in addition to the intrinsic pipeline parallelism described above. However, in order to achieve this parallelism one needs to fulfill even stronger requirements than just SIMD.

Understanding the GPU memory hierarchy is the key to maximizing computational performance. On the GPU, memory is organized in register, shared, texture, and global memory. Registers and shared memory are fastest and on-chip, while texture and global memory is maintained as slower DDR memory and on-board. Shared memory stores recently accessed data blocks for use by parallel threads, and each memory miss causes 100 or more wait clock cycles. Fortunately, GPUs hide these latencies by replacing any waiting thread by another thread that (already) has the data it needs to compute the current SIMD instruction. It is therefore desirable to (a) maintain data locality among neighboring threads in order to prevent costly cache misses overall, (b) launch a sufficient number of threads (many more than the number of available microprocessors) so the latencies incurred by cache misses can be hidden by overlapping the memory waits with computation, and (c) keep the kernel program sufficiently long to amortize setup cost, minimize synchronization overhead, and promote efficient instruction and data flow.

7.3 Methods

Most relevant in a GPU-accelerated CT reconstruction framework is to have an efficient projection and back-projection operator. The remaining operations, such as the correction computations, are simple vector operations of low complexity and can be implemented on the GPU by subtracting two 2D textures, the texture holding the acquired projections and the texture computed during projection. In the following we (a) describe an efficient parallel framework that accelerates these operations, and (b) present a new method that efficiently deals with the effects of the limited specimen coverage of the detector.

```
(1) For all $VS$ volume slices
    (2) For all $S$ ordered sets $OS_s$
        (3) For all projections $P_{sp}$ in $OS_s$
            (4) For all pixel rays $r_{sti}$ in $P_{sp}$
                Initialize ray sum $rs_{sti}$
                Set up space traversal for $r_{sti}$
                  (5) For all slice positions $p_l$ along $r_{sti}$
                      Advance $r_{sti}$ by step size $\Delta r$
                      Sum the weighted contributions from the (bilinear) neighborhood of voxels $v_j$
                      Add the interpolated value to $rs_{sti}$ using the trapezoidal integration rule
                Normalize $rs_{sti}$ by the sum of weights
```

Figure 7.1: Forward projection loop of a straightforward CPU implementation.

7.3.1 Acceleration of Forward and Backward Projection

We begin our discussion by writing the projection procedure in form of a typical CPU implementation. Assuming *S* exclusive subsets and *P* projections in total, the pseudo-code for projection is shown in Figure 7.1 (the backprojection is interleaved for each subset, but not shown here). A ray steps across a slice, interpolates the slice at the sample positions (which results in the weights), sums these contributions (and the weights) and finally divides the ray sum by the sum of weights for normalization. We pre-compute this sum of weights in a pre-processing step and store into a texture.

```
(1) For all $VS/4$ volume slices (using the RGBA channels to process 4 slices simultaneously)
    (2) For all $S$ ordered subsets $OS_s$
        (3) For all pixel rays in $OS_s$ (loop parallelized into fragments)
            Fetch ray starting location $\textbf{rayS}(r_x, r_y)$ and direction $\textbf{rayV}(r_z, r_w)$ from ray texture $TX_{ray}$
            Calculate the entry and exit points on the volume bounding box: $s_1, s_2$
            Set parametric variable $t$=0 and $raySum$=0;
            For all ray positions along $\textbf{rayV}$
                Calculate the sampling location $\textbf{s}$ along the ray and interpolate: $\textbf{s} = \textbf{rayS} + t*\textbf{rayV}$
                Interpolate sample value and add to ray sum: $raySum$ += Interpolate ($volSlice$, $\textbf{s}$)
                Increment $t$ by parametric step size $\Delta t$: $t$+=$\Delta t$ // typically $\Delta t$=0.5
            Store rendering result in rendering texture $TX_{sim}$
```

Figure 7.2: Pseudo code for sinogram-based forward projection. The first two grey lines are executed on the CPU, while the remainder is GPU-resident fragment code. Note that a fragment in this code is the equivalent of a CUDA thread.

From the code in Figure 7.1 we observe that (i) the projection procedure has 5 nested loops (indicated in blue), and (ii) the body of the final level is the longest in terms of operations. The implementation of Castano-Diez et al. maps this loop structure directly to the GPU. The body of loop (4) as well as loop (5) and its body are executed in the fragment shader, while the head of loop (4) itself is parallelized by generating a raster of fragments, one for each loop instantiation. A polygon of size T×1 is created (where T is the number of pixels within a projection) and this polygon is rasterized to the screen. Since each volume slice is processed separately, the projection data is just a set of 1D lines drawn from the set of 2D projections. This process generates one fragment per pixel and for each pixel the fragment program is executed. Given this decomposition, executing all instantiations of loop (4) then encompasses a single parallel operation (called *pass* in GLSL), and therefore one gets $VS{\cdot}P$ such passes. Furthermore, due to then-existing limits on the number of instructions that could be executed in one kernel, Castano-Diez et al. were forced to break each volume slice into *TL* tiles and computed the rays sum for each tile in a separate pass, adding the results in the end. Thus the final number of passes became even higher, that is, $VS{\cdot}P{\cdot}TL$.

Assuming SIRT and P=85, VS=1024 slices, and TL=4 tiles, this would then result in 348k passes, causing significant overhead.

This early implementation does not promote the rules set forward at the end of Section 7.2.2 which hampers performance and also scalability. There is only little parallelism, there are only few threads per pass, and the threads themselves are short. We have improved on this as follows, referring to the pseudo code given in Figure 7.2:

- **Minimize synchronization overhead:** We do not subdivide the domain into tiles, but trace all rays from entry to exit. Optionally, we pre-compute for each ray its starting locations (r_x, r_y,) at the slice boundary as well as its direction vector (r_{dx}, r_{dy}) and store these four values into a ray texture TX_{ray}.
- **Encourage latency hiding:** We launch all rays in a subset at the same time. Thus the number of threads can be controlled by the subset configuration. It is this flexibility that makes our OS-SIRT scheme so attractive and powerful for high-performance GPU computing. For this, we group all $|OS|$ 1D projections in a subset (corresponding to a certain volume slice) into a single 2D sinogram texture TX_{proj}. Then, during projection, we create a polygon of size $T\times|OS|$, and use TX_{sim} as a rendering target. This generates rays/fragments for all angles and pixels in the currently processed subset, and eliminates loop (3).
- **Exploit RGBA channel parallelism:** For this to work, all fragments in these parallel channels must exhibit the exact same mapping function – all that can be different are the data and the rendering target, with each such simultaneous pair being stored in the RBGA channels. Such a strong parallelism is readily exposed in parallel projection, and we can achieve it by storing and processing a consecutive 4-tuple of volume slices and associated projection data in the RGBA channels of their corresponding textures. This reduces the number of required passes theoretically by a factor of 4, but in practice this factor is about 3. GLSL provides a better interface than CUDA for accessing these functionalities since RGBA color is typically used for graphics rendering.

All put together, we can reduce the number of passes required for one iteration to $VS/4\cdot S$. For example, assuming classic SIRT with S=1 and VS=1024 slices as before, we would have 256 passes (if less passes are desired we could also combine equivalent rays in multiple slices). This is less than 0.1% of the implementation of Castano-Diez et al., which has a significant impact on reconstruction performance.

Equivalent to the projection code, Figure 7.3 lists the pseudo fragment code for back-projection. Similar to Castano-Diez.et al., the final two loops of the above pseudo codes are explicitly controlled and executed on the GPU and are rendered in a single pass. However, in addition, we also exploit the RGBA 4-way parallelism, reducing the total number of required passes to *VS/4 S*.

```
(1) For all $VS/4$ volume slices $VS_s$
        (2) For all $S$ ordered subsets $OS_s$
                (3) For all voxels $v$ in $VS_s$
                        (4) For all projections $P_{st}$ in $OS_s$
                                Transform voxel position ($\mathbf{v_x}$, $\mathbf{v_y}$) to sinogram position ($\mathbf{p_x}$, $\mathbf{p_y}$);
                        Sample the sinogram texture using coordinates ($\mathbf{p_x}$, $\mathbf{p_y}$);
                        Add the sample value to current voxel value;
```

Figure 7.3: Pseudo code for sinogram-based back-projection. The first two grey lines are executed on CPU, while the remainder is GPU-resident fragment code.

Note that the major difference of backprojection and forward projection is that in the former the pixel rays are processed in parallel (forming a *pixel-driven* operator), while in the latter the voxels form the threads (yielding a *voxel-driven* operator). This makes both projectors *gathering operations* which are more efficient than *scattering operations* in which every interaction would be a spreading instead of an interpolation. More concretely, a voxel-driven forward projector would have to splat (scatter, distribute) a kernel function onto the detector plane, while a pixel-driven backprojector would have to splat the corrective updates into the reconstruction grid. These already expensive operations would have to be written as kernel code, while interpolation is accelerated in special super-fast hardware circuits. Although this forms an unmatched projector-backprojector pair it has been shown by [140] to work very well in practice.

Finally, since our backprojector uses linear interpolation where the weights always sum to 1.0 for each projected voxel the post-weighting normalization in equation (2) simplifies to a division by $|OS_s|$, which is the number of projections in subset s. Equation (3.2) is then written as:

$$v_j^{(k+1)} = v_j^{(k)} + \lambda \frac{\sum_{p_i \in OS_s} \frac{p_i - r_i}{\sum_{l=1}^{N} w_{il}}}{|OS_s|} \qquad r_i = \sum_{l=1}^{N} w_{il} \cdot v_l^{(k)} \tag{7.1}$$

This reduces the need for keeping track of the sum weights in the backprojection and saves on memory and calculations.

7.3.2 Limited Detector Problem Compensation

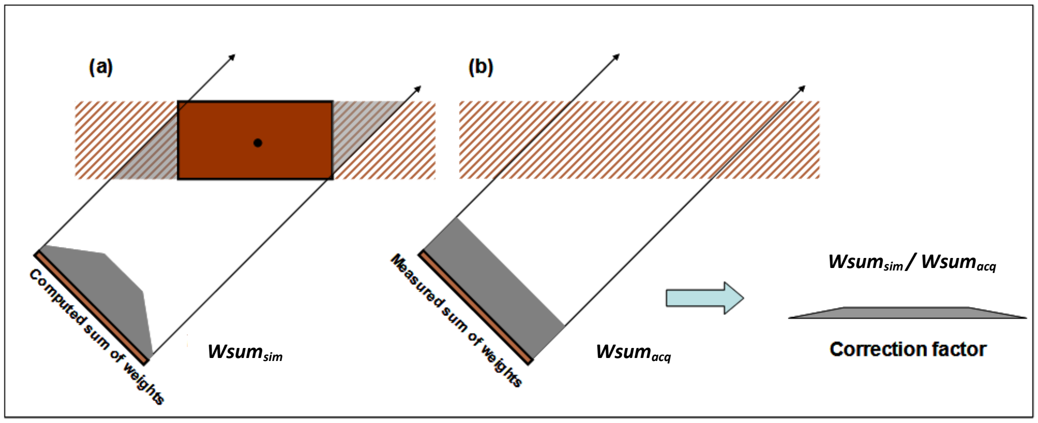


Figure 7.4: Limited detector/long-object problem: (a) the shadowed area indicated regions not reconstructed, but participating in the image formation, (b) area for acquired sum of weights term.

During the data collection stage only a small portion of the sample is imaged to obtain the tilt projections. This results in the “limited detector” or “long-object” problem as discussed in Section 7.1, and an illustration is shown in Figure 7.4a. Here an off-center acquired projection image contains ray integrals across the whole sample, but the simulated projection at the same angle does not have the complete integral since the reconstruction volume must be limited (typically by a box). In other words, voxels residing in the shadow area of the original complete sample (shown shaded in grey) participate in the projection formation during imaging, but due to the restricted reconstruction area (shown in solid red), they do not contribute in the value formation of any pixels during the reconstruction, resulting in severe vignetting effects if we do not compensate for this.

This vignetting effect is shown for three datasets in Figure 7.5a. The top row shows the reconstruction of a long slab of uniform density, while the others show two TEM datasets – a tobacco mosaic virus and the HPcere2 dataset (see below) – all after one iteration with SART. Note that these raw CT slices are vertical cross-sections of the stack of slices typically displayed for visualizing the salient biological structures – we present these more familiar cross-sectional images in Section 7.4. We observe that while the vignetting effects are most prominent for the slices at the top and bottom ends of the stack, all slices are principally affected (see the bow-tie like structures which will cause density fall-off within all cross-sectional slices).

We propose a weight correction scheme that effectively resolves this problem for iterative ET – other compensations exist for analytical algorithms [99] based on Filtered backprojection, where an extended area of around double the length of the region of interest (ROI) is used as the reconstruction target to prevent sampling artifacts. While the over-sampling approach resolves the edge problem, it introduces an extra amount of computation. Our approach does not require these extra computations, as we compensate for the missing target regions on the fly.

In a typical iterative algebraic framework, at a particular tilt angle (see Figure 7.4), the corrective update is derived as:

$$Correction = \frac{P_{acq} - P_{sim}}{Wsum_{sim}} = \frac{P_{acq}}{Wsum_{sim}} - \frac{P_{sim}}{Wsum_{sim}} \tag{7.2}$$

Here the acquired projection is denoted as P_{acq} and the simulated projection as P_{sim}. The problem with using this equation to derive a correction is that the computed sum of weights $Wsum_{sim}$ is calculated based on the bounding box which does not exist (in this closed form) in the acquired data. Therefore, this sum should not be applied towards the acquired projection P_{acq}. Instead, the acquired sum of weights $Wsum_{acq}$ (shown in Figure 7.4b) is the correct value that should be used. Using these arguments, we derive an updated correction equation as follows:

$$Correction = \frac{P_{acq}}{Wsum_{acq}} - \frac{P_{sim}}{Wsum_{sim}} = \frac{P_{acq} \cdot Wsum_{sim} - P_{sim} \cdot Wsum_{acq}}{Wsum_{acq} \cdot Wsum_{sim}} = \frac{P_{acq} \cdot \frac{Wsum_{sim}}{Wsum_{acq}} - P_{sim}}{Wsum_{sim}} \tag{7.3}$$

Consequently, an additional correction factor determined by dividing $Wsum_{sim}$ over $Wsum_{acq}$ should be computed to pre-weight the acquired projection P_{acq} before it participates in the regular correction stage. In practice, we assume that the true extent of the specimen falls within a box extending the box bounding the reconstruction region. The ratio $Wsum_{sim}/Wsum_{acq}$ could then be obtained as the ratio of the length of the parallel rays clipped to the reconstruction region's bounding box and the length of these rays fully intersecting the extended box at the given tilt angle. The latter can be computed by $L=d/cos(\alpha)$ where d is the thickness of the specimen (typically the number of voxels in that dimension assuming unit cell size) and α is the tilt angle. While using a metric ray length L to normalize P_{acq} has been already described in [60][45], these authors only used this formulation as a measure more accurate than a sum of discrete weights W. They did not differentiate the sum of weights to be used for weighting P_{acq} and P_{sim} within a limited detector scenario.

To reduce computational overhead we pre-compute L for each tilt angle and store its reciprocal into a constant texture. The ratio for a given ray is then computed by multiplying this constant by the ray's actual sum of weights $Wsum_{sim}$ obtained from the weight sum texture (see Section 7.3.1). Alternatively we may also pre-compute and store the ratios themselves as a (constant) texture.

The reconstruction results (again after one iteration with SART) presented in Figure 7.5b show that this approximation is reasonably accurate, and we find that by applying the new correction the strong vignetting artifacts present in Figure 7.6a are effectively removed.

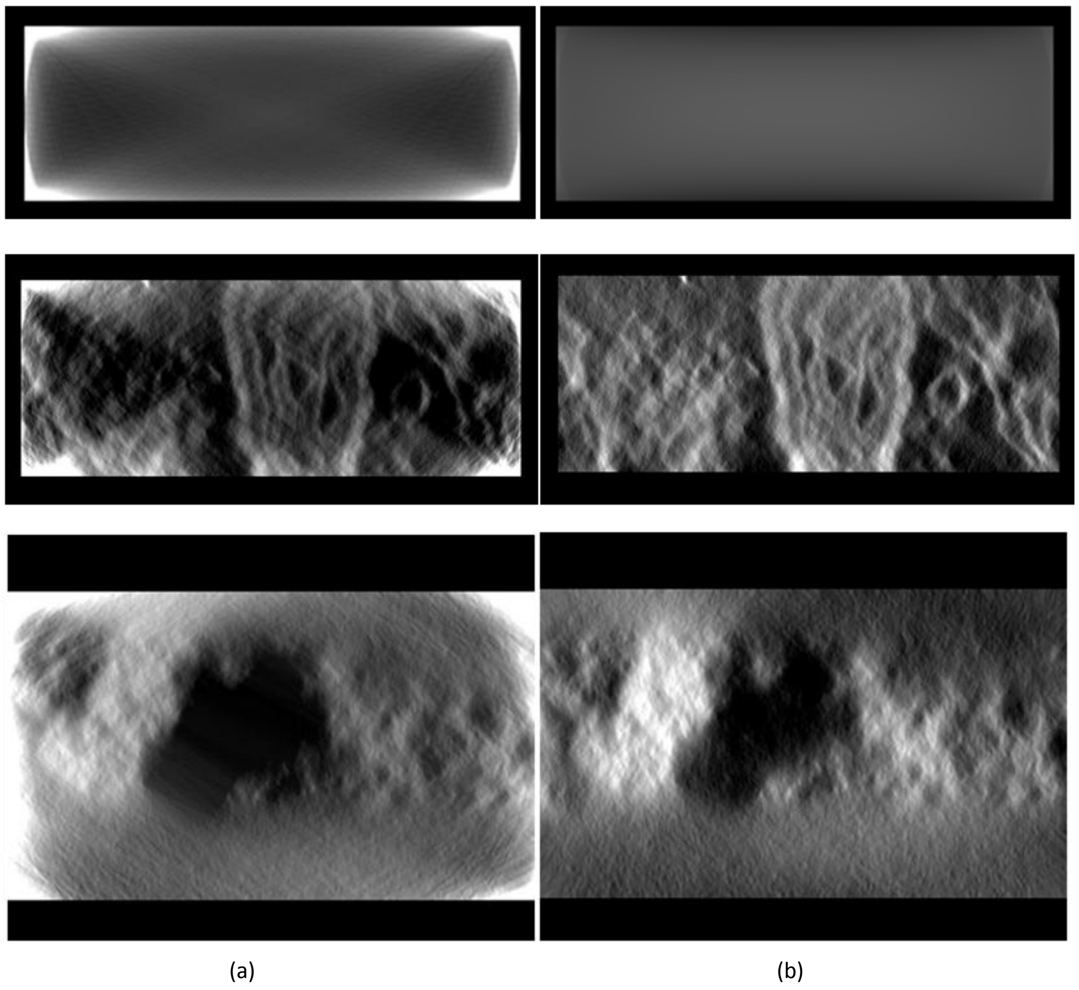


Figure 7.5: Limited detector effect: (a) without compensation, and (b) with compensation during iterative reconstruction. Top to bottom row: a uniform slab, the HPFcere2 dataset, and the tobacco mosaic virus.

7.4 Results

We have experimented with two TEM datasets: (1) a cryo dataset of a frozen hydrated tobacco mosaic virus, comprised of 61 projections of size 680×800 each and obtained at uniform spacing over a tilt angle of around 120˚ and (2) a brain synapse (the HPFcere2 dataset from the Cell Centered Database http://www.ccdb.ucsd.edu), comprised of 61 projections of size 2,242×3,340 each and obtained at uniform spacing over a tilt angle of 120˚ (these are double tilt data, but we only used a single tilt). In order to align all projections, we cropped them to size 1,424×2,024. Next we show results we have obtained with our GPU-accelerated SIRT, SART, and OS-SIRT algorithm. We first show reconstructions we obtained, then present the timing results, and end with further results on our limited-detector artifact compensation scheme.

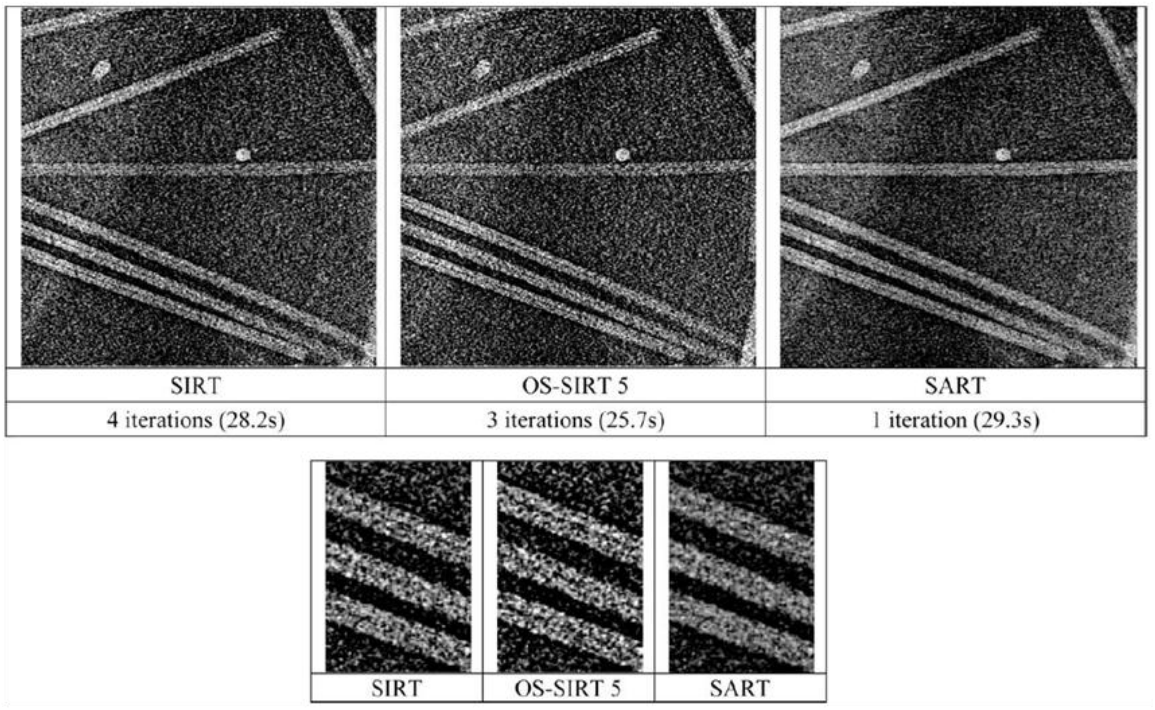


Figure 7.6: (Top row) Reconstruction results for a tobacco mosaic virus dataset using the extreme OS configurations (SIRT and SART) and OS-SIRT 5, all taking about 30s to reconstruct (intensity windowing was applied in each to boost contrast). (Bottom row) Zoom into a specific detail (again with intensity windowing). The number of projections was 61, the tilt angle 120˚, the volume size 680 ×800 ×100, and λ was set to 1.0 for SIRT, 0.5 for OS-SIRT 5, and 0.03 for SART.

7.4.1 Reconstruction Quality, in the Context of Computational Performance

For OS-SIRT we experimentally determined that OS-SIRT 5 gave the best reconstruction quality in the shortest wall clock time for the TEM datasets we tested. We used λ=1 for SIRT in all cases. For SART we used a fixed λ=0.3 for the brain synapse dataset and λ=0.05 for the noisier tobacco mosaic virus. Figure 7.6 displays reconstruction results for the tobacco mosaic virus, with all 800 slices (resolution 680×100) reconstructed via the two extreme OS configurations (SIRT and SART) and OS-SIRT 5. Here we found, similar to Dietz et al. that a single iteration was sufficient for SART to converge and that SIRT will eventually reach convergence as well with similar results, but requires many more iterations (50 or more). The top row shows the full slice view of (a linear intensity window was applied to maximize contrast) of the reconstructed volume, and the bottom row shows a detail view of the same slice (with intensity windowing). All reconstructions were obtained at the same wall-clock time of 30s, matching the time required for one iteration with SART (using the more versatile single-channel implementation, see below). It appears that OS-SIRT 5 provides somewhat better detail and feature contrast than both SART and SIRT – the tube channels seem better preserved for OS-SIRT 5. Figure 7.7 shows a similar series for the HPFcere2 dataset (with 506 slices at 356×148 resolution) obtained at a wall clock time of about 22s, with similar observations. SIRT produces significantly less converged (blurrier) images at this wall clock time. They improve (sharpen) if further iterations are allowed, which we study in Figure 7.8. There we see that 20 iterations appear to be sufficient to match the reconstruction result obtained with OS-SIRT 5 (but at more than triple the time).

Figure 7.9 compares the three different algorithms (SIRT, OS-SIRT 5, and SIRT) quantitatively using the R-factor. Although the R-factor still improves beyond the images we have shown here, we found that these improvements are not well perceived visually and so we have not shown these images here. To visualize the aspect of computational performance in the context of a quantitative reconstruction quality measure (here, the R-factor) we have inserted into Figure 7.9 a metric which we call the "computation time iso-contour". Here we used 22s for this contour – the approximate time required for 1 SART, 6 OS-SIRT, and 8 SIRT iterations (see Figure 7.7) which yielded reconstructions of good quality for OS-SIRT 5. We observe that OS-SIRT 5 offers a better time-quality performance than SART, and this is also true for other such time iso-contours (although not shown here) since the time per iteration for OS-SIRT is roughly 1/6 of that for SART. SIRT, on the other hand converges at a much higher R-factor.

Finally, Figure 7.10 presents detail results obtained from higher resolution reconstructions with OS-SIRT 5 (of the HPFcere2 dataset). We confirm that crisper detail can be obtained with higher resolution, and we will present the time overhead required in the following section.

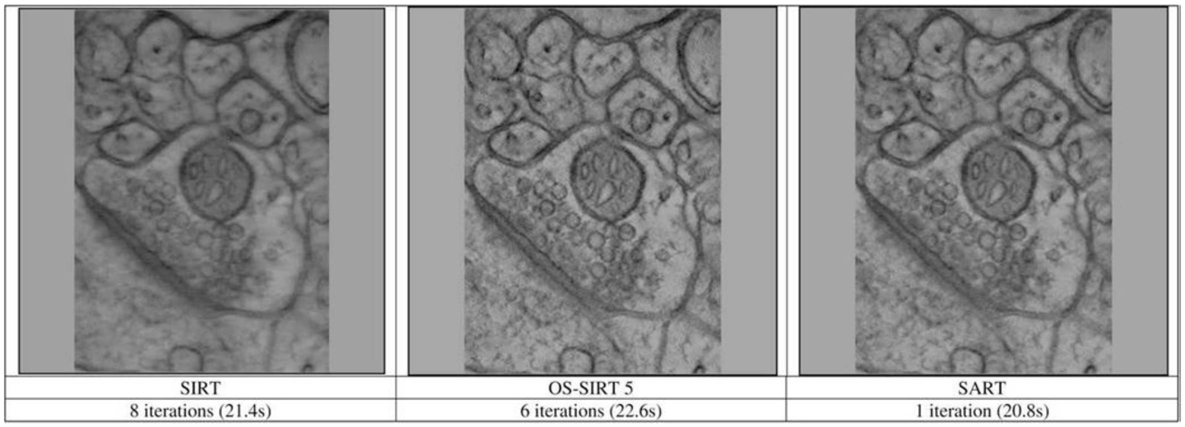


Figure 7.7: Reconstruction results for the HPFcere2 dataset using the extreme OS configurations (SIRT and SART) and OS-SIRT 5, all taking about 22s to reconstruct (intensity windowing was applied in each to boost contrast). The number of projections was 61, the tilt angle 120˚, volume size 356 ×506 ×148, and λ was set to 1.0 for SIRT, 1.0 for OS-SIRT 5, and 0.3 for SART.

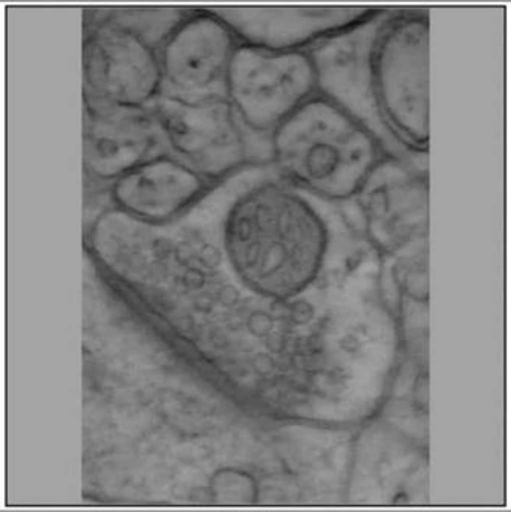
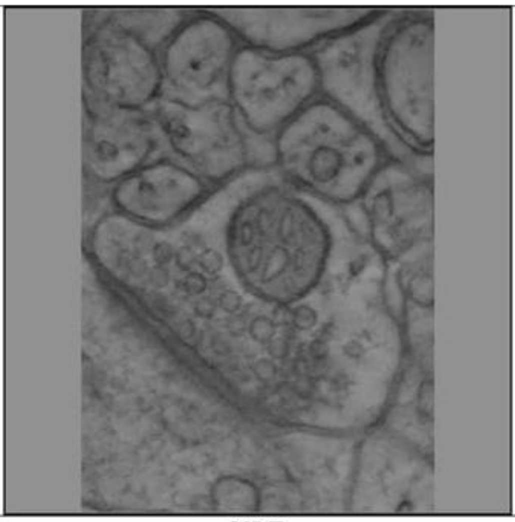
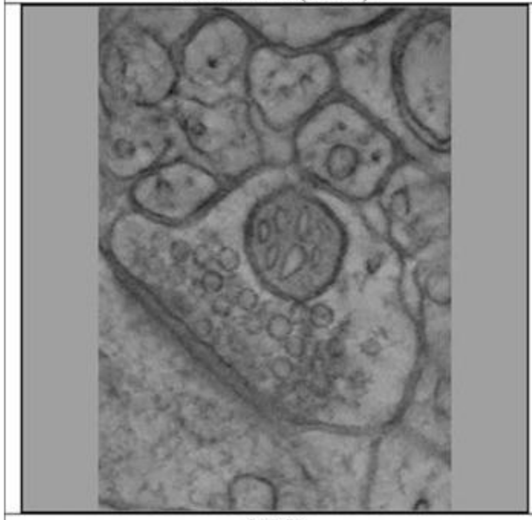
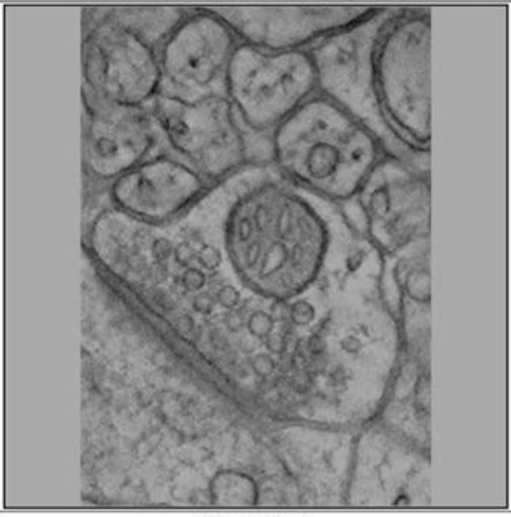
	
SIRT	SIRT
8 iterations (21.4s)	14 iterations (36.3s)
	
SIRT	OS-SIRT 5
20 iterations (51.3s)	6 iterations (22.6s)

Figure 7.8: Reconstruction results for the HPFcere2 dataset comparing the results obtained with extended iterations with SIRT and OS-SIRT 5 (intensity windowing was applied in each to boost contrast). The number of projections was 61, the tilt angle 120˚ and the volume size 356 ×506×148. We observe by visual inspection that 20 iterations with SIRT yield similar results than 6 iterations with OS-SIRT, but at double the wall-clock time. The relaxation factor λ was set to 1.0 for both SIRT and OS-SIRT 5.

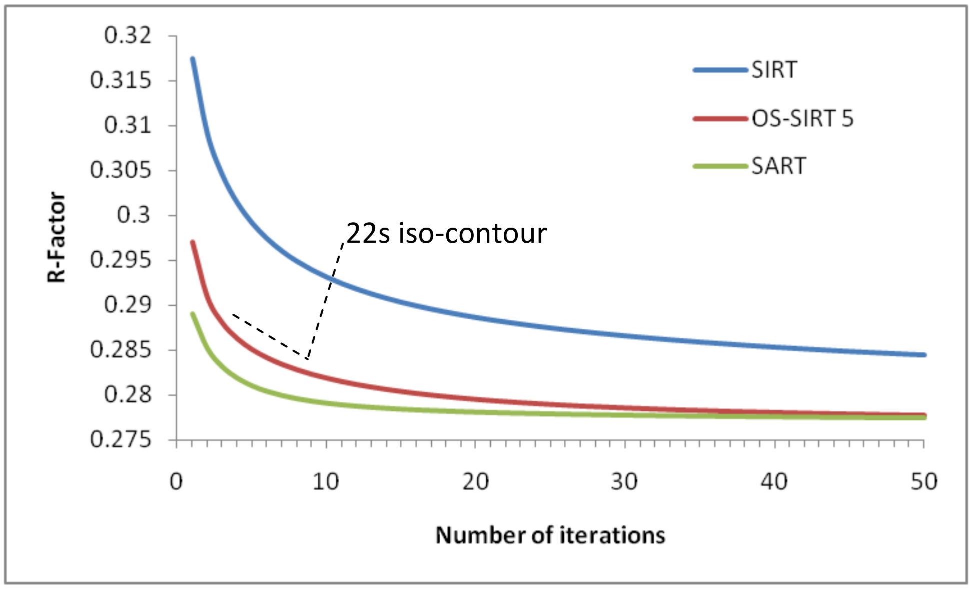


Figure 7.9: Comparison of SART, OS-SIRT 5, and SIRT in terms of quality (R-factor) and performance (22s iso-contour) for the HPFcere2 dataset.

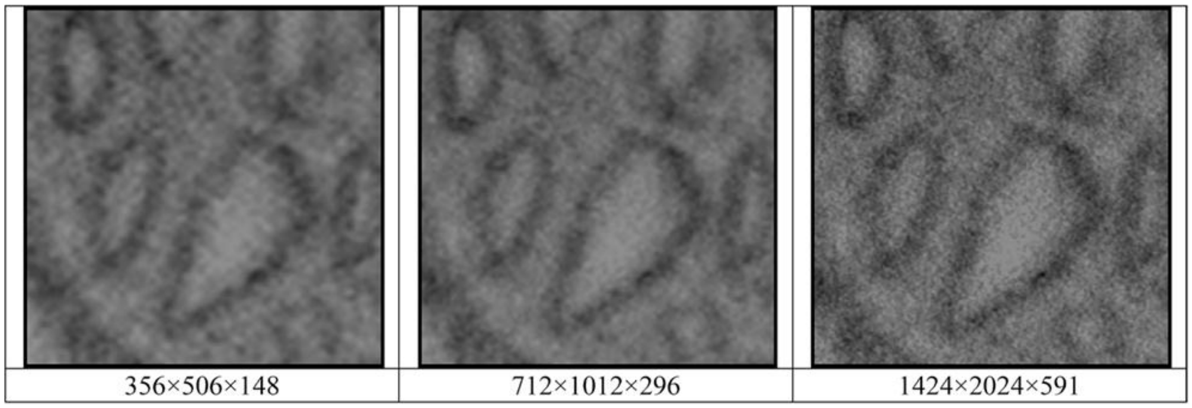


Figure 7.10: Reconstruction results for the HPFcere2 dataset using 8 iterations with OS-SIRT 5 and for increasing resolution. The number of projections was 61, the tilt angle 120˚, but the data used was at matching resolution. The relaxation factor λ was set to 1.0 for all reconstructions.

7.4.2 Absolute Computational Performance

Number of iterations	Slice resolution	(Diez et al.) Quadro 4500	7800GTX	Speedup	GTX 280	Speedup
10	256×256	N/A	0.41s	N/A	0.13s	N/A
50	256×256	N/A	2.05s	N/A	0.66s	N/A
10	512×512	9s	1.23s	7.3	0.34s	26.0
50	512×512	39s	5.32s	7.3	1.75s	22.2
10	1024×1024	32s	4.30s	7.5	1.23s	25.8
50	1024×1024	146s	21.89s	6.7	6.44s	22.7
10	2048×2048	123s	17.39s	7.1	5.06s	24.3
50	2048×2048	567s	85.30s	6.7	26.94s	21.0

Table 7.1: Timings for the reconstruction of a single volume slice at different resolutions using SIRT and parallel projections acquired at 180 tilt angles, comparing different implementations and platforms.

We now discuss the general performance of our optimized GPU-accelerated ET-framework. First, to illustrate the benefits of our new projection/backprojection scheme in general, we provide Table 7.1 which compares the running times obtained via our framework with those reported in [28]. The timings presented here refer to a 2D slice reconstruction with SIRT, using projection data from 180 tilt angles, including the time to transfer the data to the GPU. We have run our framework on GPU hardware comparable to the one employed by Castano-Diez et al., that is, the NVIDIA G70 chip (this chip forms the core of both the Quadro 4500 and the GeForce 7800 GTX boards, with only minor performance differences). Since then, newer generations of NVIDIA chips have emerged, with the latest being the G200 chip (available as the GTX 280 board) for which we also report timings. More detail on these two architectures was already presented in Section 7.2.2. We observe that the significant decrease in passes of our GPU-algorithm leads to consistent speedups of nearly an order of magnitude across all resolutions and iteration numbers (7800 GTX columns). The newer platforms yield further speedups mainly founded in hardware improvements (GTX 280 columns).

Table 7.2 studies the performance per iteration for SART, SIRT, and OS-SIRT 5 for different volume sizes/resolutions (assuming the same number of projections). In this table we also compare the timings obtained for the more versatile single-channel implementation with the RGBA (4) channel solution implemented with GLSL. We make two observations. First, we see that the 4-channel scheme pays off more as the number of subsets increases, with SART being the extreme case where it can achieve speedups between 2 and 3. Second, we observe that the performance gap of SIRT, OS-SIRT, and SART narrows (but more so for the 4-channel implementation) with increasing volume size, with eventually SIRT being only 1.5 times as fast than SART (for one iteration) in the 4-channel configuration. Since in ET practice the volume slice resolution tends to be at the order of 2k to 4k and larger, this means that the choice of subsets will be mainly determined by the traditional tradeoff between speed of convergence and noise cancelation. Figure 7.11 shows similar trends with a more standardized metric, the number of voxels (multiplied by the number of projections in the subset) processed per second. This yields a metric for the overall task complexity measured in gigavoxels/s.

Volume resolution	SIRT		OS-SIRT 5		SART	
	1-ch	4-ch	1-ch	4-ch	1-ch	4-ch
356×506×148	2.642	2.103	3.672	2.278	15.867	5.712
712×1012×296	12.822	10.665	14.625	11.053	47.993	19.341
1424×2024×591	75.708	58.471	82.055	62.135	192.337	87.042

Table 7.2: Running time for 1 iteration for the reconstruction of volumes of different sizes (resolutions) using SIRT, OS-SIRT 5 and SART, with the 1-channel and 4-channels schemes, and parallel projections acquired at 61 tilt angles.

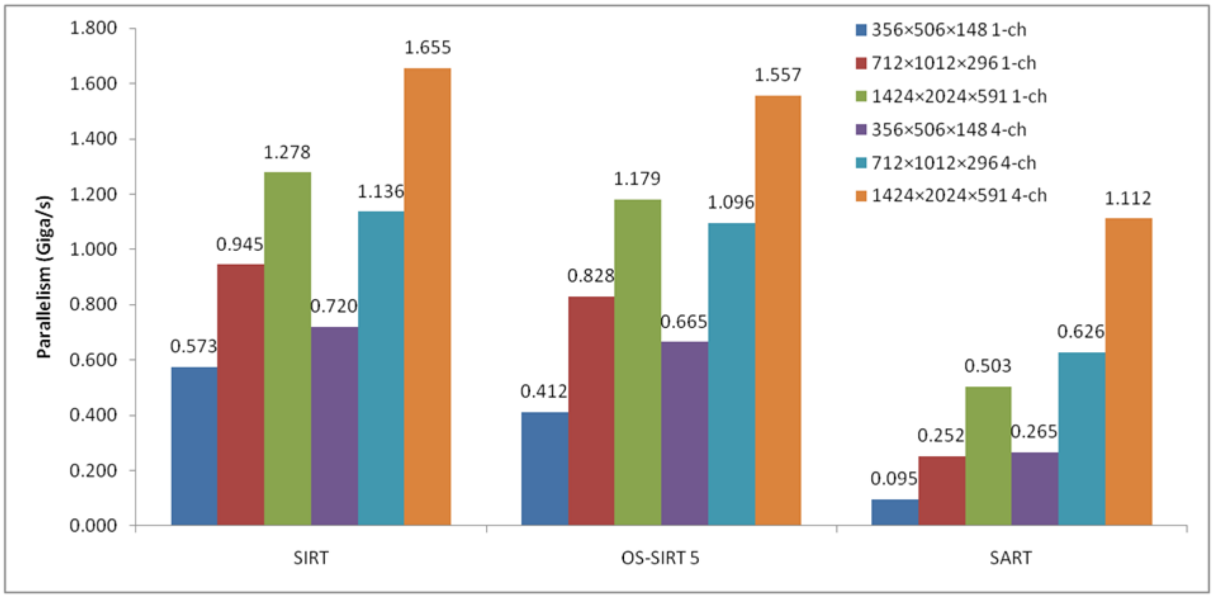


Figure 7.11: Reconstruction performance expressed in Gigavoxels/s for different volume sizes, algorithms and the 1-channel and 4-channel scheme.

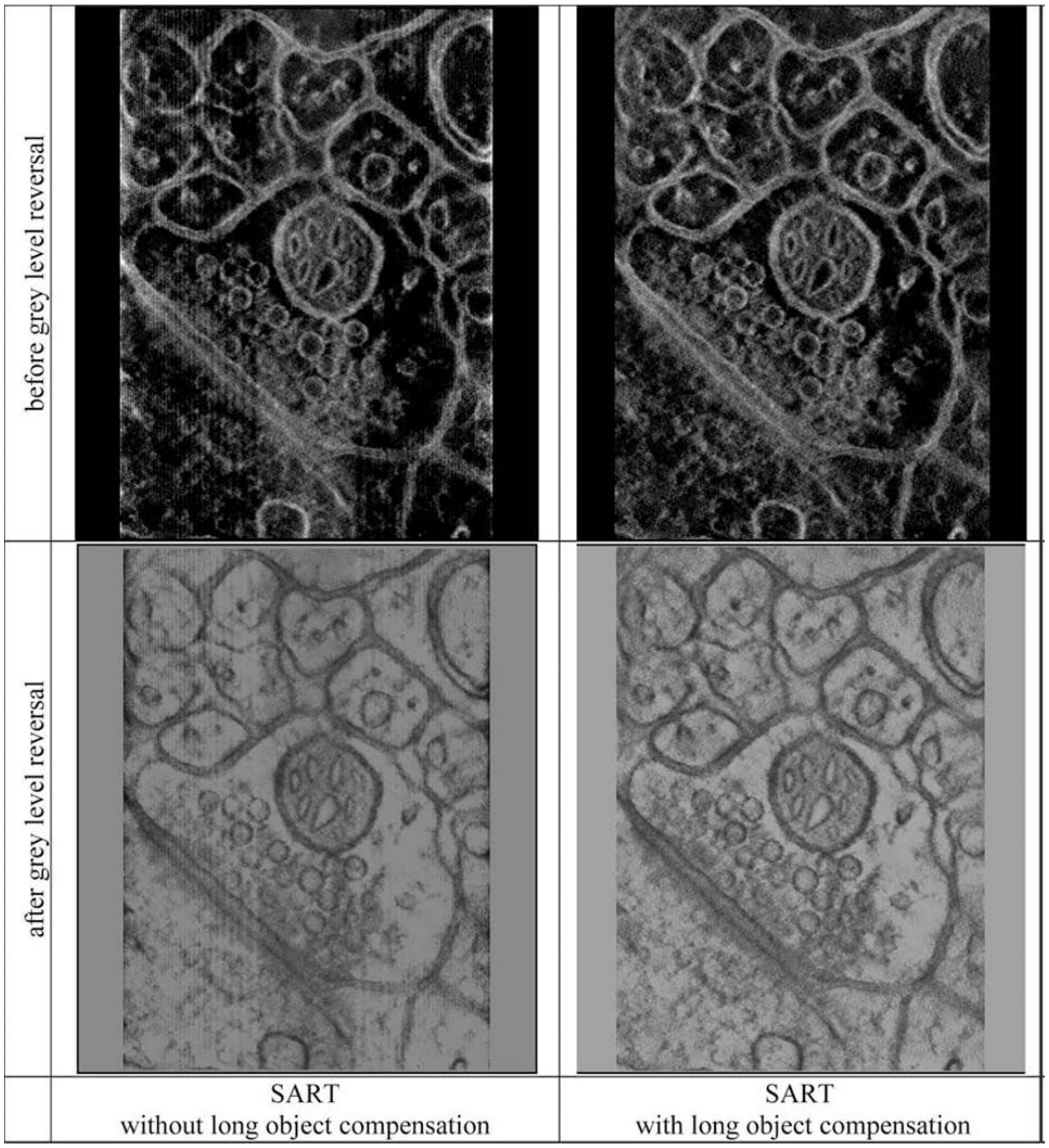

before grey level reversal
after grey level reversal
SART
without long object compensation
SART
with long object compensation

Figure 7.12: Comparing reconstruction results obtained for the HPFcere2 dataset without and with limited detector compensation for 1 iteration with SART (at the same settings than for Fig. 7.7).

7.4.3 Limited Detector Problem Compensation

Our final results present the impact of our compensation scheme on reconstruction quality. Section 7.3.2 has already shown that artifacts are effectively removed in the reconstructed thick specimen slab (which is a portion of a much wider and longer sheet). We have also mentioned that in most cases this slab is re-sliced orthogonally and then the grey level densities reversed and windowed, which yields the images typically visualized and also shown in our Figures 7.8-7.10. In Figure 7.12 we present the center slices of the re-sliced slabs of Figure 7.5 (center row, the HPFcere2 dataset) before and after grey-level reversal (top and bottom row, respectively) and without and with compensation (left and right column, respectively) after one iteration with SART. We observe, in the uncompensated images, the peripheral density fall-offs at the left and right edge and we also observe a slight vertical stripe artifact due to the bow-tie border. Both artifacts are effectively removed with our compensation scheme.

We have described new contributions and presented confirming results for these within three major areas of 3D Electron Tomography (ET): (i) the iterative reconstruction framework using algebraic methods in different subset configuration schemes, (ii) the compensation for the limited angle at which projections can be obtained, and (ii) the acceleration of ET via commodity graphics hardware (GPUs). For the latter, we have presented a novel data decomposition scheme that minimizes the number of GPU passes required, yielding speedups of nearly an order of magnitude with respect to present GPU-acceleration efforts. We also compared acceleration with a versatile single-channel scheme that is available with any GPU API with a 4-channel scheme (currently) only available with graphics APIs, such as GLSL, which offers additional speedups of up to 2 for large practical datasets (more for smaller datasets). Our GPU-accelerated framework allows full-size 3D ET reconstructions to be performed on the order of minutes, using hardware widely available for less than \$500.

7.5 Conclusions

Our GPU-accelerated ET platform allows ET researchers to achieve a major task which has so far been infeasible without expensive and extensive hardware: the iterative reconstruction of full-size 3D volumes. We have shown that it is now possible to reconstruct a $2048^2 \times 100$ volume within a few minutes, while Castano-Diez et al. report nearly an hour or more for this task. A CPU-based reconstruction would take on the order of days. We emphasize that all of our results were obtained with a single GPU solution (of a cost of less than \$500) – a multi-GPU configuration would provide even higher performance. We believe that the impact of gaining such capabilities is great, as it enables demanding iterative schemes crucial for the improvement of image resolution and contrast, such as iterative projection alignment and registration, further efforts could be planned in this direction.

Current work is directed towards determining a framework that can automatically optimize the number of subsets and determine the best relaxation factor λ for a given imaging scenario, as expressed in SNR, imaged specimen, and imaging platform. We have already obtained encouraging initial results in this direction, as recently reported in [119].

Chapter 8

Low-Dose CT Artifact Mitigation Using a Prior Scan

Low-dose CT has attracted increasing attention due to growing concerns about radiation exposure in medical scans. Although we have shown in previous chapters that iterative CT reconstruction algorithms with regularization abate the difficulty of reducing the artifacts such as noise and streaks, the frugal use of X-ray radiation in some extreme situation still makes the reconstructed images difficult to read in clinical routine. For follow-up CT exams a prior scan is often available. It typically contains the same anatomical structures, just somewhat deformed and not aligned. This work describes a two-step technique that utilizes this prior scan to achieve high-quality low-dose CT imaging, overcoming difficulties arising from noise artifacts and misalignment. We specifically focus on reducing the dose by lowering the number of projections. This gives rise to severe streak artifacts which possibly lower the readability of CT images to a larger extent than the fine-grained noise that results from lowering the mA or kV settings.

8.1 Introduction

Computed X-ray Tomography has revolutionized modern medicine and thanks to the rapid growth in scanner technology the gamut of its applications has risen at an enormous rate. In this process, buoyed by the excitement of possibilities little attention was paid to the radiation dose administered to the patient. Scans with ever-improving spatial and temporal resolutions were conducted on a routine basis and the associated CT reconstruction algorithms had the luxury of an abundance of data collected at each exam. It was only recently that the sobering results of long-term studies on the adverse radiation effects of CT imaging have dampened these developments [7][77]. Due to these studies, the harmful effects of X-ray radiation in CT scans have become publicly heard, threatening the future of this modality. To counter these concerns, campaigns such as *ImageGently* (http://www.imagegently.org) and *ImageWisely* (http://www.imagewisely.org) have been initiated that promote the optimization of the radiation dose used in both pediatric and adult medical imaging.

To reduce the radiation dose subjected to the patient one can: (1) lower the number of scans, (2) lower the number of X-ray projections per scan, and (3) lower the energy settings of the X-ray tube (kV, mA) per projection image. The first measure, i.e., reducing the number of scans, is often left at the discretion of the treating physician. The latter two options are highly detrimental to image quality, resulting in images with significant noise artifacts. They greatly challenge the conventional CT reconstruction algorithms based on analytical formulations rooted in the Inverse Radon Transform [81], and these shortcomings have recently invigorated research efforts towards methods that seek alternatives to these conventional schemes.

A popular approach to this end has been to enforce data fidelity and image quality as a joint optimization problem and solve these two parts in an iterative round-robin fashion as we stated before as the interleaved scheme. Data fidelity can be assured by ways of any CT reconstruction algorithm, iterative or analytical, but most use the former. The reduction of noise artifacts, on the other hand, can be posed as an image denoising problem. Many approaches use the method of Total Variation Minimization (TVM) [92] for this task since it is often part of general compressive sensing formulations [20] that were originally prescribed to deal with sparse data. For CT reconstruction, a number of sophisticated schemes have been developed that adapt the various parameters used in the process, such as ASD-POCS [101] and soft-threshold filtering [134].

In this work, we have attempted to devise a framework that executes the data fidelity step and the image quality step each exactly once. It is hence of lower computational complexity than the present schemes which perform these steps iteratively. We achieve this by making creative use of an artifact-free prior – constituted by an existing regular-dose scan of the patient. Such a clean prior scan is frequently available. For example, it may be a regular-dose first scan acquired before a low-dose follow-up scan or it may be a regular-dose diagnostic scan preceding a low-dose setup scan for a surgical intervention such as orthopedic spine fixation, among other scenarios.

In the present work, we focus on the second form of low-dose CT, i.e., reducing the number of X-ray projections per scan. The first step of our framework uses filtered backprojection (FBP) to reconstruct an image with significant streak artifacts which result from this low number of projections. The use of FBP to provide a quick first estimate is a common strategy. Unlike the first step of an iterative scheme, such as ART [46] and its derivatives [2][45], FBP typically reconstructs all image features at good fidelity but the high image noise makes them difficult to read. Common approaches then follow FBP by an iterative pipeline for denoising. Our second step, on the other hand, uses a single prior-based image restoration that eliminates the noise and so provides the desired viewing experience. In this step, we first register/align our prior with the FBP-estimate using an established multi-scale feature registration algorithm, i.e., SIFT flow [72]. Following, we simulate the low-dose streak artifacts of the FBP-estimate in this registered clean prior. Finally, for each pixel in the target image we use a neighborhood similarity metric to determine the best matches in the contaminated prior and then replace it using the corresponding pixels in the clean prior.

Using existing scans to support regularization is not new. Kelm et al. [55] describe an approach that reconstructs volumes at two different thicknesses, using the same acquired projection data. They reconstruct the thicker slices from bin-averaged projections which increases SNR, while the thinner (and noisier) slices are reconstructed from the original projection data. Since registration is implicit, it is relatively straightforward to use the thicker slices for neighborhood-based denoising of the thinner slices. In contrast, our method applies to settings in which the reference images are not necessarily acquired simultaneously. Yu et al. [135] present the method *Previous Scan–Regularized Reconstruction (PSRR)*. It replaces regions that are unchanged in a low-dose CT reconstruction with their direct embodiments in a normal-dose CT reconstruction, and uses a nonlinear diffusion approach for denoising in the remaining regions. This approach requires an effective strategy for feature recognition, which the authors accomplish via registration.

The approach most similar to ours is that of Ma et al. [75]. They also use a registration algorithm – a combination of rigid PCA (Principal Component Analysis) and non-rigid mutual information optimization [67] – for rough alignment. They then use a neighborhood-mechanism to locate suitable replacement candidates in a prior scan of the patient. Our work differs from theirs in the following important ways. First, while Ma et al. restore images reconstructed from projections generated at reduced mA settings, we treat artifacts caused by the reduction of projections. The resulting streak artifacts are much more severe and irregular than the random noise-artifacts caused by low-mA imaging. Second, instead of using a clean prior for matching we use a prior with simulated artifacts. We find that this affords much better accuracy, as we will proof and demonstrate. A preliminary version of the framework we describe here has been presented in [124], which predates the work by Ma et al. slightly.

Our work is organized as follows. Section 8.2 presents the algorithms we have studied, Section 8.3 presents results, and Section 8.3 ends with conclusions and points to future work.

8.2 Methods and Materials

To locate good pixel matches in the prior we use the similarity measures also employed by Non-Local Means (NLM) filtering [4]. NLM filtering can be seen as a generalization of Gaussian smoothing. It looks for structurally similar pixel neighborhoods in the smoothing site's proximity and includes them into the Gaussian filter statistics. This leads to a more robust estimate of the true pixel value and consequently to improved image restoration/denoising results. We have recently informally compared NLM filtering with TVM for low-dose imaging tasks and our results have been quite encouraging [122]. In this current work, we do not employ NLM-filtering in a conventional way as a non-local extension of Gaussian filtering. Rather, we only retain NLM's mechanism for similarity-based pixel neighborhood matching in the artifact-matched prior.

8.2.1 Standard NLM Filtering

The NLM algorithm was proposed by Buades et al. [4] for image Denoising as mentioned in chapter 4. It takes advantage of the high degree of redundancy that typically exists in an image. Given a target pixel subject to denoising, it defines a small Gaussian-weighted region around it, called a *patch*. It then searches the entire image for similar patches and accumulated them weighted by their degree of similarity. In practice, only a local neighborhood around the target pixel is searched, called *search window*. This helps performance but it also better tolerates non-stationary noise processes. Further, since we first register the prior to the target image, we do not require large search windows in any case. More formally, the updated value p_x' of a target pixel is computed as:

$$p_x' = \frac{\sum_{y \in W_x} \exp(-\sum_{t \in P} G_a(t) \left| p_{x+t} - p_{y+t} \right|^2 / h^2) \cdot p_y}{\sum_{y \in W_x} \exp(-\sum_{t \in P} G_a(t) \left| p_{x+t} - p_{y+t} \right|^2 / h^2)} \tag{8.1}$$

Here, *x* is the location of the target pixel and the *y* are the locations of the

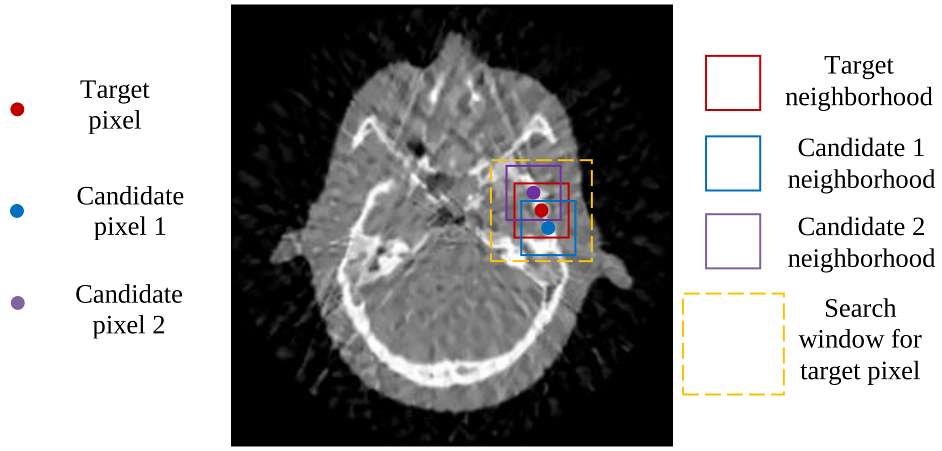
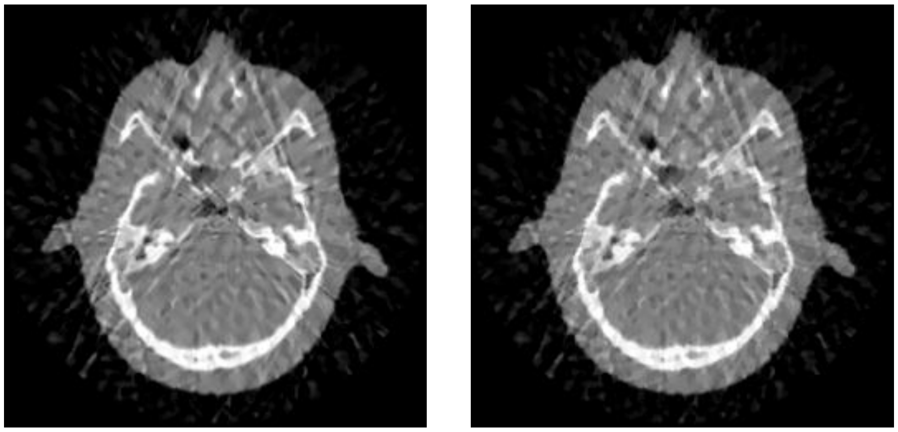


Figure 8.1: Illustration of conventional NLM for denoising.



(a) NLM-filtered (b) TVM-filtered

Figure 8.2: Filtering results obtained with (a) NLM and (b) TVM.

candidate pixels, with values p_y. W_x is the search window around x, and P is the patch size of each pixel. The patch similarity is measured by the Gaussian weighted L_2 distance between two patch vectors with t representing the index within a patch and G_a being a Gaussian kernel with standard deviation a. The exponential function converts these distances to weights, determined by a parameter h which controls the overall smoothness of the filtering. Larger values of h will result in more smoothing.

Figure 8.1 shows an illustration of this process when NLM is used in a conventional way to denoise a reconstruction with severe streak artifacts. In this particular case we used FBP to reconstruct a GE head phantom from 45 fan-beam projections acquired over 360°. The illustration shows the search window, the target pixel in the center and two candidate pixels with similar neighborhoods as the target pixel. Figure 8.2a shows the NLM-filtered result, while Figure 8.2b shows the results obtained with TVM. We can observe that both of these filters provide some amount of improvement over the original image shown in Figure 8.5a. It is likely (see for example, [9]) that repeating the fidelity and denoising steps several times would do substantially better, but since we have aimed for a non-repeating approach – one in which a prior is available to aid in the denoising – we focus on a single step scheme.

8.2.2 Registration Using the SIFT-Flow Algorithm

A crucial element in our prior-assisted framework is proper registration since without it the possibility for mismatches can be high. Nevertheless, our use of the NLM-based matching mechanism relaxes the need for tight and laborious registration of the prior image as it performs the fine registration on the fly via its search mechanism. As such it is less sensitive to spatial distortions than the PSRR approach. For registration we have made use of the SIFT-flow algorithm, recently published by Liu et al. [72]. This algorithm originates from the optical-flow algorithm which produces dense, pixel-to-pixel correspondences between two images. It extends the matching from raw pixels to SIFT feature descriptors [62]. A SIFT (Scale-Invariant Feature Transform) feature descriptor captures the histogram of gradient orientations in a local neighborhood at a given scale. It is well suited to characterize salient local and transform-invariant image structures and at the same time encode contextual information. SIFT-flow has been specifically designed for scene matching, where objects share similar scene characteristics but may have different appearances and locate at different places. This is the case in the registration of the prior to the current scan. They are certainly from the same person, i.e., they share the scene characteristics, but they will likely have different SNR and undergone distortions of the features.

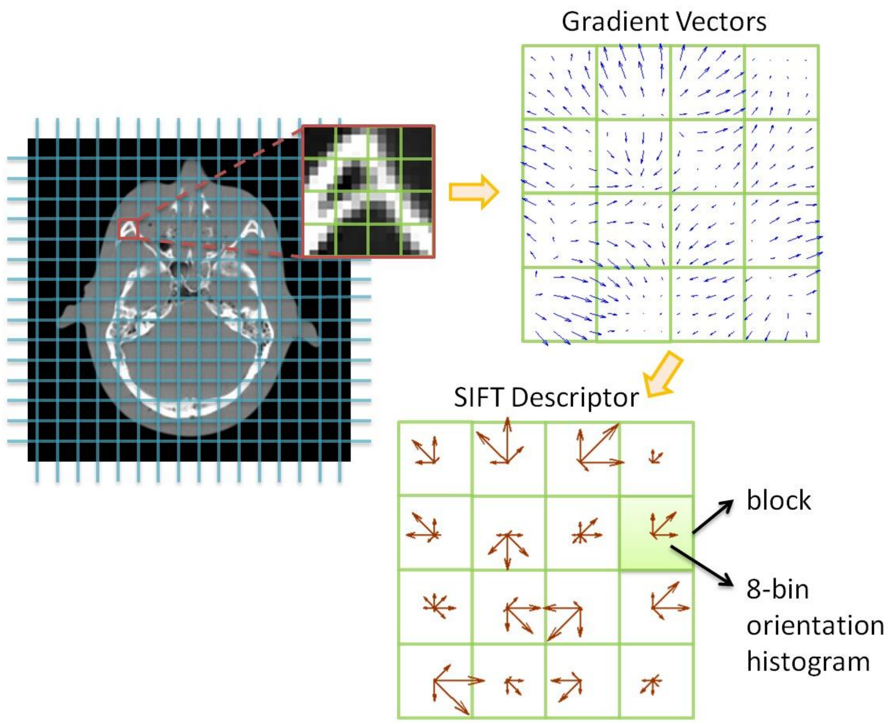


Figure 8.3: Illustration of SIFT descriptor summarizing edge orientations over 16×16 pixel area.

The implementation of SIFT-flow has two parts: (i) generate dense SIFT features where each pixel has a 128-dimensional SIFT vector, and (ii) find the correspondence of these SIFT features via discrete optimization on the image lattice to obtain the displacement field for alignment. For the first part, Figure 8.3 shows a typical SIFT feature descriptor summarizing the gradient orientations in a 16^2 pixel area (as plotted inside the red square). The gradients (shown as blue arrows) are Gaussian-smoothed according to their distance to the area center. This area is partitioned into 4^2 blocks, each of size 4^2 pixels (shown as green squares). The gradient orientations are then accumulated in each block to 8 orientation bins and are weighted by their gradient magnitudes. There are a total of 16 8-bin orientation histograms (with each red arrow representing one bin). Thus the dimension of a SIFT vector is $4\times4\times8$=128 over a 16×16=256 area. For the second part, to estimate flow, the energy function for SIFT flow is defined as below:

$$
\begin{aligned}
E(f(x,y)) = {} & \sum_{(x,y)} \min\left(\left\| s_1(x,y) - s_2(x+f_x(x,y), y+f_y(x,y)) \right\|_1, a\right) \\
& + \sum_{(x,y)} b(|f_x(x,y)| + |f_y(x,y)|) \\
& + \sum_{(x',y') \in W(x,y)} \left(\min(c\,|f_x(x,y) - f_x(x',y')|, d) + \min(c\,|f_y(x,y) - f_y(x',y')|, d) \right)
\end{aligned}
\qquad (8.2)
$$

where (x,y) and (x',y') are pixel locations, s_1 and s_2 are SIFT descriptors, f is the displacement function with f_x in x-direction and f_y in y-direction, W is the pixel neighborhood and a, b, c, d are four thresholds (with default settings). This function is designed according to three constraints: (1) the matched pixel should have similar SIFT descriptors; (2) the displacement should be as small as possible; and (3) adjacent pixels should have similar displacements to maintain flow smoothness. This discrete displacement function can be estimated by optimizing the energy function with a belief propagation algorithm [72]. Its time complexity is $O(h^2 \log h)$ where h is the width (height) of the image. Before registration we smooth the images with a 7×7 Gaussian filter (standard deviation=3). This yields more stable results. Using the Matlab implementation obtained from the author's website [147] it took less than one minute for one registration operation of two 256^2 CT scans on a quad-core Dell XPS 2.66GHz PC with 8GB of memory. In their paper, Liu et al. [72] also point out that a GPU implementation of their belief propagation algorithm could yield a further (up to) 50-time speedup which would bring the time required for the registration down to seconds.

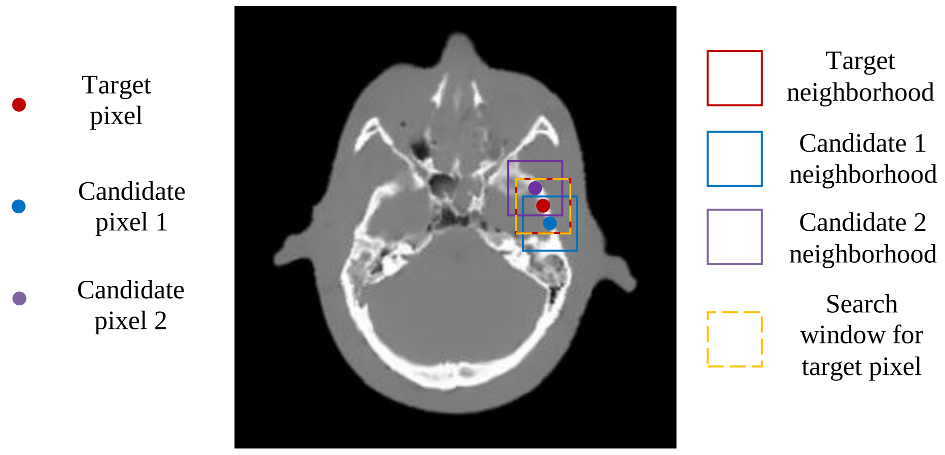


Figure 8.4: Illustration of R-NLM with the registered clean prior used for both matching and retrieval.

8.2.3 Reference-based NLM (R-NLM) Filtering

With the NLM-mechanism still being employed for the matching, we call our approach *Reference-based NLM (R-NLM)* filtering since it uses the prior image as a reference to guide the filtering. Figure 8.4 provides an illustration of this process, now using a noticeable smaller search window than in the regular NLM-case. The larger the search window the more distortion-tolerant the algorithm becomes, but the potential for lost detail and over-smoothing also rises. In experiments we found a size of 7×7 pixels for both search window and patches to represent a good compromise.

Our R-NLM algorithm first uses SIFT-flow to align the prior with the current scan, call it *target scan*. Following, it visits every pixel in the target scan, places the search window in the same location in the prior scan, and uses the NLM-algorithm to determine the update. Figure 8.5 presents some results we have obtained with our R-NLM algorithm, to motivate a further extension discussed in the next section. Figure 8.5f shows a regular-dose reconstruction obtained with 360 projections, while Figure 8.5a shows a low-dose FBP reconstruction obtained from the same data but only 45 projections – about 1/8 of the dose. Figure 8.5b shows the prior. Since this was a head phantom that could not be warped mechanically, we performed a digital warp – in this case a twirl distortion around the center of the image. This deformation field is shown in Figure 8.5h. Figure 8.5c shows the registered prior and Figure 8.5d shows the result obtained with R-NLM filtering. It is clearly better than the NLM and TVM filtered results (see Figure 8.2). This of course is a comparison that is only partially fair because R-NLM had access to a clean prior while the others did not. This procedure changes equation (1) into the following:

$$p'_x = \frac{\sum_{y \in W_x} \exp(-\sum_{t \in P} G_a(t) \left| p_{x+t} - p_{y+t}^{crp} \right|^2 / h^2) \cdot p_y^{crp}}{\sum_{y \in W_x} \exp(-\sum_{t \in P} G_a(t) \left| p_{x+t} - p_{y+t}^{crp} \right|^2 / h^2)} \tag{8.3}$$

Here the superscript *crp* indicates that the pixels originate from the clean registered prior and not from the target. However, when comparing this result with that obtained at regular dose (Figure 8.5f), we still observe some amount of blurring in the image. Edges in general appear less defined, and small features are also weakened or completely suppressed. For the latter, compare for example the intricate detail in the center of the image, to the left of the pincushion-shaped dark structure, which is barely visible in the R-NLM result. In the next section we describe an advanced scheme that overcomes these problems.

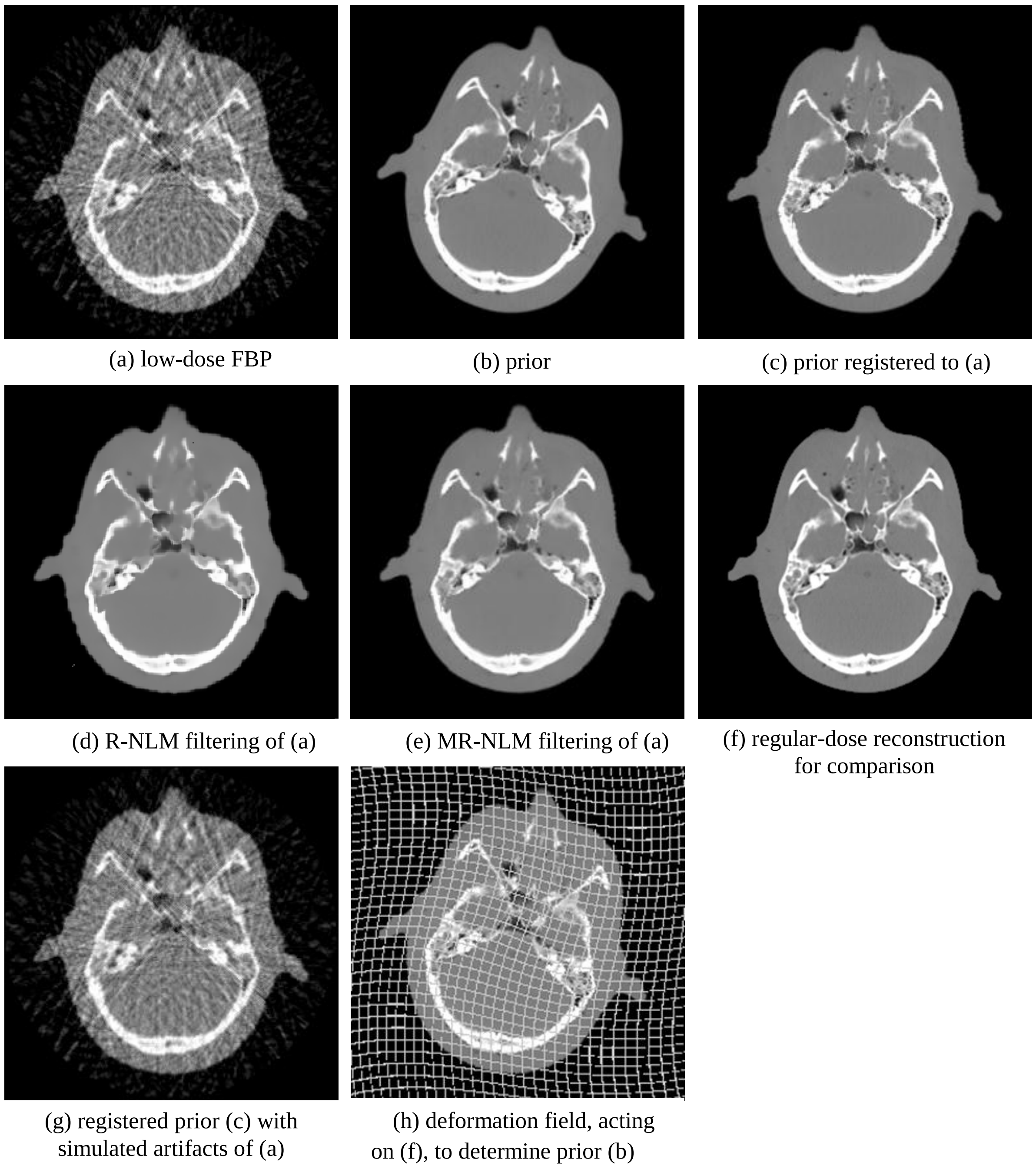

(a) low-dose FBP
(b) prior
(c) prior registered to (a)
(d) R-NLM filtering of (a)
(e) MR-NLM filtering of (a)
(f) regular-dose reconstruction for comparison
(g) registered prior (c) with simulated artifacts of (a)
(h) deformation field, acting on (f), to determine prior (b)

Figure 8.5: Results-supported illustration of reference-based NLM (R-NLM) and matched reference-based NLM (MR-NLM).

Finally, in the event that no reliable update can be found in the prior for a given target pixel, we fall back to conventional NLM-filtering using the target image. We identify this situation by a low sum of weights in equation (3). In our experiments, we have used a threshold of 0.001. We use this criterion both for R-NLM and for the advanced scheme described next.

8.2.4 Matched Reference-based NLM (MR-NLM) Filtering

Since the features are generally weakened at a scale less than the size of the NLM search window, we cannot blame the registration algorithm for these shortcomings. Rather, it is the quality of the NLM-matching that is at the heart of the problem. Consider the NLM-distance function of equations (1) and (3) used to determine the quality of a match for a specific candidate neighborhood (or patch) *P*:

$$\sum_{t \in P} G_a(t) \left| p_{x+t} - p_{y+t} \right|^2 \tag{8.4}$$

Here, $|p_{x+\mathrm{t}} - p_{y+t}|$ is the Euclidian distance of a corresponding pair of pixels parameterized by patch index *t*. The sum of these distances determines the weight that the patch *P* plays in determining the value of the target pixel, and thus it is the patch’s structural similarity that is decisive for the scaling of its contribution. While equations (1) and (3) also have a parameter *h* for scaling, it is a global parameter that scales all patches at the same weight. The difficulties we encounter with R-NLM cannot be solved just by adjusting the factor *h*, as we will demonstrate in Section II.E. Just as the best registration is achieved when the two scenes are similar in appearance, the best NLM-match is obtained when target and prior have a similar appearance. This is not the case when pairing a clean prior and a low-dose reconstruction with severe streak artifacts. Hence, we require a method that transforms the clean registered prior image into an image that bears similar artifacts as the target image. We can achieve this by first simulating projections from the registered prior and then reconstructing it under the same conditions as the target image, i.e., with a lower number of projections and at the same viewing geometry as the target. This gives rise to Figure 8.5g – which as we observe looks fairly close to the target image subject to denoising (see Figure 8.5a).

The MR-NLM reconstruction procedure is illustrated in form of pseudo-code in Figure 8.6. After registering the prior with the low-dose target, a degraded registered prior is created by simulating the low dose artifact also present in the target. The procedure then uses this degraded registered prior for NLM-matching, but copies the corresponding candidate pixels in the clean registered prior to the weighted sum. The resulting equation is, modifying equation (3):

$$p'_x = \frac{\sum_{y \in W_x} \exp(-\sum_{t \in P} G_a(t) \left| p_{x+t} - p^{drp}_{y+t} \right|^2 / h^2) \cdot p^{crp}_y}{\sum_{y \in W_x} \exp(-\sum_{t \in P} G_a(t) \left| p_{x+t} - p^{drp}_{y+t} \right|^2 / h^2)} \tag{8.5}$$

Here the subscript *crp* denotes the clean registered prior, as before, while the subscript *drp* denotes the degraded registered prior.

Figure 8.5e shows the result we obtained for our test example. We observe that the edges are now overall significantly sharper, small features are better visible, and we also see that the intricate detail in the center of the image, to the left of the pincushion-shaped dark structure is also clearly restored.

```
Input:
    Low-dose scan $L$, normal-dose prior scan $N$, low-dose degradation $D$;
Preprocessing:
1. Register $N$ to $L$ using SIFT-flow and obtain $N_R$:
        $N_R \leftarrow$ SIFT_Flow_Registration ($N$, $L$);
2. Generate projections $P$ of $N_R$ with the same low-dose degradation $D$ as
   the input:
        $P \leftarrow$ Forward_Projection($N_R$, $D$);
3. Generate the degraded version of $N_R$ – $N_{DR}$ – using FBP:
        $N_{DR} \leftarrow$ FBP($P$, $D$);
Filtering:
    Apply MR-NLM filtering to $L$ with <$N_R$, $N_{DR}$> and return the denoised filtered
result $L_F$:
    $L_F \leftarrow$ MR-NLM ($N_R$, $N_{DR}$, $L$);
```

Figure 8.6: Pseudo-code of the matched reference-based NLM (MR-NLM).

8.2.5 Comparing NLM, R-NLM, and MR-NLM

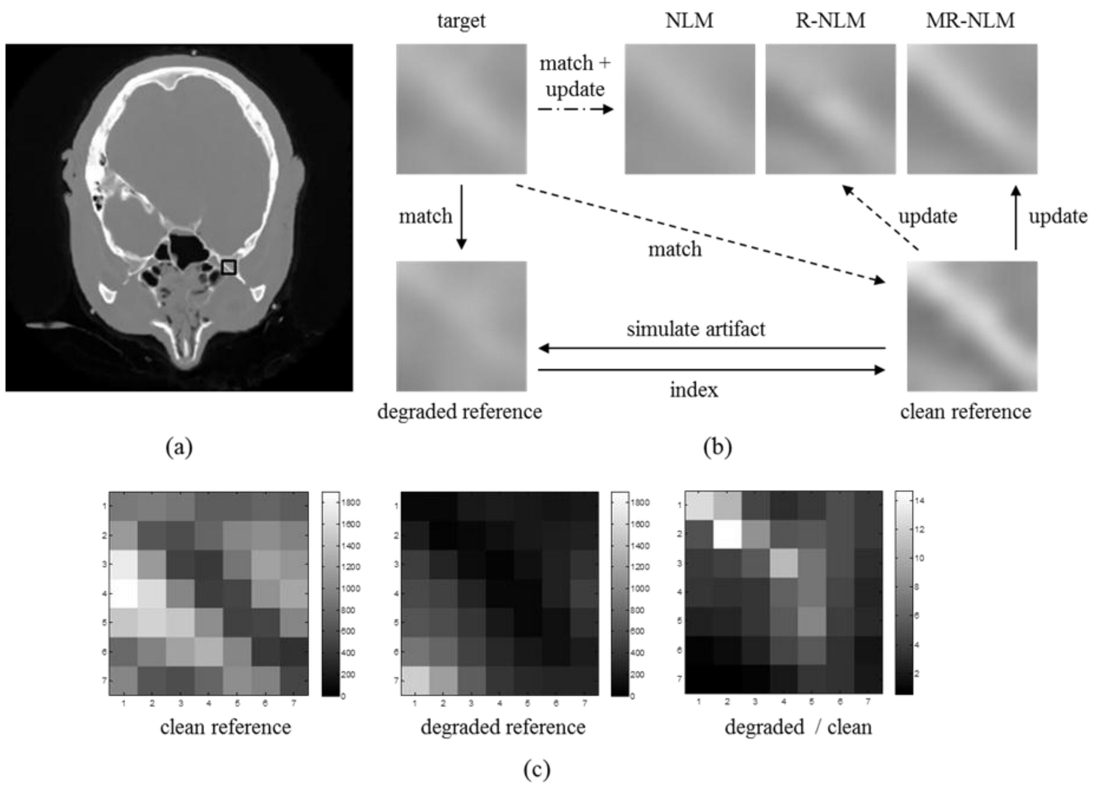


Figure 8.7: Comparing the NLM, R-NLM, and MR-NLM filtering schemes in terms of their effect on a 7×7 image region equivalent to the size of a search window. (a) Registered clean prior with black box in lower right image region indicating the region studied. (b) The three pipelines illustrated by example. (c) The map of pixel contributions for this search window. Each pixel is associated with a 7×7 patch centered on it.

Figure 8.7 compares the three different schemes, using a case study at "microscopic" detail. In Figure 8.7a we show a clean registered prior – the reference. For the matter of this discussion, we shall focus on the 7×7 image cutout – equivalent to a NLM search window – within the black box in the lower right half of this image. This cutout shows a portion of a bony structure. In Figure 8.7b we illustrate the data flow and operations of all three schemes using the cutout as an example. In this schematic, the cutout labeled 'clean reference' is a copy of the black-boxed region, while the 'target' is the corresponding cutout in the low-dose scan which is much degraded. As discussed, the MR-NLM procedure first simulates the low-dose artifacts in the clean reference producing the 'degraded reference'. It then uses this image for NLM-matching, but retains the corresponding pixels in the clean reference to update the target, yielding the cutout labeled ‘MR-NLM’. On the other hand, the R-NLM procedure (dotted lines) uses the clean reference for matching and updates the target directly, giving rise to the cutout labeled ‘R-NLM’. Finally, the conventional NLM procedure uses the target for both match and update, producing the cutout labeled ‘NLM’. The row of result cutouts demonstrates an increasing growth in quality from left to right. While the NLM cutout is quite similar to the low-dose target subject to denoising, the R-NLM cutout has somewhat sharper detail, in particular in the center. Finally, the MR-NLM cutout has the most pronounced sharpness – not quite as strong as the clean reference but fairly close.

Further insight can be obtained from visualizing the distance map for the NLM search window coinciding with the studied image cutout. Note that this search window will only resolve the value for the pixel in the center of the cutout. This distance map is used for the matching – see equation (4). Plotted in Figure 8.7c are the corresponding maps for the clean reference used in R-NLM and for the degraded reference used in MR-NLM, respectively. We can easily see that the distances in the latter map are much closer than those in the former which confirms the better correspondence. The third map, labeled ‘degraded/clean’ shows the ratio of the two maps. We clearly see that this ratio is not constant across the patch and thus a simple boosting of the *h*-parameter in the NLM equations would not be able to rectify this situation.

As for the efficiency of the three algorithms, the computational complexity is O(*NWP*) where *N*, *W* and *P* represent image size, search window size and patch size, respectively. However, in a GPU implementation, due to the high pixel independence and therefore potential parallelism, the speed could be greatly increased. As determined in [142] for the 2D case, it only takes 18ms to denoise a 512^2 image with a 11^2 search window and a 7^2 patch size. MR-NLM (R-NLM) incurs a small additional overhead for reading from two (one) other image(s) and for checking if falling back to conventional NLM-filtering is more appropriate. Overall, the entire algorithm, including FBP reconstruction (see [117]), SIFT-flow registration (see Section 8.2.2), and MR-NLM or R-NLM filtering, would most likely take on the order of seconds when accelerated on a high-performance GPU.

8.2.6 Assessing Image Quality

To evaluate the quality performance of the various reconstruction schemes we have employed two groups of metrics. The first group encompasses the traditional RMS (root mean square) and CC (correlation coefficient) measures defined as follows:

$$RMS = \sqrt{\frac{\sum_{i=1}^{N}(p_{l,i} - p_{r,i})^2}{N}} \qquad CC = \frac{\sum_{i=1}^{N}(p_{l,i} - \mu_l)(p_{r,i} - \mu_r)}{\sqrt{\sum_{i=1}^{N}(p_{l,i} - \mu_l)^2 \sum_{i=1}^{N}(p_{r,i} - \mu_r)^2}} \tag{8.6}$$

In these metrics, the $p_{l,i}$ are the pixels in the low-dose reconstruction and the $p_{r,i}$ are the pixels in the corresponding regular-dose reconstruction, in our case constituted by the originally obtained scan image. (We note that in our experiments the prior is created by non-linearly distorting the original scan, the low-dose image is created in alignment with the original scan, and the registration brings the prior back into *approximate* alignment with the original scan. Thus, the most appropriate gold standard is the originally obtained scan). The μ_l and μ_r are the averages of the low and regular dose images *l* and *r*, respectively, and *N* is the total number of pixels.

The advantage of these metrics is that they are easy to compute and have clear physical meanings. However, they reveal only little about the perceptual impact certain image differences may have. The RMS metric computes the point-wise errors and pools them across the entire image – this ignores any spatial coherence and so cannot gauge the differences in structure and contrast that may exist in local pixel neighborhoods. On the other hand, while CC does provide a statistical measure of image differences, it computes it at a global scale and it also considers only pixel intensities which are far less perceptually salient than local contrasts and edges.

As an attempt to better account for human perception when determining image quality, we have employed metrics that specifically gauge the preservation of perceptually salient information, which we define as image content to which the human visual system is most sensitive to.

The first such metric is E-CC, as defined in our earlier work [119]. It is identical to CC but operates on the edge-filtered images which we obtain using a Sobel mask. E-CC is still a global operator but it considers more perceptually salient low-level image features, i.e., edges that define the boundaries of the reconstructed objects.

Another metric we employ is the Structural Similarity Index (SSIM) devised by Wang et al. [111]. SSIM is an enhancement of the Universal Image Quality Index (UQI) [111] also recently used by Bian et al. [9]. Both UQI and SSIM combine the differences in mean intensity, contrast and structure into a single quality figure. The SSIM is computed for each image pixel at position x_j over a sliding small image window – we use an 11×11 mask – and then combined into a pooled index $\text{SSIM}_{\text{pooled}}$ by averaging the individual SSIM measurements:

$$SSIM = \left(\frac{2\mu_l\mu_r + c_1}{\mu_l^2 + \mu_r^2 + c_1}\right)\left(\frac{2\sigma_l\sigma_r + c_2}{\sigma_l^2 + \sigma_r^2 + c_2}\right)\left(\frac{\sigma_{lr} + c_3}{\sigma_l\sigma_r + c_3}\right) \qquad SSIM_{pooled} = \frac{1}{N}\sum_{j=1}^{N} SSIM(x_j) \tag{8.7}$$

Here, the subscripts l and r denote the low-dose and regular-dose images, respectively, and the μ_l and μ_r are the means of the pixels within these corresponding windows, while the σ_l and σ_r are their standard deviations and the σ_{lr} is their covariance. The constants c_1, c_2, c_3 are typically small (see [111]) and prevent numerical instabilities when a denominator is close to zero – the UQI does not have these constants which can lead to wrong estimates when these adverse conditions are met. Finally, to avoid blocking artifacts Wang et al. recommend a Gaussian-weighting of the samples under a SSIM window. Since SSIM is the generally accepted name of the metric, we will use it throughout the work but it is understood that we use its pooled version.

The first term in equation (8.7) is quite consistent with the Just-Noticeable Intensity Difference (JND) metric often used in perceptual quality studies. The second term compares the local contrasts that exist in the sliding window. Finally, the third terms evaluates the structural similarity after the differences in means and contrasts have been accounted for. The SSIM is quite powerful – studies that ask human observers to rank images with identical scenes, but corrupted with different artifacts, in terms of quality show that these ranking correlate exceedingly well with the SSIM outcome. Furthermore, it is also interesting that in these studies all images had the same RMS error. Finally, large experiments [68] have shown that SSIM is particularly well suited to detect distortions caused by noise. It also detects spatially correlated noise, which in CT images could mimic false features.

8.3 Results

We have run the algorithms described above on two datasets: a head phantom and a human lung. The head phantom is part of a body phantom scanned with a GE LightSpeed scanner. The human lung scan was obtained from the “Give a Scan” dataset collection (http://www.giveascan.org). Specifically we used the first dataset series 2 of patient p0015 obtained with a GE LightSpeed16 scanner. For all examples we used the original floating-point reconstructions for three purposes. First, they served as the basis for a high-quality projection simulation in fan-beam geometry (fan angle = 20°). We then picked a subset of these projections and reconstructed the reduced-projections low-dose imagery studied in this work. Second, we also used them to generate the priors. For this, we applied various non-linear distortions on them and subsequently registered them to the low-dose reconstructions. Third, they represented the gold standard for all numerical quality assessment via the various metrics described in Section 8.2.6.

	W/O	TVM		NLM		R-NLM		MR-NLM	
	N/A	λ=30	%	h=220	%	h=200	%	h=120	%
RMS	165	123	25.4	126	2.4	57	54.8	45	21.0
CC	0.96	0.98	2.1	0.98	0	0.99	1.0	0.99	0
E-CC	0.62	0.73	17.7	0.74	1.4	0.94	27.0	0.96	2.1
SSIM	0.43	0.53	23.2	0.53	0	0.95	79.2	0.97	2.1

Table 8.1: Numerical comparison of the results obtained for the head phantom via various metrics. The percentage figure for a metric measures the improvement with respect to the method to its immediate left. To the left of the % cell, above the scores, we list the optimal parameter setting for each algorithm which we obtained by manual tuning.

We begin with the head phantom already examined in Section 8.2 to illustrate the outcomes of the various algorithms. Table 8.1 and Figure 8.8 compare the results obtained for MR-NLM, R-NLM, NLM, TVM, and no filtering, as gauged by the RMS, CC, E-CC, and SSIM error metrics. Table 1 also gives the settings for the various algorithm parameters which we manually tuned for optimal performance. The first observation we make is that all metrics show similar trends (but we also observe that CC is much less sensitive to the changes in image quality). In general, for the CC, E-CC, and SSIM the maximum possible value is 1.0, while for the RMS error the optimal value is 0. In this particular experiment, all metrics reach their best values for the MR-NLM algorithm – around 0.97 for the perceptual metrics – and their worst values when no filtering is applied. It also appears that NLM and TVM reach quite similar scores for each metric, with a slight advantage for NLM. This can be verified by comparing the images (see Figure 8.2) which look fairly similar. The improvement of R-NLM over NLM is significant for both RMS (54%) and the perceptual metrics (27% for E-CC and 79% for SSIM). The improvement for MR-NLM over R-NLM is another 21% for RMS and 2% for the two perceptual metrics.

The absolute difference images presented in Figure 8.9 show similar trends. Marked improvements can be observed for R-NLM over the prior-less schemes, and more moderately for MR-NLM over R-NLM. The improvement achieved by R-NLM is mainly in streak removal. MR-NLM, on the other hand, adds sharpness and detail definition which can be appreciated by the overall much smaller errors, in particular at edges and sharp corners.

In order to better explore the performance of the various algorithms at the local detail level we have conducted an ROI-based analysis (see Figure 8.10). Figure 8.10a depicts the locations of 4 ROI regions and Figure 8.10b plots the corresponding SSIM scores. We observe a 7-8% improvement for R-NLM over NLM in all 4 ROIs and another 7-8% in ROI 2, 3, and 4 for MR-NLM over R-NLM. ROI 1 contains a fairly structured and high-contrast feature which is less in need of the added fidelity of the MR-NLM scheme. Lastly, Figure 8.11 presents the ROI study visually in form of cutout details, which best demonstrate the considerable benefits the MR-NLM restoration method provides. We observe that all ROIs show significantly more detail for MR-NLM, as opposed to R-NLM. In fact, the reconstructions are quite close to the ideal image. On the other hand, the differences of R-NLM vs. the prior-less methods are also significant, but not as marked as for MR-NLM vs. R-NLM.

	W/O	TVM		NLM		R-NLM		MR-NLM	
	N/A	λ=20	%	h=260	%	h=180	%	h=120	%
RMS	222	154	30.6	153	0.6	128	16.3	120	6.2
CC	0.92	0.96	4.3	0.96	0	0.97	1.0	0.98	1.0
E-CC	0.6	0.69	15.0	0.71	2.9	0.81	14.1	0.85	4.9
SSIM	0.49	0.61	24.5	0.61	0	0.66	8.2	0.69	4.6

Table 8.2: Numerical comparison of the results obtained for the human lung via various metrics.

Next, Figure 8.12 show the same sequence of results obtained with the human lung, reconstructed from 90 projections over 360°. The distortion applied was a fisheye warp. Figure 8.13 presents difference images, and Table 8.2 lists and Figure 8.14 plots overall evaluations with the various metrics. We make similar observations as for the head phantoms but note that in this test case the quantitative improvements for R-NLM and MR-NLM are more balanced and the MR-NLM/R-NLM gain is about double than that for the head phantom. Finally, Figure 8.15 depicts ROI-definitions and the SSIM-scores for the studied restoration schemes. Figure 8.16 shows the cutout details. Again, we see that MR-NLM significantly improves the fidelity of small detail and in fact it is even able to restore some of the original CT image noise texture that was part of the prior.

Finally, Figure 8.17 explores the effect of different search window sizes, for the human lung. The size needed mainly depends on how well the registration performs and how much the overall structure changes between low-dose and normal-dose scan. A larger size results in more smoothness and a reduction of detail, but it also increases robustness when the registration is not perfect. As mentioned, we have used a 7×7 window for all experiments and as this figure demonstrates this window size provides a good trade-off on smoothness and detail preservation.

8.4 Conclusions

We have presented an efficient non-iterative framework for low-dose CT image reconstruction that utilizes an available prior regular-dose scan to assist in the NLM-based regularization of a filtered backprojection reconstruction plagued with significant low-dose artifacts. We have specifically addressed the case when dose reduction is achieved with a lesser number of projections which typically results in severe streak artifacts. Therefore the reduction of the dose is directly related to the reduction of projections. We find that a crucial element in this effort is to simulate the same low-dose artifacts also in the registered prior to facilitate a more accurate structure matching for subsequent regularization with samples in the registered clean prior. The overall purpose of this work is to make the low-dose image faster readable by reducing the streak artifacts and increase the visibility of the features. While all of these image features can also be seen in the low-dose image, recognizing them requires a time-intensive visual inspection which reduces diagnostic throughput and also makes clinical reasoning much more difficult.

The main limitation in using our method for reducing the number of views is the registration of the prior with the degraded reconstruction image. We have chosen the SIFT-flow method which we found to perform better than the Demon algorithm [104] in the presence of noisy data. We suspect that this might be because SIFT-flow uses a structure-sensitive feature descriptor at multiple scales and might ignore noise artifacts better. One item of future work is to test other registration methods and see if they perform even better and so allow a further reduction of projections. It would also be interesting to use the SIFT-flow registration technique for noisy reconstructions obtained with low mA or kV settings and compare the outcome with those obtained using the EMP-MI approach of Ma et al. [75].

Another topic of study is to see how sensitive the method is with regards to newly appearing or vastly changing features in the follow-up low-dose scan. Our NLM-based matching scheme is designed to fall back to conventional NLM if no reliable match can be found in the registered neighborhood. Future work will study this fail-safe design using a wide set of structures in a rigorous fashion. Further, we also aim to perform a detailed study with regular-dose and subsequent low-dose projection data directly obtained from a scanner, which we did not have access to for this present work. But nevertheless, we believe that our results clearly demonstrate the conceptual merit of our method.

Further, we find that the perceptual image quality metrics track quite well what can also be visually observed from the reconstructed images. Specifically, we find that our efficiently computed E-CC metric shows similar trends as the more computationally involved SSIM metric. It therefore represents a good alternative when image evaluation must be fast. For future work we plan to involve clinicians in the image assessment to further validate the suitability of E-CC and SSIM for our purposes.

Current work also focuses on accelerating the R-NLM and MR-NLM frameworks entirely on the GPU in order to increase appeal to clinical applications, also in conjunction with our clinician-based validation and fine tuning.

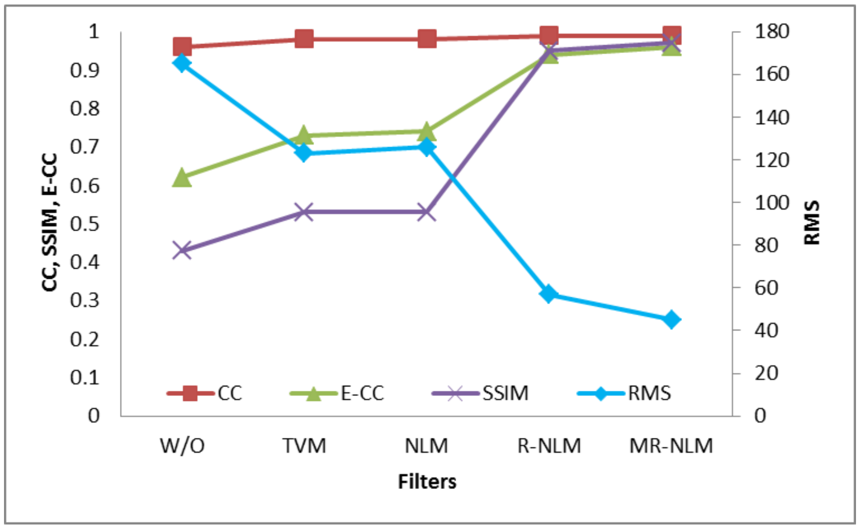


Figure 8.8: Graphical comparison of the results obtained for the head phantom via various error metrics.

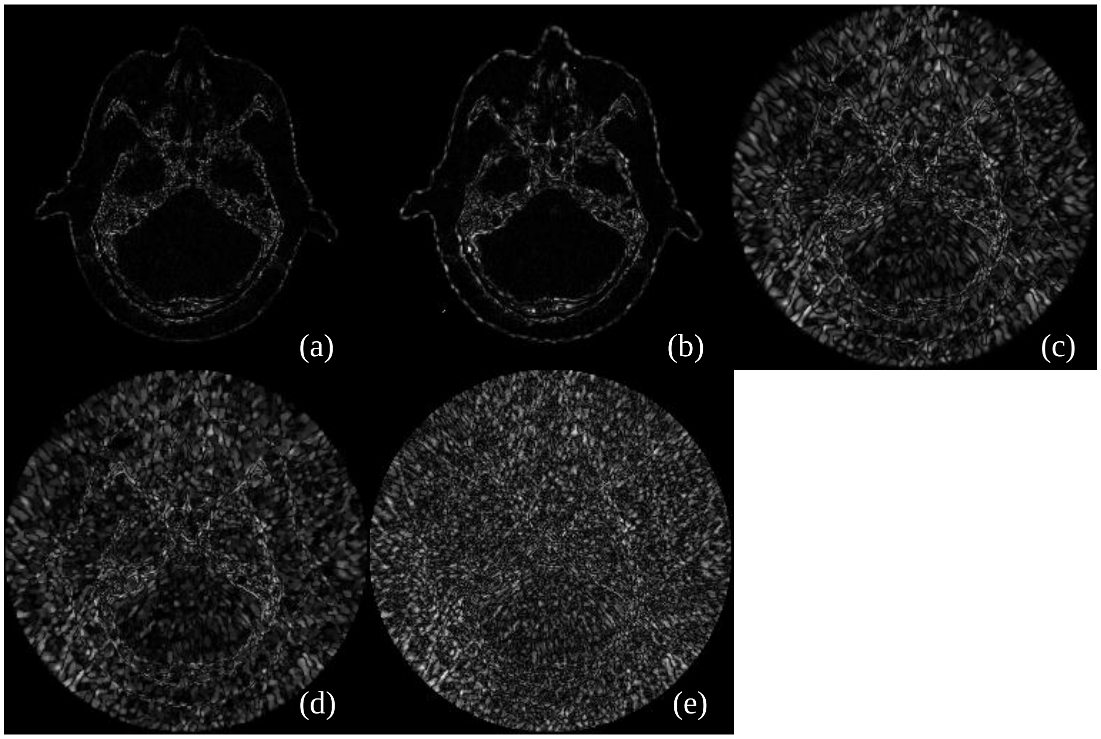
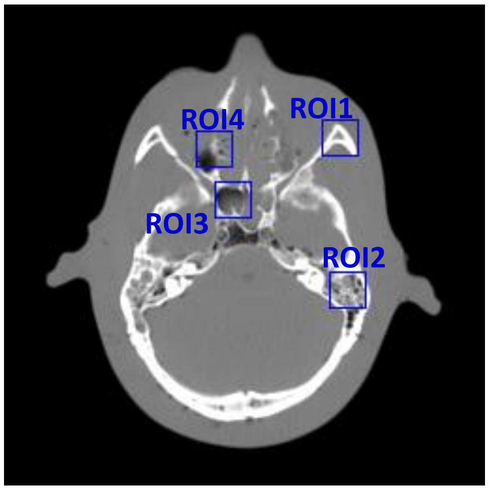
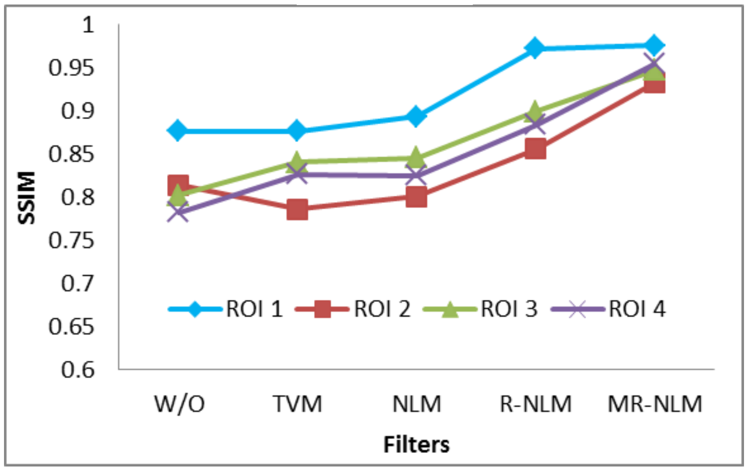


Figure 8.9: Difference images for the head phantom reconstructions. (a) MR-NLM; (b) R-NLM; (c) NLM; (d) TVM; (e) without filtering.



(a)



(b)

Figure 8.10: ROI-based analysis for one slice of the head phantom. (a) ROI locations; (b) SSIM evaluation.

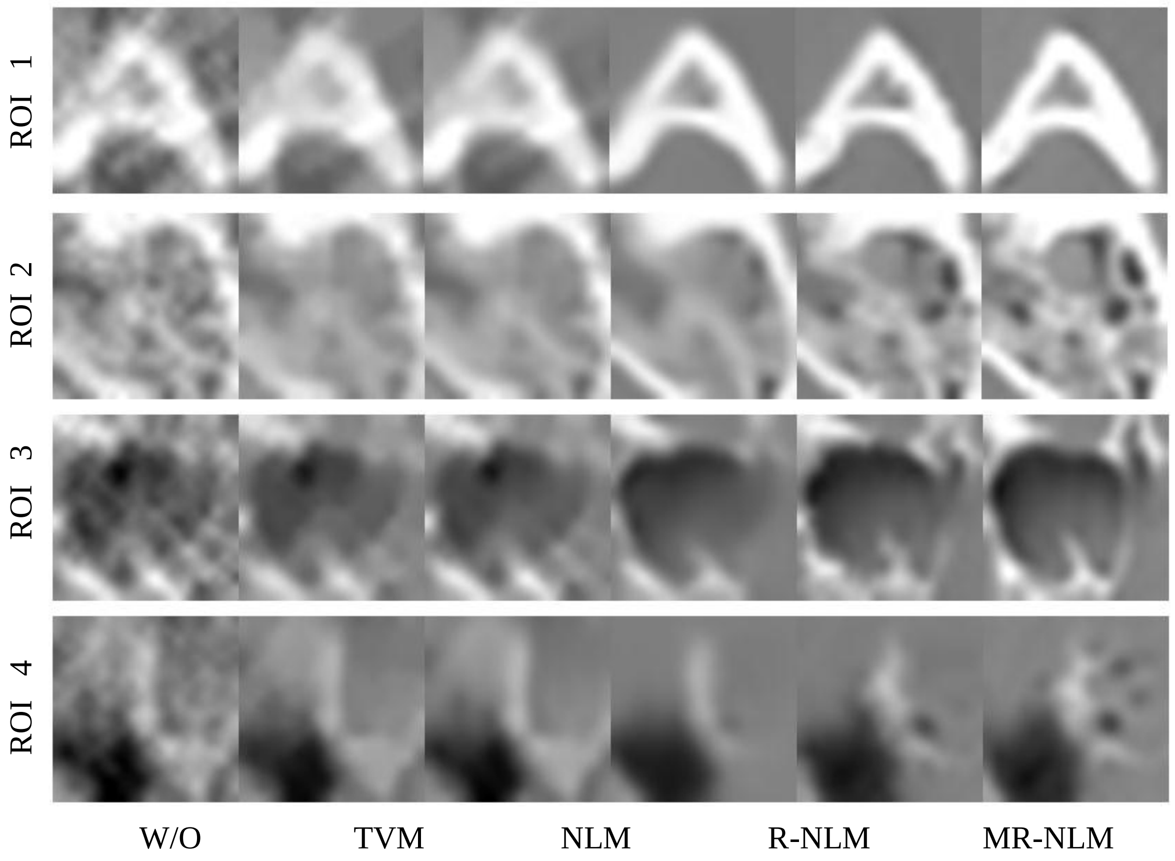


Figure 8.11: Comparing the four ROIs defined in Figure 8.9 visually.

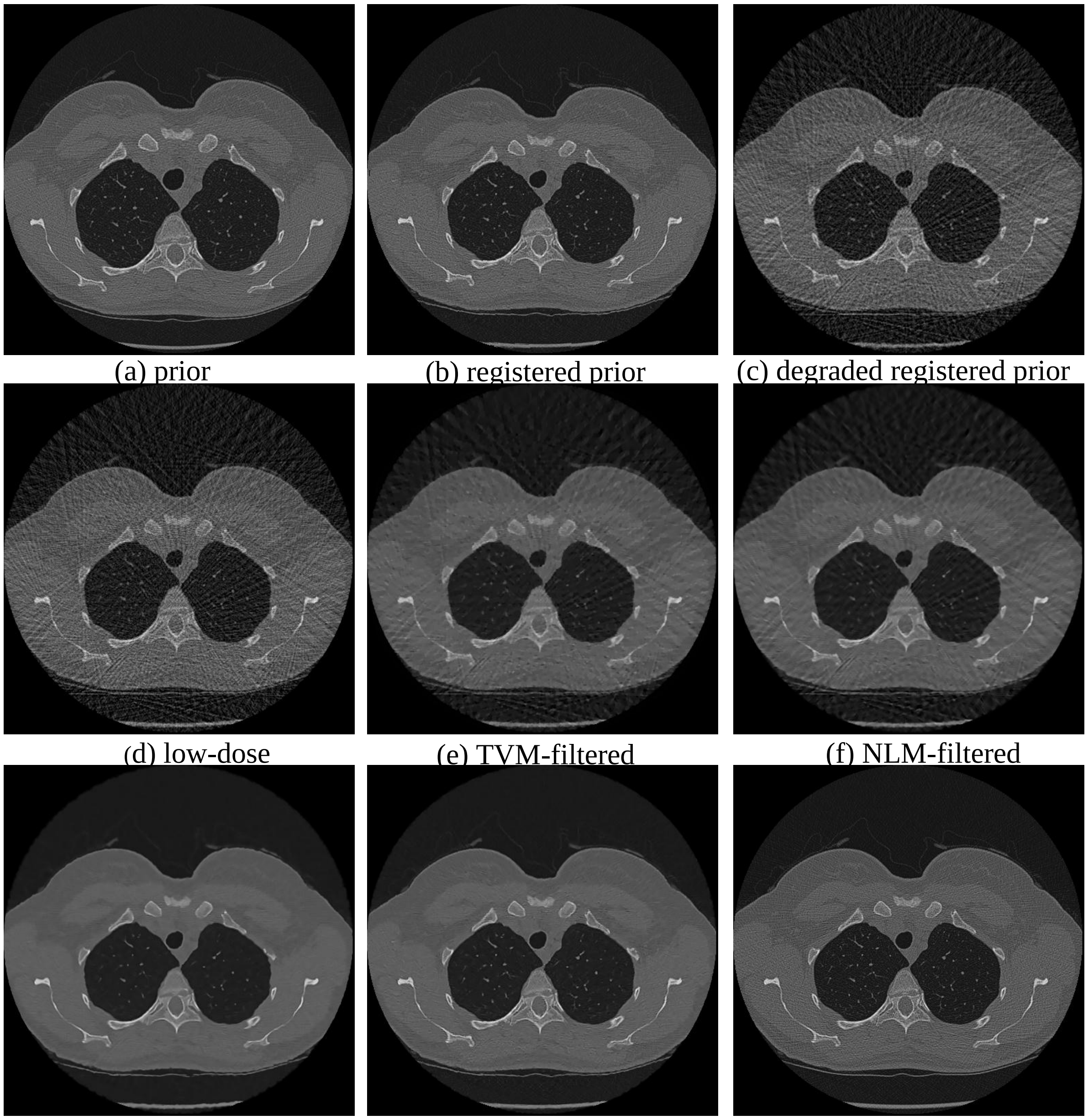


Figure 8.12: Results obtained for the human lung.

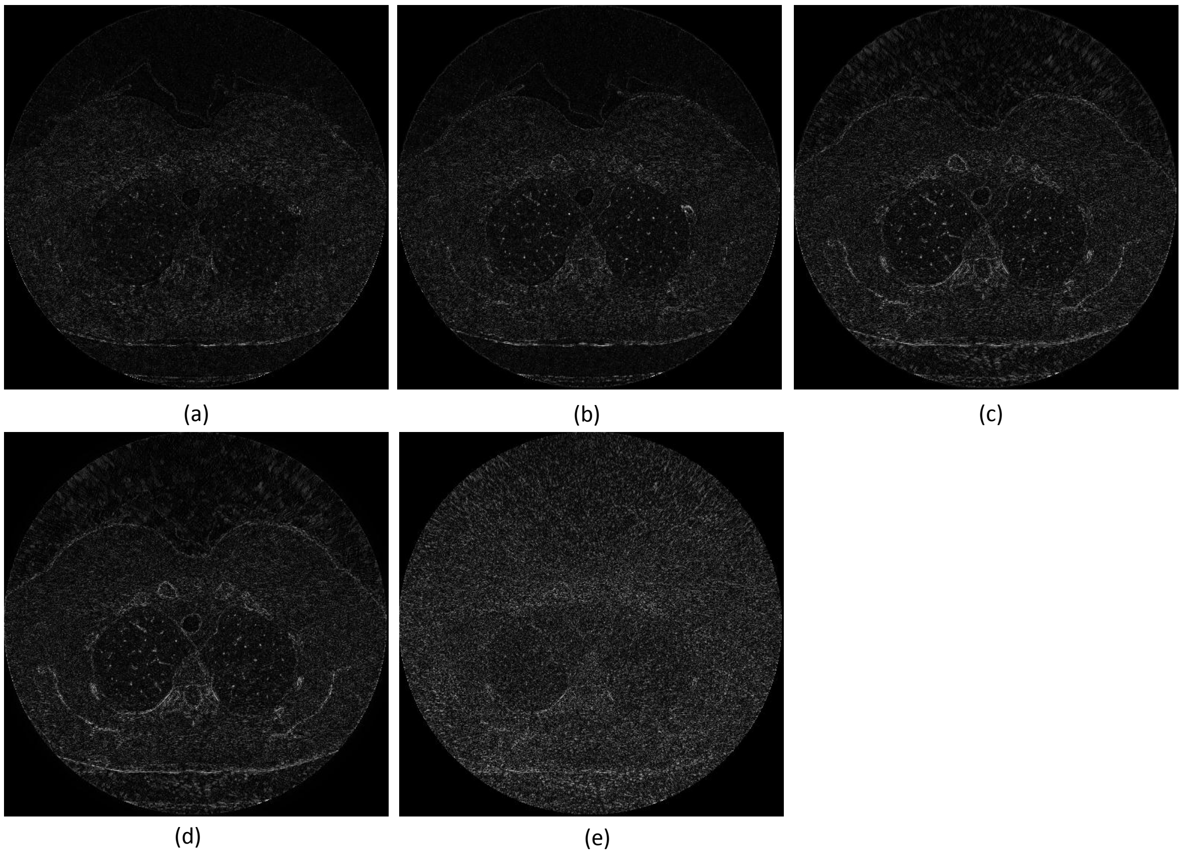


Figure 8.13: Difference images for the human lung. (a) MR-NLM; (b) R-NLM; (c) NLM; (d) TVM; (e) without filtering.

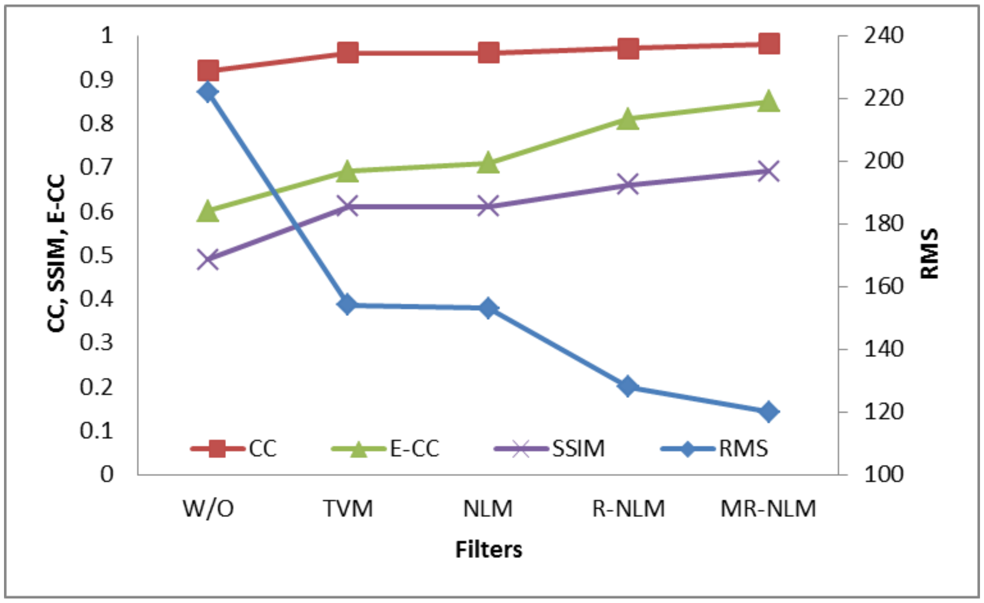
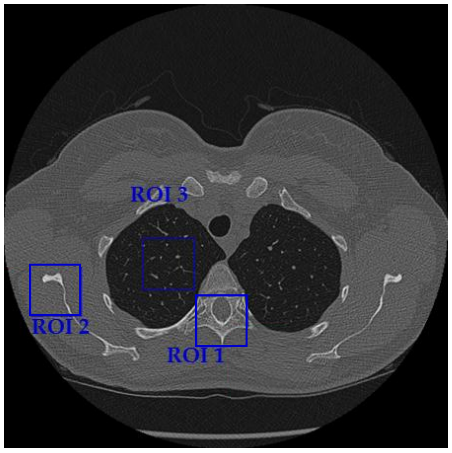
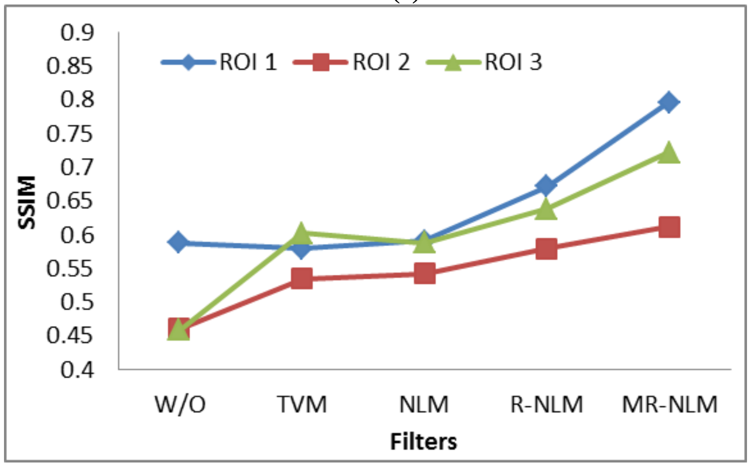


Figure 8.14: Graphical comparison of the results obtained for the human lung via various metrics.



(a)



(b)

Figure 8.15: ROI-based analysis for one slice of the human lung. (a) ROI locations; (b) SSIM evaluation.

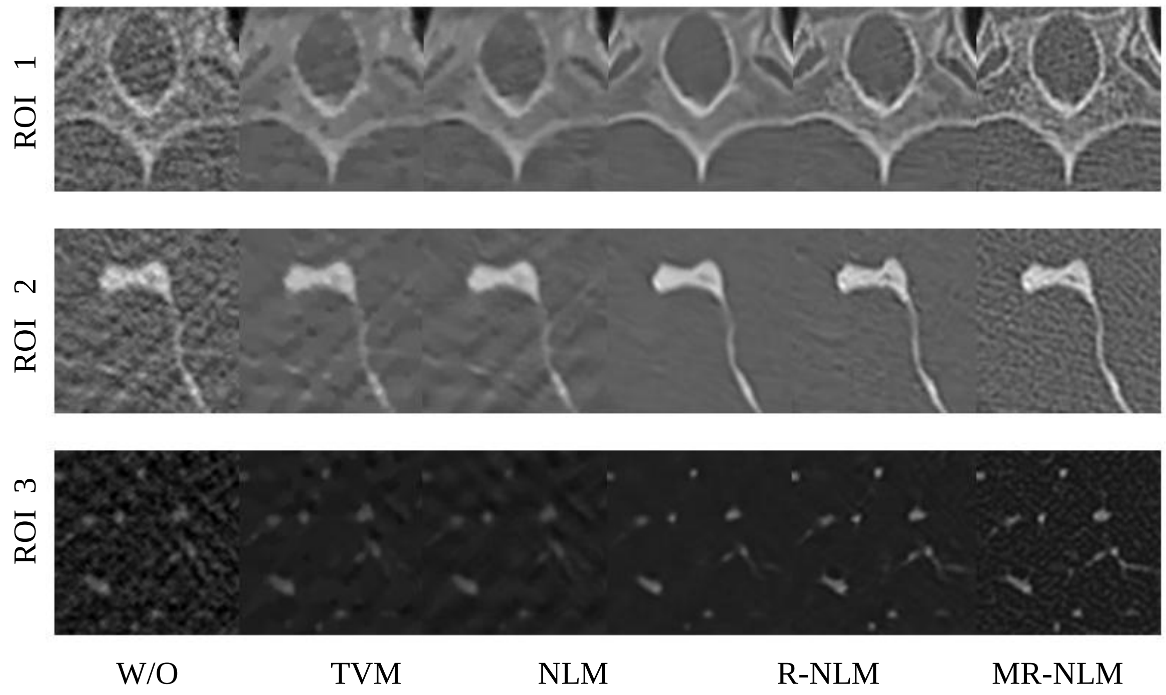


Figure 8.16: Comparing the two ROIs defined in Figure 8.14 visually.

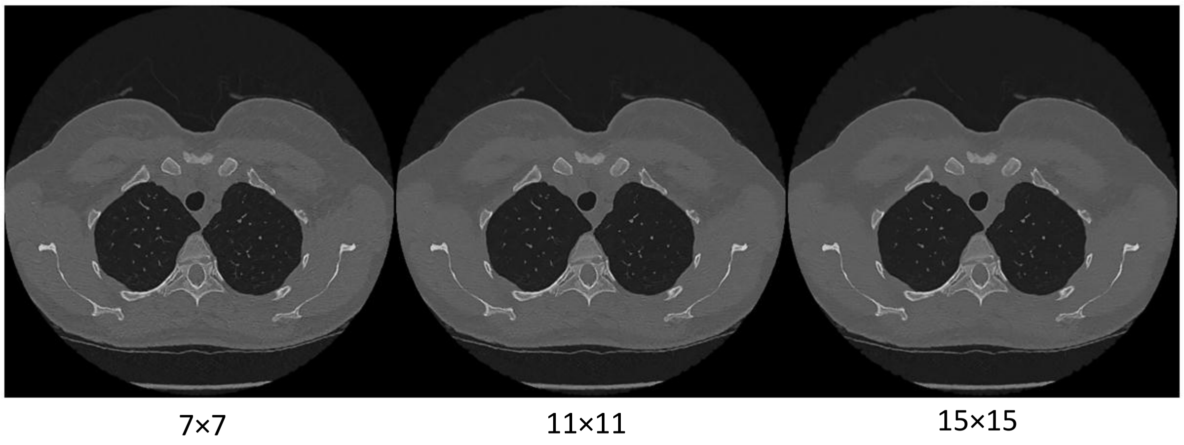


Figure 8.17: Comparing the results obtained with three search window sizes.

Chapter 9

Database-Assisted CT Image Restoration

With the success of exploiting prior CT scan of the same patient to do image restoration for low-dose CT, we move forward in this chapter to a more general situation – when the prior of the same patient is not available. The process to seek for appropriate reference images could be treated as a pre-step of the two-step framework presented in last chapter. Thus the framework presented here focuses on method to implement this extra module and meanwhile includes some improvements to the rest modules.

9.1 Introduction

Motivated by the need to minimize the radiation exposed to patients, a growing number of research efforts have been dedicated to the topic of *low-dose CT*. Lowering the radiation dose can be achieved either by reducing the number of X-rays, their energy, or both. However, a direct effect of these dose reduction efforts are CT images with strong noise artifacts, streaks and reduced feature detail – all of which impede image readability in diagnostic tasks. To overcome these problems one can either apply iterative reconstruction schemes with the goal of optimizing the reconstruction given the limited data [21][52][101] or one can try to reduce the artifacts in the image domain via a suitable image restoration method [71].

For the latter option, neighborhood filters, in particular the Non-Local Means (NLM) filter [4], have shown great promise for the restoration of degraded low-dose CT imagery [71][122]. Originally devised for general image de-noising tasks, NLM is essentially an extended Gaussian filter. It updates a given pixel by looking for pixels with statistically similar local neighborhoods in the image and then Gaussian-weighs their contributions by the degree of similarity. The extent of the search is specified by a *search window*, while the size of the neighborhood used for similarity matching is called a *neighborhood patch*. A more recent trend in CT reconstruction has been to extend the search window beyond the image subject to restoration. Schemes have been devised that utilize a prior scan [75][129][135][125] of the same patient to search for high-quality updates. Other work has successfully constrained a reconstruction by images of the same dynamic scan [21]. While all this produces excellent results, such a prior scan or dynamic scan of the same patient may not always be available.

To meet this inherent shortcoming we propose the idea of extending the search window even further, namely, to a collection of images of *different* patients. This approach, in fact, is quite alike the psycho-physical processes that occur in medical professionals when viewing degraded imagery. They also borrow from their extensive medical training and experiences to see the “true patterns behind the noise”. These recognition tasks, however, take valuable time and can also lead to frustration, and it is for this and other reasons that artifact reduction by image processing and algorithmic means is an important mission.

Using collections of clean images to reduce noise or blur in degraded images is not a new idea, at least not in general. There are in fact two rather disjoint schools of thought, and both aim to cope with the extremely large space of possible image detail. The first approach first constructs a large-scale database of possible detail at some level and then uses a sophisticated matching strategy to retrieve the detail of interest from this database [48]. The other approach is based on *sparse coding*. It first constructs a dictionary of representative base patterns which must then be optimally combined for reproducing the desired detail of interest [32]. While the first approach is a top-down search, the second is bottom-up. Both strategies can be justified by theories on how humans perform visual search, which likely is a conjunction of both [51].

Our work expands on two workshop papers [124][128] where we have presented preliminary thoughts as well as encouraging experimental results using the first – the database – approach for low-dose artifact mitigation. Parallel to our work, another team of researchers [129][132] has pursued with similar success the alternative – the sparse coding – approach for the same purpose. Since the vote is still out, on an even grander scale, about which of the two strategies is better or more likely, we refrain from making such claims here. Our sole purpose is rather to formalize the framework we conceived, expose results on what is currently possible when using it for low-dose CT, and point out current shortcomings that warrant future work.

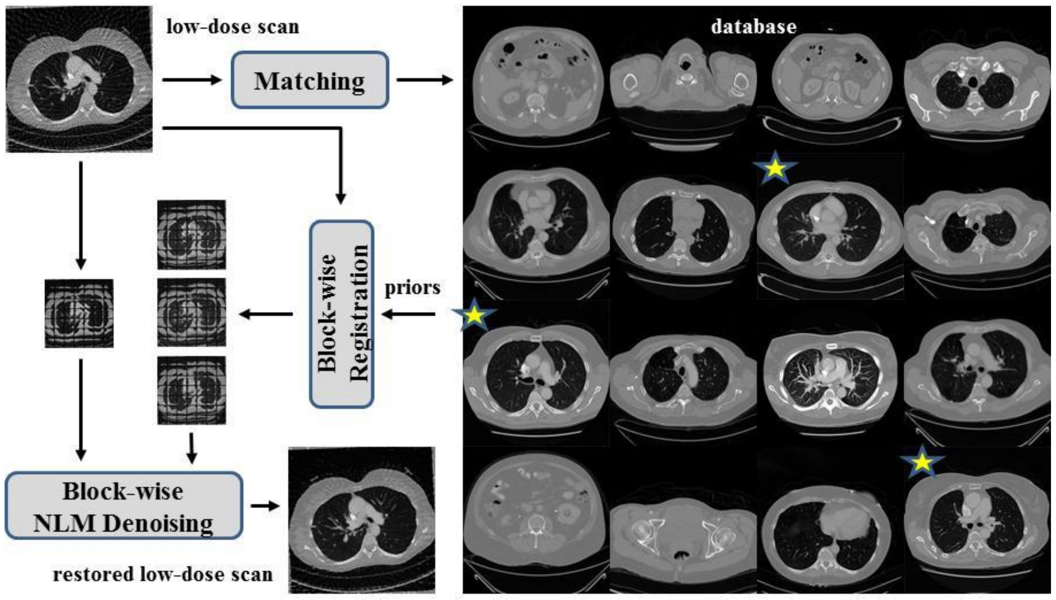


Figure 9.1: Illustration of our framework by ways of a (small) lung database example.

Figure 9.1 illustrates our framework by ways of an example: the restoration of a low-dose lung (target) scan using a database of regular-dose lung images. First, we match the target scan with the images in the database and select a set of images (marked with stars) containing similar anatomical content as the target. We then register these images to the target to form the set of artifact-free priors. Finally, using these priors we apply the extended NLM-filter scheme to de-noise the target via a block-wise update strategy.

This chapter is organized as follows. Section 9.2 describes both methodology and technical detail, Section 9.3 presents results, and Section 9.4 ends with conclusions and future work.

9.2 Methods and Materials

The workflow of our method consists of three major stages: offline database construction, online prior-search and online de-noising as shown in Figure 9.2. In the offline database construction stage, we create a global image feature descriptor *G* to represent each image in the given image database. This forms the global feature database. A visual vocabulary *V* summarizing the local image features is also learned in this stage. Then in the online prior-search, we generate *G(I)* with *V* for the target image *I*. Following, we use *G(I)* to query the global feature database to find the *M* nearest neighbors as regular-dose priors. These priors have (artifact-free) anatomical content that is most similar to the (degraded) anatomical content of the target. Next, in the online de-noising, we first align the regular-dose priors to the target in a block-wise manner. These images form the set of clean registered prior (*CRP*) blocks. Using these blocks, we run what we call *R-NLM (Reference-based NLM)* [125], where we use the prior blocks for NLM matching and look up the pixel values in the corresponding *CRP block*. We now describe each of these components in closer detail.

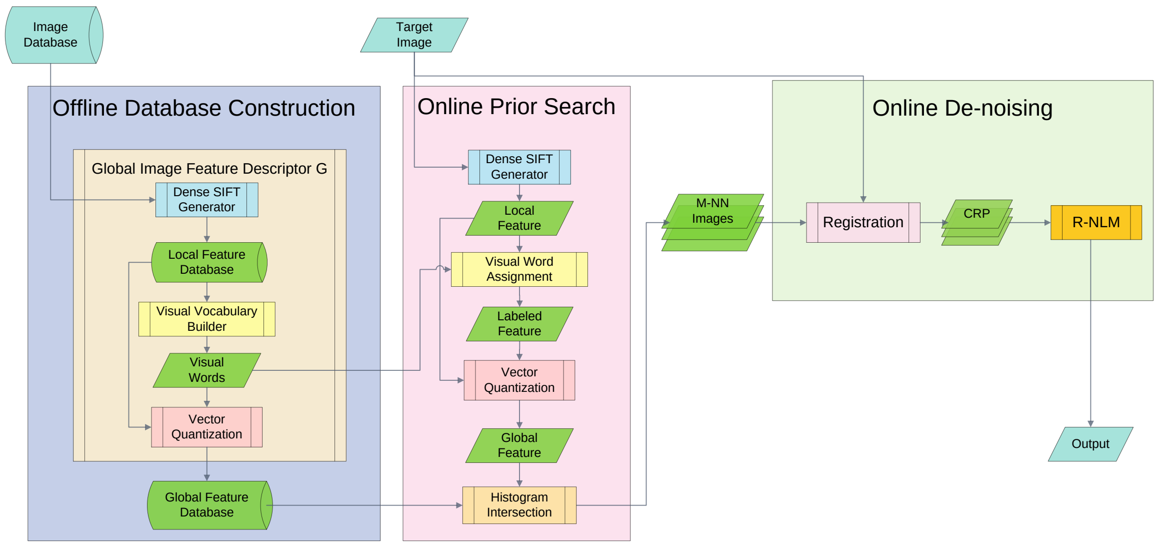


Figure 9.2: Overall workflow of our framework.

9.2.1 Local Image Feature Descriptor

Image matching is a fundamental operation in computer vision and image processing and it is often used for scene and object recognition. Typically, the image is expressed as a high-dimensional feature vector and the matching occurs in this high-D feature space. Since we wish to match image features in the presence of significant noise and streaks artifacts we require a feature descriptor that is insensitive to these degradations. The scale-invariant feature transform (SIFT) [62] is such a feature descriptor. It captures the histogram of edges in a local neighborhood at multiple levels of scale, characterizes salient local and transform-invariant image structures and encodes contextual information. A SIFT feature descriptor is usually a 128-D vector encoding 8-orientation histograms of edges over 4×4 blocks with each block of size 4×4, serving as a local descriptor of the image. In its original definition, only keypoint locations are selected. However, it was shown that dense SIFT vectors (dSIFT) on a regular spaced grid could provide more correspondence and reveal more details also in flat texture areas and are thus more robust [70][72].

We found that a grid spacing of 8 pixels works well for dense SIFT in our application. Thus, for an image of size 256^2 we get 32×32 SIFT vectors, while for size 512^2 we get 64×64 SIFT vectors. We use these local image features to compute the visual vocabulary, as shown next.

9.2.2 Global Image Feature Formation

To form a global image feature descriptor from local ones, traditional dense SIFT algorithms follow the bag-of-feature method [13]. We use the following algorithm to accomplish this.

First, extract the local feature descriptors: generate a set of SIFT local feature descriptors $\{S_0, S_1, .., S_{N-1}\}$ to represent each image. As explained in last subsection, the local feature vector is extracted for each grid point. Therefore each image will have a fixed number of local feature vectors, with each vector summarizing the local features inside a 16×16 neighborhood.

Second, build the visual vocabulary: randomly select the local feature descriptors of all images in the database and perform k-means clustering to learn K cluster centers as visual words $\{V_0, V_1, \ldots, V_{K-1}\}$. This forms the visual vocabulary V of the database. The number of cluster centers K is dependent on the complexity of the features. As the CT images are more simplified than natural images, a smaller K is usually enough.

Third, label the local features to the visual words: for each image, assign the index of the closest visual word to each local feature vector. This step extremely reduces the dimension of the description of the local features. Now only a fixed number of indices as labels is stored to represent each image.

Last, perform vector quantization to generate a global feature descriptor: compute the histogram of visual words in each image $\{H_0, H_1, \ldots, H_{K-1}\}$ and concatenate the weighted histogram series into a long vector to form the global feature descriptor.

One drawback of this algorithm is that the feature's location information in the original 2D image space is discarded. To make use of this spatial information and keep track of it in multi-resolution, we exploit a spatial pyramid scheme [70] to implement a "stronger" feature description. The multi-resolution layers are formed by recursively subdividing the image into $b\times b$ blocks. In a layer L, for each block, only the feature vector extracted from that block is aggregated into the histogram of its specific visual word. In this way, the clustering is still performed in feature space while the histogram pyramid is built in 2D image space. The weight for each histogram is inversely proportional to its block width. We call the resulting histogram sets the *spatial pyramid-based histograms*. Finally these histograms are concatenated layer-wisely, block-wisely and visual word-wisely to form the complete global feature vector.

There are four parameters associated with this part: the number of visual words *K*, the number of layers *L* and the block size *b*. In our experiments, we used 60 visual words of the local feature descriptors, and each image was represented with a spatial pyramid-based histogram composed of a single layer with 5×5 blocks. These settings showed the best query precision in the online prior search stage. The next section provides further detail.

9.2.3 Online Prior Search

In this stage, the global feature vector of the target scan is generated online following the same steps as outlined for the off-line stage: extract local features, label them with indices of visual words, and concatenate the set of histograms. This long feature vector is then used to search for similar priors in the database. These priors anatomically characterize the same content as the target scan but may contain small variations in scale, rotation, and deformation. We found that histogram intersection performs better than the Euclidean distance for gauging the similarity for matching. The histogram intersection operates within a spatial pyramid, i.e., the intersection is counted both block-wise and visual word-wise and is then summed up to form a single value [70].

To ensure that the priors contain a wide variety of diverse anatomical features, we perform the searching process patient by patient. We then construct a ranked-list by selecting and sorting only the top ranked priors in each patient. Finally, additional search within the top-ranked priors in the list further expands and refines the list [24].

9.2.4 Registration

Once the regular-dose prior (or reference) scans are found, the online de-noising process can be executed. The first step is to register the priors with the target scan. Different from our initial work [124] we perform both the registration and the de-noising in a block-wise fashion. This provides better local control which is needed since the database priors have less correspondence to the target than priors coming from the same patient. More specifically, we create a small block of size 129×129 and shift that block with a step size of 64 in raster-scan order. Figure 9.3 illustrates this process. For each block, we perform the registration by aligning that block with the corresponding blocks in the prior scans (red boxes in Figure 9.3). This local registration relaxes the strict requirements of a global image registration and allows for priors to only partially match the target, which is likely since they come from different patients.

We used the SIFT-flow registration algorithm [72] for the registration. SIFT-flow is a state-of-the-art registration method that originates from the optical-flow algorithm and produces dense, pixel-to-pixel correspondences between two images. It extends optical-flow matching from raw pixels to SIFT features, which significantly improves robustness when registering artifact-rich with artifact-free images. In order to further improve the quality of the registration we pre de-noise the target scan via Gaussian filtering.

9.2.5 R-NLM De-noising

R-NLM [125] follows the standard NLM filtering scheme but uses the artifact-free registered prior images, *CRP*, instead of the target itself. Thus, the pixel weights are computed by comparing patches in the target with patches in the prior images. More formally:

$$p_x^{'} = \frac{\sum_{y \in W_x} \exp(-\sum_{t \in P} \left| p_{x+t} - p_{y+t}^{crp} \right|^2 / h^2) \cdot p_y^{crp}}{\sum_{y \in W_x} \exp(-\sum_{t \in P} \left| p_{x+t} - p_{y+t}^{crp} \right|^2 / h^2)} \tag{9.1}$$

Here *x* is the location of the target pixel and *y* are the locations of the candidate pixels with values p_y. W_x is the search window around *x*, and *P* is the patch size of each pixel. The patch similarity is measured by the L_2 distance between two patch vectors with *t* representing the index within a patch. The factor *h* controls the overall smoothness of the filtering. In our case *h* is larger than typically used for standard NLM to accommodate higher noise level. The superscript *crp* indicates that the pixels originate from the artifact-free registered priors *CRP*.

Equation (9.1) differs from the one used in [125] in that we removed the Gaussian weighting in the patch similarity measurement. We found in experiments that this leads to better matches since a greater patch neighborhood influences it. The direct consequence of better matches is an increased sharpness at edges, as demonstrated in [125] for the matched artifact MR-NLM scheme. We could not use MR-NLM in the work presented here since it proofed difficult to simulate realistic streak artifacts in a block. Noise would have been easier but we strived for a scheme that applies to both types of artifacts unilaterally. Eliminating the Gaussian kernel and replacing it by a wide box filter seems to better "see through" the noise and capture the true pattern underneath more faithfully.

For pixels for which no similar patch can be found within the search window we perform standard NLM with a smaller *h*. The pixels are detected by comparing their denominators in (9.1), which represents the summation of contributions from pixels in the search window, with a pre-set threshold – we used 0.3. For these pixels the de-noising falls back to standard NLM. This conservative approach ensures that no false ill-fitting features are introduced.

We perform the R-NLM in a block-wise, raster-scan fashion, as shown in Figure 9.3. For each block, when the R-NLM has finished, the pixels in the overlapping regions are *feathered* in the raster-scan order. Feathering retains the edges but adjusts the grey levels such that newly added regions blend well with existing ones [86]. It allows one to not only retain local contrast among neighbor but also to remove remaining noise at some loss of sharpness.

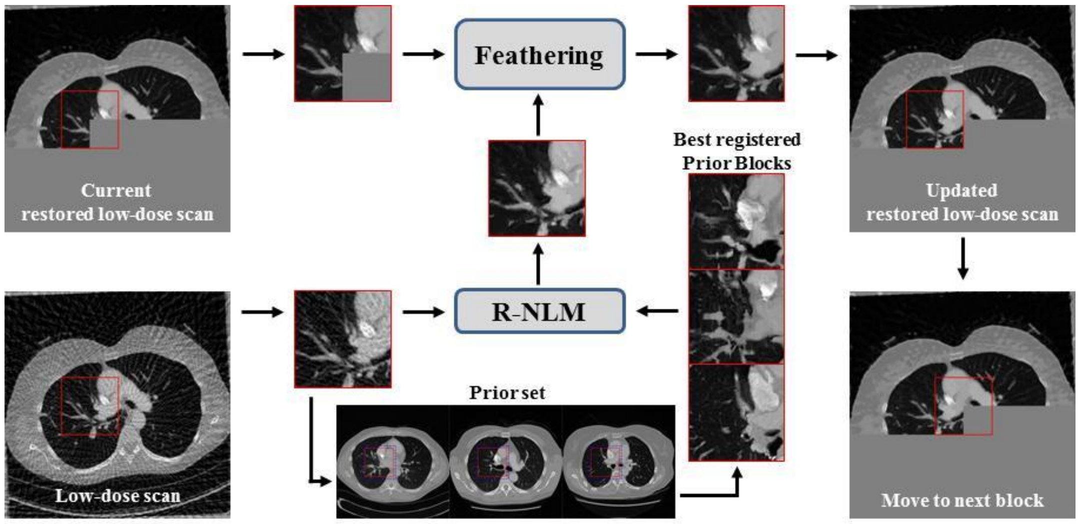

Feathering
Current restored low-dose scan
Best registered Prior Blocks
Updated restored low-dose scan
R-NLM
Prior set
Low-dose scan
Move to next block

Figure 9.3: Block-wise registration and R-NLM de-noising workflow.

9.3 Experimental Results

We constructed a human lung database of 30 patients (41,138 512^2 images) from an online human lung database (http://www.giveascan.org). The images were not pre-aligned. To match the quality of the CT scans, all scans were re-generated from 720 projections over 360^o with a fan-beam geometry (fan angle = 20^o). The 720 projections were the point at which there was no more quality improvement, as gauged by RMSE with the original scan. We used these scans as the gold standard in all experiments. We then picked subsets of these projections to generate the low-dose scans with streak artifacts. To simulate the noise artifacts, we added various levels of noise (dB, signal-to-noise ratio (SNR)) to the sinogram of the gold standard images. To create a new scan different from any scan in the database (even if they were already from different patients), the selected scan was first deformed or rotated (to mimic a real clinical situation), forward projected, and then reconstructed with the low-dose condition under study.

In the following sections, we report on two experiments to evaluate and validate the proposed algorithm. In all experiments, we used a patch size of 7×7 with a 13×13 search window for the NLM filter. Then, both the overall smoothness parameter, h, and the threshold parameter which decides between standard NLM and R-NLM, were experimentally chosen by inspecting the quality of the restored image for each streak and noise reduction task.

9.3.1 Priors Quality and Diversity

We performed experiments at which the database-assisted restoration occurred at three levels of difficulty, gauged as a function of closeness of the available priors to the target anatomy. In these experiments the database contained, among other scans: (1) one almost identical CT scan of the same patient; (2) two somewhat similar CT scans of the same patient; and (3) only CT scans of other patients. Target images were generated at the following two low dose conditions: (1) reduced data (only 86 projection – this is about 11% of the gold standard data and represents a dose reduction of 88%), and (2) low mA imaging (30 dB SNR Gaussian noise was added to the sinogram). Figure 9.4 shows the restoration results for these two conditions using the different types of priors described above, and compares them with results obtained with the standard NLM method. We observe that the prior-based scheme significantly improves image quality even with “foreign” priors from different patients. The edges are sharper and detail is better preserved.

Figure 9.5 shows that having a more diverse set of priors can improve the outcome tremendously. In the example given, we needed at least three priors to successfully restore the structure pointed to by the arrow. This is not unlike the case in which a more experienced radiologist “reads” a noisy image. And indeed, the need for a massive database has been confirmed in research that aimed to remove unwanted structures in photographs [48]. Constructing such a large database is our current goal, with a need for “big data” management.

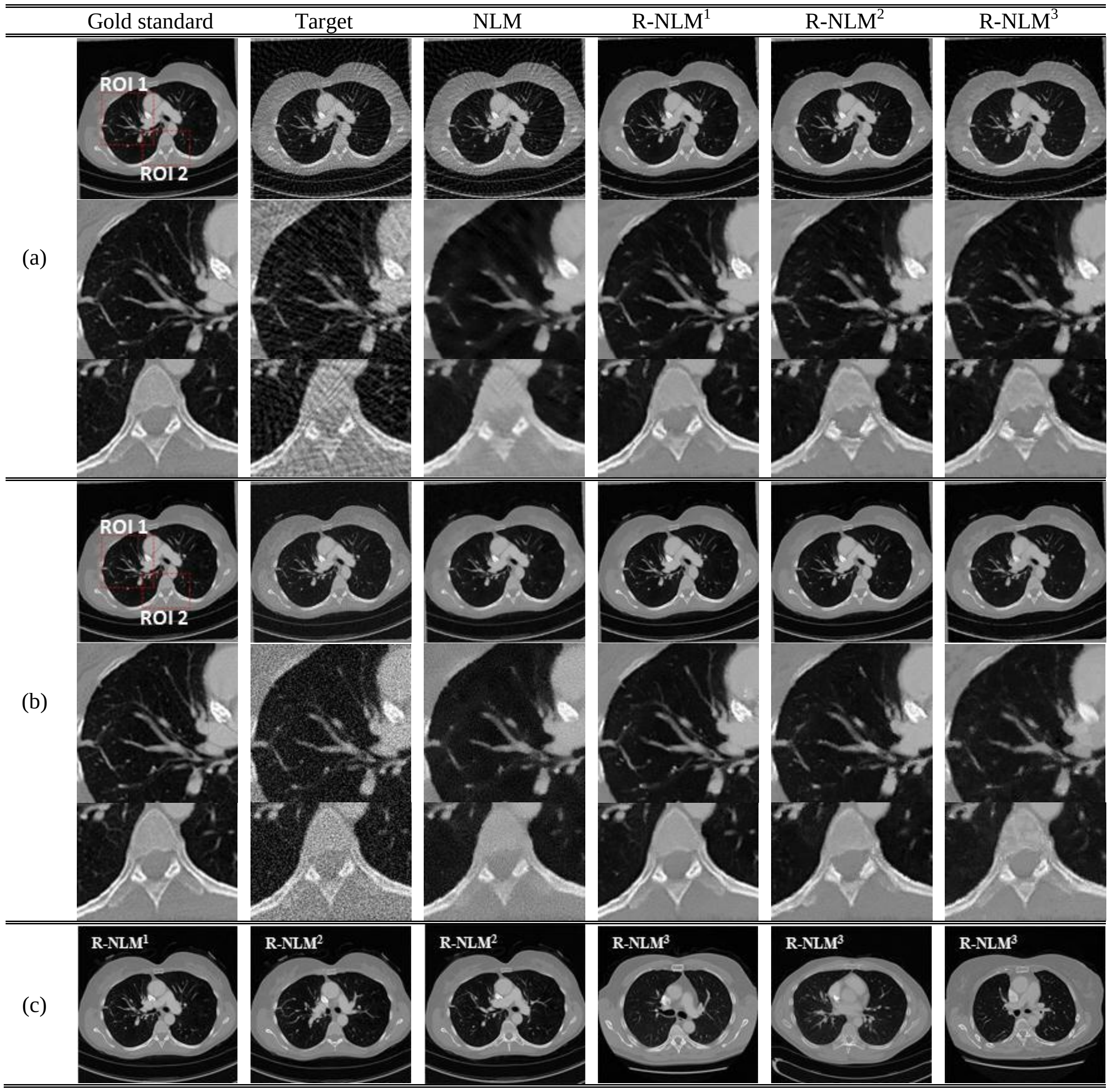
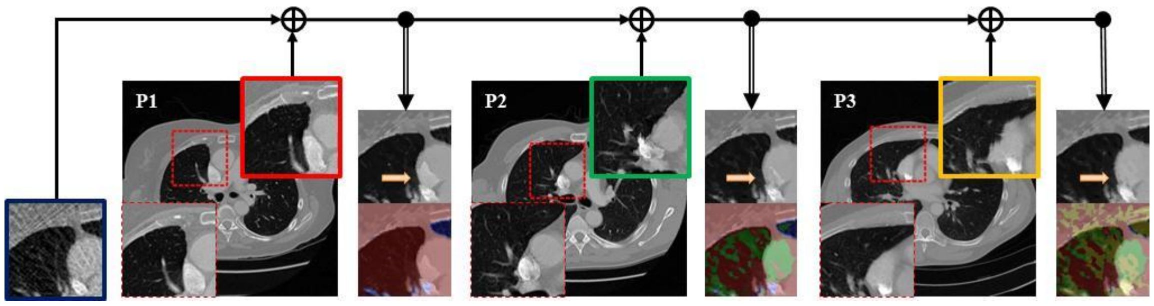


Figure 9.4: Restoration results for (a) streaks (86 projections) and (b) noise (30 SNR dB) using (c) different sets of priors. R-NLM^1 is a prior image from the same patient – this image prior is from scan of the same slice and thus very similar to the target. R-NLM^2 are two different image priors also from the same patient but from different slices and so less similar to the target. R-NLM^3 are three priors from three different patients located automatically in our data base. The corresponding restorations are shown in rows (a) and (b) along with results obtained with the standard NLM approach.



(a) (b) (c) (d)

Figure 9.5: Multiple priors: effect and benefits. By increasing the number of priors used for scan restoration (here, streak reduction), we gain a much wider range of anatomical features. This prevents the borrowing of pixel values from wrong structures, or the failure of finding structures at all, during the restoration process. (a) Target block subject to denoising. (b-d) Restoration using: (b) only prior P1, (c) prior P1 and P2, and (d) priors P1, P2, and P3. In each sub-figure, the prior is shown with the matched block marked by a red-dotted outline. The two inserted (zoomed) images are: (bottom left) the original block and (top right) the block aligned to (a). The two stacked images to the immediate right are (top) the restoration result and (bottom) the same block with the pixels colored by the prior from which they originate. The (blue, green, yellow) pixels come from prior (P1, P2, P3). We observe that only when using all three priors the structure pointed to by the arrow gets restored in the most plausible way, according to its noisy counterpart in (a).

9.3.2 Data Quality

To test robustness, we lowered the quality of the data by approximately 20% and observed the restoration outcome obtained with a database that only contained scans from other patients (case 3 above). In one experiment we reduced the number of projections from 86 to 70, which is a further dose reduction of about 20%. In the other experiment we increased the level of sinogram noise further, from 30dB to 25dB SNR. To calculate the decrease in dose we can use the relationship SNR=$N/\sigma=N/\sqrt{N}=\sqrt{N}$, where σ is the level of quantum noise and N is the number of X-ray photons. Since the dose is directly and linearly related to N we achieve a further reduction in dose by 30%. Our experimental results are shown in Figure 9.6. We observe that the outcome is still acceptable – all major features are still well preserved. This is especially true for the noise case, while the streak case would probably benefit from an alternate iterative CT reconstruction scheme such as SART (as opposed to the current FDK).

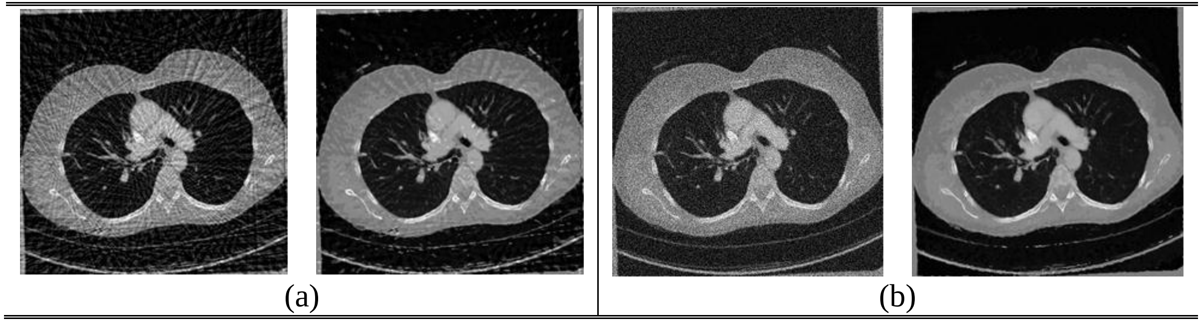


Figure 9.6: Robustness of (a) streak, and (b) noise reduction: The restorations use the three priors introduced in Figure 9.4 (denoted R-NLM^3). The target images are generated with 20% less projections (70) (a, left) or 20% more noise (25 SNR dB) (b, left) than in the study of Figure 9.4. The restored images for each (a) and (b) are shown on the target's right.

9.4 Conclusions

We proposed a general framework for high quality restoration of low-dose CT scans with the help of a general CT image database. We believe our approach is attractive because once the database has been established the online restoration process is quite fast. The restoration process itself, once the priors have been selected from the database, is just a minor extension to standard NLM filtering which is easy to implement and efficient to run.

Our results point out that a sufficiently elaborate database is crucial to the success of our method. Since modern PACS systems now have massive CT data on cheap disks we do not see this as a major obstacle. Future work will focus on enriching our database with more data, also of pathologies, and so create a system that requires our conservative fallback NLM scheme only in rare cases. In addition, in order to still maintain a manageable set of priors we are currently working on transforming our present image-based prior set to a patch-based set. This will eliminate redundancies and provide for a more diverse feature set to base the matching on. Also, we intentionally did not embed our method into an iterative reconstruction pipeline such as SART or ASD-POCS [101]. We wanted to test how far a single restoration step can take us. Next, we will iterate the databases-assisted restoration with a data-driven reconstruction which will likely improve the results further and also serve for verification.

We close by stating that the foremost purpose of a CT scan is to gain insight into a patient's state of health. Low-dose imaging is also important, but it should not be at the expense of increasing the likelihood of false positives or negatives. Our approach uses techniques from machine learning to improve the quality of low-dose CT scans. It grows by the quality of the examples it is taught with and the sophistication of its algorithms. Without sounding too futuristic, this is not too dissimilar from the educational process of a radiologist. We have clearly a long way to go, and we will likely never be able to match the tremendous capabilities of the human brain, but we may achieve a reliable digital doctor’s assistant.

Chapter 10

Conclusions

In this dissertation, we have presented a new GPU-accelerated framework targeting many unsolved issues with low-dose CT reconstruction and restoration problems in consideration of both efficiency and quality. A set of iterative reconstruction methods generalizing two popular algebraic methods was devised and mapped to the architecture of GPUs. We find that the special architecture and programming model of GPUs add extra constraints on the real-time performance of ordered subsets algorithms, counteracting the speedup benefits of smaller subsets observed on CPUs. This gives rise to new relationships governing the optimal number of subsets as well as relaxation factor settings for obtaining the smallest wall-clock time for reconstruction – a factor that is likely application-dependent. A high performance 3D reconstruction framework for electron tomography (ET) was shown as a specific low-dose CT application. We specially designed a framework to gain more benefit in performance and we also proposed limited detector/long object compensation method to address the vignetting issue appeared in ET reconstruction.

To handle issues arisen from low-dose CT such as noise and streak artifacts, starting from a successful introduction of bilateral filter to the regularization of the iterative reconstruction process, we explored and employed several nonlinear neighborhood filters as regularization step and compared their performance for both time and quality to the traditional method total variation minimization. Among the filters, patch-based technique such as non-local means with optimal adaptive setting is a powerful approach not just for denoising but also for other applications such as metal artifact reduction.

Then we focused on the effective approaches on this framework to search the optimal parameter settings for the reconstruction process balancing multiple performance objectives. We provided two variants to learn parameters associated with the regularized iterative algorithms – GPU acceleration-assisted exhaustive benchmark testing and multi-objective optimization providing a set of tradeoff options. A parameter space visualizer was introduced as an interactive parameter selection tool to enable user interaction. We believe that our parameter learning approach has much more general applications than shown here. Parameters are commonplace in many applications and finding optimal settings for these can be a laborious task. By using GPUs to test all possible parameter settings in parallel optimal settings can be found fast and accurately.

While summarizing the research on the performance of the neighborhood filters, we found that the non-local means filter provided a nice platform to incorporate prior information into the filtering process. Eventually, this prior information was installed as the extended search window providing both the weight and high-quality content references. This led to a new platform for reference-based image restoration. Based on this platform, when the prior scan of the same patient is available, a simple two-step image restoration method was developed. To consider the case more generally, the use of the database of other patients was also experimentally developed which shows its potential to improve the quality even when the scan of the same patient is unavailable.

As future work, a CUDA based implementation will bring further acceleration to the performance of the framework especially for the image restoration part. In fact, to keep the-state-of-the-art technology updated steers the leading role of the framework more easily. While following the new GPGPU programming techniques on the latest GPU boards, new features of the algorithms might be discovered. Another interesting direction is to research on the database reference-based method. The registration algorithm SIFT-flow could work well only when two images are not deformed or different too much from each other. This demands the database to be complete enough while in practice this is not always feasible. It becomes the bottleneck to the development of the restored quality. Although our block-wise approach loosens the restriction of the registration, it still has space of development. In the future, considering removing the registration operation while not affecting the matching of the features from two images would probably direct the research to a brand new way. As pointed in last chapter, the image database could be replaced with the patch database. Finally, we hope to show the framework to doctors to demonstrate its application value in clinical practice.

Bibliography

[1] A. Andersen, "Algebraic reconstruction in CT from limited views," *IEEE Trans. on Medical Imaging*, Vol. 8, pp. 50-55, 1989.

[2] A. Andersen, A. Kak, "Simultaneous Algebraic Reconstruction Technique (SART): a superior implementation of the ART algorithm," *Ultrasonic Imaging*, Vol. 6, pp. 81-94, 1984.

[3] E. C. Beckmann, "CT scanning the early days," *The British Journal of Radiology*, Vol. 79 (937), pp. 5–8, 2006.

[4] A. Buades, B. Coll and J.-M. Morel, "A Review of Image Denoising Algorithms with A New One," *Multi-scale Modeling Simulation*, Vol. 4(2), pp. 490-530, 2005.

[5] A. Buades, B. Coll and J.-M. Morel, "The Staircasing Effect in Neighborhood Filters and its Solution," *IEEE Trans. Img. Proc.*, Vol. 15(6), pp. 1499-1505, 2006.

[6] T. M. Benson and J. Gregor, "Modified simultaneous iterative reconstruction technique for faster parallel computation," *IEEE Nuclear Science and Medical Imaging Symposium*, pp. 2715-2718, 2005.

[7] D. Brenner, E. Hall, "Computed tomography - an increasing source of radiation exposure," *New England Journal of Medicine*, Vol. 357, pp. 2277-2284, 2007.

[8] N. Burningham, Z. Pizli, et al, "Image Quality metrics," *Conf. Image Processing, Image Quality, Image Capture Systems*, pp. 598-616, 2003.

[9] J. Bian, J. Siewerdsen, X. Han, E. Sidky. J. Prince, C. Pelizzari, and X. Pan, “Evaluation of sparse-view reconstruction from flat-panel-detector cone-beam CT,” *Phys. Med. Biol*., Vol. 55, pp. 6575–6599, 2010.

[10] A. Chambolle, "An algorithm for total variation minimizations and applications,*" J. Math. Imaging Vis*., Vol. 20(1-2), pp. 89-97, 2004.

[11] Y. Censor, "On block-iterative maximization," *J. Information and Optimization Science*, Vol. 8, pp. 275-291, 1987.

[12] C. A. Coello, "A short tutorial on evolutionary multiobjective optimization," *Proc. 1st Int. Conf. Evolutionary Multi-Criterion Optimization*, pp. 21-40, 2001.

[13] G. Csurka, C. Bray, and L. Fan et al., “Visual categorization with bags of keypoints,” *ECCV Workshop on Stat. Learn. in Comp. Vis*. 2004.

[14] B. Cabral, N. Cam and J. Foran, "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware," *ACM Symposium on Volume visualization*, pp. 91-98, 1994.

[15] K. Chidlow and T. Möller, "Rapid emission tomography reconstruction," *Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics*, pp. 15-26, Tokyo, Japan, 2003.

[16] M. R. Charest and P. Milanfar, "On Iterative Regularization and Its Application," *IEEE Trans. Circuits and Systems for Video Tech.*, Vol. 18(3), pp. 406-411, 2008.

[17] T. Chan and F. Park, "Data Dependent Multiscale Total Variation Based Image Decomposition and Contrast Preserving Denoising," *UCLA CAM Report*, pp. 04-15, 2004.

[18] J. Chen, S. Paris and F. Durand, "Real-time Edge-aware Image Processing with the Bilateral Grid," *ACM Trans. on Graphics*, Vol. 26(3), pp. 103-111, 2007.

[19] A. Criminisi, P. Pérez, and K. Toyama, "Region Filling and Object Removal by Exemplar-based Inpainting," *IEEE Trans. Image Process*, Vol. 13(9), pp. 1200-1212, 2004.

[20] E. Candes, J. Romberg, T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency Information," *IEEE Trans. Information Theory*, Vol. 52, pp. 489-509, 2006.

[21] G. Chen, J. Tang and S. Leng, "Prior Image Constrained Compressed Sensing (PICCS): A Method to Accurately Reconstruct Dynamic CT Images from Highly Undersampled Projection Data Sets," *Med. Phys*., Vol. 35(2), pp. 660-663, 2008.

[22] P. Choudhury and J. Tumblin, "The Trilateral Filter for High Contrast Images and Meshes," *Proc. EG Symp. on Rendering*, pp. 1-11, 2003.

[23] T. F. Chan, Y. Wang and H. M. Zhou, "Multiscale Total Variation and Multiscale Anisotropic Diffusion Algorithms for Image Denoising," submitted to *Applied and Computational Harmonic Analysis*, 2008.

[24] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval, " *Proc. ICCV*, 2007.

[25] P. Coupe, P. Yger, and S. Prima et al., "An optimized blockwise non local means denoising filter for 3D magnetic resonance images," *Trans. on Med. Imag*., Vol. 27, pp. 425–441, 2008.

[26] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, "A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II," *Proceedings of sixth international conference on parallel problem solving from nature*, pp.18-20, 2000.

[27] D. Castano-Diez, D. Moser, A. Schoenegger, S. Pruggnaller, and A.S. Frangakis, "Performance evaluation of image processing algorithms on the GPU," *J. Struct. Biol.*, Vol. 164, pp. 153-160, 2008.

[28] D. Castano-Diez,, H. Mueller, and A.S. Frangakis, "Implementation and performance evaluation of reconstruction algorithms on graphics processors," *Journal of Structural Biology*, Vol. 157(1), pp. 288-295, 2007.

[29] M. Defrise, F. Noo, and H. Kudo, "A solution to the long-object problem in helical cone-beam tomography," *Physics in Medicine and Biology*, Vol. 45, pp. 623-644, 2000.

[30] C. Castano-Diez, A. Seybert, and A.S. Frangakis, "Tilt-series and electron microscope alignment for the correction of the non-perpendicularity of beam and tilt-axis," *Journal of Structural Biology*, Vol. 154, pp. 195-205, 2006.

[31] M. Elad, "On the Bilateral Filter and Ways to Improve It," *IEEE Trans. Img. Proc.*, Vol. 11(10), pp. 1141-1151, 2002.

[32] M. Elad, M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image Proc.*, Vol. 15(12), pp. 3736-3745, 2006.

[33] E. Eisemann and F. Durand, "Flash Photography Enhancement via Intrinsic Relighting," *ACM Trans. Graphics*, Vol. 23(3), pp. 673-678, 2004.

[34] A. Efros and W. Freeman, "Image Quilting for Texture Synthesis and Transfer," *Proc. Siggraph*, pp. 341-346, 2001.

[35] T. Emrich, F. Graf, and H.-P. Kriegel et al., “CT Slice Localization via Instance-Based Regression,” *Proc. SPIE*, vol. 7623, 76232O, 2010.

[36] J.-J Fernández, "High performance computing in structural determination by electron cryomicroscopy," *Journal of Structural Biology*, Vol. 164, pp. 1-6, 2008.

[37] J. Frank, "Methods for Three-Dimensional Visualization of Structures in the Cell," *Electron Tomography*, 2006.

[38] R. Fattal, M. Agrawala and S. Rusinkiewicz, "Multiscale Shape and Detail Enhancement from Multi-light Image Collections, " *Proc. Siggraph*, 2007.

[39] J.-J. Fernández, J.-M. Carazo, and I. García, "Three-dimensional reconstruction of cellular structures by electron microscope tomography and parallel computing," *Journal of Parallel and Distributed Computing*, Vol. 64, pp. 285-300, 2004.

[40] L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone beam algorithm," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, pp. 612-619, 1984.

[41] K. Fatahalian and M. Houston, "A closer look at GPUs," *Communications of the ACM*, Vol. 51(10), pp. 50-57, 2008.

[42] R. Fernando and M.J. Kilgard, "The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics," *Addison-Wesley Professional*, 2003.

[43] J. Frank and B.F. McEwen, "Alignment by Cross-Correlation," *Electron Tomography*, pp. 205-214, 1992.

[44] S. Fu, Q. Ruan, W. Wang and Y. Li, "Adaptive Anisotropic Diffusion for Ultrasonic Image Denoising and Edge Enhancement," *Int'l J. Inform. Tech.*, Vol. 4(2), 2004.

[45] P. Gilbert, "Iterative methods for the 3D reconstruction of an object from projections," *J. Theoretical Biology*, Vol. 76, pp. 105-117, 1972.

[46] R. Gordon, R. Bender and G. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography," *J. Theoretical Biology*, Vol. 29, pp. 471-481, 1970.

[47] G. Herman and R. Davidi, "Image reconstruction from a small number of projections," *Inverse Problems*, Vol. 24(4), 2008.

[48] J. Hays, A. Efros, "Scene completion using millions of photographs," *ACM Trans. on Graphics*, Vol. 26(3), pp. 4-10, 2007.

[49] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE TSAP*, vol. 3, no. 6, pp. 613-623, 1973.

[50] H. Hudson and R. Larkin, "Accelerated Image Reconstruction Using Ordered Subsets of Projection Data," *IEEE Trans. Medical Imaging*, Vol. 13, pp. 601-609, 1994.

[51] L. Itti, C. Koch, E. Niebur, "A model of saliency-based visual attention for rapid scene analysis." *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 20(11), pp. 1254-1259, 1998.

[52] X. Jia, Y. Lou, R. Li, W. Song, S. Jiang, "GPU-based fast cone beam CT reconstruction from undersampled and noisy projection data via total variation," *Medical Physics*, vol. 37, pp. 3441-3447, 2010.

[53] C. Kervrann and J. Boulanger, "Optimal Spatial Adaptation for Patch-based Image Denoising," *IEEE Trans. Img. Proc.*, Vol. 15(10), pp. 2866-2878, 2006.

[54] J. Kole, F. Beekman, "Evaluation of accelerated iterative x-ray CT image reconstruction using floating point graphics hardware", *Physics in Medicine and Biology*, Vol. 51, pp. 875-889, 2006.

[55] Z. Kelm, D. Blezek, B. Bartholmai, B. Erickson, "Optimizing non-local means for denoising low-dose CT," *IEEE Symp. on Biomedical Imaging (ISBI)*, pp. 662-665, 2009.

[56] A. Konak, D. W. Coit and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, Vol. 91, pp. 992-1007, 2006.

[57] B. Keck, H. Hofmann, et al., "GPU-accelerated SART reconstruction using the CUDA programming environment," *Proc. SPIE*, Q7258, 2009.

[58] S. Keeling and R. Stollberger, "Nonlinear Anisotropic Diffusion Filtering for Multiscale Edge Enhancement," *Inverse Problems*, Vol. 18, pp. 175-90, 2002.

[59] V. Kwatra, A. Schodl, I. Essa, G. Turk and A. Bobick, "Graphcut Textures: Image and Video Synthesis using Graph Cuts," *Proc. Siggraph*, 2003.

[60] A. C. Kak and M. Slaney, "Principles of Computerized Tomographic Imaging," *IEEE Press*, 1988.

[61] M. C. Lawrence, "Least-squares method of alignment using markers," pp. 197-204, *Electron Tomography*, 1992.

[62] D. Lowe, "Object recognition from local scale-invariant features," *Inter. Conf. on Computer Vision*, pp. 1150-1157, 1999.

[63] A. Lawrence, S. Phan, R. Singh. "Parallel Processing and Large-Field Electron Microscope Tomography," *WRI World Congress on Computer Science and Information Engineering*, Vol. 3 pp. 339-343, 2009.

[64] R. L. Lagendijk, J. Biemond, and D. E. Boekee, "Regularized Iterative Image Restoration with Ringing Reduction," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 36, pp. 1874-1887, 1988.

[65] R. L. Lagendijk and J. Biemond, "Iterative Identification and Restoration of Images," *Boston, MA: Kluwer*, 1991.

[66] V. Lucic, F. Forster, W. Baumeister, "Structural studies by electron tomography: from cells to molecules," *Annu Rev Biochem*., Vol. 74, pp. 833-865, 2005.

[67] Z. Lu, Q. Feng, P. Shi, W. Chen, "A fast 3-D medical image registration algorithm based on equivalent meridian plane*," IEEE Intern. Conf. on Image Processing*, pp. 357-360, 2007.

[68] W. Lin, C.-C. Jay Kuo, "Perceptual visual quality metrics: a survey," *J. Visual Communication and Image Representation*, Vol. 22(4), pp. 297-312, 2011.

[69] Z. Lin and Q. Shi, "An Anisotropic Diffusion PDE for Noise Reduction and Thin Edge Preservation," *Proc. 10th Int. Conf. Image Analysis and Processing*, pp. 102-107, 1999.

[70] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” *Proc. CVPR*, vol. 2, pp. 2169–78, 2006.

[71] Z. Li, L. Yu, J. Trzasko, J. Fletcher, C. McCollough, A. Manduca, "Adaptive Nonlocal Means Filtering Based on Local Noise Level for CT Denoising," *Proc. SPIE 8313,* 2012.

[72] C. Liu, J. Yuen, A. Torralba, “SIFT flow: dense correspondence across scenes and its applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 33(5), pp. 978-994, 2011.

[73] Y. Liu, J. Wang, X. Chen, Y. Guo and Q. Peng, "A Robust and Fast Non-Local Means Algorithm for Image Denoising," *Journal of Computer Science and Technology*, Vol. 23(2), pp. 270-279, 2008.

[74] B. Miao, R. Jeraj, S. Bao, and T. R. Mackie, "Adaptive Anisotropic Diffusion Filtering of Monte Carlo Dose Distributions," *Phys. Med. Biol.*, Vol. 48, pp. 2767-2781, 2003.

[75] J. Ma, J. Huang, Q. Feng, H. Zhang, H. Lu, Z. Liang, W. Chen, “Low-dose computed tomography image restoration using previous normal-dose scan,” *Medical Physics*, Vol. 38(10), pp. 5713, 2011.

[76] M. Mahmoudi and G. Sapiro, “Fast Image and Video Denoising via Nonlocal-Means of Similar Neighborhoods,” *IEEE Signal Processing Letters*, Vol. 12(12), pp. 839–842, 2005.

[77] F. Mettler, B. Thomadsen, M. Bhargavan, D. Gilley, J. Gray, J. Lipoti, J. McCrohan, T. Yoshizumi, M. Mahesh, "Medical Radiation Exposure in the U.S. in Preliminary Results," *Health Physics*, Vol. 95(5), pp. 502-507, 2008.

[78] K. Mueller and F. Xu, "Practical considerations for GPU-accelerated CT," *IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1184-1187, Arlington, VA, 2006.

[79] K. Mueller, R. Yagel, and J.J. Wheller, "Anti-aliased 3D cone-beam reconstruction of low-contrast objects with algebraic methods," *IEEE Transactions on Medical Imaging*, Vol. 18, pp. 519-537, 1999.

[80] K. Mueller, R. Yagel, "Rapid 3D cone-beam reconstruction with the Algebraic Reconstruction Technique (ART) by using texture mapping hardware," *IEEE Transactions on Medical Imaging*, Vol. 19(12), pp. 1227-1237, 2000.

[81] F. Natterer, "The Mathematics of Computerized Tomography," *Wiley/Teubner*, 1886.

[82] J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A.E. Lefohn, T. Purcel, "A Survey of General-Purpose Computation on Graphics Hardware," *Proc. Eurographics*, pp. 21-51, 2005.

[83] A. Oliva and A. Torralba, “Building the gist of a scene: The role of global image features in recognition,” *Visual Perception, Progress in Brain Research*, vol. 155, 2006.

[84] S. Paris and F. Durand, "A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach," *Int'l J. Comp. Vision*, Vol. 81(1), pp. 24-52, 2009.

[85] S. Paris, P. Kornprobst, J. Tumblin and F. Durand, "Bilateral Filtering: Theory and Applications," *Foundations and Trends in Computer Graphics and Vision*, Vol. 4(1), pp. 1-73, 2008.

[86] P. Pérez, M. Gangnet, A. Blake, "Poisson image editing, " *ACM Trans. on Graphics*, Vol. 22(3), pp. 313-318, 2003.

[87] T. Pock, M. Unger, D. Cremers, H. Bischof, "Fast and Exact Solution of Total Variation Models on the GPU," *Comp. Vis. Patt. Reco.*, pp. 1–8, 2008.

[88] T. Pock, M. Unger, D. Cremers and H. Bischof, "Fast and Exact Solution of Total Variation Models on the GPU," *Comp. Vis. Patt. Reco.*, pp. 1–8, 2008.

[89] T. Q. Pham and L. J. van Vliet, "Separable Bilateral Filtering for Fast Video Preprocessing," *Proc. IEEE Int'l Conf. on Multimedia and Expo*, 2005.

[90] E. T. Quinto, U. Skoglund, O. Oktem, "Electron lambda-tomography," *Proc Natl Acad Sci*, 2009.

[91] M. Radermacher, "Weighted back-projection methods," *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell*, pp. 245-273, 2006.

[92] L. Rudin, S. Osher and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D*, Vol. 60, pp. 259–268, 1992.

[93] P. Suetens, "Fundamentals of Medical Imaging", *Cambridge University Press*, March 2002.

[94] D. M. Strong, P. Blomgren and T. F. Chan, "Spatially Adaptive Local Feature-Driven Total Variation Minimizing Image Restoration," *Proc. SPIE*, Q3167, 1997.

[95] T. Schiwietz, T. Chang, P. Speier, and R. Westermann, "MR image reconstruction using the GPU," *Proc. SPIE*, Vol. 6142, pp. 1279-1290, 2006.

[96] D. Strong and T. Chan, "Edge-preserving and scale-dependent properties of total variation regularization," *Inverse Problems*, Vol. 19, pp. S165–S187, 2003.

[97] E. Y. Sidky, C.-M. Kao, X. Pan, "Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT," *J. X-ray Sci.Tech.*, Vol. 14, pp. 119–139, 2006.

[98] M. Schmeisser, B.C. Heisen, M. Luettich, B. Busche, F. Hauer, T. Koske, K.H. Knauber, H. Stark. "Parallel, distributed and GPU computing technologies in single-particle electron microscopy," *Acta Crystallogr. D Biol. Crystallogr.*, Vol. 65, pp. 659-671, 2009.

[99] K. Sandberg, D.N. Mastronarde, and G. Beylkina, "A fast reconstruction algorithm for electron microscope tomography," *Journal of Structural Biology*, Vol. 144, pp. 61–72, 2003.

[100] U. Skoglund, L.G. Ofverstedt, R.M. Burnett, and G. Bricogne, "Maximum-entropy three-dimensional reconstruction with deconvolution of the contrast transfer function: a test application with adenovirus," *J Structural Biology*, Vol. 117, pp. 173-188, 1996.

[101] E. Sidky, X. Pan, “Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization,” *Phys. Med. Biol*, Vol. 53(17) pp. 4777-4807, 2008.

[102] R.H.M. Schoenmakers, R.A. Perquin, T.F. Fliervoet, W. Voorhout, H. Schirmacher. "New software for high resolution, high throughput electron tomography," *Microscopy and Analysis*, Vol. 19(4), pp. 5-6, 2005.

[103] L. Shepp, Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Trans. on Medical Imaging*, Vol. 1, pp. 113-122, 1982.

[104] J. Thirion, "Image matching as a diffusion process: an analogy with Maxwell’s demons," *Medical Image Analysis*, Vol.2 pp. 243–60, 1998.

[105] T. Tasdizen, "Principal neighborhood dictionaries for nonlocal means image denoising," *IEEE Trans. Img. Proc.*, Vol. 18(12), pp. 2649-2660, 2009.

[106] M. Tomassini, "A survey of parallel genetic algorithms," *World Scientific III*, pp. 87–118, 1995.

[107] C. Tomasi, R. Manduchi, "Bilateral filtering for gray and color images," *Intern. Comf. Comp. Vis.*, pp. 839–846, 1998.

[108] T. Thaipanich, P. Wu and C.-C. Kuo, "An Adaptive Nonlocal Means Scheme for Medical Image Denoising," *SPIE Medical Imaging*, 2010.

[109] B. Weiss, "Fast Median and Bilateral Filtering," *ACM Trans. on Graphics*, Vol. 25(3), pp. 519–526, 2006.

[110] A. Wong, "Adaptive Bilateral Filtering of Image Signals using Local Phase Characteristics," *Signal Processing*, Vol. 88(6), pp. 1615-1619, 2008.

[111] Z. Wang, A.C. Bovik, et al, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. on Img. Proc.*, 2004.

[112] Z. Wang, G. Han, T. Li, and Z. Liang, "Speedup OS-EM image reconstruction by PC graphics card technologies for quantitative SPECT with varying focal-length fan-beam collimation," *IEEE Transactions on Nuclear Science*, Vol. 52, pp. 1274-1280, 2005.

[113] G. Wang, M. Jiang, "Ordered-subset simultaneous algebraic reconstruction techniques (OS-SART)," *Journal of X-Ray Science and Technology*, Vol. 12, pp. 169-177, 2004.

[114] X. Xue, A. Cheryauka, D. Tubbs, "Acceleration of fluoro-CT reconstruction for a mobile C-Arm on GPU and FPGA hardware: a simulation study," *Proc. SPIE*, Vol. 6142, pp. 1494–501, 2006.

[115] F. Xu, and K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," *IEEE Transactions on Nuclear Science*, Vol. 52, pp. 654-663, 2005.

[116] F. Xu, and K. Mueller, "A comparative study of popular interpolation and integration methods for use in computed tomography," *IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1252-1255, 2006.

[117] F. Xu, and K. Mueller, "Real-Time 3D Computed Tomographic Reconstruction Using Commodity Graphics Hardware," *Physics in Medicine and Biology*, Vol. 52, pp. 3405–3419, 2007.

[118] F. Xu, K. Mueller, M. Jones, B. Keszthelyi, J. Sedat, D. Agard, "On the Efficiency of Iterative Ordered Subset Reconstruction Algorithms for Acceleration on GPUs," *Workshop on High-Performance Medical Image Computing & Computer Aided Intervention (MICCAI)*, New York, 2008.

[119] W. Xu, K. Mueller, "Learning Effective Parameter Settings for Iterative CT Reconstruction Algorithms," *Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, pp. 251-254, 2009.

[120] W. Xu and K. Mueller, "A Performance-Driven Study of Regularization Methods for GPU-Accelerated Iterative CT," *Proc. Fully 3D Image Recon. in Rad. and Nucl. Med.*, pp. 20-23, 2009.

[121] W. Xu, K. Mueller, "Accelerating Regularized Iterative CT Reconstruction on Commodity Graphics Hardware (GPU)," *IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1287-1290, 2009.

[122] W. Xu and K. Mueller, "Evaluating Popular Non-Linear Image Processing Filters for their Use in Regularized Iterative CT," *IEEE Medical Imaging Conference*, 2010.

[123] W. Xu and K. Mueller, "Parameter space visualizer: an interactive parameter selection interface for iterative CT reconstruction algorithms," *Proc. SPIE on Medical Imaging*, Vol. 7625, 76251Q, 2010.

[124] W. Xu, K. Mueller, "A reference image database approach for NLM filter-regularized CT reconstruction," *Proc. Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, pp. 116-119, 2011.

[125] W. Xu, K. Mueller, "Efficient low-dose CT artifact mitigation using an artifact-matched prior scan, " *Medical Physics*, Vol. 39 (8), pp. 4748-4760, 2012.

[126] F. Xu, W. Xu, M. Jones, B. Keszthelyi, J. Sedat, D. Agard, K. Mueller, "On the Efficiency of Iterative Ordered Subset Reconstruction Algorithms for Acceleration on GPUs," *Computer Methods and Programs in Biomedicine*, Vol. 98(3), pp. 261-270, 2010.

[127] W. Xu, F. Xu, M. Jones, B. Keszthelyi, J. Sedat, D. Agard and K. Mueller, "High-Performance Iterative Electron Tomography Reconstruction with Long-Object Compensation using Graphics Processing Units (GPUs)," *J. Structural Biology*, Vol. 171(2), pp. 142-153, 2010.

[128] W. Xu, S. Ha, K. Mueller, "Database-assisted low-dose CT image restoration," *Proc. 2nd International Conference on Image Formation in X-ray Computed Tomography*, pp. 111-114, Salt Lake City, UT, July 2012.

[129] Q. Xu, H. Yu, X. Mou and G. Wang, “Dictionary learning based low-dose X-ray CT reconstruction,” *Proc. Fully3D*, pp. 258-261, Germany, 2011.

[130] Z. Wang, A. Bovic, "A universal image quality index," *IEEE Signal Processing Letters*, Vol. 9(3), pp. 81-84, 2002.

[131] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. on Image Processing*, Vol. 13(4), pp. 600-612, 2004.

[132] L. Yang, J. Zhao, G. Wang, “Few-view image reconstruction with dual dictionaries,” *Phys. Med. Biol.*, Vol. 57, pp. 173–189, 2012.

[133] D. C. Youla, "Generalized image restoration by the method of alternating orthogonal projections," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 694–702, 1978.

[134] H. Yu and G. Wang, "A soft-threshold filtering approach for reconstruction from a limited number of projections," *Phys. Med. Biol.*, Vol. 55(13), pp. 3905-3916, 2010.

[135] H. Yu, S. Zhao, E. Hoffman, G. Wang, "Ultra-low dose lung CT perfusion regularized by a previous scan," *Academic Radiology*, Vol. 16, pp. 363-373, 2009.

[136] M. E. Zervakis, "Nonlinear Image Restoration Techniques," Ph.D. dissertation, Univ. Toronto, 1990.

[137] B. Zhang and J. P. Allebach, "Adaptive Bilateral Filter for Sharpness Enhancement and Noise Removal," *IEEE Trans. Image Proc.*, Vol. 17(5), pp. 664-678, 2008.

[138] S. Q. Zheng, M. B. Braunfeld, J. W. Sedat, and D. A. Agard, "An improved stategy for automated electron microscopic tomography," *Journal of Structural Biology*, Vol. 147, pp. 91-101, 2004.

[139] H. Zhou, M. Chen, M.F. Webster, "Comparative Evaluation of Visualization and Experimental Results Using Image Comparison Metrics," *IEEE Visualization*, pp. 315-322, 2002.

[140] G. Zeng, G., G. Gullberg. "Unmatched projector/backprojector pairs in an iterative reconstruction algorithm," *IEEE Transactions on Medical Imaging*, Vol. 19(5), pp. 548–555, 2000.

[141] S. Q. Zheng, B. Keszthelyi, E. Branlund, J.M. Lyle, M.B. Braunfeld, J.W. Sedat, and D.A. Agard, "UCSF tomography: An integrated software suite for real-time electron microscopic tomographic data collection, alignment, and reconstruction," *Journal of Structural Biology*, Vol. 157, pp. 138-147, 2006.

[142] Z. Zheng, W. Xu and K. Mueller, "Performance Tuning for CUDA-Accelerated Neighborhood Denoising Filters," *The 3rd Workshop on High Performance Image Reconstruction (held with Fully 3D 2011)*, pp. 52-55, Potsdam, Germany, 2011.

[143] http://www.nvidia.com/object/cuda_develop.html

[144] http://www.fltk.org/

[145] http://www.gpgpu.org

[146] http://graphics.stanford.edu/projects/gpubench

[147] http://people.csail.mit.edu/celiu/SIFTflow