ABSTRACT

Essential Elements of Proton Computed Tomography for Practical Applications

Blake Edward Schultze, Ph.D.

Mentor: Keith Evan Schubert, Ph.D.

The work presented in this dissertation all pertains to developments of proton computed tomography (pCT) and the elements essential to its viability as a clinical imaging modality. This includes methodological and implementational developments for reducing reconstruction time and improving pCT image quality, each advancing pCT towards clinical viability. The corresponding methods are presented in the chronological order of their development.

Hull-detection, a method for differentiating voxels internal and external to an object, is presented first. Hull-detection was specifically developed for pCT as a preferable means for obtaining a binary image of the object, a preconditioning step often referred to as object detection. The concept of hull-detection, similar to the way a sculptor chisels away portions of material to produce the desired sculpture, is that voxels along the paths of protons that completely miss the object can be carved away to reveal the object hull. However, this neglects to account for the ramifications of uncertainties in the data, which was accounted for in different ways. Several hull-detection algorithms were developed and compared to the classic object detection method based on thresholding the filtered backprojection image.

The second topic presented is efficiently implementing the most-likely path (MLP) formalism for pCT. This formalism was developed to more accurately approximate proton paths within an object, increasing the achievable spatial resolution. Computing the MLP

is, by far, the most computationally expensive task performed during image reconstruction, making it the biggest hurdle to achieving clinically viable image reconstruction times (below 10 minutes). A computationally efficient implementation of the MLP was developed by simplifying the associated equations and incorporating several software design principles to reduce the number of compute operations and improve numerical stability.

The final topic presented is the incorporation of recent advancements of total variation superiorization (TVS) into pCT. A fixed parameter version of TVS was initially incorporated into the feasibility-seeking algorithms of pCT, which included a step verifying successful TV reduction. Presented here is the modern version of TVS applied to pCT, with user-control of parameters, removal of the verification step, and additional option to perform repeated perturbations.

Essential Elements of Proton Computed Tomography for Practical Applications

by

Blake Edward Schultze, B.S., M.S.

A Dissertation

Approved by the Department of Electrical and Computer Engineering

_____

Kwang Y. Lee, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of

Doctor of Philosophy

Approved by the Dissertation Committee

_____

Keith E. Schubert, Ph.D., Chairperson

_____

Robert J. Marks II, Ph.D.

_____

Jeffrey S. Olafson, Ph.D.

_____

Reinhard W. Schulte, Ph.D.

Accepted by the Graduate School

August 2021

_____

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Special thanks go to my long term mentor, Dr. Keith Schubert, and his wife Kym Schubert for their support throughout my academic journey. I would also like to thank my fellow Baylor student colleagues, professors, and support staff for their roles in my success, particularly Minnie Simcik, Brian Sitton, Pat Hynan, Kay Riddering, Robert Marks II, Paniz Karbasi, and Ritchie Cai. I would also like to extend my gratitude to those at CSUSB that offered guidance and support along the way, particularly Dr. Paul K. Dixon, Dr. Laura Woodney, and Dr. Ernesto Gomez.

Additional thanks go out to the pCT Collaboration and its members for their contributions to and support of my research; in particular, I would like to thank Reinhard W. Schulte, Vladimir A. Bashkirov, Yair Censor, Ford Hurley, Valentina Giacommetti, Christina Sarosiek, George Coutrakon, and Robert P. Johnson.

The research in proton CT presented in this dissertation was supported by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) of the National Institute of Health (NIH) and the National Science Foundation (NSF) award no. R01EB013118, and the United States - Israel Binational Science Foundation (BSF) grants no. 2009012 and no. 2013003. The content of this paper is solely the responsibility of Blake Edward Schultze and does not necessarily represent the official views of NBIB, NIH, or BSF. The support of UT Southwestern and State of Texas through a Seed Grants in Particle Therapy award is also gratefully acknowledged.

ATTRIBUTIONS

(1) B. E. Schultze, M. Witt, Y. Censor, K. E. Schubert, and R. W. Schulte, "Performance of hull-detection algorithms for proton computed tomography reconstruction," in *Infinite Products of Operators and Their Applications*, ser. Contemporary Mathematics, S. Reich and A. Zaslavski, Eds., vol. 636. American Mathematical Society, 2015, pp. 211–224.

- I (B. E. Schultze) was the sole investigator and primary author of the publication.
- M. Witt provided the simulated data sets used for the initial hull-detection investigations.
- Y. Censor, K. E. Schubert, and R. W. Schulte acted in a supervisory role on editing, content accuracy, and approval of the final form submitted for publication.

(2) B. E. Schultze, Y. Censor, P. Karbasi, K. E. Schubert, and R. W. Schulte, "An Improved Method of Total Variation Superiorization Applied to Reconstruction in Proton Computed Tomography," *IEEE Transactions on Medical Imaging*, vol. 39, no. 2, pp. 294–307, 2020

- I (B. E. Schultze) was the sole investigator and primary author of the publication.
- Y. Censor is among the original developers of the superiorization methodology, which serves as the theoretical framework of total variation superiorization, and requested the investigations be performed for pCT. He also provided approval of the investigation results and final form submitted for publication.
- P. Karbasi was a colleague working independently on a related topic who participated in discussions to ensure our investigations did not overlap. She also assisted in the editing of the final form submitted for publication.
- K. E. Schubert and R. W. Schulte acted in a supervisory role on editing, content accuracy, and approval of the final form submitted for publication.

CHAPTER ONE

Introduction

Interest in proton computed tomography (pCT) has increased due to the expanding use of proton therapy for cancer treatment and the potential for reducing range uncertainties in proton therapy using pCT for treatment planning and pretreatment verification [1, 2]. At present, x-ray CT imaging is used to develop a treatment plan for proton therapy, introducing additional range uncertainty due to the conversion from Hounsfield units (HU) (sometimes called CT numbers) to stopping power relative to water, referred to as relative stopping power (RSP). It has been common practice to account for uncertainties by adding a margin of 3.5% plus 1 mm to the nominal range of a proton beam [3], but this yields larger uncertainty margins between clinical and planning target volumes than those associated with photon therapy. Treatment planning using pCT images offers the potential to remove the HU-RSP conversion uncertainty by reconstructing RSP values directly, which can be far more accurate than the conversion in some materials. The low radiation dose of pCT imaging also presents the opportunity to perform weekly, or even daily, scans to track changes in patient anatomy and RSP distributions for adaptive proton therapy.

The possibility of pCT imaging had been proposed in the 1960s but, due to the complicated paths of protons and inadequate measurement technologies, it was deemed infeasible for decades. By the mid 2000s, high energy physics detector technology had advanced to the point where it was feasible to track individual protons and measure their energy loss. Coupled with the emerging interest in proton therapy, and ion therapy in general, this sparked the development of a theoretical framework for pCT and a Phase I preclinical proton scanner system at Loma Linda University [4, 5, 6]. The success of this project as a proof of concept subsequently led to the development of a Phase II preclinical scanner system which drastically improved the quality of scanner data [7, 8, 9].

Although the images reconstructed using the data from the Phase I/II preclinical scanner system successfully demonstrated a proof of concept, there were several image reconstruction hurdles remaining in advancing pCT to clinically feasibility. The image reconstruction software developed for the Phase I/II preclinical scanner had computation times far exceeding the clinically appropriate 10 min threshold. There were also aspects of the reconstruction software, both theoretical and practical, that could be improved to yield an increase in image quality. Around the time when the Phase II scanner was being completed, the image reconstruction software was advanced by two separate groups: one group at Northern Illinois University (NIU) and one group initially at California State University, San Bernardino (CSUSB) and later moving to Baylor University (BU).

The group at NIU initially focused primarily on reducing computation time down to clinically feasible times by restructuring the reconstruction software to run efficiently on an internode computation cluster. The CSUSB/BU group focused on rewriting the software from scratch, simultaneously identifying and correcting any theoretical or implementation errors and optimizing the code for execution on multi-GPU compute nodes. The early work performed at CSUSB culminated in the thesis of an MS in Computer Science. However, the majority of the pCT software design and development took place at BU and it is this work that is the topic of this PhD dissertation. Clinically feasible reconstruction times have been achieved on affordable computation hardware as a result of the work presented here. Image quality has also been improved, to the point where pCT can now be argued to be a viable imaging modality for proton treatment planning; the methods by which this was achieved are also presented here.

This dissertation includes a historical review and literature survey on the general technological development of pCT, which appears as Chapter 2. The methodological and implementational aspects of pCT that are common and relevant to each of the presented topics, as well as the associated image reconstruction software/hardware, are presented inChapter 3 (Methods). Due to the fact that several unrelated theoretical and practical

aspects of pCT are addressed, the corresponding work on each of these topics is presented in separate Chapters. Hence, in addition to the broad historical review and literature survey of pCT in Chapter 2, each Chapter also contains a narrow historical review and literature survey germane to the associated topic. Chapter 3 provides a general overview of the methods of pCT that are relevant to every topic presented in this dissertation. Information that is too long or otherwise inappropriate for inclusion within the Chapters, are provided as Appendices in the dissertation's back matter. The appendices are organized according to the order in which the corresponding material is referenced within the main text, with the content relevant to each chapter included in distinct, alphabetically numbered sections of the appendix.

The order in which Chapters 3–6 appear reflects the order in which the corresponding tasks are performed during image reconstruction; coincidentally, this is also the chronological order in which the associated methods were developed. To maintain conceptual flow, each topic has its own methods, results, and discussion/conclusions Sections, rather than collecting these from each topic and presenting them as separate chapters. There are Chapters for each of the three topics of pCT presented here, each corresponding to the methods developed for a particular task (or discrete step) of pCT image reconstruction. These topics are hull-detection, most-likely path (MLP) calculations, and total variation superiorization (TVS).

Chapter 4 presents hull-detection, a method for identifying the portions of the image containing the scanned object and excluding those corresponding to air. The early work on hull-detection took place while at CSUSB, but the first practically useful hull-detection method which is presented here, modified silhouette carving (MSC), was developed while at BU. The development of hull-detection also required the development of a three-dimensional line (or voxel) walking algorithm, so this is presented along with hull-detection. The object hull image is also useful in defining the initial iterate of image reconstruction, where it is applied as a mask to the median-filtered filtered backprojection

(FBP) image to set the RSP of all voxels outside the object to zero. This is not only of practical convenience, but it also enables the subsequent iterative image reconstruction to be performed within a reduced image space, thereby eliminating the unnecessary computations and undesired influence of voxels outside the object and improving computational efficiency.

Chapter 5 presents the details of how the most-likely path (MLP) formalism, developed and later refined for the Phase I scanner system, was implemented to reduce its computational burden. The MLP calculations represent the single largest computational burden in pCT image reconstruction and, thus, the biggest challenge to achieving clinically viable reconstruction times on a multi-GPU node. The scattering of a proton at a particular point along its path through an object depends on the energy it possesses at that point. Hence, the integral equations for the elements of the MLP's prior and posterior likelihood scattering matrices include an energy dependent term, which appears in the denominator of the integrands. An accurate analytical model of this relationship does not exist and is currently infeasible, so the energy dependent term is approximated by fitting a fifth-degree hexanomial to the scattering observed by conducting Monte Carlo simulations of 200 MeV protons. Replacing the energy dependent term of each integrand with the hexanomial fit yields integrable expressions for each element of the scattering matrices, but the definite integration of the approximated integrand results in a large number of polynomial terms of various degrees. definite integration of the product of the hexanomial with the other integrand terms elements the MLP calculations were expanded and simplified and the approach to storing and transferring the resulting MLP data was specifically designed to reduce both the amount of data and the computation time.

Chapter 6 presents the work associated with implementing and investigating the most recent version of total variation superiorization (TVS) as it applies to pCT. An older and more rigid form of TVS was used in the initial image reconstruction software for the

Phase I scanner, but this version lacked the flexibility to change parameter values and included safety checks that had been removed in other applications. There was also a structural change to the algorithm that had been proposed. The work presented here discusses each of the algorithmic changes to TVS and demonstrates their effectiveness as applied to pCT specifically. This work was published, in an alternative and abridged form, as:

The conclusory Chapter 7 provides a cumulative summary of the presented work and conclusions regarding the merit and impact of these contributions to pCT. Following this is a prospective on the future and advancement of pCT. This includes an overview of several advancements that have already been developed and implemented in the BU software, but have not yet been published.

Terminology that may be unfamiliar to readers and is not (adequately) defined within the text is italicized, indicating that the corresponding word(s) are defined for the present context in Appendix A.1. Likewise, mathematical notation relevant to pCT and the content of this dissertation is defined in Appendix A.2.

CHAPTER TWO

Historical Review and Literature Survey

Ion radiation therapy has only recently become a clinical option for cancer treatment, which has renewed interest in the development of proton computed tomography (pCT or proton CT), but the potential of these technologies were first proposed more than 50 years ago. In 1946, the physicist Robert Wilson published a paper citing the potential advantages of high energy (fast) protons and other heavier ions for radiation therapy [10]. In 1963 and 1964, Allan Cormack published papers introducing the concept of tomographic image reconstruction [11, 12], for which he was awarded the 1979 Nobel Prize in Medicine. Although Cormack understood the benefits of using protons for tomographic imaging, but he felt that the Coulombic scattering of protons would result in blurry images, leading him to the conclusion that x-rayswould be more useful than protons for tomographic imaging.

Despite Cormack's initial skepticism on the efficacy of tomographic imaging with protons, the 1970s and 1980s saw an increase in the technological development of proton and ion imaging. In 1975, a team at Lawrence Berkeley National Laboratory demonstrated the ability to track 900 MeV helium ions using a multi-wire proportional chamber (MWPC) and measure their residual range with a stack of plastic scintillators [13]. This provided the first indication that the concept of a particle-tracking CT scanner was technologically feasible. The team also used the acquired helium data to generate tomographic reconstructions, which were then compared to those generated with the commercially available x-ray CT scanners of the time. The dose of the single-particle-based helium CT was up to 50 times lower than that of the conventional x-ray CT, a fact that the team emphasized in discussing its potential beyond their proof of concept.

In the following year (1976), Cormack and his colleague at the Harvard Cyclotron Lab, Andreas Koehler, published a paper demonstrating the first experimental pCT images [14]. A scan of radially symmetric acrylic phantom, 9.52 cm in diameter, was performed with 158 MeV protons and the energy loss was measured. Analysis of the pCT images generated from this data demonstrated superior performance in terms of density resolution than the commercial x-ray CT scanners at that time; density differences as small as 0.006 g/cm^3 could be distinguished, relative to the bulk acrylic material (1.17 g/cm^3), in the pCT images as compared to the approximately 1% density resolution of x-ray CT at comparable imaging dose.

From the late 1970s to early 1980s, Kenneth Hanson and a team at the Los Alamos Meson Physics Facility (LAMPF) designed the first series of pCT experiments with the intention of closing the technological gap between physics laboratories and clinics. The first of the experimental pCT systems Hanson developed used an MWPC to measure the downstream exit position and a hyper-pure germanium detector (HPGe) to measure the residual energy of each proton. The system was later modified to include a range telescope composed of a stack of plastic scintillators. The performance and viability of the systems were assessed in terms of achieving the minimum dose at the highest feasible count rate, the results of which were presented in a series of landmark publications [15, 16, 17]. After an upgrade of the readout electronics, a subsequent experiment was performed in which sample human tissues were scanned and reconstructed as a proof of concept of the clinical feasibility of pCT. These papers led to several important conclusions about pCT, the first being that spatial resolution could be further improved by measuring the trajectory in addition to the position of protons exiting the phantom. Furthermore, it was concluded that the dose advantage of pCT, relative to x-ray CT, was approximately 4:1 for a 20 cm diameter phantom and 8:1 for a 30 cm diameter phantom for ideal systems.

Hanson's experiments at Los Alamos represented the end of the early proton and ion CT exploration phases. It would be nearly 20 years before development of the next phase

began. This renewed interest was motivated by the emerging clinical use of protons, which had begun to expand into hospital-based facilities at that time. As experience in proton therapy increased, it became clear that the range uncertainties introduced into treatment planning by the HU-RSP conversion of x-ray CT images would need to be addressed. Coincidentally, at this time there was also a marked absence of image guidance technology in the treatment rooms where proton therapy was performed. The resulting inability to track tissues that have shifted in position and/or RSP, either between treatments or as a result of internal organ movement during a treatment, leading to even larger range uncertainties. Knowledge of Hanson's work on pCT and its potential to reduce, or eliminate, some of the sources of range uncertainty led to an increase in support for further development of pCT.

A two day meeting at Brookhaven National Laboratory led to the formation of the U.S. pCT collaboration in early 2003. At this initial meeting, the collaboration discussed plans to develop a pCT system using the most advanced particle tracking and energy detector technology available at the time. The collaboration also identified the most-likely path (MLP) concept, which had been proposed by Schneider and Pedroni [18], for further development. The concept incorporates particle tracking information into the MLP formalism [19], which is subsequently combined with iterative image reconstruction techniques. The pCT collaboration successfully built a preclinical pCT scanner, funded by a 2011 grant award from the National Institute of Biomedical Imaging and Bioengineering (NIBIB), which was able to achieve the 1 million protons per second tracking and the 10 minute threshold on a 360 ° pCT acquisition of a head phantom objectives of the grant [20, 21].

A number of groups have emerged in the field of proton/ion imaging for proton/ion therapy over the last 20 years (e.g. see [22, 23]), including some than have developed different approaches than the pCT collaboration. A recent review provides an overview of the different approaches [24]. Proton data acquisition approaches are typically divided into two modes (types): particle tracking (list) mode and integration mode. Particle tracking mode

tracks and measures the energy loss of individual protons or ions traversing an object. Integration mode measures an integrated beam current, which depends on the water-equivalent thickness (WET) of proton or ion beams traversing the patient and is typically acquired with existing dosimetry equipment, using the lower intensity modes established for therapeutic beam delivery. The particle tracking mode is more complicated and more expensive than the integration mode, but it yields superior spatial resolution and lower patient dose [24], both of which are attractive attributes of a clinical pCT system.

CHAPTER THREE

Methods

Each of the following Chapters correspond with a particular aspect of pCT and the associated methods developed for each task. The order in which they are presented here coincides with the order in which the corresponding method(s) are performed during execution of the image reconstruction software, i.e., their execution order.

3.1 Image Reconstruction Software

The image reconstruction software developed and used at BU is designed such that each high level task, or step, of pCT is implemented as a separate functional unit. The main pCT program file calls these functional units in the appropriate order to enforce the computational structure of pCT. The partitioning of the image reconstruction steps into functional units is intended to capture and encapsulate the basic theoretical structure of pCT image reconstruction, a structure which is independent of advancing theory and methods, from those portions of the program that can be adapted now or in the future. There are varying methods for accomplishing each task, so methods are designed as independent modules, functionally equivalent to what are often called *helper functions*. The particular methods that are called at execution depend on the value of the associated *control variables*; control variables are boolean variables in the case of methods that can be turned on/off and enumerated type variables in cases where there are multiple methods that can be chosen. This programmatic structure provides a consistent framework for the use and advancement of pCT image reconstruction.

A great deal of effort was spent on parallelizing every method for which this was feasible. There are some portions of code that must be executed sequentially but for various reasons, are executed by a single thread of the GPU instead of by the host CPU. For

example, sequential calculations performed between two GPU *kernels* and that are dependent on existing GPU data may be more efficiently executed by a single GPU thread rather than transferring the dependent data to the host for computation and back to the GPU for use. This approach, where a single thread executes a sequential section of code, is herein referred to as *sequential gap parallelization (SGP)*. In some cases, data dependencies that prevent parallelization can be eliminated by separately generating the required data on each thread; often times this results in threads calculating one or more unnecessary values, but this has negligible importance relative to achieving parallelization. This approach is herein referred to as *dependency distribution parallelization (DDP)*. Note that it is possible to have a combination of these two cases, such as when the sequential portion of code generates data that the threads are subsequently dependent on, in which case the DDP approach is typically the most appropriate. As a result of the natively parallel and forcibly parallel steps in pCT, nearly all computation is performed on the GPU(s) and the host CPU operates primarily as the master in evaluating conditional executions and initiating data transfers.

The work presented in this dissertation is not only presented in the order it is performed during reconstruction, but this also the chronological order in which they were developed. For the purpose of performance comparisons, all methods unrelated to the method under development remained the same, providing the means to compare the newly developed method to the one previously used in accomplishing a particular task. Since this remained important throughout the development of each method presented here, the image reconstruction program was configured with the same control variable values in each case. Although many of the methods selected for these image reconstructions do not warrant discussion here, there are some methods that have a bearing on performance and could yield slightly different results than those presented here, so these are discussed here. Details on the acquisition, preprocessing, and preconditioning of pCT data for iterative image reconstruction can be found in the recently published article “Particle-Tracking Proton Computed

Tomography—Data Acquisition, Preprocessing, and Preconditioning" [25]. The terminology used in the following sections and the associated mathematical notation are defined in the Glossary and Notation sections, respectively, of Appendix A.

3.2 Methods Selected for Consistency

The BU image reconstruction program has the ability to perform several *feasibility-seeking* algorithms from the *fully simultaneous*, *block-iterative*, and *string-averaging* classes, as well as hybrids of these algorithmic classes. However, the feasibility-seeking algorithm used for iterative image reconstruction throughout the entirety of the following work is the *fully simultaneous* diagonally-relaxed orthogonal projection (DROP) algorithm, herein referred to as FS-DROP. This fully simultaneous projection method is based on the classic Cimmino algorithm [26]. The structural form of the Cimmino and FS-DROP algorithms is illustrated in Figure 3.1.

Although the block-iterative and string-averaging algorithms yield superior reconstructed image quality, there are historical and practical reasons why the FS-DROP algorithm remained the preferred approach in the work presented here. When the image reconstruction program was initially developed, the technological state of GPU hardware and the CUDA software platform made it difficult to attain computation times within the clinically viable 10 min threshold. Since the fully simultaneous algorithms require less frequent data transfers between host and GPU and are much more flexible in the way they can be parallelized, these algorithms were the natural choice. The Cimmino algorithm applies an equal weighting, $1/m$, to the contributions from each of the m hyperplane projections in the calculation of the image update. However, in the case of a highly sparse system matrix $\boldsymbol{A}$, such as those in pCT, it has been shown that a column (i.e. voxel) dependent weighting scheme based on the number of nonzero values in the jth column yields superior performance [27]. Hence, the FS-DROP algorithm was chosen instead of the Cimmino algorithm.

There are practical reasons why the use of the FS-DROP algorithm continued in later work. The fully simultaneous methods do not depend on the order in which projections

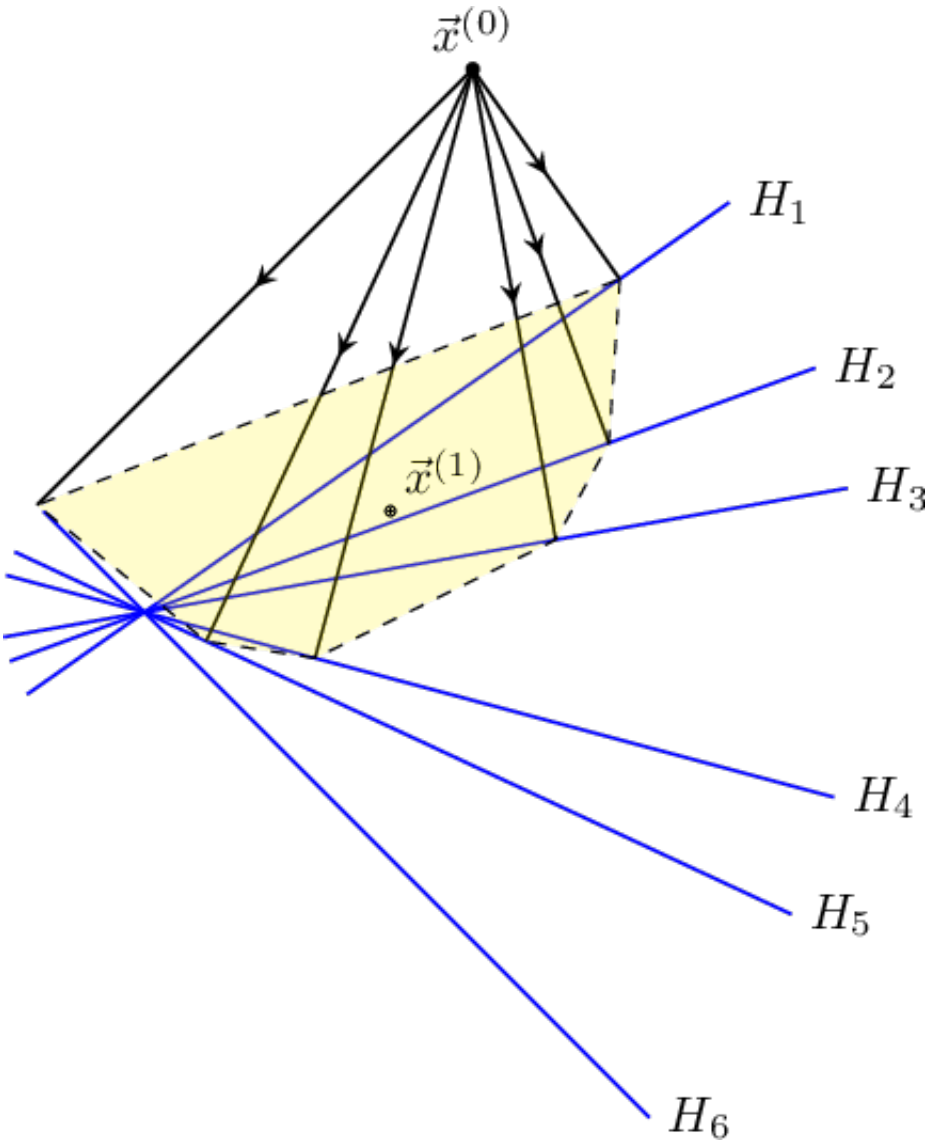


Figure 3.1: Illustration of the first *full iteration*, starting from the initial iterate $\vec{x}^{(0)}$, of a fully-simultaneous projection algorithm with $\lambda^{(0)} = 1$.

are performed, unlike block-iterative and string-averaging algorithms, yielding consistent performance across data sets and eliminating history ordering methods from consideration in the development and analysis of methodological changes. Since this development proceeded roughly chronologically, it also made sense to retain the methods employed in previous work such that the impacts of new methods can be isolated from other potential sources.

Although particular reconstruction algorithms have slightly better performance, either in terms of speed or image quality, each feasibility-seeking algorithm exhibits the same general behavior, particularly those that use a similar weighting scheme such as the component weighted algorithms used in pCT. Hence, approaches that yield improved image quality with one algorithm can be expected to yield similar, but not necessarily equal, improvements with other algorithms. Therefore, replacing FS-DROP with a block-iterative or string-averaging algorithm will not negate the legitimacy of any of the methodological developments or conclusions presented here.

The FS-DROP algorithm operates as follows: given a data set with m measured *proton histories* and an image vector $x^{(k)}$ composed of n voxels, the kth iteration proceeds by performing simultaneous orthogonal projections of the kth iterate onto each of the m hyperplanes. The residual error is then calculated for each hyperplane, subsequently normalized by its length, and then summed over all m hyperplanes. For each voxel j, the sum of normalized residuals is divided by the number of hyperplanes with nonzero jth column. The resulting component weighted sums are then added to the corresponding columns of the kth iterate to produce the $(k+1)$th iterate.

This can be expressed by the following equation:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + \lambda^{(k)} \boldsymbol{D}^{(k)} \sum_{i=1}^{m} \frac{b_i - \left\langle \vec{a}_i, \vec{x}^{(k)} \right\rangle}{\left\| \vec{a}_i \right\|^2} \vec{a}_i^T \tag{3.1}$$

$$\boldsymbol{D}^{(k)} = \underset{1 \le j \le n}{\operatorname{diag}} \left(\min \left(1, \frac{1}{d_j^{(k)}} \right) \right)$$

where $d_j^{(k)}$ is equal to the number of proton paths intersecting the jth voxel.

CHAPTER FOUR

Hull-Detection

Portions of the content presented in the following Chapter was previously published, in an alternative and abridged form (Attribution 1), in the following AMS book:

B. E. Schultze, M. Witt, Y. Censor, K. E. Schubert, and R. W. Schulte, "Performance of hull-detection algorithms for proton computed tomography reconstruction," in Infinite Products of Operators and Their Applications, ser. Contemporary Mathematics, S. Reich and A. Zaslavski, Eds., vol. 636. American Mathematical Society, 2015, pp. 211–224.

4.1 Introduction

Proton computed tomography (pCT) is an attractive alternative to x-ray CT in the planning of proton radiation therapy since it has the potential to more accurately predict the range of proton beams delivered to the patient than those predicted using x-ray CT images [28].

The data acquired from a pCT scanner includes position tracking information and downstream energy measurements for individual protons traversing the target object. The object is staged on a rotating platform such that the pCT data is acquired from many different directions, ensuring an ample coverage of the target object is obtained with the fixed beam (which may also *wobble*). Modern pCT reconstruction has primarily moved away from the direct use of energy measurements for image reconstruction and instead converts energy detector measurements to water-equivalent path length (WEPL) values. In the context of proton therapy, "water-equivalent" denotes that the objective measurement

for a proton is the "path length" through water that would, on average, yield the same energy loss as that observed for the proton after traversing the object. These measurements are then used to reconstruct the RSP according to the formula $RSP = S_{mat}/S_{water}$, where $S_{mat} = -dE/dL$ is the stopping power of a material, defined as the mean differential energy loss (dE) of protons per unit path length (dL) through this material.

The viability of pCT image reconstruction for treatment planning depends on whether tractable methods can be developed for the computationally intensive calculations involved in the solution of the large sparse linear system of equations governing pCT imaging. As with other imaging modalities, pCT images can be expressed as the solution $\vec{x}$ of the linear system $\boldsymbol{A}\vec{x} = \vec{b}$. The system matrix $\boldsymbol{A}$ contains the proton path information, where $A_{i,j} = dL_{i,j}$ for each voxel j intersected by the ith proton; voxels that aren't intersected are denoted by the assignment $A_{i,j} = 0$. The vector $\vec{b}$ is composed of the WEPL values associated with each proton. The solution $\vec{x}$ is the vector representation of the reconstructed pCT image, whose components have the desired units of RSP as a result of the choice of representing energy loss with WEPL. At the time when hull-detection was developed, the size of the linear pCT system was on the order of $100 \cdot 10^6 \times 10^6$, which already represented a computational challenge to achieving the reconstruction speeds necessary for the clinical use of pCT. Moreover, despite having demonstrated that good quality pCT images could be reconstructed using iterative projection (feasibility-seeking) methods [29, 30], image quality was the most common criticism and clinical vulnerability cited in critiques of pCT. Thus, improving the quality of reconstructed images remained a vital objective of the pCT Collaboration.

If the quality of pCT images is to be not only maintained, but improved, then approximations and other simplifications that would typically be employed to reduce computational costs will not be viable options in the development of clinical pCT. The investigation of iterative projection method algorithms included the use of parallelization schemes, which were implemented on a single GPU, to demonstrate the impact this could have on

reconstruction times [29, 30]. Although reconstruction times were still well beyond clinically appropriate time thresholds, these works laid the foundations for modern pCT and demonstrated the direction that future developments would advance toward. Subsequent developments were nearly all based on the assumption that efficient pCT reconstruction would only be achieved with a parallelizable image reconstruction algorithm, and that image reconstruction software must be implemented on a distributed computing system.

Hull-detection was one of the developments following soon after the initial GPU based pCT software implementation. Accurate knowledge of the target object's hull, i.e. the smallest bounded region enclosing the object, can be used to expedite image reconstruction. The object hull is represented as a binary (or Boolean) image, where the value 1 (or TRUE) represents voxels belonging to the object hull and the value 0 (or FALSE) represented voxels outside the object hull (i.e. surrounding air). For a discrete target object $\mathcal{X} \subset \mathbb{N}^3$, representing a finite set of voxels, within a corresponding discrete (image) space $\mathcal{V} \subset \mathbb{N}^3$ (such that $\mathcal{X} \subseteq \mathcal{V}$), the object's hull, $\mathcal{H}$, is defined as the smallest subset $\mathcal{H} \subseteq \mathcal{V}$ such that $\mathcal{X} \subseteq \mathcal{H}$. Of course, the objective of hull-detection is to obtain an object hull $\mathcal{H}$ that is identical to the target object $\mathcal{X}$, i.e. $\mathcal{H} = \mathcal{X}$, but there is only a requirement that $\mathcal{X} \subseteq \mathcal{H}$ to prevent complications with image reconstruction. Hence, in practice the object hull is the smallest obtainable bounded region that completely encloses the object.

The MLP calculations that accompany image reconstruction, which are an important step, require knowledge of the object boundary [19]. This boundary has classically been acquired by thresholding, based on RSP value, the filtered backprojection (FBP) image generated during preconditioning. However, the RSP values of an FBP image below a threshold is not a reliable means for identifying object voxels since FBP images are noisy and, partly due to proton scattering, suffer from artifacts that obscure edges.

Accurate knowledge of the object hull can also be used to remove voxels outside the object from the FBP image, which is used as the initial iterate of image reconstruction, and exclude them from consideration in the subsequent image reconstruction. This is achieved

by spatially filtering the FBP image using the object hull $\mathcal{H}$ as a mask. This process effectively reduces the image space in which the pCT image $\vec{x}$ is reconstructed and, thus, the number of columns of the system matrix $\boldsymbol{A}$. On the other hand, the object image generated by FBP thresholding can contain "holes" in the interior of the object, preventing it from being used as mask without additional image processing.

Solutions of a linear system with m rows (i.e. proton histories) and n columns (i.e. voxels) obtained using iterative solvers have (computation) time complexity $O(mnK)$, where K is the total number of iterations performed. At the time hull-detection was being developed, the pCT system typically had $100n \geq m \geq 10n$. The largest number of voxels a proton can pass through occurs when it passes diagonally through the reconstruction volume, which corresponds with approximately $\sqrt[3]{n}$ voxels. This serves as an upper bound on the number of nonzero elements of the system matrix $\boldsymbol{A}$, thereby reducing the time complexity from $O(mnK)$ to $O(n^{1.333}K)$. The amount of memory required by the iterative image reconstruction algorithms, i.e. the space complexity, is also reduced by the voxel removal process; the less voxels misidentified as belonging to the hull, i.e. the closer the hull approaches $\mathcal{H} = \mathcal{X}$, the greater the reduction in space complexity. This can be particularly important for parallelized iterative image reconstruction algorithms, especially in the case of GPU implementations given their memory size constraints.

Hull-detection offers the opportunity to more efficiently perform pCT image reconstruction, improving both time and space complexity. It is also potentially more useful for MLP calculations and defining the initial iterate than previous methods, both of which have an impact on reconstructed image accuracy. Therefore, the preliminary objective is the development of a hull-detection algorithm for pCT that can begin to realize its aforementioned theoretical benefits. The work presented in this Chapter describes the initial silhouette carving algorithm developed at CSUSB, the modified silhouette carving (MSC) and space modeling (SM) algorithms developed later at BU, and compares and contrasts their effectiveness versus that of the classic FBP thresholding method. Note that although

"silhouette carving" is the more accurate descriptor, it was initially called "space carving". The term "silhouette" was later added, but to balance accuracy with the desire for terminological consistency, these intermediate publications used the term "silhouette/space carving". This dissertation will drop the previous terms and continue only with the more apt "silhouette carving" name.

4.2 Input Data

There were two sources of data for this work. Initial feasibility testing of the concept and underlying theory of hull-detection was performed using a simulated digital head phantom, which was custom designed for pCT [31]. Subsequent testing of the practical applicability of hull-detection was conducted using the pCT data acquired from an experimental scan of a pediatric head phantom using the Phase I prototype pCT scanner at Loma Linda University Medical Center [6, 32, 33]. Details of these phantoms and corresponding data sets are described below.

4.2.1 Simulated Data

In the development of algorithms for use in pCT, it is difficult to separate out the uncertainties arising from proton scattering and assess only the theoretical limits of an algorithm itself. Hence, a pCT data simulator generating deterministic proton paths was specifically designed for the developmental testing and analysis of algorithms [31]. The simulator allows the user to define the internal/external anatomy and size of a non-homogenous elliptical object (NEO), intended to approximate a head phantom. The user can add internal anatomical features, such as ventricles and the frontal sinus, which are represented by simplified geometric shapes. Similarly, geometrically simple representations of external anatomy, such as the ears and nose, can also be added to the NEO.

The digital phantom constructed and used in this work, shown in Figure 4.1, is composed of isotropic 1 mm^3 voxels . Its anatomical features include an outer elliptical region representing skull bone, two inner elliptical regions representing fluid-filled, and

Figure 4.1: Digital head phantom used to generate simulated data in this work.

all other enclosed regions representing brain matter. The simulator assigns realistic RSP values to anatomical features, which for this phantom composition are: 1.6 for skull bone, 1.04 for brain matter, and 0.9 for the fluid-filled ventricles.

Two simulated scans of the NEO were produced to assess the performance and viability of each hull-detection algorithm. The resulting simulated pCT data sets were each composed of 11,796,480 proton histories from a uniform 200 MeV proton cone-beam, approximated by randomly distributing each proton about the central beam axis. Simulations include the effect that multiple Coulomb scattering (MCS) within an object has on exit displacement and trajectory, relative to its entry displacement and trajectory, by considering these to be bivariate normal random variables. However, the simulator does not incorporate the impact that MCS has on the path inside the object, opting instead for the more easily characterized straight-line path (SLP) between its object entry and exit points. The intersection length, i.e. the path length between the points where a proton enters and exits a voxel (also referred to as chord length), is defined to be 1 mm for every voxel along the proton's SLP, yielding a system matrix $\boldsymbol{A}$ whose elements are either $A_{i,j} = 1$ or $A_{i,j} = 0$. The WEPL value assigned to proton i is then defined as the sum, over all voxels intersected by its SLP, of the product of the true RSP of voxel j and the corresponding chord length $A_{i,j} = 1$; since the $A_{i,j} = 1$, the WEPL calculation is reduced to simply the sum of each

intersected voxel's RSP. This data set is herein referred to as the "noiseless" simulated data set.

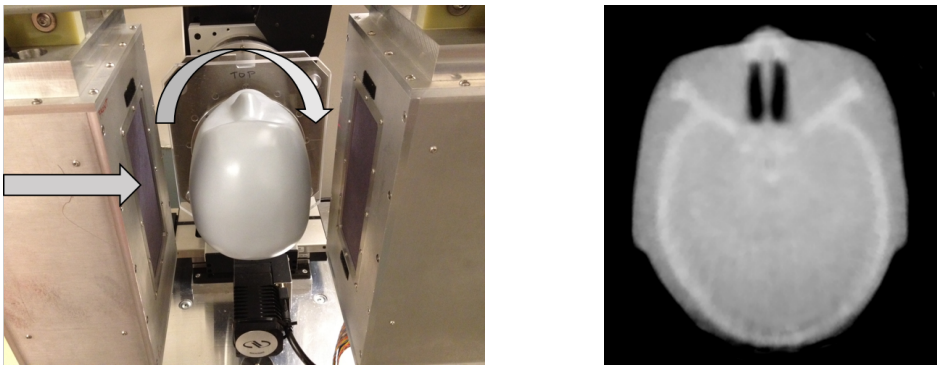
The second simulated data set was generated by adding noise to the "noiseless" data set, generating what is herein referred to as the "noisy" simulated data set. This is accomplished by converting the WEPL values of the noiseless data set into exit (residual) energy, adding normally distributed noise with a standard deviation defined according to Tschalar's energy straggling theory [34], and subsequently converting the now noisy energy value back to WEPL. The WEPL to exit energy conversion, as well as the exit energy to WEPL conversion, is based on data from ICRU Report 49 [35].

4.2.2 Experimental Data

The experimental pCT data set was used to validate the practical efficacy of hull-detection and determine the impact that realistic data has on the performance of each algorithm. This data set, composed of 50,897,953 proton histories, was acquired by scanning an anthropomorphic pediatric head phantom (model HN715, CIRS, Norfolk, VA, USA[1]) using the Phase I prototype pCT detector system illustrated in Figure 4.2. The medical proton accelerator at Loma Linda University Medical Center was used to generate a cone-beam composed of approximately 200 MeV protons. The head phantom was staged on a rotating platform and iteratively rotated in 4 ° increments through one complete revolution, i.e. 360 °, relative to the fixed horizontal beam line axis.

Note that the cited number of proton histories refers to the number that the image reconstruction software receives as input, not necessarily the number that are used for hull-detection or subsequent iterative reconstruction. Unlike the pCT data simulator, which is designed to solely generate useful data, experimental pCT data will include proton histories that are unsuitable for hull-detection and/or pCT image reconstruction. For the purposes of image reconstruction, only those protons that solely experienced MCS are sought; protons that underwent other physical interactions, such as elastic larger angle scattering or

[1] http://www.cirsinc.com/products/all/36/pediatric-anthropomorphic-training-phantoms



(a) (b)

Figure 4.2: (a) Annotated image indicating the direction of rotation (curved arrow), relative to the direction of the fixed horizontal beam line (straight arrow), of the pediatric head phantom when mounted on the rotating platform of the Phase I prototype pCT scanner system (Loma Linda University Medical Center). (b) a representative slice of the reconstructed pCT image of the pediatric head phantom.

inelastic nuclear interactions, are removed (cut) from the data set. Another common source for unsuitable proton histories is proton *pile-up* in the energy detector (calorimeter), i.e. proton(s) entering the detector before the residual energy from a previous proton has dissipated, resulting in this residual energy inappropriately being added to the subsequent proton's energy measurement. Efforts are also taken to identify and cut these from the data set as well.

Hence, the number of protons that remain after identifying and removing as many unsuitable proton histories as possible will be lower, up to half as many, than the number acquired from the pCT scanner system. A detailed description of the various sources/types of unsuitable data, as well as the preprocessing and preconditioning steps developed to identify and remove the corresponding proton histories, can be found in "Particle-Tracking Proton Computed Tomography — Data Acquisition, Preprocessing, and Preconditioning" [25].

4.3 Hull-Detection Algorithms

A hull-detection algorithm seeks, for an object $\mathcal{X} \subset \mathbb{N}^3$ with hull $\mathcal{H} \subseteq \mathcal{V} \subset \mathbb{N}^3$, an approximate hull, $\mathcal{H}'$, such that $\mathcal{H} \subseteq \mathcal{H}' \supseteq \mathcal{V}$ and the cardinality of the difference between the sets $\mathcal{H}$ and $\mathcal{H}'$, $|\mathcal{H}' \setminus \mathcal{H}|$, $\mathcal{H}' \setminus \mathcal{H}$, approaches as close to zero as possible. In

other words, a hull-detection algorithm must generate an approximate hull $\mathcal{H}'$ that includes every voxel of $\mathcal{H}$ and a minimal number of voxels from outside the hull $\mathcal{H}$, $\mathcal{V} \setminus \mathcal{H}$.

Three hull-detection algorithms were developed and are presented in this Chapter: silhouette carving (SC), modified silhouette carving (MSC), and space modeling (SM). Each of these are subsequently compared to the FBP thresholding approach to object detection. FBP is a full image reconstruction algorithm and is used as such for other imaging modalities. The reconstructed FBP image has been used in pCT as well, not only for object detection but also as the basis for the initial iterate of the subsequent iterative reconstruction algorithms [29, 30]. However, FBP reconstruction is performed along straight rays (lines), which is inconsistent with the curved paths of protons due to MCS, so FBP images of pCT data are particularly noisy and prone to artifacts. In early hull-detection work, the FBP thresholding approach was compared to SC in terms of computation time and reconstructed image quality [36], but not a direct voxel-by-voxel comparison between the true and approximate hull, as is done in this work to assess algorithm performance independent of other factors.

4.3.1 Filtered Backprojection (FBP)

FBP was first proposed as an alternative to Fourier transform methods for the reconstruction of CT data in 1971 by Ramanchandran and Lakshminarayanan [37]. A variant of the FBP algorithm that assumes straight ray paths (or SLPs) with a cone-beam geomtry, known as the Feldkamp Davis Kress (FDK) algorithm [38], is implemented here using a Shepp-Logan filter [39] (as in previous pCT work). The FDK algorithm is performed using a 4° angular bin spacing, a 1 mm lateral bin size, and a 5 mm vertical bin size. The image space in which reconstruction is performed is is 200 mm $\times$ 200 mm$\times$ 9.6 cm. Each slice is 3 mm thick, yielding 32 reconstructed tomographic slices. The resulting image was then thresholded to generate the approximate hull. Any voxel with RSP $\geq$ 0.6 was assumed to belong to the object and was assigned an RSP value of one. Voxels with RSP values below this threshold were assigned an RSP value of zero.

For the purposes of object detection, a threshold of RSP $\geq$ 0.6 was applied to the FBP image to generate a binary object image. This method does not strictly produce a hull, since voxels within the object boundary may also be thresholded and become "holes", but it will be considered comparable to a hull for the purpose of comparison analysis. Note that the unsuitable data was cut, using the same statistical binning intervals as FBP, prior to generating and using the FBP image for object detection. Details on the statistical cuts and the use of the FBP image as a preconditioner are provided in the same publication as the aforementioned unsuitable data discussion [25].

4.3.2 Silhouette Carving (SC)

SC is a hull-detection algorithm, developed specifically for pCT, that generates an approximate hull by iteratively carving away undesired voxels from an initially entirely filled image space, which can be visualized as a similar process as a sculptor chiseling away portions from a solid block of material [36]. The concept was conceived based on shape and space carving methods [40, 41, 42, 43, 44] and the fact that protons that never enter the object, passing solely through air, experience negligible scattering and energy loss. Hence, such protons traverse the reconstruction volume along nearly SLPs. If these protons can be identified, and there is a sufficient number and angular distribution of them, then the SLP approximation of their paths can be used as the geometric shape (line) for a carving method. At the time this algorithm was developed, pCT scans acquired 10 million or more proton histories, so the more challenging concern is how easily can they be identified. Perhaps even more importantly, how well can protons that did traverse the object be excluded.

In principle, the lack of scattering (lateral displacement or angular deviation) or energy loss (WEPL), either of individual protons or the mean for the protons in a bin, can be used to identify the protons useful for SC. The initial development posited the use of both scattering and energy as selection criteria. However, proton paths aren't reliable since a proton may have been measured to have identical angle before and after the object, but this doesn't guarantee that there was no scattering between them; after all, the expected

deviations are expected to be normally distributed around zero. Therefore, a threshold on the WEPL values (energy loss) is used to identify protons (or proton bins) that entirely missed the object.

The value WEPL = 0 was initially selected as the RSP threshold for SC since this implies that the proton went through 0 mm of water-equivalent material, i.e. missed the object. However, WEPL values are based on an analysis of calorimeter readout measurements and a subsequent WEPL calibration process [45]. A more accurate interpretation of WEPL values is that they correspond to the central average, or expected value, of a distribution of possible calorimeter ADC value measurements. In other words, a proton that missed the object may have WEPL > 0 or WEPL < 0 but the calibration ensure that, on average, their WEPL $= 0$. Similarly, protons that pass through small portions of the object will sometimes have WEPL 0. To prevent such proton histories from being used for SC, WEPL $= -1$ mm was used: protons at or below this threshold value are used for hull-detection, whereas those above it are excluded.

The SC algorithm (see Algorithm 1 below) seeks an approximation, $\mathcal{H}_1$, of the object hull, $\mathcal{H}$, using the protons that satisfy the threshold WEPL$le-1$ mm. The approximate hull, $\mathcal{H}_1$, is initialized as encompassing the entire image space, i.e. $\mathcal{H}_1 = \mathcal{V}$, represented as a binary image composed entirely of the value 1 (TRUE). For each proton below the WEPL threshold, the voxels along the SLP defined by connecting the proton's image entry and exit points are carved from $\mathcal{H}_1$ by assigning the corresponding voxels the value 0 (FALSE). For a particular scan angle, the projected paths of protons missing the object produce a silhouette of the object. Repeating this for successive scan angles advances $\mathcal{H}_1$ closer to its final approximation of $\mathcal{H}$. Note that once a voxel is carved, it does not need to be carved again, so the number of voxels carved for successive scan angles tends to decrease. This can also be visualized as the superposition of the silhouettes obtained for each scan angle, where many of the silhouettes overlap. The nonzero voxels of $\mathcal{H}_1$ remaining after performing SC for each scan angle then defines the detected hull.

The preceding was a conceptual description of the SC algorithm as implemented for development. A formal definition of the general framework of the SC algorithm is now presented in terms of set theory and operations, followed by the corresponding pseudocode implementation. Note that although investigations of SC have not included a threshold on the angular deviations of proton paths, future investigations may find some threshold on angular deviation that aids the identification of suitable proton histories for SC. Hence, this flexibility is included in the following definition of the SC algorithm. Furthermore, the threshold selection criteria is stated in general terms of energy loss, rather than the specific WEPL representation. These decisions were made to provide a complete and generic framework for the SC algorithm such that the underlying theory is completely conveyed and the specific implementation decisions presented in this dissertation, which somewhat depend on current pCT theory and data acquisition, are more likely to be reevaluated by future investigators.

The notation used in the following definition of the SC algorithm is as follows:

- p_i: the ith proton (or proton bin)
- $\Delta E(p_i)$: (mean) energy loss of the i^{th} proton (bin)
- $\Delta\angle(p_i)$: (mean) angular deviation of the ith proton (bin)
- E_L: user-defined energy loss threshold for identifying protons the missing object
- θ_L: user-defined angular deviation threshold for identifying protons missing object
- $\mathcal{I} = \{\ 1, 2, 3, \cdots, m\}$: sequential set of indices of the m protons (or proton bins)
- $\mathcal{V}$: the set composed of each voxel in the image space.
- v_j: the jth voxel of the image space $\mathcal{V}$

For each proton (or proton bin) i, if $\Delta E(p_i) \leq E_L$ and $\Delta\angle(p_i) \leq \theta_L$, then proton (or protons in bin) p_i are identified as having missed the object. The set, $\mathcal{I}_L$, composed of the indices of all protons (or proton bins) i satisfying the E_L and θ_L thresholds and subsequently used in SC, is then defined as:

$$\mathcal{I}_L = \{\ i \in \mathcal{I} \mid \Delta E(p_i) \leq E_L \wedge \Delta\angle(p_i) \leq \theta_L\} \tag{4.1}$$

For all $i \in \mathcal{I}_L$, the line, L_i, is defined as the chord connecting the points where proton (or proton bin) i entered and exited the image space $\mathcal{V}$. The set, $\mathcal{A}_i$, is composed of each voxel v_j along the path L_i, identified according to a maximum distance constraint d_0 placed on the (user-defined) distance measure $d(L_i, v_j)$; i.e. each voxel v_j within distance d_0 of the line L_i comprises the set $\mathcal{A}_i$:

$$\mathcal{A}_i = \left\{ v_j \in \mathcal{V} \;\middle|\; d(L_i, v_j) \leq d_0 \right\} \tag{4.2}$$

Therefore, the resulting hull $\mathcal{H}_1$ generated by SC is defined as:

$$\mathcal{H}_1 = \mathcal{V} \setminus \cup_{i \in \mathcal{I}_L} \mathcal{A}_i. \tag{4.3}$$

Given these definitions, a pseudocode definition of the SC algorithm is as follows:

Algorithm 1 Silhouette Carving (SC)

$\mathcal{I}_L \leftarrow \emptyset$
for all $i \in \mathcal{I}$ **do**
    if $\Delta E(p_i) < E_L$ **and** $\Delta\angle(p_i) < \theta_L$ **then**
        $\mathcal{I}_L \leftarrow \{\mathcal{I}_L, i\}$
    end if
end for
for all $v_j \in \mathcal{V}$ **do**
    $\mathcal{H}_1(v_j) \leftarrow 1$
end for
for all $i \in \mathcal{I}_L$ **do**
    $\mathcal{A}_i \leftarrow \emptyset$
    for all $v_j \in \mathcal{V}$ **do**
        if $d(L_i, v_j) \leq d_0$ **then**
            $\mathcal{A}_i \leftarrow \left\{ \mathcal{A}_i, v_j \right\}$
        end if
    end for
    $\mathcal{H}_1 \leftarrow \mathcal{H}_1 \setminus \mathcal{A}_i$
end for

The protons used as input to the SC algorithm were the same as those used in the FBP thresholding approach, i.e. after applying statistical data cuts with the same bin sizes, to remove unsuitable proton histories. The same image space and voxel dimensions were also used for SC, the latter of which were used to define each proton path L_i. A threshold $E_L = 1.0$ mm was placed on the mean WEPL of each bin, rather than the mean energy loss, to identify proton bins for subsequent use in SC. If the mean WEPL of a bin satisfied the threshold E_L, the protons in that bin were identified as travelling exclusively through air; the voxels v_j along each L_i (approximated by an SLP using the bin's angle and displacements), i.e. $\mathcal{A}_i$, were excluded from the approximate hull $\mathcal{H}_1$. The algorithm developed to identify the voxels intersected by an SLP is presented in Appendix B. No angular deviation threshold was employed in the SC algorithm implemented for these investigations.

Previous work [36] demonstrated that, although rare relative to the sizes of the date sets, some unsuitable proton histories survive statistical data cuts (particularly those that pass near to or through a small amount of the object surface) and result in some lines through the object mistakenly being removed from the hull $\mathcal{H}_1$. To account for such possibilities and refill these portions, an averaging filter with 5×5 structural element was applied to $\mathcal{H}_1$, where the filter assigns voxels with neighborhood average exceeding 0.4 the value 1 (TRUE) and 0 (FALSE) otherwise. The 0.4 threshold corresponds to 10 of the 25 neighboring voxels having value 1 (TRUE) before filtering, but this is allowed to vary as a user-defined parameter.

4.3.3 Modified Silhouette Carving (MSC)

The first investigations of the SC algorithm used a WEPL threshold $E_L = 1.0$ mm, but it was quickly observed that this resulted in nearly all voxels in the image space being carved from the hull, i.e. $H_1 = \emptyset$, and the threshold was changed to 0.0 mm $\leq$ WEPL $\leq$ 1.0 mm [36]. This resulted in highly accurate hull approximations from simulated data sets, better than the FBP thresholding approach with both the noiseless and noisy data, but was still inadequate for experimental data. Since there will inevitably be some unsuitable

data that eludes statistical cuts and other attempts to remove undesired data, an effective hull-detection algorithm must have some level of tolerance to their presence. The conclusion drawn from this early work was that there were two basic options for nullifying, e.g. averaging out, the impact of unsuitable data in future hull-detection development: (1) use proton bin data, rather than individual protons, to identify the lines to carve along in the SC algorithm or (2) develop alternative algorithms, either from scratch or as an adaptation of SC. The former of these corresponds to the implementation of SC described above. The latter led to an adaptation of SC, herein referred to as MSC.

MSC exploits the fact that the amount of unsuitable data remaining after cuts is only a small fraction of the total number of proton histories. Unlike the SC algorithm, which immediately removes all $v_j \in \mathcal{A}_i$ from the hull $\mathcal{H}_1$, MSC maintains a running count of the number of times, N_j, a line L_i passed through voxel v_j. The assumption is that although voxels inside the object may lie along one or more lines L_i associated with unsuitable proton data, the count N_j for these interior voxels will be small relative to those that truly lie outside the object. Furthermore, the unsuitable data do not contribute much to the count N_j of any voxel v_j, whether inside or outside the boundary of the object, thereby reducing the impact that unsuitable data can have. Therefore, MSC uses the N_j to differentiate between voxels within the object hull $\mathcal{H}$ and those outside of it ($\mathcal{V} \setminus \mathcal{H}$), thereby minimizing the number of object voxels mistakenly excluded from the approximate hull $\mathcal{H}_1$.

The MSC algorithm (see Algorithm 2 below) seeks an approximation of the hull $\mathcal{H}$, herein referred to as $\mathcal{H}_2$, by deferring the voxel carving steps until the number of times, N_j, each voxel v_j was identified as external to the object has been determined. Then for each voxel $v_j \in \mathcal{V}$, $C(v_j)$ is defined as the set composed of the indices of each line, L_i, that passed through v_j:

$$C(v_j) = \left\{ i \in \mathcal{I}_L \;\middle|\; v_j \in \mathcal{A}_i \right\} \tag{4.4}$$

Similarly, for each voxel v_j, the set, $\mathcal{B}(v_j)$, composed of each voxel, w_j, neighboring v_j is defined as:

$$\mathcal{B}(v_j) = \left\{ w_j \in \mathcal{V} \;\middle|\; d(v_j, w_j) \leq 1 \right\} . \tag{4.5}$$

Let $N(v_j) = |C(v_j)|$ and $N(w_j) = |C(w_j)|$ be defined as the cardinalities of $C(v_j)$ and $C(w_j)$, respectively. Given a strict threshold N_t on the cardinality difference $N(v_j) - N(w_j)$, the approximate hull $\mathcal{H}_2$ generated by MSC is given by:

$$\mathcal{H}_2 = \left\{ v_j \in \mathcal{V} \;\middle|\; \max_{w_j \in \mathcal{B}(v_j)} N(v_j) - N(w_j) < N_t \right\} . \tag{4.6}$$

Unlike the FBP thresholding and SC algorithms, the MSC algorithm was specifically designed to be performed prior to the statistical data cuts that remove the bulk of the unsuitable proton histories. The reason for this deviation with the MSC algorithm is that it was developed with the objective of providing a hull-detection algorithm that is tolerant to the unsuitable proton histories present in data acquired from an experimental pCT scanner. A pseudocode definition of the MSC algorithm, written in terms of the preceding definitions, appears in Algorithm 2.

The implementation of the MSC algorithm used individual protons histories, selecting as those that satisfied the threshold WEPL ≤ -1.0 mm, rather than the proton bins used in SC. ns on whether protons missed the object were based on analysis of individual WEPL values rather than bin averages. Thus, proton histories were not binned in this case. Proton histories whose WEPL values were less than 1.0 mm were assumed to have missed the object, which is the same WEPL cutoff value used for SC. A strict cardinality difference threshold $N_t = 50$ was used to generate the results presented here since this was found to be insensitive to the varying sizes of the data sets. Note that a threshold on individual cardinalities, $N(v_j)$, was found to vary too much between slices and phantoms, so defining an a priori threshold in future scans of difference objects was deemed to be infeasible and

abandoned. On the other hand, the cardinality difference, $N(v_j) - N(w_j)$, between neighboring voxels proved to be far more stable and a more robust means for differentiating hull voxels from surrounding (air) voxels.

Algorithm 2 Modified Silhouette Carving (MSC)

$\mathcal{I}_L \leftarrow \emptyset$
for all $i \in \mathcal{I}$ **do**
    if $\Delta E(p_i) < E_L$ **and** $\Delta\angle(p_i) < \theta_L$ **then**
        $\mathcal{I}_L \leftarrow \{ \mathcal{I}_L, i \}$
    end if
end for
for all $v_j \in \mathcal{V}$ **do**
    $\mathcal{H}_2(v_j) \leftarrow 1; N(v_j) \leftarrow 0$
end for
for all $i \in \mathcal{I}_L$ **do**
    for all $v_j \in \mathcal{V}$ **do**
        if $d(L_i, v_j) \leq d_0$ **then**
            $N(v_j) \leftarrow N(v_j) + 1$
        end if
    end for
end for
for all $v_j \in \mathcal{V}$ **do**
    for all $w_j \in \mathcal{B}(v_j)$ **do**
        if $N(v_j) - N(w_j) \geq N_t$ **then**
            $\mathcal{H}_2(v_j) \leftarrow 0$
        end if
    end for
end for

4.3.4 Space Modeling (SM)

Another hull-detection technique, developed along with MSC as alternatives to SC, is SM. Rather than using protons that miss the object to carve away voxels external to the object hull, SM only uses protons that pass through the object to construct, or model, an approximate hull. Protons passing through the object tend to experience increasing amounts of scattering and energy loss as their penetration depth within the object increases. Hence, similar to the selection of protons for SC, SM can identify protons that passed

through the object according to their energy loss (WEPL) and/or angular deviations. In this case, SM selects protons that exceed a minimum bound (rather than a threshold) placed on the energy loss (WEPL) and angular deviation measurements. In other words, the protons that are selected for use are those whose energy loss (WEPL) and/or angular deviations exceed the associated minimum constraints defined for these measures.

Protons that satisfied the minimum energy and/or angular deviation constraints are subsequently projected along straight lines, defined according to their individual position and angle measurements, and the voxel intersected along these paths are determined. Similar to MSC, SM maintains a record of the number of times, M_j, each voxel v_j is intersected by a proton passing through the object, thereby reducing the impact of unsuitable histories on the misidentification of protons passing through the object. Unlike MSC, however, this count is a requirement with SM since the projected lines pass through both voxels within the object and external to it, so it is the values of M_j for each voxel that are used as the entire basis for constructing an approximate hull.

As predicted, it was found that the M_j decline more sharply at the interface between the object and its surrounding air than anywhere else. Hence, for each tomographic slice, SM proceeds to automatically locate the voxel edge with largest gradient in M_j and subsequently records the largest value of M_j on this edge as M_t. These M_t values define the threshold applied to the M_j of each voxel v_j in that slice to select voxels for membership in the approximate hull; i.e. for each slice, every voxel v_j for which $M_j \geq M_t$ is defined as belonging to the object hull. After performing this procedure for each slice, the approximate hull has been constructed.

The user-defined parameters used in the following definition of the SM algorithm, and absent from preceding definitions, are defined as follows:

- E_H: minimum energy loss constraint for identifying protons traversing the object
- θ_H: minimum angular deviation constraint for identifying protons traversing the object

The SM algorithm (see Algorithm 3 below) seeks a robust approximation of the hull, $\mathcal{H}_3$, by backprojecting the silhouette of the object and recording the number of times, M_j, each voxel v_j was intersected by a backprojected line. If $\Delta E(p_i)$ or $\Delta\angle(p_i)$ exceed the minimum constraints E_H and θ_H, respectively, then the proton (or proton bin), p_i, is considered to have traversed the object. The set, $\mathcal{I}_H$, composed of the indices of these protons (or proton bins) is then defined as:

$$\mathcal{I}_H = \{\, i \in \mathcal{I}. \mid \Delta E(p_i) > E_H \} \ . \tag{4.7}$$

In this case (SM), for $i \in \mathcal{I}_H$, let L_i be the line that connects the entry and exit points of the ith proton (or proton bin) traversing the object (rather than missing the object, as in SC and MSC). Given a (possibly different) distance measure $d(\cdot,\cdot)$ and a maximum distance constraint d_0, the set, $\mathcal{A}_i$, composed of each voxel v_j along L_i is defined as:

$$\mathcal{A}_i = \left\{ v_j \in \mathcal{V} \;\middle|\; d(L_i, v_j) \leq d_0 \right\} \ , \tag{4.8}$$

Then for each voxel $v_j \in \mathcal{V}$, $C(v_j)$ is defined as the set composed of the indices of each line, L_i, that passed through v_j:

$$C(v_j) = \left\{ i \in \mathcal{I}_H \;\middle|\; v_j \in \mathcal{A}_i \right\} \ . \tag{4.9}$$

Let $M(v_j) = |C(v_j)|$ and $M(w_j) = |C(w_j)|$ be defined as the cardinalities of $C(v_j)$ and $C(w_j)$, respectively. Given a strict minimum constraint M_t on the cardinality $M(v_j)$, the approximate hull $\mathcal{H}_3$ generated by SM is given by:

$$\mathcal{H}_3 = \left\{ v_j \;\middle|\; M(v_j) > M_t \right\} \tag{4.10}$$

Given the preceding definitions, a pseudocode definition of the SM algorithm is as follows:

Algorithm 3 Space Modelling (SM)

$\mathcal{I}_H \leftarrow \emptyset$
for all $i \in \mathcal{I}$ **do**
    if $\Delta E(p_i) > E_H$ **or** $\Delta\angle(p_i) > \theta_H$ **then**
        $\mathcal{I}_H \leftarrow \{\mathcal{I}_H, i\}$
    end if
end for
for all $v_j \in \mathcal{V}$ **do**
    $\mathcal{H}_3(v_j) \leftarrow 0; M(v_j) \leftarrow 0$
end for
for all $i \in \mathcal{I}_H$ **do**
    for all $v_j \in \mathcal{V}$ **do**
        if $d(L_i, v_j) \leq d_0$ **then**
            $M(v_j) \leftarrow M(v_j) + 1$
        end if
    end for
end for
MaxSlope $\leftarrow 0$, index $\leftarrow 0$
for all $v_j \in \mathcal{V}$ **do**
    for all $w_j \in \mathcal{B}(v_j)$ **do**
        if $M(v_j) - M(w_j) \geq$ MaxSlope **then**
            MaxSlope $\leftarrow M(v_j) - M(w_j)$; index $\leftarrow v_j$
        end if
    end for
end for
$M_t \leftarrow M$(index)
for all $v_j \in \mathcal{V}$ **do**
    if $M(v_j) > M_t$ **then**
        $\mathcal{H}_3(v_j) \leftarrow 1$
    end if
end for

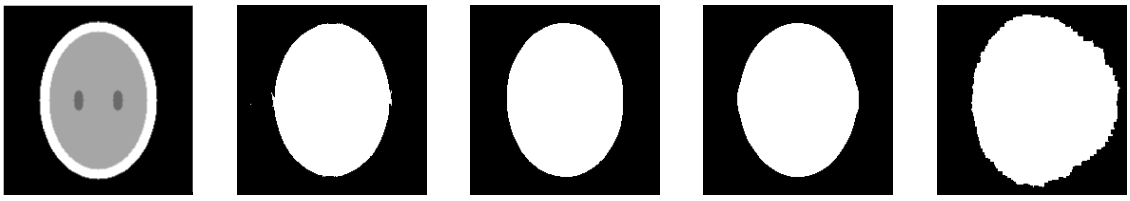
As with the MSC algorithm, unsuitable proton histories were not cut prior to hull-detection and protons traversing the object were identified by the WEPL values of individual protons (rather than mean WEPL values, as with SC).A minimum energy constraint $E_H = 5.0$ mm was used, effectively eliminating the potential misidentification of protons

that actually missed the object. The neighborhood cardinality comparison used in MSC was not effective in SM. Instead, the maximum distance constraint M_t was determined, for each individual slice, using a modified version of the Canny edge-detection algorithm [46].

4.4 Results

4.4.1 Simulated Data Results

Figure 4.3 shows, for each hull detection algorithm, the binary image of the approximate (NEO object) hull generated from the noiseless simulated data. Each image slice is 200 voxels × 200 voxels, representing a 200 mm × 200 mm area. Table 4.1 summarizes the performance of each algorithm in terms of computation time and the quality of the hull approximation. Hull approximation quality is evaluated in terms of (1) the number of target object (NEO) voxels missing from the approximation and (2) the number of voxels in the approximate hull that do not belong to the target object (NEO) hull. This evaluation is performed using a voxel-by-voxel comparison of the digital NEO head phantom and each approximate hull.



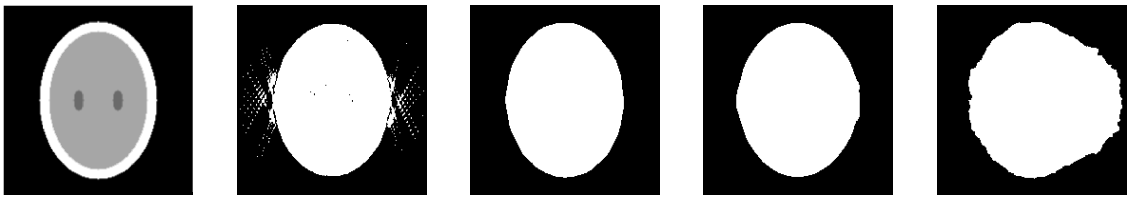
(a) Phantom (b) FBP (c) SC (d) MSC (e) SM

Figure 4.3: (a) Original digital NEO head phantom; (b)-(e) approximate object hulls generated by each hull-detection algorithm from the noiseless simulated data set.

Table 4.1: Performance comparison of hull-detection algorithms for noiseless simulated data set

Measurement	FBP	SC	MSC	SM
Computation Time	16.70 s	<0.10 s	5.95 s	5.52 s
Missing Voxels	50	0	0	0
Extra Voxels	116	345	488	5802

The same performance analysis, for each algorithm, was applied to the hull approximations from the noisy simulated data set. The results of the noisy simulated data analysis are shown in Figure 4.4 and in Table 4.2, respectively.



(a) Phantom (b) FBP (c) SC (d) MSC (e) SM

Figure 4.4: (a) Original digital NEO head phantom; (b)-(e) approximate object hulls generated by each hull-detection algorithm from the noisy simulated data set.

Table 4.2: Performance comparison of hull-detection algorithms for noisy simulated data set

Measurement	FBP	SC	MSC	SM
Computation Time	16.72 s	<0.10 s	6.14 s	5.86 s
Missing Voxels	88	0	0	0
Extra Voxels	831	461	716	4563

The FBP thresholding approach was the only algorithm, for both the noiseless and noisy simulated data sets, that generated an approximate hull that did not include every voxel of the NEO object, i.e. voxels were missing from the approximate hull. Additionally, the streak artifacts often seen in FBP images resulted in voxels well outside the object being misidentified as part of the hull, yielding similar artifacts in the hull approximations in Figure 4.3(b) and, particularly, Figure4.4(b).

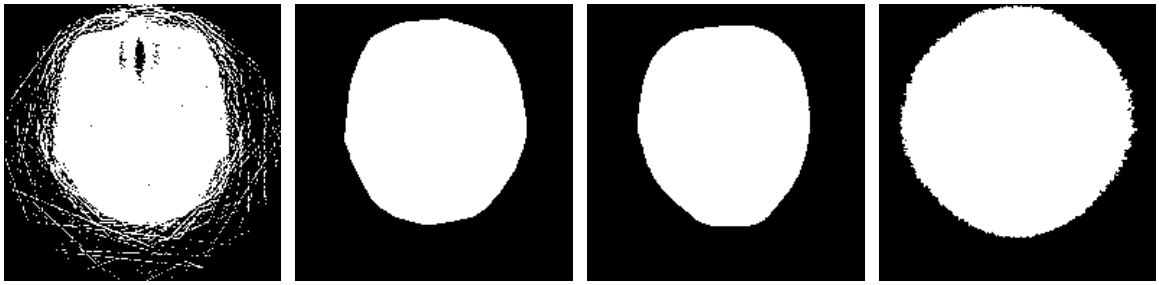
On the other hand, the SC, MSC, and SM algorithms all successfully identified each voxel belonging to the target object. The primary difference between the three hull-detection algorithms is the number of voxels that were misidentified as belonging to the object, which varied considerably. SC generated the fewest and SM generated, by far, the most, whereas MSC performed close to SC and considerably better than SM. With the

noiseless simulated data, the FBP thresholding approach generated fewer extra voxels than any of the three hull-detection algorithms, but with the noisy data, it generated the second most (after SM) extra voxels.

In terms of computational performance (see Tables 4.1 and 4.2), SC was by far the fastest with computation times two orders of magnitude lower than any of the other algorithms. MSC and SM, as could be predicted given the additional steps they perform compared to SC, were slower than SC but comparable to one another. On the other hand, the FBP thresholding approach was consistently slower than any of the hull-detection algorithms. As a full image reconstruction algorithm, which must subsequently be thresholded to approximate the object hull, it is understandably slower.

4.4.2 Experimental Data Results

Figure 4.5 shows, for each hull detection algorithm, the binary image of the approximate (pediatric head phantom object) hull generated from the experimental pCT data set. Each image slice is 192 voxels $\times$ 192 voxels, representing a 200 mm $\times$ 200 mm area. Note that a direct voxel-by-voxel comparison is not possible in this case since there is no "true" image of the head phantom to compare results to.



(a) FBP (b) SC (c) MSC (d) SM

Figure 4.5: (a)-(d) Object hull approximations generated by the various hull-detection algorithms using the experimental data set from the scan of the pediatric head phantom.

As with the simulated data, the FBP thresholding approach generated an approximate hull containing artifacts, more so than with the simulated data, and with missing target object voxels. The approximate hull also contains three of the aforementioned "holes" in

the interior of the hull, corresponding with the (low RSP) nasal passages, which in actuality, is only two passages separated by the nasal septum.

The SC and MSC algorithms (Figures 4.5(b) and 4.5(c), respectively) both generated an approximate hull that visually closely matches the shape of the pediatric head phantom and are free of artifacts. The SM algorithm, however, generates an approximate hull (Figure 4.5(d)) that is clearly too large and doesn't appear to match the shape of the head phantom well.

4.5 Discussion

The development of the three hull-detection algorithms investigated as potential alternative to FBP-based object detection is an extension of early work [36]. This early work demonstrated that the number and spatial/angular distribution of proton histories in contemporary pCT data sets was sufficient for carving voxels at the resolution of typically pCT images, i.e. there was enough coverage of the image space and enough projection lines (L_i) generated to completely carve voxels out to the edges of the image. However, for experimental pCT data, the SC algorithm generated hull approximations that indicated protons traversing the object had been misidentified as missing the object and subsequently used to errantly carve out voxels actually belonging to the target object. Two sources were found to explain this performance deficiency: (1) the WEPL value threshold used to identify protons was too inclusive and (2) proton histories unsuitable for image reconstruction also interfered with hull-detection. The results of the investigations of hull-detection algorithms presented here indicate that the issues encountered with the original SC algorithm can potentially be eliminated.

To isolate the impact of unsuitable proton histories on SC, the statistical data cuts performed as part of preconditioning image reconstruction were applied prior to performing SC, thereby eliminating all identifiable unsuitable data. The binned proton histories associated with statistical cuts were retained for use in the SC algorithm, where each bin's mean WEPL value was used to identify protons missing the object, rather than individual

WEPL values as had been done previously. To account for potential missing voxels in the interior of the object hull, a blurring filter was used to "fill" such holes and enlarge the outer boundary slightly. These advancements were shown to be effective for their intended purpose.

Other schemes for overcoming the issues encountered in early SC work also led to the development of two novel hull-detection algorithms: MSC and SM. These continued the use of individual proton histories for identifying protons that miss the object as in the early SC work. Similarly, unsuitable data was not removed from consideration for MSC and SM. For the simulated data sets, MSC and SM both generated approximations of the NEO hull that contained every voxel of the NEO object. It was not possible to determine this for the hull approximations from experimental data and remains to be investigated. However, the results suggest that the binning of protons and removal of unsuitable data is not a requirement of MSC or SM. This makes it possible to perform hull-detection with MSC or SM in an online mode setting during pCT scans. Their computational inefficiency relative to SC would be moot in this case, since hull-detection would be complete prior to the point in reconstruction when SC can be used. The immediate availability of an approximate hull after a pCT scan is an attractive property of the MSC and SM algorithms.

For these data sets, SC appears to have the superior performance and is certainly a viable hull-detection algorithm. It is also clear that SM is unlikely to be a viable option since its projection paths intersect both voxels internal and external to the hull, making it difficult to differentiate between the two. However, there remains much to investigate of MSC and its performance approached that of SC in terms of hull-detection accuracy. The scheme for converting the voxel intersection counts is not quite mature and has ample opportunity for improvement. Whether energy constraints can be adjusted to improve performance remains to be investigated, but it is clear that the FBP threshold approach is inferior to both SC and MSC in terms of computational efficiency and quality of hull approximations.

4.6 Conclusion

This work has investigated the suitability of two existing and two new hull-detection algorithms for pCT reconstruction. The results obtained with SC, MSC, and SM are promising and represent a significant step toward an effective and robust hull-detection algorithm. SC performed best, but MSC and SM could be further improved. FBP was not adequate for efficient and accurate hull-detection.

CHAPTER FIVE

Most-Likely Path (MLP): Simplifications and Implementation Design

Portions of the content presented in the following Chapter is currently being prepared, but has not yet been submitted or approved, for publication. This forthcoming publication will serve as a companion to, and provide the remaining details of current pCT technology which were not presented in, the recently published journal article:

B. E. Schultze, P. Karbasi, C. Sarosiek, G. Coutrakon, C. E. Ordo˜nez, N. T. Karonis, K. L. Duffin, V. A. Bashkirov, R. P. Johnson, K. E. Schubert, and R. W. Schulte, "Particle-tracking proton computed tomography—data acquisition, preprocessing, and preconditioning," *IEEE Access*, vol. 9, pp. 25 946–25 958, 2021.

5.1 Introduction

The spatial resolution that can be obtained with pCT depends on how accurately proton paths through the object can be determined from their entry and exit information. The MLP formalism provides for more accurate approximations of proton paths through an object than those approximated with other means (e.g. cubic splines) [50]. Since the definition of the object produced by hull detection is not in the form of an equation defining the object boundary, but a binary object image, there is no algebraic solution for the proton's entry and exit points. Hence, the entry and exit points are calculated using the same 3D-DDA (voxel walk) algorithm developed for hull-detection (see Appendix B). The voxel walk begins at the points where the proton enters and exits the reconstruction volume, which are calculated early in preconditioning [25]. The voxel walk proceeds from one voxel intersection to the next until a voxel belonging to the binary object is encountered. The edge of the voxel at the interface between the object and surrounding air is used defined the coordinates of the entry and exit points. These two points define the starting- and end-point of the MLP and the subsequent iterative image reconstruction is performed between these two points.

In the initial image reconstruction program [51, 29], reconstruction was also performed along the portions of the entry and exit SLPs between the object and reconstruction cylinder. This was partially motivated by the fact that since hull-detection was not performed during preconditioning, it was necessary to employ some method to remove the artifacts outside the object in the FBP image and reduce their RSPs down close to zero. This can indeed be accomplished by reconstructing along the SLP portions of a proton path, but the WEPL measurement associated with the proton should not be used to do so. A WEPL calibration procedure is performed prior to each scan to account for potentially time-varying beam line conditions. Calibrated WEPL values are defined such that, on average, WEPL = 0 when there is no object present. In other words, the air outside the object is accounted for in the WEPL calibration. Hence, WEPL values associated with proton histories correspond to energy losses solely within the object. By considering the WEPL measurement to be associated with the MLP and the entry/exit SLP, the WEPL of each proton traversing the object was undervalued and, consequently, the convergence rate of feasibility-seeking is reduced.

Of course, there are a large number of protons that pass solely through air and help reduce the RSPs of voxels outside the object without affecting internal WEPL assumptions. This helps to drive the RSP of voxels outside the object towards zero fairly quickly, thereby reducing the negative impacts of assigning WEPL to the SLPs. However, although the negative impacts are primarily restricted to the first few iterations, since it does not conform to theory and can only be detrimental, this was corrected in the BU reconstruction software. Since hull-detection provides a binary object image, the voxels outside the object are removed from consideration by applying the hull as a mask to the FBP image.

Protons not removed by the preconditioning cuts [25] are influenced by repeated small-angle multiple Coulomb scattering (MCS), resulting in a curved path [52, 18]. There are no abrupt changes in direction due to large-angle scattering events as these are likely to be removed by the statistical cuts described in [25]. The MLP formalism was developed

assuming a bivariate normal distribution of T and Θ as a function of depth to describe proton scattering in water [53, 54]. The resulting 2-dimensional vector y_{MLP} that maximizes the prior and posterior likelihoods results in the following formula [19]:

$$y_{\text{MLP}} = \begin{bmatrix} T_1 \\ \Theta_1 \end{bmatrix} = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{R}_1^T \boldsymbol{\Sigma}_2^{-1} \boldsymbol{R}_1\right)^{-1} \left(\boldsymbol{\Sigma}_1^{-1} \boldsymbol{R}_0 \vec{y}_0 + \boldsymbol{R}_1^T \boldsymbol{\Sigma}_2^{-1} \vec{y}_2\right) \tag{5.1}$$

where

$$\vec{y}_0 = \begin{bmatrix} T_0 \\ \Theta_0 \end{bmatrix} \quad \boldsymbol{R}_0 = \begin{bmatrix} 1 & U_1 - U_0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} \sigma_{T_1}^2 & \sigma_{T_1\Theta_1}^2 \\ \sigma_{T_1\Theta_1}^2 & \sigma_{\Theta_1}^2 \end{bmatrix} \tag{5.2}$$

$$\vec{y}_2 = \begin{bmatrix} T_2 \\ \Theta_2 \end{bmatrix} \quad \boldsymbol{R}_1 = \begin{bmatrix} 1 & U_2 - U_1 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} \sigma_{T_2}^2 & \sigma_{T_2\Theta_2}^2 \\ \sigma_{T_2\Theta_2}^2 & \sigma_{\Theta_2}^2 \end{bmatrix} \tag{5.3}$$

The elements of the covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, given by $\sigma_{T_i}, \sigma_{\Theta_i}, and \sigma_{T\Theta_i}$ for $i = 1$ and $i = 2$, respectively, are defined as:

$$\sigma_{T_1}^2(U_0, U_1) = C(U_1 - U_0) \int_{U_0}^{U_1} \frac{(U_1 - U)^2}{\beta^2(U) p^2(U)} \frac{dU}{X_0} \tag{5.4}$$

$$\sigma_{\Theta_1}^2(U_0, U_1) = C(U_1 - U_0) \int_{U_0}^{U_1} \frac{1}{\beta^2(U) p^2(U)} \frac{dU}{X_0} \tag{5.5}$$

$$\sigma_{T_1\Theta_1}^2(U_0, U_1) = C(U_1 - U_0) \int_{U_0}^{U_1} \frac{U_1 - U}{\beta^2(U) p^2(U)} \frac{dU}{X_0} \tag{5.6}$$

and

$$\sigma^2_{T_2}(U_1, U_2) = C(U_2 - U_1) \int_{U_1}^{U_2} \frac{(U_2 - U)^2}{\beta^2(U)p^2(U)} \frac{dU}{X_0} \tag{5.7}$$

$$\sigma^2_{\Theta_2}(U_1, U_2) = C(U_2 - U_1) \int_{U_1}^{U_2} \frac{1}{\beta^2(U)p^2(U)} \frac{dU}{X_0} \tag{5.8}$$

$$\sigma^2_{T_2\Theta_2}(U_1, U_2) = C(U_2 - U_1) \int_{U_1}^{U_2} \frac{U_2 - U}{\beta^2(U)p^2(U)} \frac{dU}{X_0} \tag{5.9}$$

where

$$C(U) = E_0^2 \left[1 + 0.038 \ln\left(\frac{U}{X_0}\right) \right]^2 \tag{5.10}$$

with $E_0 = 13.6$ MeV/c and $X_0 = 36.1$ cm is the radiation length of water.

5.2 Mathematical Improvements of MLP Calculations

The scattering of the proton at a particular depth depends on its kinetic energy, which decreases as the proton traverses the object. Accurate calculation of the proton energy without a priori knowledge of the object composition is computationally expensive and currently not implemented. Hence, scattering calculations assume the object has a homogeneous composition of water; the energy-dependent term was approximated by a 5th degree hexanomial as follows:

$$\frac{1}{\beta^2(U)p^2(U)} \approx a_0 + a_1 U + a_2 U^2 + a_3 U^3 + a_4 U^4 + a_5 U^5 \tag{5.11}$$

The coefficients of this hexanomial are determined by conducting Monte Carlo simulations for protons with the expected accelerator energy and performing a best fit analysis to the observed scattering. Note that although the fit obtained for a particular proton energy

remains adequate for a reasonable range of energies, such simulation studies should be performed separately for significant energy differences, particularly for lower energies where even small energy differences have a more significant impact on scattering. The coefficients obtained for 200 MeV protons are:

- $a_0 = 7.457 \times 10^{-6}$
- $a_1 = 4.548 \times 10^{-7}$
- $a_2 = -5.777 \times 10^{-8}$
- $a_3 = 1.301 \times 10^{-8}$
- $a_4 = -9.228 \times 10^{-10}$
- $a_5 = 2.687 \times 10^{-11}$

Equations 5.4-5.9 are translationally and rotationally invariant. This provides freedom in the definition of the local coordinate system (t, u, θ) of each proton used for the proton's MLP calculation. This has the advantage that many of the terms in the polynomial after integration become zero and the y_{MLP} terms simplified by assigning the origin at the object entry point and aligning the coordinate system with the incoming proton trajectory, or mathematically, $u_0 = t_0 = 0$ and $\theta_0 = 0$ (Figure 5.1).

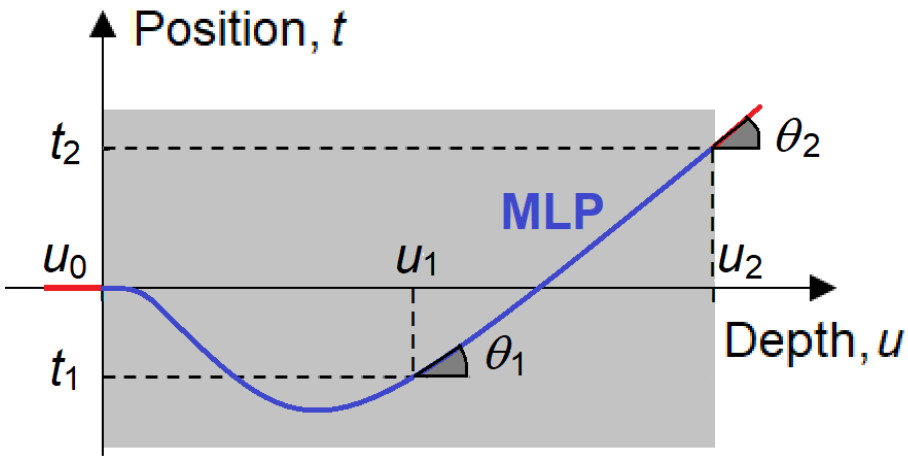


Figure 5.1: Illustration of the MLP in the t-u plane of the proton's reference system. The proton MLP starts at $u_0 = t_0 = 0$ and is initially parallel to the u-axis ($\theta_0 = 0$). A similar path can be drawn in the u-v plane.

Replacing the energy dependent term with the polynomial approximation of Equation 5.11 and expanding the integral yields the following equations:

$$\sigma^2_{t_1}(u_0,u_1) = C(u_1)\int_{u_0}^{u_1}\frac{(u_1-u)^2}{\beta^2(u)p^2(U)}\frac{du}{X_0}$$

$$= C(u_1)P_1(u_1) \tag{5.12}$$

$$\sigma^2_{\theta_1}(u_0,u_1) = C(u_1)\int_{u_0}^{u_1}\frac{1}{\beta^2(u)p^2(u)}\frac{du}{X_0}$$

$$= C(u_1)P_2(u_1) \tag{5.13}$$

$$\sigma^2_{t_1\theta_1}(u_0,u_1) = C(u_1)\int_{u_0}^{u_1}\frac{u_1-u}{\beta^2(u)p^2(u)}\frac{du}{X_0}$$

$$= C(u_1)P_3(u_1) \tag{5.14}$$

and

$$\sigma^2_{t_2}(u_1,u_2) = C(u_2-u_1)\left[P_1(u_2) - u_2^2P_3(u_1) + 2u_2P_4(u_1) - P_5(u_1)\right] \tag{5.15}$$

$$\sigma^2_{\theta_2}(u_1,u_2) = C(u_2-u_1)\left[P_2(u_2) - u_2P_3(u_1) + P_4(u_1)\right] \tag{5.16}$$

$$\sigma^2_{t_2\theta_2}(u_1,u_2) = C(u_2-u_1)\left[P_3(u_2) - P_3(u_1)\right] \tag{5.17}$$

where the polynomials $P_N(u)$ are defined as follows:

$$\begin{bmatrix}P_1\\P_2\\P_3\\P_4\\P_5\end{bmatrix} = \begin{bmatrix}0 & 0 & \frac{a_0}{3} & \frac{a_1}{12} & \frac{a_2}{30} & \frac{a_3}{60} & \frac{a_4}{105} & \frac{a_5}{168}\\ 0 & \frac{a_0}{2} & \frac{a_1}{6} & \frac{a_2}{12} & \frac{a_3}{20} & \frac{a_4}{30} & \frac{a_5}{42} & 0\\ a_0 & \frac{a_1}{2} & \frac{a_2}{3} & \frac{a_3}{4} & \frac{a_4}{5} & \frac{a_5}{6} & 0 & 0\\ 0 & \frac{a_0}{2} & \frac{a_1}{3} & \frac{a_2}{4} & \frac{a_3}{5} & \frac{a_4}{6} & \frac{a_5}{7} & 0\\ 0 & 0 & \frac{a_0}{3} & \frac{a_1}{4} & \frac{a_2}{5} & \frac{a_3}{6} & \frac{a_4}{7} & \frac{a_5}{8}\end{bmatrix}\begin{bmatrix}u\\u^2\\u^3\\u^4\\u^5\\u^6\\u^7\\u^8\end{bmatrix} \tag{5.18}$$

These equations are evaluated at incrementally increasing depths within the object, over the open interval $u_1 = (0, u_2)$, to produce a sufficiently smooth series of coordinates to reliably identify the voxels intersected by the MLP. The $P_N(u)$ and $C(u)$ terms are relatively expensive to compute, particularly given the number of depths u_1 at which they must be recalculated. This is compounded by the cost of manipulating the memory for its data dependencies and output path data. With respect to the other preconditioning and reconstruction tasks, MLP represents the dominant portion of reconstruction time. Depending on the computational hardware used for reconstruction, the explicit implementation of these equations can incur a prohibitive cost. However, several approaches can be employed to optimize computation and reduce the resource burden, which will be discussed in detail in Chapter 5.4.

The impact that the coordinate system translations and rotation have on the vectors $\vec{y}_0$ and $\vec{y}_2$ characterizing the path of a proton at the object entry and exit points is derived in Appendix C.3.2. The resulting equations which are relevant to the MLP calculations are given by:

$$\vec{y}_0 = \begin{bmatrix} t_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.19}$$

$$\boldsymbol{R}_0 \vec{y}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.20}$$

$$\vec{y}_2 = \begin{bmatrix} t_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \sin\Theta_0 \left(U_2 - U_0\right) + \cos\Theta_0 \left(T_2 - T_0\right) \\ \Theta_2 - \Theta_0 \end{bmatrix} \tag{5.21}$$

5.3 Converting MLP Data into Rows of System Matrix

For each proton i, we have the following relation:

$$\sum_{\text{entry}_i}^{\text{exit}_i} \Delta\ell_{i,j} \text{RSP}_j = \text{WEPL}_i \tag{5.22}$$

where RSP_j is the RSP of object voxel j, $\Delta\ell_{i,j}$ is the path length of the ith proton within voxel j, and WEPL_i is the WEPL measured for the ith proton. The objective of pCT reconstruction is to determine the RSP_j from the WEPL measurements and each proton's path. This problem takes the well-known form of $\boldsymbol{Ax} = \boldsymbol{b}$, where Equation 5.22 is the ith equation of the linear system.

For an acquisition scan of m individual protons and an n-dimensional RSP vector space x, the system matrix $\boldsymbol{A}$ is an $m \times n$ matrix whose components are $a_{i,j} = \Delta\ell_{i,j}$. Voxels that are not intersected by the proton are assigned $a_{i,j} = 0$. Each voxel j intersected by the ith proton is determined by its MLP. The MLP is divided into incremental steps Δu and the corresponding voxel j is recorded at each discrete depth $u_1 = u_0 + k\Delta u$. Note that depending on the spacing Δu, voxels intersected with $a_{i,j} < \Delta u$ can be missed. Hence, steps should be made based on the $a_{i,j}$ lengths deemed important to be included in reconstruction. Note that a voxel can be encountered at multiple consecutive steps but is only recorded once.

Calculating the exact path length $a_{i,j}$ of the MLP for each intersected voxel requires the coordinates of its entry and exit points and an integrable expression for the curved path. This is a nontrivial task given the complexity of the analytical expression governing the MLP. Since the MLP has low curvature, the path length is well approximated using a straight-line path inside the voxel, herein referred to as the *chord length*.

The endpoints of the chord are also difficult to determine. It is possible to employ an iterative search algorithm, such as successive projections onto the voxel boundary plane. A parametric expression for the MLP could also be formulated, allowing the endpoints to be calculated analytically. However, performing such tasks for each intersected voxel is computationally expensive. Thus, the approach currently in use is that of an *effective mean chord length*, as described and formulated in (Mean Chord Length 2009) [].

The effective mean chord length is calculated for each proton i, based on voxel dimensions, the MLP step length Δu, and representative $T - U$ and $U - V$ angles for the overall MLP trajectory with respect to the image coordinate system (X, Y, Z). The derivation

considers the chord length in the $T-U$ axial plane and applies a 3D scaling factor for the $U-V$ skew, taking into account the probability of an undetected voxel, i.e. an MLP step skips over an intersected voxel. The objective is for the sum of all the $a_{i,j}$ to be a good approximation of the total path length. The approach is to augment the *mean detected chord length* with a correctional term given by the *mean undetected chord length* weighted by the probability of undetection. The derived expression for the effective mean chord length, $\bar{\Delta}_{\text{eff}}$, is:

$$\bar{\Delta}_{\text{eff}}(\theta) = \bar{\Delta}_{\text{d}}(\theta) + p_u(\theta)\bar{\Delta}_{\text{u}}(\theta) \tag{5.23}$$

where $\bar{\Delta}_{\text{d}}$ is the mean detected chord length, $\bar{\Delta}_{\text{u}}$ is the mean undetected chord length, p_u is the probability of undetection, and θ is the effective angle of the MLP.

The effective mean chord length $\bar{\Delta}_{\text{eff}}$ is subsequently assigned to each voxel j intersected and successfully detected, i.e. $a_{i,j} = \bar{\Delta}_{\text{eff}} \forall\, j \in \text{MLP}_i(\Delta u)$.

Note that the inaccuracies in the approximated $a_{i,j}$ effectively increases the inconsistency in the data and, hence, the size of the feasible region. This is an undesired effect, but it was shown to have minor impact on the solution [55]. This may not continue to be true in the future, particularly once the convergence rate of the pCT reconstruction algorithms can be accelerated.

5.4 Computational Design and Implementation of MLP Calculations

The simplified mathematical expressions presented in Section are computationally more efficient to evaluate in software, even in the case of a naive implementation lacking basic software design principles, due simply to the fact that there are fewer terms and some of the remaining terms are common to two or more scattering elements. For example, note that several of the $P_N(u)$ terms appear as components of multiple scattering elements (e.g. $P_3(u_1)$ appears in four elements), providing an opportunity to compute these terms once per MLP step to reduce the number of compute operations. The $P_N(u_2)$ terms need only

be calculated once per MLP (proton) since u_2 remains constant. For those polynomials that need to be evaluated, rather than naively computing the individual terms of a polynomial separately, it is considerably more efficient to evaluate a polynomial using Horner's rule [56]:

$$\begin{aligned} &a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n \\ &\quad = a_0 + x\Big(a_1 + x\Big(a_2 + x(a_3 + \cdots + x(a_{n-1} + x\, a_n)\cdots)\Big)\Big). \end{aligned} \tag{5.24}$$

Furthermore, since the majority of factors comprising the MLP are a combination of constants and depth $0 < u_1 < u_2$ dependent terms, the operation count can be drastically reduced by precalculating $C(u)$ and the $P_N(u)$ over a sufficiently large range of depths (e.g., the longest feasible path within the reconstruction cylinder). Submillimeter depth granularity yields a lookup table (LUT) occupying ≈ 10 MB for each tabulated term, which fits easily in constant (or texture) GPU memory for quick and easy access. Accuracy within numerical error can be achieved without dipping into micrometer granularity.

The system matrix $\boldsymbol{A} \in {\mathbb{R}_+}^{m\times n}$, with typical values of $m \sim 350\times10^6$ proton histories and $n \sim 2 \times 10^6$ voxels, consumes a large volume of data when the entire matrix is stored in memory. This is alleviated by the fact that any single proton only passes through a small number of the n voxels, resulting in a highly sparse $\boldsymbol{A}$, but the size is still a challenge for parallelization given current GPU capacities. Fortunately, construction of the entire $\boldsymbol{A}$ matrix is not required for image reconstruction, since the iterative reconstruction algorithms project onto individual rows of $\boldsymbol{A}$, either one row at a time or a group of rows simultaneously. The fact that projections are independently performed onto each hyperplane provides a natural framework for the parallelization of the image reconstruction algorithms: each row of the system is assigned to a single GPU thread on which all calculations for that row are performed. This also provides some leeway in how row (hyperplane) data is managed and effectively parallelized by the image reconstruction software.

There are a number of schemes for storing a sparse matrix more efficiently, e.g. compressed sparse row (CSR, a.k.a. compressed row storage (CRS) or Yale) format or compressed sparse column (CSC, a.k.a. compressed column storage (CCS)) format and variations of both. There are other sparse matrix formats, but they are typically more useful in efficient matrix construction, not for the purpose of efficiently accessing matrix elements or performing efficient matrix operations as their use in pCT would demand. The modern CUDA platform for (Nvidia) GPU computing now includes support for sparse matrices and there are a number of highly efficient, task specific CUDA implementations made freely (e.g. via Creative Commons license) available by members of the GPU computing community. Such resources were an attractive option in the development of an efficient MLP computation routine, but the potential use of these was abandoned after careful consideration. If, at some point in the future, pCT can make use of the entire system matrix $\boldsymbol{A}$ for some, as yet, unidentified benefit, sparse matrix formatting schemes would become highly beneficial.

At present, however, operations performed on rows of $\boldsymbol{A}$ are computationally independent of each other. Aside from the accumulation (i.e. summation operations) of update contributions from each row, the rows are also data independent. The objective MLP data represents the set of voxels intersected by the corresponding proton, which is simply a binary assignment of set membership, and their associated path (chord) lengths. However, the path lengths are currently assigned according to the effective mean chord length, which is calculated based on the path angle and subsequently assigned to each voxel intersected by the path. Hence, it is redundant to store this path length for every voxel. Therefore, the practical objective is to determine an efficient scheme for storing the set of intersected voxels.

Set membership is a logical, or binary, property whose explicit (Boolean) storage format would identify an intersected voxel by the value one (TRUE) and the value 0 (FALSE) for all other voxels. An important caveat is that the order of the members of

this set is inconsequential since summations are unaffected by the order of its terms; i.e. there is no restriction on the order of the rows $\vec{a}_i$, which can be inferred from Equation 3.1 and remains true for other iterative projection methods used in pCT (e.g. block-iterative DROP, component-averaged row projections (CARP)).

Consideration of each of the aforementioned properties led to the development of a row storage format with a similar scheme as the list of lists (LIL) format. The LIL format stores, for each row, a list composed of tuples representing the column index and corresponding value. Unlike the LIL format, the lists for each row are stored separately, not stored in a higher-level list. Consequently, since each column of a row has the same value, the format chosen for pCT is a list solely composed of column indices, where each index uniquely identifies a particular voxel in the image space. This will henceforth be referred to as the pCT row (data) format. At some point in pCT development, path lengths will likely be calculated for each individual voxel. The pCT row format can be extended to accommodate this in one of two ways: either by storing the column index/value tuple as in LIL or as an entirely separate, but equivalently ordered, list. The CUDA platform has increased support for C++ data structures, but this flexibility often comes at the cost of efficiency. Therefore, the most reliably efficient option is to store the column index lists, and column value lists if applicable, as individual (C language) arrays. Each array is stored in the local memory of the GPU thread that allocated it, which provides for considerably more efficient access to the array data.

Given an MLP data storage scheme, the next design decisions are the parallelization of MLP computations and the efficient communication of its output path data. Solutions of linear $\boldsymbol{A}\vec{x} = \vec{b}$ systems whose system matrix $\boldsymbol{A}$ or measurement vector $\vec{b}$ are composed of elements with uncertain values, often seek to iteratively reduce the uncertainties and update the system. This is not currently employed in pCT, so the system matrix $\boldsymbol{A}$ remains unaltered throughout iterative image reconstruction. Hence, the ideal situation is calculating

MLP data once and then storing and reusing it throughout the remainder of reconstruction. Unfortunately, this currently isn't a feasible option because there simply isn't enough global GPU memory to store all m rows, even by utilizing the pCT row format. The primary reason for this is the sheer amount of data required as input to MLP calculations. In fact, although FS-DROP is theoretically a fully-simultaneous algorithm, the number of hyperplanes (rows) that can be projected onto at one time. The GPU memory constraint results in an inability to explicitly perform FS-DROP in the form defined by Equation 3.1. The actual implementation of FS-DROP on a GPU currently requires the m hyperplanes to be partitioned into groups that will each fit on the GPU, then successively process each group. This implementation is mathematically equivalent to FS-DROP but takes the form:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + \lambda \boldsymbol{D}^{(k)} \sum_{g=1}^{G} \sum_{i=1}^{S_g} \frac{b_i - \left\langle \vec{a}_i, \vec{x}^{(k)} \right\rangle}{\left\| \vec{a}_i \right\|^2} \vec{a}_i^T \tag{5.25}$$

$$\boldsymbol{D}^{(k)} = \underset{1 \le j \le n}{\text{diag}} \left(\min \left(1, \frac{1}{d_j^{(k)}} \right) \right)$$

where G is the number of groups (i.e. partitions), g is the group index, and all other parameter remain the same. The GPU memory constraint also places limits on block sizes and string lengths in other pCT reconstruction algorithms, but these details will not add to the current discussion and have been omitted.

One option that was considered was calculating the MLP data on the host (i.e. CPU) once and storing the results in host memory, which is typically large enough to accommodate this. This data would then be transferred to the GPU each time it was needed for image reconstruction. Alternatively, MLP data could be generated on the GPU for each group g, then transferred to the host and accumulated from all G groups. In this case, all m rows would be stored on the host, more like the LIL format except in the presumed data structure. Implementing this entails a fairly complicated programmatic design, which differs between first iteration and all other iterations, but it is a manageable difficulty. An attractive consequence of such an approach is that most of the data transferred to the GPU

for MLP calculations doesn't need to be retained after the first iteration. This is only true for the current approach using effective mean chord length, however, otherwise it would be needed to calculate individual path lengths. This is also only applicable if the system matrix $\boldsymbol{A}$ is constant, not if it is improved and updated.

Given some of the shortcomings of this approach, especially in terms of future compatibility, ultimately it was deferred until it could be determined whether the impact of all of the MLP work presented here yielded sufficiently efficient reconstructions or not. To this end, tremendous effort was put into minimizing the number of operations performed in calculating MLP, going so far as to include LUTs for sine and cosine values and reusing variables in $tu-$ and $uv-$coordinate plane MLP calculations. The results were an 80%+ reduction in total compute operations and among the remaining operations, several were simply LUT accesses, exceeding even the most optimistic expectations. With this implementation, the Baylor pCT software was able to perform image reconstructions under the clinical viability constraint of 10 minutes for the first time.

5.5 Conclusion

The addition of the MLP formalism, which models proton behavior within a target volume, represented an important step forward in the development of pCT. The MLP more accurately approximates the path of a proton through an object than other methods, such as cubic splines, thereby increasing the achievable spatial resolution. Although the improved MLP model is only moderately more accurate the cubic splines and much more computationally demanding, since spatial resolution is a key measure of image quality and the most commonly cited deficiency of pCT, the additional computational cost of MLP was deemed to be an acceptable tradeoff.

Calculating each proton's MLP is the most compute intensive task of pCT and has been the primary factor affecting pCT image reconstruction time. The objective of the work presented in this Chapter was to alleviate the resource and computational burden of MLP calculations as part of an effort to drive reconstruction times below the practical 10 minute

limit. The combination of mathematical manipulations of MLP equations (Section 5.4) and a computationally efficient MLP implementation (Section 5.4) resulted in image reconstruction times that were under 10 minutes for the first time.

The effectiveness of the MLP implementation far exceeded expectations. Prior to this development, it was still unclear whether the Baylor software, which is designed for execution on a single GPU node, could ever perform pCT image reconstruction sufficiently quickly. Advancements in GPU hardware and the Baylor pCT software since then have further reduced reconstruction times, in some cases generating pCT images in under 2 min. These performance capabilities provide an opening for potential pCT advancements, both methodologically and practical applications. For example, in an effort to reduce complexity, several approximations are performed at various steps in the reconstruction process, but these should be revisited to determine whether each is still worthwhile.

The replacement of effective mean chord length with more accurate voxel dependent path lengths is of particular interest; although it was determined to have negligible impact on early pCT reconstructions, the advancements in iterative projection algorithms currently being developed are likely to render this false. Other possibilities include (1) the iterative improvement of the system matrix $\boldsymbol{A}$ (typically alternating with updates of $\vec{x}$) and (2) the extension of the MLP formalism to inhomogeneous material compositions. Each of these were considered at one point and rejected in favor of less expensive approaches, but it may now be possible to incorporate one or more of these improvements while still satisfying the 10 minute constraint. Reconstruction times in the 1-2 minute range could also be useful for patient alignment and verification procedures. In any case, the fact that pCT images can be generated from a typically sized data set in under 10 minutes on a relatively low cost compute system (i.e. \$5k-\$10k) is an important achievement in the development of a clinical pCT system.

CHAPTER SIX

Total Variation Superiorization (TVS)

The content presented in the following Chapter was previously published, in an alternative and abridged form, and corresponds to Attribution 2. In accordance with the "Dissertation Reuse" guidelines of IEEE (available at `https://www.ieee.org/content/dam/ieee-org/ieee/web/org/pubs/permissions_faq.pdf`), we note that this chapter contains content that was originally published in the following IEEE journal article [57]:

B. E. Schultze, Y. Censor, P. Karbasi, K. E. Schubert, and R. W. Schulte in the *IEEE Transactions on Medical Imaging* journal under the title "An Improved Method of Total Variation Superiorization Applied to Reconstruction in Proton Computed Tomography"

Furthermore, figures and tables appearing in this Chapter that were published in the preceding article include the copyright notice "© 2020 IEEE" in the corresponding caption.

6.1 Introduction

The hull-detection methods and computational advancements of the MLP formalism presented thus far have primarily represented efforts towards achieving the objective of reconstructing pCT images on a single GPU node in under 10 minutes using the BU pCT software. This capability is a minimum requirement for a clinical pCT system and having achieved this is an important step in the development of pCT. Naturally, the quality of reconstructed pCT images, in terms of both RSP accuracy and noise, is also of paramount clinical importance. The use of hull-detection for object detection was shown to have a positive impact on RSP accuracy, reducing the error to below 1% [58]; RSP accuracy is determined based on region of interest (ROI) analyses of the RSP within different material regions. However, the noise in the reconstructed RSP values is also an important measure of pCT image quality.

Noise is a particularly challenging aspect to deal with in pCT, which has a number of sources of noise. In addition to the typical noise associated with experimental detector measurements, the scattering of protons as they traverse an object results in moderately large variations in trajectory and energy. The pCT imaging model is based on the MCS theory of proton scattering, with considerable effort placed in the removal of pCT data associated with protons that underwent any other scattering. The selection of MCS proton histories narrows the distribution of proton variations, but it is still a considerable source of uncertainty. There isn't much that can be done to reduce the variations in calibrated WEPL values, but the MLP formalism was developed to reduce the uncertainty from MCS in proton paths. The voxels that a proton passed through can be identified fairly accurately from MLP calculated paths, but the actual path through each voxel has more variation.

The combination of uncertainties in proton paths and energies (WEPL), which manifest in the system (path) matrix $\boldsymbol{A}$ and (WEPL) measurement vector $\vec{b}$, yield a moderately inconsistent linear system. The use of an effective mean chord length, rather than individual path length approximations, for all voxels intersected by a proton further increases the inconsistency of the system (as well as the condition number of $\boldsymbol{A}$). The accuracy of the solutions obtained with the feasibility-seeking algorithms used in pCT is fairly tolerant of data inconsistencies, but the noise is propagated through each iteration and eventually grows to dominate the resulting image. In other words, the algorithms are quite sensitive to RSP variations, which provide for good spatial RSP accuracy, but tends to magnify noise with compounding effect through successive iterations. This noise propagation ends up being the basis for halting reconstruction, rather than simply approaching convergence as closely as possible.

Although noise propagation occurs at each feasibility-seeking iteration, it becomes more apparent once the solution has converged to the point where the noise is a larger portion of the subsequent RSP updates. At this point, the standard deviations in RSP, σ_{RSP}, within each ROI reaches a minimum and subsequent iterations are accompanied by

a sharp increase in σ_{RSP}. The exact behavior observed varies depending on the data set and configuration of the image reconstruction software, e.g. size and segementation (voxel dimensions) of the image space and the particular reconstruction algorithm and associated parameter values selected, but the general behavior is ubiquitous in pCT. Since RSP noise is not only an impediment in using pCT image for treatment planning, it effectively places a limit on the number of feasibility-seeking iterations that can be performed and, consequently, RSP accuracy. This situation demanded the development of some method for reducing the noise, or the propagation of noise, in pCT imaging.

A recently developed approach that has been applied in other medical physics applications is the superiorization methodology (SM), which lies somewhere between feasibility-seeking and constrained optimization [59]. The superiorization methodology has also been tested in pCT, where it was performed in conjunction with DROP and shown to improve image quality, particularly in terms of noise [30]. The objective of superiorization is to reduce, but not necessarily minimize, the value of a cost function (often called target function in SM literature) while maintaining compatibility with the problem constraints. Compatibility is maintained by analyzing the perturbation resilience of the underlying algorithm, in our case feasibility-seeking. The iterates of feasibility-seeking are then perturbed accordingly, steering the solution towards a feasible point with a lower cost function value. The computational cost of adding these perturbation steps, when implemented appropriately, is negligible.

The mathematical principles underlying the SM, over general consistent "problem structures" with the notion of bounded perturbation resilience, were introduced in 2010 [60]. The SM framework was subsequently expanded to apply to the case of inconsistent data, such as pCT (sometimes referred to as ill-posed problems), using the concept of strong perturbation resilience [61, 62]. Subsequent investigations demonstrated its efficacy in comparison to the projected subgradient method in their application to constrained

minimization [62]. There are a number of works on the topic of superiorization, from generalized theory and applications [63, 64] to its use in single- and multi-energy x-ray CT imaging[65, 66, 67, 68]. A comprehensive list of SM publications, updated continuously with the latest in active research, appears on a website managed by the foremost expert and originator of the work [69].

In the application of the SM to pCT, the cost function selected is the total variation (TV) of RSP in reconstructed images, where total variation is defined as:

$$\mathrm{TV}^{(k)} = \sum_{j=1}^{n} V_j^{(k)} \tag{6.1}$$

where $V_j^{(k)}$ is the neighborhood RSP variation of the jth voxel, calculated as the normalized sum of (RSP) subgradients into and out of the voxel, and TV is the sum of these variations over all n voxels. In other words, TV is a measure of local RSP variations summed over all voxels in the image $\vec{x}^{(k)}$. A stepwise description of the calculations performed in TVS (e.g. $V_j^{(k)}$) is provided in Appendix D.4). For a more detailed introduction to the use of TV for image analysis, see [70].

With TV defined as the cost function, the implementation of SM is referred to as total variation superiorization (TVS). Although TVS has been applied previously to pCT, showing respectable improvements, the theory and implementations of SM have advanced considerably since then. Hence, the objective here is to investigate these advancements and determine their applicability to TVS applied to pCT image reconstruction. These investigations include both assessment of image quality (RSP accuracy and associated σs and TV) and the computational efficiency of TVS after incorporating the recent novel adaptations of SM.

6.2 Motivation

Iterative feasibility-seeking projection methods have been shown to be effective pCT image reconstruction algorithms [71], but the noise content in the reconstructed images appears as local RSP variations that the algorithms cannot reduce. The aforementioned statistical variations in WEPL manifest as locally (neighborhood) correlated fluctuations in RSP. The feasibility-seeking projection algorithms were specifically selected for their higher noise tolerance [72], as compared to transform methods such as filtered backprojection (FBP) [73, 37], but these algorithms do propagate the noise introduced by WEPL uncertainties. As a result, the noise content of successive iterations of reconstruction exhibit increasing levels of noise, i.e. each feasibility-step amplifies the current noise content. Hence, after a certain number of iterations, which depends on the particular algorithm and its parameters, image quality begins to be degraded by each subsequent iteration. This is explained by the fact that as feasibility-seeking approaches convergence, the updates applied to the image decrease and the noise becomes the dominant content of the updates. This behavior forces image reconstruction to be halted well before the solution has converged to a steady-state. The iterations performed before noise sharply increases are sometimes referred to as "useful iterations".

Given the unavoidable WEPL uncertainty and the accumulating effect of noise propagation, any approach that provides the means to reduce noise before and/or during feasibility-seeking will have compounding benefit. The potential benefit is not only the reduction of RSP fluctuations (clearer images) but can also have an accelerating effect on the convergence behavior. Improving convergence rate is a key objective of future pCT development, as it permits the options to either achieve the same accuracy with fewer iterations (speed benefit) or achieve better accuracy with the same number of iterations (image quality benefit.

Although the feasibility-seeking algorithms used for pCT have a tendency to accentuate the noise content, i.e. reduce the signal-to-noise (SNR) ratio, their sensitivity to

RSP fluctuations also results in a sharpening of the boundaries (or edges) between disparate materials. This can be observed in RSP line profiles through one or more material regions, which demonstrate the effect that successive feasibility-seeking iterations have on RSPs at the interface between two materials. Note that the RSP gap (discontinuity) between neighbors represents an impulse, which are known to be impossible to construct through any finite actions and are prone to behaviors like "ringing" (i.e. Gibb's phenomena); the typical voxel dimensions of pCT images are too large to observe ringing clearly, but the general s-curve shape can be observed at RSP transitions. Since the sensitivity to RSP variations improves transitional RSP edges, which has a direct impact on achievable spatial resolution, it is important that this behavior is not disturbed by any noise reduction method.

The reason TVS was selected for noise reduction is because of its so called "edge preserving" property. This is not strictly true or an inalienable property of TVS, but within the context of superiorized feasibility-seeking with appropriately chosen TVS parameter values, it retains edges quite well. On the other hand, poorly selected TVS parameters can result in inappropriately large perturbations, which can not only blur edge RSP but, perhaps more alarmingly, can also effectively shift the location of the edge. Thus, the investigations of TVS herein include a thorough search of the TVS parameter space. When TVS is incorporated into feasibility-seeking projection algorithms the resulting reconstruction algorithms are referred to as the superiorized (version of) feasibility-seeking algorithms. These superiorized feasibility-seeking algorithms interleave feasibility-steps with TVS perturbation steps in alternating succession. For the purposes of these investigations, feasibility-seeking steps precede TVS perturbation steps.

There are two metrics used herein to approximate the noise content in the reconstructed pCT images: the TV of the image as a whole and the σ_{RSP} within specific ROIs. The TV is an effective measure of the prevalence and magnitude of local RSP variations, but TV alone is unable to differentiate between RSP variations caused by noise and those that are faithful to the object composition. To isolate and analyze the RSP variations due

solely to noise, regions known to be composed of homogenous material (with known RSP) are selected for ROI analysis. A statistical analysis of the reconstructed RSP values within these ROIs provides measures of regional RSP accuracy and the standard deviations associated with regional fluctuations in RSP. These two measures provide the basis for assessing the performance of TVS for different structural and parameter configurations.

6.3 Methods

6.3.1 TVS Algorithm

The first application of TVS to pCT image reconstruction demonstrated the efficacy of the approach [30]. The SM has evolved since then (see Appendix of [74]), particularly in terms of algorithmic structure, and these advancements are potentially beneficial for pCT. There were also some existing aspects of the SM that were not included in these initial investigations. The objective of the work presented here is to incorporate the existing and newly established features of TVS into the BU pCT software and perform analyses of the structural changes under a breadth of parameter conditions. Note that the original TVS implementation was rewritten and incorporated into the BU software to permit appropriate comparisons. For the purposes of the forthcoming discussions, the previous implementation of TVS will be referred to as the "old" TVS (OTVS) algorithm and the version investigated here will be referred to as the "new" TVS (NTVS) algorithm. Note, however, this is not a recommended terminology outside the context of this dissertation. The usage of the terms OTVS/NTVS herein is merely for the sake of brevity and clarity, but in contexts outside these investigations they will henceforth be known simply as TVS.

The mathematical notation associated with the forthcoming discussions is defined in Appendix D.1. Pseudocode definitions of the NTVS and OTVS algorithms are provided in this notation in Appendix D.2 and D.3, respectively. As previously mentioned, there is also a step-by-step definition of the calculations involved in the NTVS algorithm in Appendix D.4.

6.3.2 NTVS Algorithm

The NTVS algorithm is an amalgamation of advancements from a scattered set of publications on the use of TVS in x-ray CT (see Appendix of [74]). Although each of the newly developed properties included in the NTVS algorithm have been investigated for x-ray CT, this is the first time they have been combined into a single implementation of TVS for either x-ray CT or pCT. The properties that were incorporated into the OTVS algorithm to generate the NTVS algorithm are enumerated below:

(1) Exclude step (10) of OTVS (Appendix D.3) verifying TV reduction.

(2) Use a perturbation kernel α and perturbation step counter ℓ to control the perturbation step-size $\beta^{(k)}$.

(3) Add capability to perform a user-selected number N of perturbation steps (step (8) of NTVS (Appendix D.2)) per feasibility-seeking step.

(4) Add the formula $\ell^{(k+1)} = \text{rand}\left(k, \ell^{(k)}\right)$ to (randomly) decrease the perturbation kernel exponent $\ell^{(k)}$ (step (6) of NTVS (Appendix D.2)) used to calculate the perturbation step-size $\beta^{(k)} = \alpha^{\ell^{(k)}}$ between feasibility-seeking steps

The TV reduction verification step of OTVS, step (10) of OTVS (Appendix D.3), after each perturbation is not a particularly expensive task, but such conditional branching can be detrimental to the computational efficiency of a GPU implementation of TVS in the BU software. The majority of the computations comprising an implementation of TVS are parallelizable, but some steps have data dependencies and/or represent innately sequential operations. Fortunately, these parallelization impediments were successfully eliminated by rearranging and/or reformulating some of the governing formulae and employing SGP and DDP parallelization schemes. Furthermore, if the TV reduction verification step can safely be removed without a degradation of pCT image quality, the NTVS algorithm can be implemented to achieve up to a 30% reduction in sequential operations and eliminate the need to perform repeated perturbations until the TV reduction constraint is fulfilled. This

yields a reduction in computation time and, in the latter case, an increase in computational efficiency.

Verification steps like the TV reduction requirement are often added to a theoretical model as a part of a proof of its behavior in the presence of conceivable outlier cases. Mathematical rigor is an important part of algorithm development, but in practical applications such verification steps primarily serve as "safety checks" that are often unnecessary, or even detrimental. In fact, the classical TVS algorithm includes a computationally expensive feasibility proximity verification step [75, 76, 77], but investigations were performed during the development of the OTVS algorithm which found it could safely be removed [30]. The removal of the TV reduction verification step, henceforth referred to by the acronym TVRVS, represents a continuation of the process of removing such safety checks. Since the TVRVS is a structural feature of the TVS algorithm, rather than simply a manageable parameter, there are two versions of the NTVS algorithm written in pseudocode in Appendix D.2: the first without the TVRVS and the second with the TVRVS. These two versions of NTVS are investigated as a function of each TVS parameter for every combination of parameter values.

The OTVS algorithm is implemented with initial perturbation step-size $\beta_0 = 1$, which is subsequently halved after each perturbation step. In terms of the NTVS notation, this halving corresponds with a perturbation kernel $\alpha = 0.5$. This limits the adaptability of the algorithm and provides no control over the aggressiveness of the applied perturbations. For example, in cases where image reconstruction is performed using *block-iterative* or *string-averaging* algorithms with TVS applied to each block or each step along a string, respectively, the equivalent $\alpha = 0.5$ of OTVS is much too aggressive and results in highly smoothed feature reconstructions. Hence, for these scenarios with frequent perturbation steps, the OTVS algorithm simply isn't a viable option and can only be applied once per *full iteration*. However, a perturbation step once per feasibility-seeking step yields a decently sized initial perturbation step, but a halving of the step-size rapidly decreases $\beta^{(k)}$ to

the point where, after the first few feasibility-seeking steps, subsequent perturbation steps are too small to have a meaningful impact on image quality or the suppression of propagated noise. This scenario represents a sub-optimal utilization of TVS. Therefore, the NTVS algorithm includes the parameter α to provide user-control of the rate of decay of the perturbation step-size $\beta^{(k)}$. The performance of TVS is investigated for a range of α values to determine whether values of $\alpha > 0.5$ yield improved image quality relative to OTVS. Note that the convergence of the underlying feasibility-seeking algorithm is preserved by imposing the constraint $0 < \alpha < 1$.

The permutation kernel α has two related impacts on TVS performance: α dictates (1) the size of perturbation steps and (2) the rate of decay (with respect to k) of the size of perturbation steps. Hence, a larger α will result in larger initial perturbation steps but the selected value also dictates a slower decaying perturbation step-size β. This inseparable relationship prevents TVS from being configured to, e.g., apply large initial perturbation steps which quickly decay in successive perturbations. Another feature of the NTVS algorithm that helps to counteract the inseparability is the introduction of a parameter N that specifies the number of consecutive perturbations to perform per feasibility-seeking iteration. This provides the ability to impart larger perturbations without altering α, thereby achieving larger TV reductions with the same β decay rate.

A potential drawback of performing N moderately sized perturbation steps, as opposed to a single large step, is the fact this causes the perturbation kernel exponent to be incremented N times per feasibility-seeking step. Since perturbation step-size $\beta^{(k)} = \alpha^{\ell}$ is governed by ℓ, larger values of N cause perturbations to decay more quickly towards zero. Consequently, as $\beta^{(k)}$ approaches zero, which can occur early in feasibility-seeking depending on the value of N, subsequent applications of TVS are characterized by a general lack of any sizeable benefit. To counteract this accelerated $\beta^{(k)}$ decay and preserve meaningful perturbations throughout reconstruction, the perturbation kernel exponent $\ell^{(k)}$

is reassigned an integer value selected via random sample of the closed interval $\left[k, \ell^{(k)}\right]$; i.e. $\ell^{(k+1)} = \text{rand}\left(k, \ell^{(k)}\right)$.

This approach was developed and investigated for biomedical applications [78, 79], but it has also been applied to maximum likelihood expectation maximization (MLEM) [80] and linear superiorization (LinSup) algorithms [74, Algorithm 4]. In cases where the selected N is small, the interval $\left[\ell^{(k)}, k\right]$ is also small, so a random sample from the narrow (closed) interval will generate roughly comparable perturbation step-size. For $N = 1$ specifically, the interval is only nonzero if TVS includes the TVRVS step and at least one of the perturbation steps failed, otherwise $\ell^{(k)} = k$ throughout reconstruction. On the other hand, in cases where the selected N is large, the difference between iteration k and $\ell^{(k)}$ is also large, so a random sample from the wider (closed) interval will, on average, yield a value that results in a considerably larger perturbation step. The importance of this random decrease in $\ell^{(k)}$ become increasingly apparent as N increases, which causes perturbation step-sizes to decay to negligible levels increasingly early in feasibility-seeking. The relative behavior at $N = 1$ and $N >> 1$ coincide with the features desired for permitting larger perturbations that persist for an increased number of feasibility-steps, which is not possible to decouple in the case without N. At $N = 1$, the TVS behavior is effectively unaltered, where its convergence property is preserved by defining k as the lower bound on the sampling interval, thereby guaranteeing that the perturbation step-size will decay to zero since k is incremented sequentially.

Note that the preceding discussion of the recently developed TVS features included in the NTVS algorithm indicate relationships between the various TVS parameters. Therefore, investigations of the efficacy and performance of the NTVS algorithm include detailed analyses of the impact that each parameter has on one another. For the purposes of these analyses, although the TVRVS step is more aptly classified as an algorithmic structure, it is regarded as another TVS parameter with binary (or Boolean) values: Inclusion of the TVRVS corresponds to the value 1 (or TRUE) and exclusion of the TVRVS is represented

by the value 0 (or FALSE). Details of the TVS parameter space, as well as the pCT data and image reconstruction software configurations, are each presented in the following subsections.

6.3.3 Phantoms and Data Sets

6.3.3.1 Simulated pCT Data.

The simulated pCT data set was generated, using the simulation toolkit geant4 [81], with proton cone beam projections at 90 fixed angular step intervals of 4 degrees ranging from 0 to 356 degrees. The simulation was configured such that the resulting data set contained approximately 120 million proton histories. The simulated data was used for preliminary investigations of NTVS, particularly to assess the variations introduced by the random decrease in $\ell^{(k)}$ between feasibility-seeking steps (step 6 of Algorithm 9). A full description of the platform developed to produce data for pCT has been published [82].

6.3.3.2 Experimental pCT Data.

The experimental pCT data sets were generated, using an experimental pCT scanner [83], with the phantoms mounted on a (platform) stage rotating relative to a fixed horizontal proton beam line producing a rectangular field (using a magnetically wobbled beam spot) over a continuous range of projection angles between 0 and 360 degrees. The configuration of the pCT scanner results in the production of data sets with approximately 250 million proton histories. The experimental data sets were used for practical viability investigations of NTVS in the case of two, anatomically and compositionally different, phantoms.

The definitive investigations were then performed using two experimental data sets: (1) the same Catphan® CTP404 phantom and (2) an HN715 pediatric anthropomorphic head phantom, both obtained with an experimental pCT scanner [83] yielding approximately 250 million proton histories for each object.

6.3.3.3 *Catphan® CTP404 Phantom.*

A Catphan® CTP404 phantom module (The Phantom Laboratory Incorporated, Salem, NY, USA) was selected, for both simulated and experimental data sets, due to the simplicity of its anatomical structures and the accuracy to which their sizes, locations, and compositions (in terms of RSP) are known (see Figure 6.1).

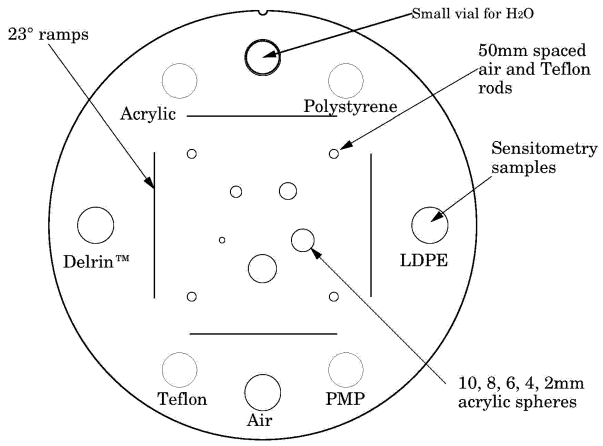


Figure 6.1: Catphan® CTP404 phantom composition and geometry of the material inserts. © 2020 IEEE

It is a 15 cm diameter by 2.5 cm tall cylindrical phantom composed of an epoxy material with RSP $\approx$ 1.144. In the geant4 simulation, the epoxy RSP was explicitly set to that of water (RSP = 1.0) because, at the time, the RSP of the epoxy had not been determined. The RSP values used in the simulation and the experimentally measured RSP values for each phantom material are defined in Table 6.1.

Table 6.1: RSP of the material inserts for the simulated and experimental Catphan® CTP404 data sets. © 2020 IEEE

Data Set	Air	PMP	LDPE	Epoxy
Simulated	0.0013	0.877	0.9973	1.024
Experimental	0.0013	0.883	0.979	1.144

Data Set	Polystyrene	Acrylic	Delrin	Teflon
Simulated	1.0386	1.155	1.356	1.828
Experimental	1.024	1.160	1.359	1.79

The phantom is composed of three geometrically different types of contrasting material inserts, each of whose geometric centers are evenly distributed around circular patterns of varying diameter d. The inserts of interest for the present investigations are those associated with the eight 12.2 mm diameter cylindrical holes in the phantom at $d = 120$ mm. In this case, two of the holes are not filled, resulting in two air-filled regions. The other six are each filled with a unique material of known composition: acrylic, polymethylpentene (PMP), low density polyethylene (LDPE), Teflon®, Delrin®, and polystyrene.

6.3.3.4 HN715 Pediatric Anthropomorphic Head Phantom.

The pediatric anthropomorphic head phantom (model HN715, CIRS, Norfolk, VA, USA) is designed to accurately represent the head of a young human. The phantom contains anatomically accurate features composed of tissue equivalent materials. For the purposes of these investigations, the materials of interest are soft tissue, brain tissue, and trabecular bone; the RSP of each of these materials is provided in Table 6.2. Since the geometry of the anatomical features cannot be accurately characterized by simple geometric shapes and positions, these are not given. The method for identifying ROIs will be explained in the following subsection.

Table 6.2: RSP of the tissue/bone regions of interest analyzed in the pediatric head phantom. © 2020 IEEE

Data Set	Soft Tissue	Brain Tissue	Trabecular Bone
Experimental	1.037	1.047	1.108

6.3.4 Image Reconstruction Details

Configuration of the BU pCT software is the same as that described in Chapter 3. The details of the BU software implementation used in these investigations has been described in a previous publication [58]. Details of the WEPL calibration, preprocessing,

and preconditioning steps have also been described, in detail, in a recent publication [25]. Image reconstruction was performed within a $20 \times 20 \times 5$ cm^3 volume and partitioned into $1.0 \times 1.0 \times 2.5$ mm^3 voxels, thereby yielding 200×200 images for each axial slice.

The implementation of the random decrease of $\ell^{(k)}$ made use of a random number generator which was given a random seed (based on the Julian time of BU pCT software execution). This ensures that a different set of random numbers is generated for each image reconstruction, thereby preventing biased effects in the pCT images which would skew analysis.

6.3.5 Reconstruction Parameter Space

The TVS parameters and corresponding values that were systematically investigated are described in the following. Note that, for the purposes of this investigation of NTVS viability, each parameter was given fixed values for the entirety of the corresponding image reconstruction.

6.3.5.1 Inclusion or Exclusion of TVRVS.

To isolate the impact of removing the TVRVS constraint of OTVS from the other changes in NTVS, this constraint is considered another parameter variable. Hence, a second version of NTVS (Algorithm 10) that includes the TVRVS was defined solely for the purpose of comparing the impact of excluding TVRVS for all combinations of parameters and parameter values.

6.3.5.2 The Number of TV Perturbations per Feasibility-Seeking Iteration.

Preliminary investigations revealed a considerable degradation of image quality when exceeding $N \approx 10$. To account for this, while providing some flexibility for the potential differences with experimental data or other phantoms, the investigated range of values is $1 \leq N \leq 12$.

6.3.5.3 *The Perturbation Kernel Coefficient.*

Previous (unpublished) investigations with the OTVS algorithm indicated that $\alpha = 0.5$ had only produced negative results when it was applied for each block of a block-iterative feasibility-seeking algorithm, which would correspond to 10s-1000s of individual perturbation steps per full iteration of feasibility-seeking. Hence, the decision was made not to test any values lower that the $\alpha = 0.5$ equivalent to OTVS. However, $\alpha = 0.5$ was included in these investigations to allow overlap with OTVS. Hence, the values that are investigated are $\alpha = \{ 0.5, 0.65, 0.75, 0.85, 0.95\}$.

6.3.5.4 *The Selection of Relaxation in Feasibility-Seeking.*

As the relaxation parameter λ increases beyond its optimal fixed value (associated with a particular block or group size), the noise content (in terms of ROI standard deviations in RSP) of the resulting images increases. Increasing λ results in faster rates of convergence, but also increasing levels of noise. Hence, investigations only include the optimal and larger than optimal values of λ to investigate its interaction with TVS parameters and determine if NTVS would reduce noise enough to permit larger λ. Therefore, the relaxation parameter values investigated are $\lambda = \{ 0.0001, 0.00015, 0.0002\}$. Note that a direct comparison of performance for different values of λ is impossible since λ dictates the rate of convergence of the feasibility-seeking algorithm. Hence, comparisons of performance after a fixed number of feasibility-seeking steps (iterations) correspond to wildly different point of convergence for each λ. The approach taken here is to define an RSP accuracy based on $\lambda = 0.0001$ after $k = 12$ iterations and determine the number of iterations it takes for each of the other λ values to reach a comparable accuracy. The comparison of performance is then performed using the results obtained with the other λ after the number of iterations k that produced comparable RSP. Of course, the RSP accuracies are not identical or completely consistent across all ROI, but are (perhaps surprisingly) reasonably close to each other to be sufficient for general behavior inferences.

6.4 Computation Hardware and Performance Analysis

The BU pCT software was executed on a single node of the Tardis GPU cluster at the Baylor Research and Innovation Collaborative (BRIC) facility. The program read the input pCT data from a local SSD and performed all parallel computation on a single NVIDIA k40 GPU. Including the time dedicated to reading pCT data from disk and writing reconstructed images to disk, reconstruction times were about 6 minutes for the most demanding cases (i.e. $k = 12$ feasibility-seeking steps, each with $N = 12$ perturbation steps) and slightly faster for less demanding cases.

Analyses of the CTP404 phantom were performed on the central slices of the reconstructed images. This slice was selected due to the presence of five acrylic spheres (diameters 2,4,6,8,10 mm) arranged more centrally than cylindrical inserts ($d = 30$ mm) at the middle height of the phantom (i.e. 1.25 cm down its cylindrical axis). Their presence results in a greater variation in material composition traversed by the protons, and, hence, a greater variation in their paths and WEPL values. As previously noted, such variations manifest in the image as noise, resulting in these central slices possessing higher RSP fluctuations. Therefore, these slices were selected for their greater levels of noise as an ideal scenario for comparative analyses of the ability of the OTVS and NTVS algorithms to reduce noise content. A representation reconstruction of this central slice is shown below in Figure 6.2.

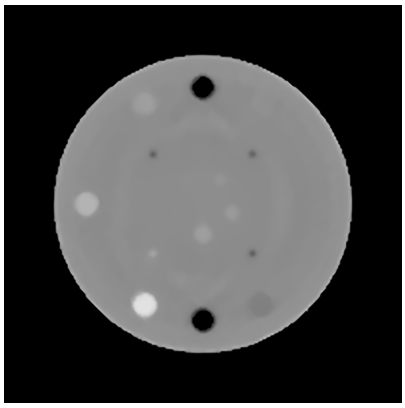


Figure 6.2: Representative reconstruction of the central slice of the CTP404 phantom from simulated data. © 2020 IEEE

The quantitative analyses of pCT image quality were performed using an automated analysis program developed specifically for pCT image analysis at BU. This program performs higher level tasks and interfaces with macros from the image analysis program ImageJ2 1.51r [84] to utilize its ROI analysis tools. For the CTP404 phantom, the cylindrical inserts were analyzed using a circular ROI with 10 mm diameter (from insert center) and calculating the mean and standard deviation in RSP associated with the voxels lying entirely within the ROI. For the HN715 head phantom, ROI selections are composed of polygonal shapes (i.e. polygonal chains or, colloquially, polylines) which attempt to select a region containing only the desired material, with as few voxels of disparate composition as possible. The resulting ROIs defined for analysis are illustrated as shaded regions with annotated labels in Figure 6.3.

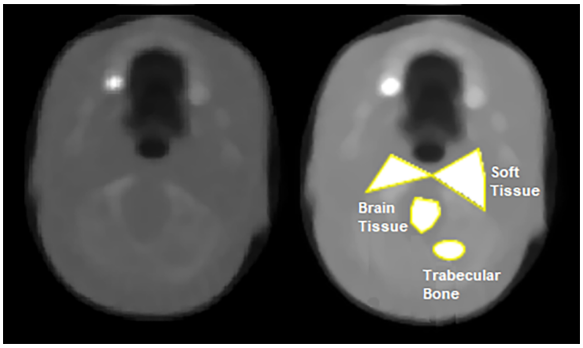


Figure 6.3: Representative reconstruction of the slice of the pediatric head phantom containing the analyzed regions of interest (left); the analyzed regions of interest are filled in white and labeled in the image on the right.

The RSP error, calculated as a percentage for each ROI (given arbitrary indices i), used for these analyses is defined as:

$$E_{\mathrm{RSP},i} = \frac{\overline{\mathrm{RSP}_i} - \mathrm{RSP}_i}{\mathrm{RSP}_i} \times 100 \tag{6.2}$$

where $\overline{\mathrm{RSP}_i}$ is the mean measured RSP within the ith ROI, and RSP_i is the known RSP (i.e. Tables 6.1 and 6.2) of the ith ROI material.

In accordance with [70], total variation was calculated according to Equation 6.1.

6.5 Results

The results of the multi-parameter space investigation of NTVS, including the relationships between TVS parameters, is presented in the following three subsections. The preliminary investigation of NTVS using the simulated CTP404 is presented first, followed by the investigations using the experimental pCT data sets for, in order of presentation, the CTP404 and HN715 phantoms. The following presentation is in the context of image reconstruction and, thus, feasibility-seeking steps are henceforth referred to as feasibility-seeking iterations. Note that the data points appearing in plots in the following presentation each correspond to an independently executed image reconstruction with associated fixed value parameters. The parameter values and combinations are also fixed, i.e., the same set of parameter configuration are investigated for each of the three investigated pCT data sets. For the sake of brevity, in some cases the acronym FS is used to refer to feasibility-seeking in a general context, but this does not apply to the "FS" in the acronym FS-DROP, which remains as defined in Chapter 3.2. Similarly, in discussions of performance with respect to the standard deviation in RSP within a particular material insert ROI, the ROI is simply referenced by its constituent material; e.g. standard deviation in LDPE or standard deviation (LDPE) refer to the standard deviation in RSP within the ROI defined for the LDPE material insert.

6.5.1 Simulated CTP404 Data Set

6.5.1.1 Number of TVS Steps (N).

The number of TV perturbations per FS-DROP iteration, N, was varied between 1 and 12 in increments of 1. Figure 6.4 shows the dependence of TV as a function of N for each of the first four FS-DROP iterations with the TVRVS excluded. As will be shown later, a similar pattern was observed with the TVRVS included. The general effect of increasing N was a reduction in TV that leveled off after $N \geq 5$ steps, as best seen in the $k = 1$ iteration plot (top left of Figure 6.4). An irregular oscillation in TV as a function of

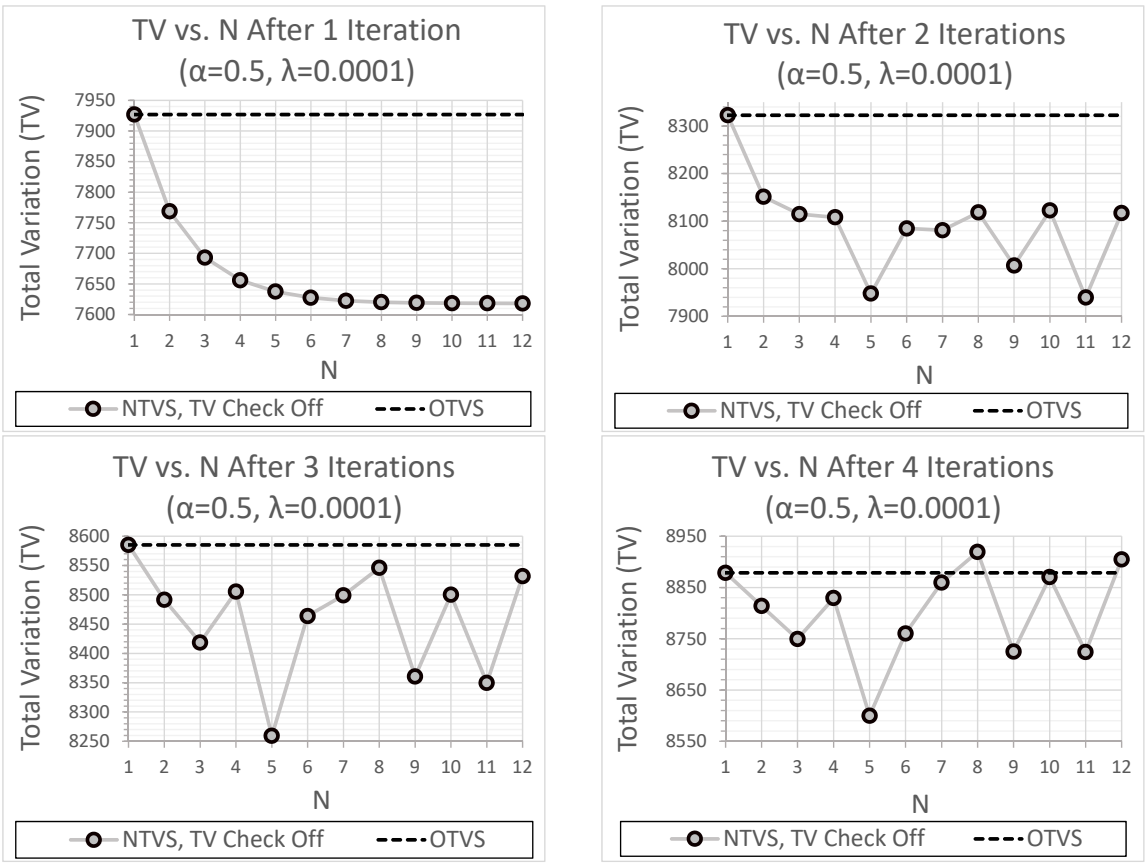


Figure 6.4: TV as a function of N after each of the first 4 FS-DROP iterations for the simulated CTP404 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$ and $\alpha = 0.5$.

increasing N appeared for $k \geq 2$ and increased in magnitude as the number of FS-DROP iterations k increased.

To determine whether the observed fluctuations were random, an analysis of 8 separate reconstructions with $N = 5$, $\alpha = 0.5$, and the TVRVS excluded were performed for $k = 12$ FS-DROP iterations. The standard deviation within the LDPE insert varied between reconstructions with a standard deviation of $\sigma_{LDPE} = 0.00038$ (shown as an error bar on the point at $N = 5$ in Figure 6.5(b)); similar variations were also seen in the ROI of the other materials. Note that the standard deviation obtained within the LDPE insert at $N = 5$ with the TVRVS excluded was nearly $2\sigma_{LDPE}$ less than that obtained with the requirement included and just under $4\sigma_{LDPE}$ less than that obtained with OTVS. In addition, the standard deviation obtained with $N = 5$ was at least $1.5\sigma_{LDPE}$ less than that obtained with any other

value of N. These differences are large enough to conclude that the observed fluctuation in standard deviation as a function of N was not random.

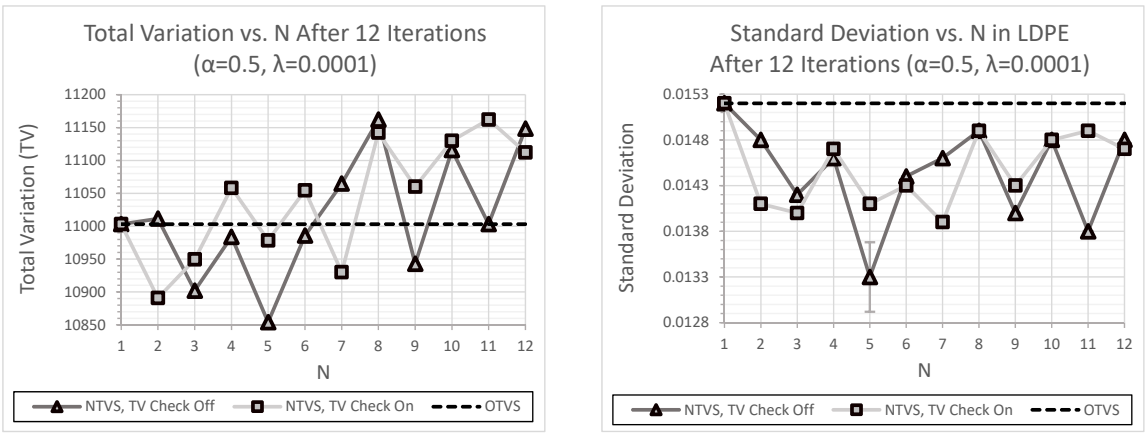
For $3 \leq N \leq 6$, there was a benefit from NTVS compared to OTVS, which persisted throughout all twelve FS-DROP iterations (see Figure 6.5(a) and (b)). However, for $N \geq 7$ the benefits of NTVS were increasingly lost as N and k increased. This can be explained by the decreasing magnitude of TV reducing perturbations with increasing N and the overall increase in TV from each FS-DROP iteration. Although not shown here, a similar dependence on N and k was seen for regional standard deviations. However, the benefit of NTVS in terms of standard deviation was consistently seen, including for $N \geq 7$, after twelve FS-DROP iterations (see, e.g., Figure 6.5(b)).

6.5.1.2 Inclusion/Exclusion of TVRVS.

To determine if the exclusion of the TVRVS in the definition of the NTVS algorithm (Appendix D.2) is an appropriate decision, reconstructions were also performed with a variation of the NTVS algorithm that included the TVRVS; the definition of the algorithm used for these investigations is provided for reference at the end of Appendix D.2.

Figures 6.5(a) and 6.5(b) show the comparison of TV and standard deviation, respectively, for OTVS and NTVS with relaxation parameter $\lambda = 0.0001$, median filter radius $r = 2$ applied to the initial iterate [58], and 12 FS-DROP iterations. In each plot, the results for NTVS with and without inclusion of the TVRVS are shown as a function of N. The horizontal line corresponds to the result of OTVS ($N = 1$, $\alpha = 0.5$).

In the range of $3 \leq N \leq 6$, including the TVRVS had practically no benefit, whereas its removal yields up to a 5.7% reduction in the standard deviation in RSP within the LDPE material insert and up to a 1.2% reduction in overall TV. Similar results were obtained for other values of α, λ, and, in the case of standard deviation, for different materials. One can conclude that imposing the TVRVS does not provide a consistent benefit in terms of TV and standard deviation. Therefore, for the remainder of the parameter space exploration, the TVRVS was excluded.

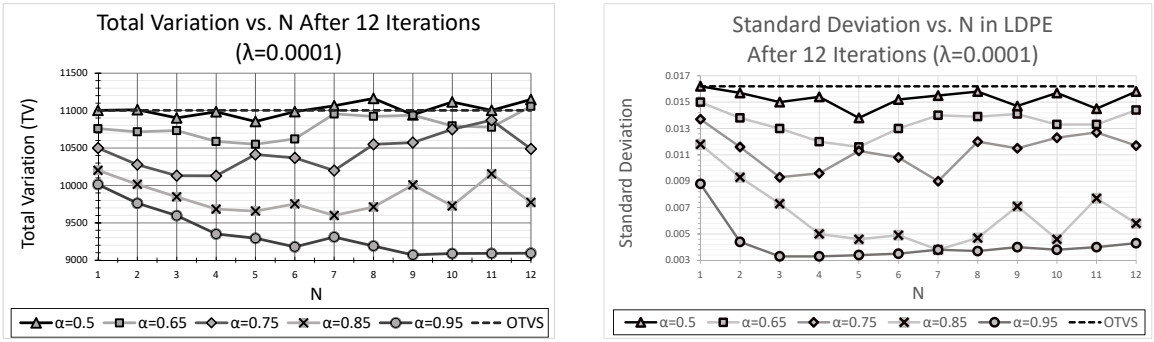


(a) (b)

Figure 6.5: (a) TV and (b) standard deviation (LDPE) as a function of N after 12 FS-DROP iterations for the simulated CTP404 data set using OTVS and NTVS including and excluding the TVRVS with $\lambda = 0.0001$ and $\alpha = 0.5$. The error bar at $N = 5$ denotes the variation in standard deviation ($\sigma = 0.00038$) between 8 repetitions of reconstruction with $N = 5$.

6.5.1.3 Perturbation Kernel (α).

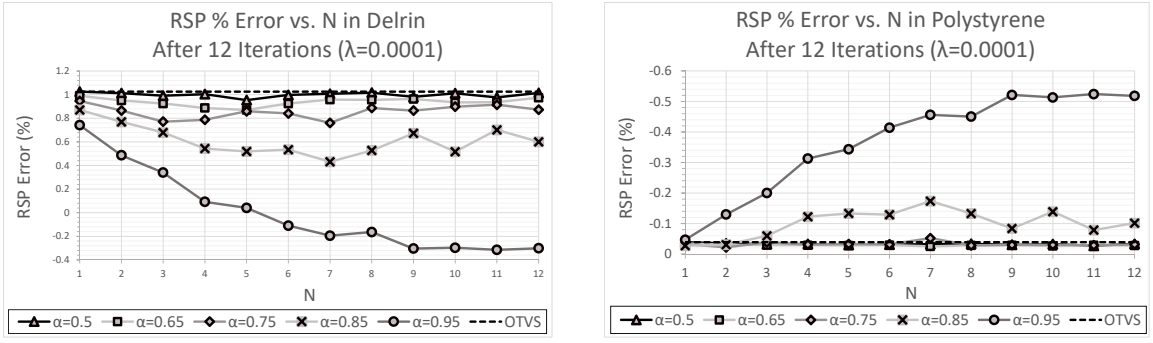
Further investigations were performed to determine the effect of the perturbation kernel α (see step (10) of the NTVS algorithm in Appendix D.2) on TV and standard deviation for $0.5 \leq \alpha \leq 0.95$ and $1 \leq N \leq 12$. Increasing α produces larger perturbations and results in the perturbation magnitude $\beta^{(k)}$ converging to zero more slowly. Thus, one can expect a larger reduction of TV and standard deviation for larger values of α. Figures 6.6(a) and 6.6(b) demonstrate this effect.



(a) (b)

Figure 6.6: (a) TV and (b) standard deviation (LDPE) as a function of N after 12 FS-DROP iterations for the simulated CTP404 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$ and $\alpha = 0.5$.

Figures 6.7(a) and 6.7(b) show the effect of α on the accuracy of reconstructed RSP values in the Delrin and polystyrene inserts, respectively. These two materials were chosen because they were most affected by the value of α. From these plots, one can see that for $\alpha > 0.75$, perturbations have a growing effect on RSP accuracy as α and N increase. This leads to changes in error greater than 1% for Delrin and greater than 0.5% for polystyrene. Although increasing α to decrease TV and standard deviation is a worthwhile goal, one cannot do so without considering its effect on RSP error. On the other hand, increasing α from $\alpha = 0.5$ to $\alpha = 0.75$ yielded up to a 39.3% reduction in the standard deviation in RSP within the LDPE material insert and up to an 8.2% reduction in overall TV without negatively impacting RSP error.



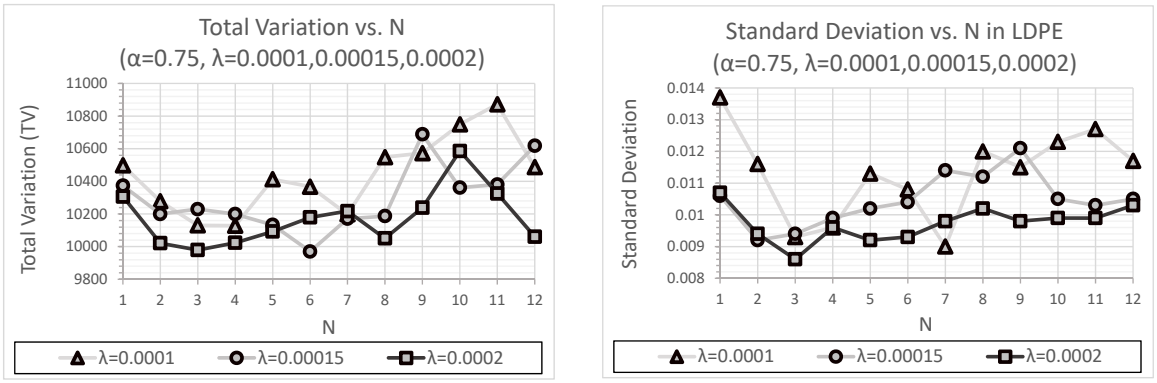
(a) (b)

Figure 6.7: RSP error in the (a) Delrin and (b) polystyrene ROIs as a function of N after 12 FS-DROP iterations for the simulated CTP404 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$ and varying α.

6.5.1.4 Relaxation Parameter (λ).

Increasing the relaxation parameter accelerates the rate of convergence of the FS-DROP algorithm. To investigate the impact of NTVS independent of convergence rate, the number of iterations was adjusted for $\lambda = 0.00015$ and $\lambda = 0.0002$ to obtain the same RSP accuracy as for $\lambda = 0.0001$ and 12 iterations. For this comparison, $\alpha = 0.75$ was chosen.

Figures 6.8(a) and 6.8(b) show TV and standard deviation within the LDPE ROI, respectively, for three combinations of λ and k. For most values of N, the relative improvements in TV and standard deviation increased as λ increased. Note that the trend for standard deviation was not as pronounced for other materials, but increasing λ consistently produced comparable or larger reductions in TV and standard deviation in each material region.



(a) (b)

Figure 6.8: (a) TV and (b) standard deviation (soft tissue) as a function of N for $\lambda = 0.0001$, $k = 12$; $\lambda = 0.00015$, $k = 8$; and $\lambda = 0.0002$, $k = 6$ iterations, respectively, and $\alpha = 0.75$ for the simulated CTP404 data set.

6.5.2 Experimental CTP404 Data Set

6.5.2.1 Number of TVS Steps (N).

Figure 6.9 shows plots of TV as a function of $N = 1..12$ for the first four FSDROP iterations when the TVRVS is excluded. As compared to the corresponding plots from the simulated data set, TV can be seen to exhibit the same general behavior but with less fluctuation: increasing N is accompanied by a monotonic TV reduction for the first iteration, but the TV for larger values of N can be seen to increase with successive iterations and eventually surpass that obtained with OTVS. Repeating TVS $3 \leq N \leq 6$ times per FS iteration yielded a consistent improvement relative to OTVS, similar to the observations

for the simulated data, and can be deemed advantageous. On the other hand, $N \geq 7$ can be seen to be too many TVS repetitions to be beneficial throughout reconstruction.

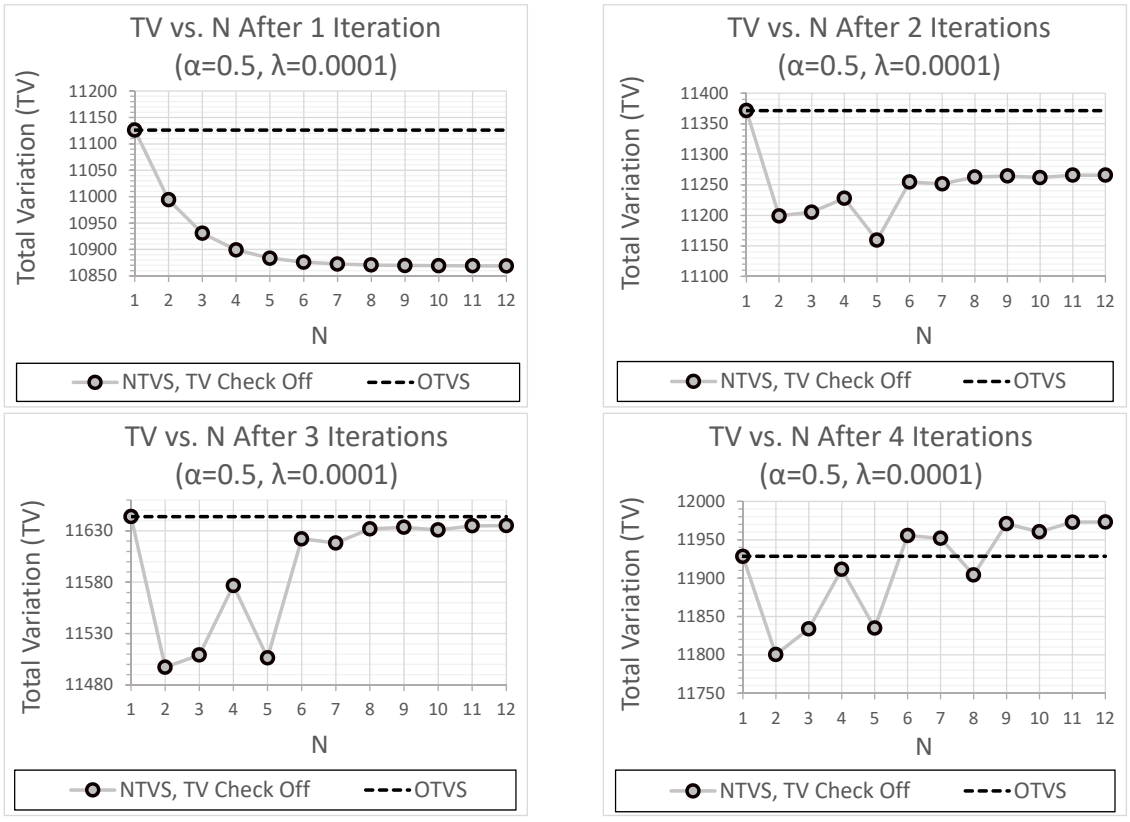
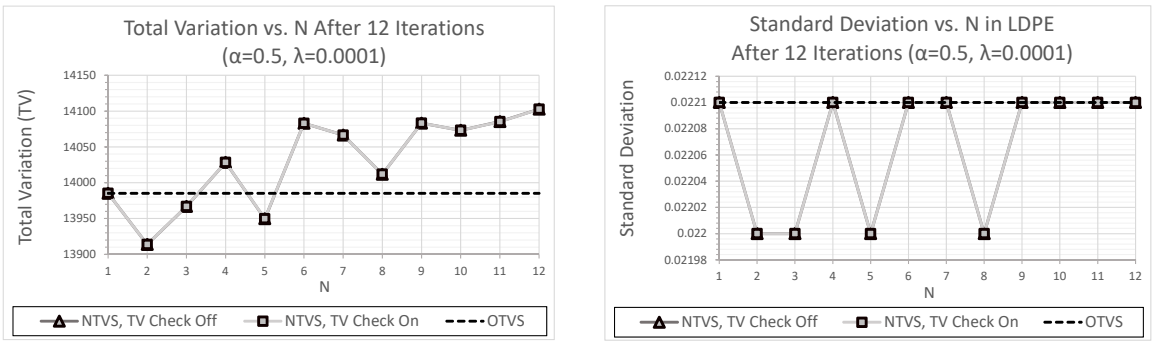


Figure 6.9: TV as a function of N after each of the first 4 FS-DROP iterations for the experimental CTP404 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$ and $\alpha = 0.5$. © 2020 IEEE

6.5.2.2 Inclusion/Exclusion of TVRVS.

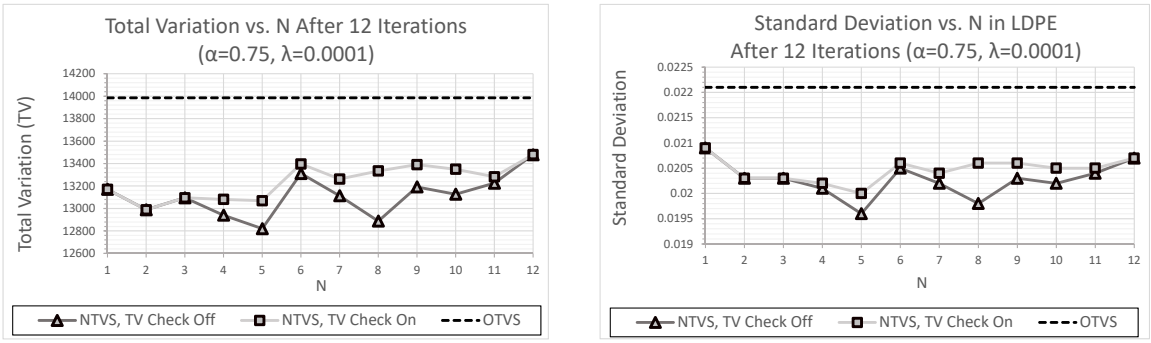
Selected results of an investigation of the exclusion of the TVRVS in NTVS are shown for $\lambda = 0.0001$ in Figures 6.10(a) and (b) for $\alpha = 0.5$ and Figures 6.11(a) and (b) for $\alpha = 0.75$, demonstrating a comparison of performance in terms of TV and standard deviation (LDPE), respectively, for inclusion and exclusion of the TVRVS as a function of N.

For $\alpha = 0.5$, the plot lines for inclusion and exclusion of the TVRVS are not distinguishable and completely overlap, which was also observed for each of the other values of



(a) (b)

Figure 6.10: (a) TV and (b) standard deviation (LDPE) as a function of N after 12 FS-DROP iterations for the experimental CTP404 data set using the OTVS algorithm and the NTVS algorithm including and excluding the TVRVS with $\lambda = 0.0001$ and $\alpha = 0.5$ (note that the 2 NTVS curves overlap). © 2020 IEEE



(a) (b)

Figure 6.11: (a) TV and (b) standard deviation (LDPE)as a function of N after 12 FS-DROP iterations for the experimental CTP404 data set using the OTVS algorithm and the NTVS algorithm including and excluding the TVRVS with $\lambda = 0.0001$ and $\alpha = 0.75$. © 2020 IEEE

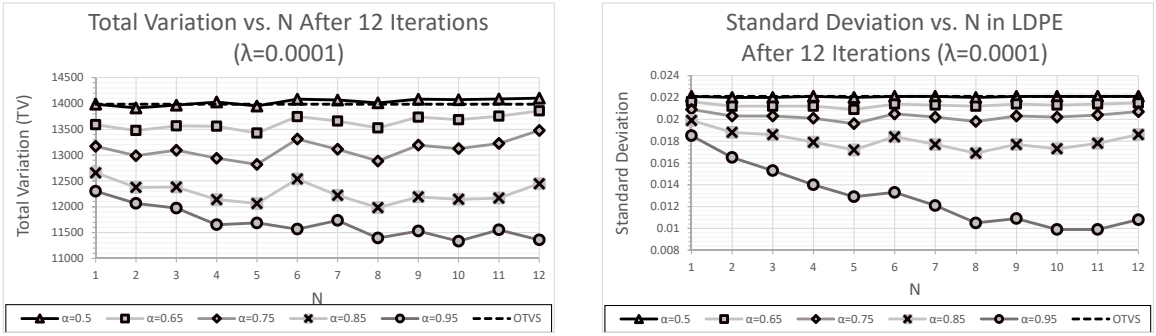
λ. As for standard deviation, the same overlap was observed for each material insert ROI. On the other hand, for $\alpha = 0.75$, excluding the TVRVS consistently yielded lower TV and standard deviation (LDPE). Again, this behavior was observed for each value of λ and for each material insert ROI.

Notice the difference in the scale of the $\alpha = 0.5$ and $\alpha = 0.75$ plots, indicating (relative to OTVS) relatively small improvements in TV and standard deviation for $\alpha = 0.5$ but sizeable improvements for $\alpha = 0.75$ even for $N \geq 7$. The 50% increase in α yielded improvements relative to OTVS with approximately the same magnitude as the

peak difference in performance as a function of N. The range $3 \leq N \leq 6$ exhibited this behavior more prominently, where the improvement in standard deviation (LDPE) relative to OTVS was more than double the peak difference in standard deviation for the N in this range. This behavior was also observed, with slight variation in magnitudes, for each material ROI.

6.5.2.3 Perturbation Kernel (α).

Selected results of an investigation of the perturbation kernel α of NTVS (TVRVS excluded) are shown for $\lambda = 0.0001$, demonstrating a comparison of performance in terms of TV (Figure 6.12(a)), standard deviation (LDPE,Figure 6.12(b)), and RSP error (Delrin, Figure 6.13) as a function of N.



(a) (b)

Figure 6.12: (a) TV and (b) standard deviation (LDPE) as a function of N after 12 FS-DROP iterations for the experimental CTP404 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$ and different values of α. © 2020 IEEE

Figures 6.12(a) and (b) demonstrate performance that coincides with expectations: increasing α, which governs perturbation step-sizes and decay) yields increasing reductions in TV and standard deviation. The improvements are not as large as those for the simulated data, but the same general trend as a function of N is observed: incrementally increasing α by 0.1 yields increasingly larger improvements each time. However, it can also be seen that as N increases beyond some point, the result of incrementing $\ell^{(k)}$ so many times per

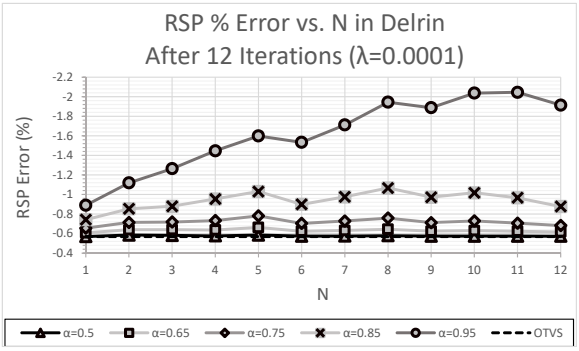


Figure 6.13: RSP error (Delrin) for each value of α as a function of N after 12 FS iterations for the experimental CTP404 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$. © 2020 IEEE

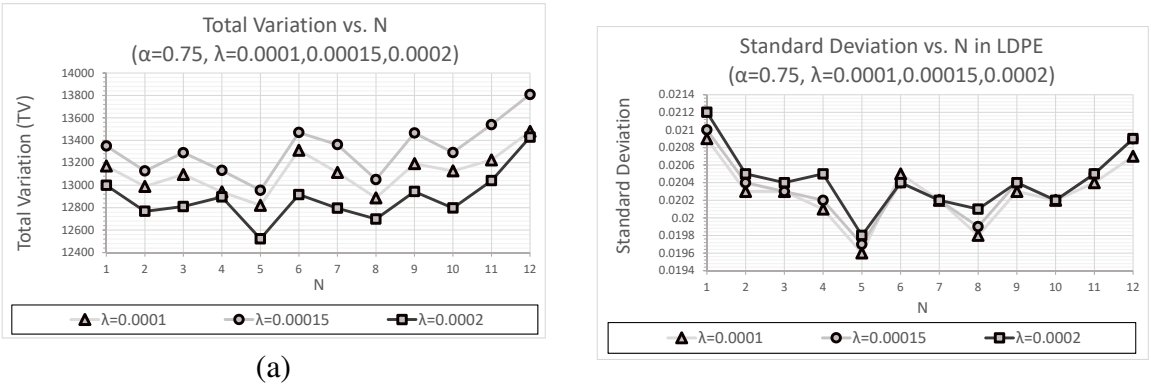
iteration causes diminishing improvements for each additional increment of N. In other words, at some (inconsistent) value of N the TV and standard deviation begin to increase.

The behavior of NTVS as a function of α essentially mirrors expectations. The primary unknown was the value of α that resulted in perturbations larger than the resilience of FS-DROP, thereby actively affecting the reconstructed RSP values. This was observed for each material insert, but was particularly profound for the Delrin insert (Figure 6.13), where values of $\alpha >\approx 0.75$ have a pronounced effect on reconstructed RSP error. For $\alpha > 0.75$, RSP error was seen to increase for every material ROI. However, such RSP errors effects are not always omnidirectional, on the contrary, results indicating increases and decreases in RSP error were observed for different material ROIs with the simulated data. For example, the error in Delrin here is seen to increase, whereas for the simulated data, the error was seen to decrease for the same ROI.

6.5.2.4 Relaxation Parameter (λ).

Selected results of an investigation of the relaxation parameter λ are shown for $\alpha = 0.75$ and the TVRVS excluded in Figures 6.14(a) and (b), demonstrating a comparison of performance in terms of TV and standard deviation (LDPE), respectively, as a function of N. The RSP accuracy for $\lambda = 0.0001$ after $k = 12$ iterations was comparably obtained with $k = 8$ iterations for $\lambda = 0.00015$ and $k = 6$ iterations for $\lambda = 0.0002$. The TV comparison

plot demonstrates larger reductions in TV as λ increases. This is primarily due to the fewer number of iterations performed in obtaining the same RSP accuracy, as FS tends to propagate and amplify noise with each successive iteration. However, in contrast to the simulated data results, the standard deviation (LDPE) plot demonstrates a slight increase in standard deviation for the same LDPE insert. The best results for TV and standard deviation were obtained with $N = 5$ for each λ investigated, which was consistently observed for each material ROI. In fact, each λ yielded nearly identical behavior as a function of N for every material ROI.



(a)

Figure 6.14: (a) TV and (b) standard deviation (LDPE) as a function of N for $\lambda = 0.0001$, $k = 12$; $\lambda = 0.00015$, $k = 8$; and $\lambda = 0.0002$, $k = 6$ FS-DROP iterations, respectively, and $\alpha = 0.75$ for the experimental CTP404 data set. © 2020 IEEE

6.5.3 Experimental HN715 Pediatric Head Phantom Data Set

6.5.3.1 Number of TVS Steps (N).

Figure 6.15 shows plots of TV as a function of N for the first four FS-DROP iterations for the case where $\lambda = 0.0001$ and the TVRVS is excluded. These results are very similar to those of the experimental CTP404 phantom, particularly for $N \leq 6$, but unlike for both the simulated and experimental CTP404 data sets, the benefits of NTVS do not degrade as quickly for $N \geq 7$ and continue to outperform OTVS for all values of N. However, the optimal values of N after 4 FS-DROP iterations occur at $N = 2$ and $N = 5$ for all 3 data sets.

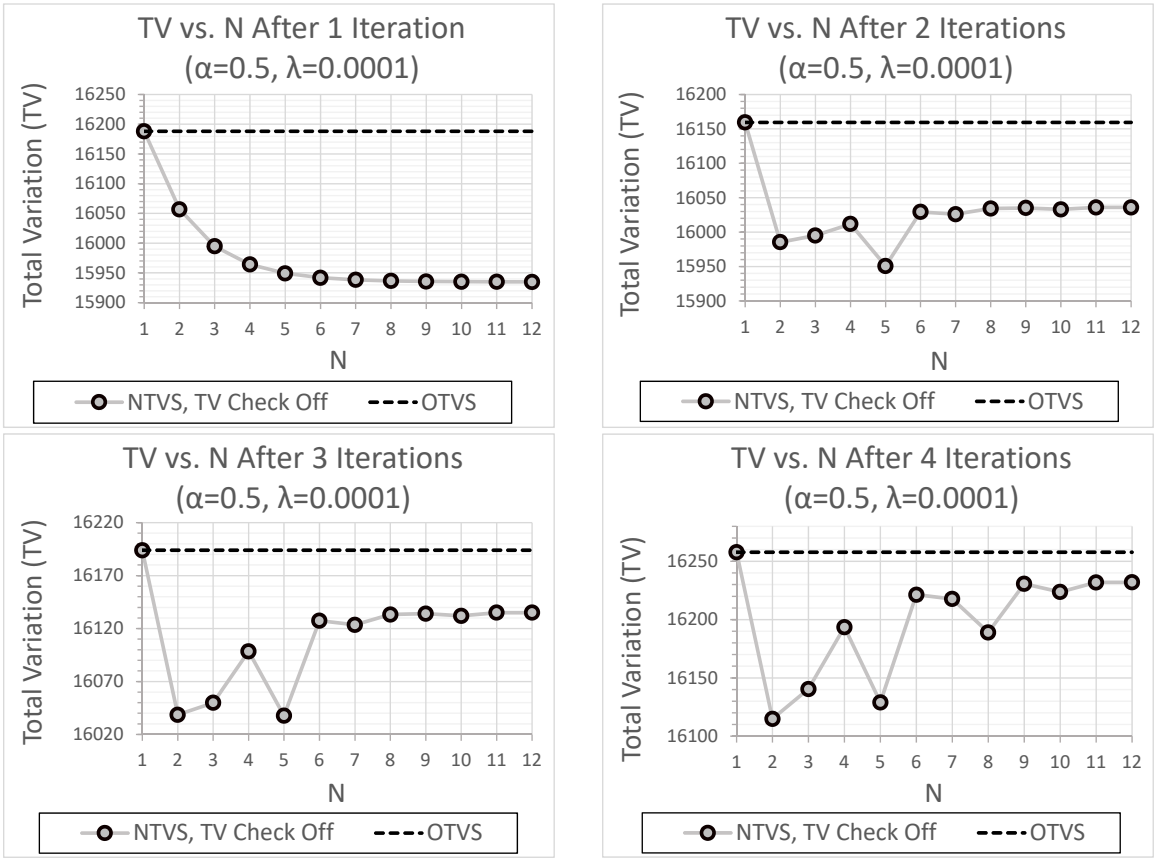


Figure 6.15: TV as a function of N after each of the first 4 FS-DROP iterations for the experimental HN715 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$ and $\alpha = 0.5$. © 2020 IEEE

As previously noted, repeated reconstructions with the same value of N yield slight variations in TV and standard deviation. Again, however, the difference in TV and standard deviation as a function of N is seen to be a property of the algorithm and its relationship with FS-DROP, not the result of the variations arising from random decreases in $\ell^{(k)}$. The objectives of FS-DROP and TVS are somewhat opposed; FS-DROP tends to amplify noise, thereby increasing TV, while each TVS perturbation may drive the solution to a more or less feasible solution. The resulting push back and forth begins to produce small differences in TV between successive values of N after the first two FS-DROP iterations and these subsequently increase as each additional FS-DROP iteration amplifies the resulting differences. Simultaneously, TV perturbations and updates applied by FS-DROP both decrease in magnitude as k increases, diminishing their ability to counteract the impact of a

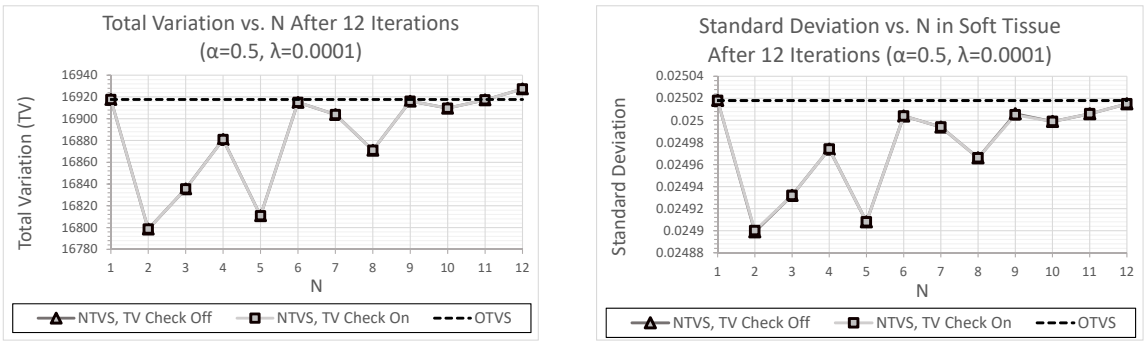
previous, less optimal solution. Hence, a solution that is less optimal after the first few iterations will rarely overcome its performance deficit and will far more often become increasingly suboptimal, particularly if parameter values are held fixed and not adapted based on performance as in the present case. Hence, values of N that yield a larger reduction in TV early in reconstruction also experience a lesser amplification of noise at each FS-DROP iteration, resulting in a compounding effect that accounts for the relatively large differences in TV between similar values of N.

As can be seen in Figure 6.16, showing the TV and standard deviation within the soft tissue ROI as a function of N after all 12 FS-DROP iterations for $\lambda = 0.0001$ and $\alpha = 0.5$, NTVS including and excluding the TVRVS both yield larger reductions in TV and standard deviation for every value of N except for the slight increase in TV obtained with $N = 12$. Repeating these reconstructions with $\alpha = 0.75$ consistently yields images with significantly larger reductions in both TV and standard deviation for every value of N, with similar standard deviation results obtained for every material ROI. These results also demonstrate that the smallest reductions in TV and standard deviation obtained with $N = 1$ and $N = 12$ were approximately 50% larger than the largest difference between varying values of N and more than twice as large for $3 \leq N \leq 6$.

6.5.3.2 Inclusion/Exclusion of TVRVS.

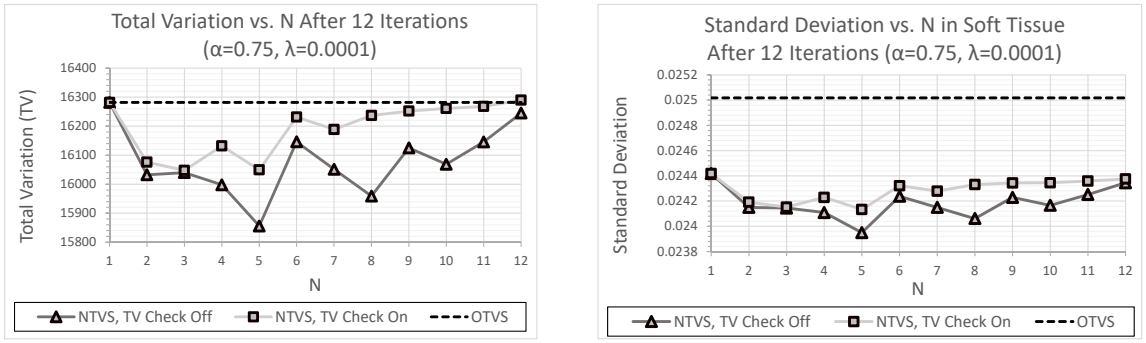
Selected results of an investigation of the exclusion of the TVRVS in NTVS are shown for $\lambda = 0.0001$ in Figures 6.16(a) and (b) for $\alpha = 0.5$ and Figures 6.17(a) and (b) for $\alpha = 0.75$, demonstrating a comparison of performance in terms of TV and standard deviation (soft tissue), respectively, for inclusion and exclusion of the TVRVS as a function of N.

For $\alpha = 0.5$, the plot lines for inclusion and exclusion of the TVRVS are not distinguishable and completely overlap, which was also observed for each of the other values of λ. As for standard deviation, the same overlap was observed for each material insert ROI. These results mirror those observed for the experimental CTP404 phantom.



(a) (b)

Figure 6.16: (a) TV and (b) standard deviation (soft tissue) as a function of N after 12 FS-DROP iterations for the experimental HN715 data set using the OTVS algorithm and the NTVS algorithm including and excluding the TVRVS with $\lambda = 0.0001$ and $\alpha = 0.5$ (note that the 2 NTVS curves overlap). © 2020 IEEE



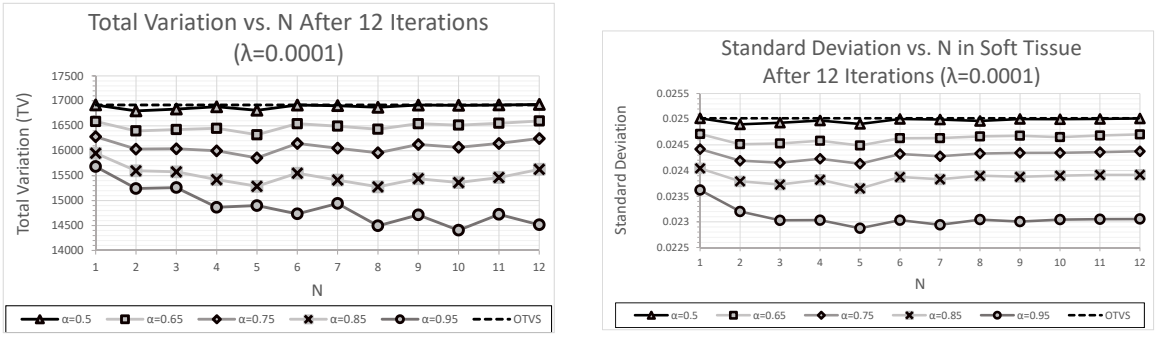
(a) (b)

Figure 6.17: (a) TV and (b) standard deviation (soft tissue) as a function of N after 12 FS-DROP iterations for the experimental HN715 data set using the OTVS algorithm and the NTVS algorithm including and excluding the TVRVS with $\lambda = 0.0001$ and $\alpha = 0.75$. © 2020 IEEE

On the other hand, for $\alpha = 0.75$, excluding the TVRVS consistently yielded lower TV and standard deviation (LDPE, for each value of λ and for each material insert ROI, but larger values of N yielded results increasingly closer to, and exceeding slightly beyond in the case of TV with $N = 12$, those of OTVS. This is in sharp contrast to the experimental CTP404 results, which showed marked improvement for all values of N. As with the previous data sets, all subsequent analyses for this data set were performed using the NTVS algorithm with the TVRVS excluded (i.e. as defined in Appendix D.2).

6.5.3.3 *Perturbation Kernel (α).*

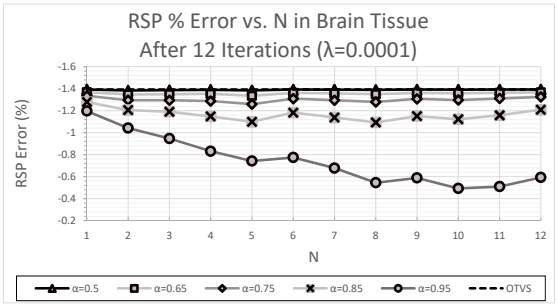
Selected results of an investigation of the perturbation kernel α of NTVS (TVRVS excluded) are shown for $\lambda = 0.0001$, demonstrating a comparison of performance in terms of TV (Figure 6.18(a)), standard deviation (soft tissue,Figure 6.18(b)), and RSP error (brain tissue, Figure 6.19(a) and soft tissue, Figure 6.19(b)) as a function of N. Figures 6.12(a) and (b) demonstrate performance that coincides with expectations: increasing α, which governs perturbation step-sizes and decay) yields increasing reductions in TV and standard deviation. These results mirror those obtained with the simulated and experimental CTP404 data sets, indicating the same trend in TV and standard deviation (soft tissue) as a function of α, but the standard deviation was less sensitive to N than those observed with the CTP404 data sets.



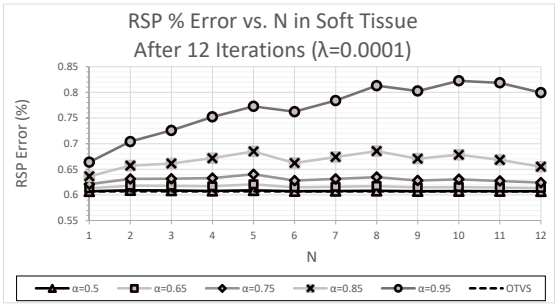
(a) (b)

Figure 6.18: (a) TV and (b) standard deviation (soft tissue) as a function of N after 12 FS-DROP iterations for the experimental HN715 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$ and different values of α. © 2020 IEEE

Similarly, RSP error as a function of α exhibits the same trends as those observed with the simulated and experimental CTP404 phantoms: for $\alpha > 0.75$, RSP error was seen to increase for every material ROI. The direction that RSP was driven was more similar to those with the simulated CTP404 data, varying between increasing and decreasing RSP for different material ROI, thereby improving RSP error in some material ROIs and worsening RSP error in others.



(b)

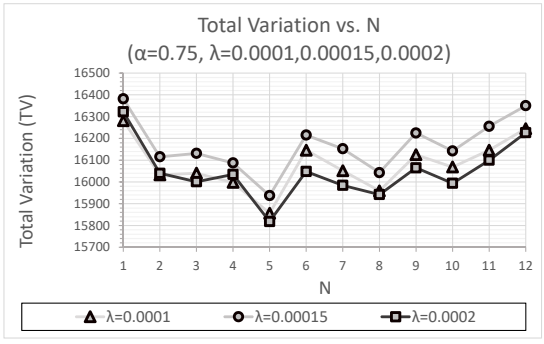


(b)

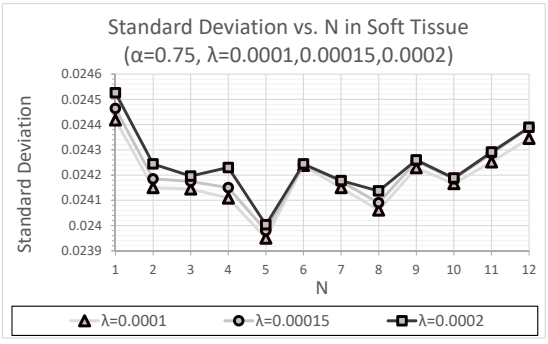
Figure 6.19: RSP error in the (a) brain tissue and (b) soft tissue ROIs as a function of N after 12 FS-DROP iterations for the experimental HN715 data set using OTVS and NTVS (TVRVS excluded) with $\lambda = 0.0001$ and different values of α. © 2020 IEEE

6.5.3.4 Relaxation Parameter (λ).

Selected results of an investigation of the relaxation parameter λ are shown for $\alpha = 0.75$ and the TVRVS excluded in Figures 6.20(a) and (b), demonstrating a comparison of performance in terms of TV and standard deviation (soft tissue), respectively, as a function of N. The RSP accuracy for $\lambda = 0.0001$ after $k = 12$ iterations was comparably obtained with $k = 8$ iterations for $\lambda = 0.00015$ and $k = 6$ iterations for $\lambda = 0.0002$. Plots of standard deviation for the ROI of other materials displayed the same dependence on N and λ.



(a)



(b)

Figure 6.20: (a) TV and (b) standard deviation (soft tissue) as a function of N for $\lambda = 0.0001$, $k = 12$; $\lambda = 0.00015$, $k = 8$; and $\lambda = 0.0002$, $k = 6$ iterations, respectively, and for $\alpha = 0.75$ for the experimental HN715 data set. © 2020 IEEE

These results mirror those observed with the CTP404 data sets: increasing λ consistently yielded larger TV reductions for each value of N. The standard deviation (soft

tissue) exhibited similar behavior as those observed with the experimental CTP404 data set: a slight increase in standard deviation as a function of λ for each value of N. However, the TV and standard deviation (soft tissue) results obtained for $3 \leq N \leq 6$ are consistently and considerably better for each value of λ than those obtained with NTVS for $N = 1$ and with OTVS for the less noise sensitive $\lambda = 0.0001$ (see Figure 6.17(a) and (b)).

Permissibility of larger λ with NTVS can be observed by comparing the performance in terms of standard deviation (trabecular bone) for $\lambda = 0.0002$ with NTVS and OTVS for $\lambda = 0.0001$ and $\lambda = 0.0002$, which is demonstrated in Figure 6.21. For OTVS, the performance for $\lambda = 0.0001$ is noticeably better than for $\lambda = 0.0002$ for the same (comparable) RSP accuracy. This coincides with previously observed behavior, which has been used to justify using $\lambda = 0.0001$ for FS-DROP. On the other hand, the performance of NTVS for $\lambda = 0.0002$ and $\alpha = 0.75$ demonstrates considerably lower standard deviations (trabecular bone) for each value of N. Combining these observations with the fact that larger TV reductions are obtained with $\lambda = 0.0002$, it is reasonable to posit that the previously inappropriately large $\lambda = 0.0002$ is now an appropriate option when the NTVS algorithm is employed to replace OTVS.

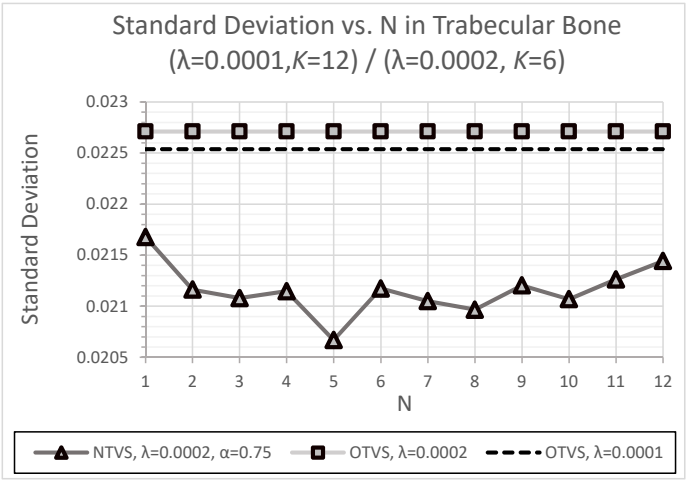


Figure 6.21: Standard deviation in the ROI of trabecular bone as a function of N after 6 FS-DROP iterations for $\lambda = 0.0002$ as compared to OTVS with $\lambda = 0.0001$ and $\lambda = 0.0002$. © 2020 IEEE

6.6 Discussion

The objective of these investigations has been to assess whether the recently developed features of TVS (as well as those existing but not included in OTVS), reflected in the NTVS algorithm, were beneficial to its use in pCT. Preliminary investigations were performed using simulated data of the cylindrical CTP404 phantom, followed by practical viability investigations using experimental data of the same CTP404 phantom as well as an anatomically realistic head phantom. Improvements in noise measures (TV and standard deviation) obtained with NTVS are small to moderately sized (less than 5% in most cases), but in some cases there was no observed improvement. However, such improvements are not meaningless and can have real world impact for treatment planning. Improvements are also expected to be proportionally larger in the presence of greater noise content, such as in the case of very low fluence or fluence-modulated pCT [85, 86]. Given the consistent improvements with NTVS, especially the potentially detrimental and unnecessary TVRVS constraint, and the absence of drawbacks, it can be concluded that the innovative features of NTVS should be incorporated into pCT image reconstruction.

For each of the three data sets, investigations of all combinations of values in the reconstruction parameter space ($2\{\text{TVRVS on/off}\} \cdot 5\{\alpha\} \cdot 3\{\lambda\} \cdot 12\{N\}$) required 360 individual reconstructions. A striking observation from these investigations is an alternating oscillation in TV and standard deviation as a function of N, which was observed with each data set. Furthermore, this behavior was observed to be independent of the other parameters, which consist of 30 combinations of parameter values for each value of N. The most likely cause was thought to be the random decrease in $\ell^{(k)}$, but this was eliminated from consideration after performing repeated reconstructions (8-10) with the same combination of parameters and observing that the resulting performance fluctuations were not nearly large enough to account for the observed behavior. A series of discussions with member

of the pCT and superiorization communities regarding the larger set of results and the potential theoretical explanations yielded no definitive explanation, but it did establish a basic supposition that the behavior is a result of the competing objectives of FS and TVS.

Although the performance oscillated as a function of N, the benefits of NTVS were observed to have a recognizable trend dependent on N. The largest benefits of NTVS were consistently obtained for $3 \leq N \leq 6$, with $N = 5$ generating the best results most often and consistently near best otherwise. On the other hand, the benefits of NTVS continually decreased as a function of FS iteration k for $N \geq 7$ except for $\alpha \geq 0.85$. However, $\alpha \geq 0.85$ demonstrated an unpredictable effect on reconstructed RSP, a wholly undesirable feature of superiorization that indicates perturbations beyond the resilience of the underlying FS algorithm. This is due to the effect that α has on the perturbation step-size $\beta^{(k)} = \alpha^{\ell^{(k)}}$. Larger α not only generates larger perturbations, but as $\alpha \to 1.0$, the size of the perturbation steps decays too slowly and produce perturbations that are inappropriately large when nearing convergence. The same effects can be seen with smaller α when N is relatively large (e.g. $N \geq 7$). In this case, since $\ell^{(k)}$ is incremented after each of the N perturbations, $\beta^{(k)} = \alpha^{\ell^{(k)}}$ can decrease too quickly and produce meaningful perturbations for only the first few FS iterations, resulting in an under utilization of TVS and an inability to reduce error propagation in all subsequent FS iterations. In either scenario, it can be inferred that the perturbation step-size $\beta^{(k)} = \alpha^{\ell^{(k)}}$ has a complicated relationship with α and N and, therefore, selecting a desired value for one necessitates consideration for selecting the value of the other. The investigations presented provide a cursory view of this relationship, but higher resolution parameter value investigations are needed.

When the TVRVS constraint is included, the perturbation kernel exponent $\ell^{(k)}$ is incremented after each TVRVS failure, resulting in successively smaller perturbations each time. Failure of the TVRVS constraint typically only occurs early in reconstruction (first few FS iterations) before convergence directions have stabilized for each voxel. Hence, by the time convergence behavior has settled, the $\ell^{(k)}$ increments result in smaller perturbation

step-sizes and an under utilization of TVS throughout all subsequent FS iterations. This effect becomes even more pronounced as N increases. Excluding the TVRVS avoids this early reduction in perturbation step-sizes, thereby maintaining meaningful perturbations throughout reconstruction and generating larger reductions in TV and standard deviation. Naturally, the increases in TV that this permits are an undesired result, but the persistence of larger perturbations for the remainder of reconstruction more than offset any such temporary increases in TV. For example, for $3 \leq N \leq 6$, excluding the TVRVS yielded at least comparable, and often superior, TV and standard deviation for all three data sets. Comparable results were typically associated with $\alpha = 0.5$, whereas the results for $\alpha = 0.75$ were consistently superior and most clearly observed with the experimental data sets.

Excluding the TVRVS also allowed for a fully parallelized implementation of the NTVS algorithm, resulting in a more computationally efficient TVS due to the elimination of the TVRVS conditional branch and the successive perturbations it performs until TVRVS is satisfied. The data dependencies within TVS (e.g. discrete ℓ_2 norm) that act like a computational bottleneck were eliminated via DDP schemes, and SGP schemes were employed in the integration of TVS with FS-DROP. The overall reduction in sequential compute operations due to the full parallelization resulted in an approximately 30% reduction in computation time.

One of the most influential features included in NTVS is user control of the perturbation kernel α, which in OTVS had been a fixed hardcoded value equivalent to $\alpha = 0.5$. Larger reductions in TV and standard deviation are observed as α increases, but $\alpha > 0.75$ was shown to affect reconstructed RSP is an unpredictable direction depending on the material ROI. In some cases, the effect on the RSP of an ROI was in opposite directions for the simulated and experimental CTP404 data sets. These results indicate that the direction that RSP is driven by inappropriately larger perturbations is not simply a property of the phantom or its material composition, nor is it a product of any single parameter or parameter combination. In accordance with the observed behaviors with respect to α and

its relationship with other parameters, a practical suggestion for safely reliable and beneficial behavior is $\alpha = 0.75$ since this yields nearly optimal reductions of TV and standard deviation without the potentially detrimental effects observed for larger α.

The most potentially rewarding benefit of NTVS is its larger permissible relaxation parameter λ values relative to OTVS. It was demonstrated that NTVS was able to achieve the same RSP error with $\lambda = 0.0002$ after $k = 6$ FS-DROP iterations without the accompanying larger standard deviations observed with OTVS. The larger standard deviation observed for larger λ using OTVS led to the use of $\lambda = 0.0001$ in previously published work involving simulated CTP404 data [87, 58]. Generating pCT images in fewer iterations has twofold benefit: fewer iterations requires less computation time and yields fewer error propagation steps, thereby yielding superior noise performance. Hence, the capability of using larger λ represents a potentially invaluable development in pCT.

These NTVS investigations represent a thorough assessment of TVS parameter impacts and interactions, as well as their relationship with relaxation parameter λ, and provide a good reference to guide future investigations. The most obvious directions for future investigations are (1) a finer resolution of parameter values within the ranges suggested or inferred here and (2) allowing parameter values to vary during reconstruction in different ways.

Other directions to explore involve the decoupling of the relationship between α and N. It is currently unclear whether the diminished benefits for $N \geq 7$ are solely a result of an over utilization of TVS or a consequence of $\beta^{(k)}$ decreasing too quickly (due to N increments of $\ell^{(k)}$ per FS step), causing an under utilization of TVS.

A more flexible implementation would provide for the independent control of α and the decay rate of $\beta^{(k)}$ as a function of k. These two features are currently inseparable since $\beta^{(k)} = \alpha^{\ell^{(k)}}$ and $\ell^{(k)}$ implicitly depends on r and k. One possible solution is the introduction of another parameter $\gamma^{f(k)}$, where $f(k)$ is some function of k with desired

properties, that controls the $\beta^{(k)}$ rate of decay as a function of k. With this additional parameter, α would then be responsible for controlling only the $\beta^{(k)}$ rate of decay as a function of $r = 1, 2, \ldots, N$. This yields the equation $\beta^{(k,r)} = \alpha^r \gamma^{f(k)}$ for the perturbation step-size. Note the elimination of the perturbation step counter $\ell^{(k)}$, as this is no longer needed to ensure decaying perturbations for the preservation of convergence. Note that to guarantee convergence is preserved, $0.0 < \gamma < 1.0$ and $f(k)$ must be chosen such that $\lim_{k\to\infty} f(k) = \infty$ (e.g., $f(k) = k$). With this new feature, the decay rate of $\beta^{(k,r)}$ can be independently controlled as a function of k and r without disturbing the superiorization perturbation constraint $\lim_{k\to\infty} \beta^{(k,r)} = 0$.

6.7 Conclusions

These investigations demonstrate the efficacy of the NTVS algorithm and the clear advantages its use offers, in terms of both image quality and computational efficiency, in comparison to the OTVS algorithm. Future investigations should include (1) an assessment of whether or not the TVS parameters can be varied as a function of iteration k in some way so as to yield superior performance and/or (2) whether the perturbation step-size rate of decay between FS iterations as a function of k and within FS iterations as a function of N can be controlled independently to increase the performance benefits of NTVS.

The investigations performed in this work demonstrate that the modifications implemented by the NTVS algorithm provide clear advantages over the OTVS algorithm in terms of both quality and computational cost. Future work should include investigating whether varying parameters during reconstruction or controlling the decrease of the perturbation magnitude independently during iterations and repeated perturbation steps can further increase the advantages of the NTVS algorithm.

CHAPTER SEVEN

Conclusion

Confidence in the feasibility of clinical pCT continues to rise as new advancements in detector technologies and image reconstruction methods are developed. The work presented in this dissertation represents several important contributions toward viable pCT image reconstruction software for clinical applications. The pCT image reconstruction software used to generate the pCT images in this dissertation was primarily developed at Baylor University over the course of several years. The contributions that were presented here were developed at the time when the associated pCT task was being implemented in software. Hence, although the same software was used for each contribution to the dissertation, other software and hardware improvements were being made throughout this same time period, so the software was more mature for each successive topic of investigation. However, this fact does not obscure the validity of any of the separate works since each of these were performed in isolation of external influence. Furthermore, the three steps of pCT image reconstruction that were selected for development do not influence each other and all comparison data/images were generated using the same software and configurations as the data/images under investigation. Given that the primary objective for developing the pCT software entirely from scratch and that computation time was the largest deficiency in the initial pCT software developed at Loma Linda University, computational concerns were addressed to some degree in every contribution.

Hull-detection, as presented here, has taken considerable strides since its initial conception and investigatory work. The improvements of the SC algorithm and the development of the MSC algorithm were proposed as two possible options to resolve the poor performance observed in early SC work. The SC and MSC algorithms both performed well in terms of successful identification of every target object voxel, but they have room

for improvement. The other algorithm proposed, SM, performed considerably worse than the other hull-detection algorithms. Whereas SC and MSC both represent possible foundations for a fully mature hull-detection algorithm, it is hard to conceive of a way to adjust or modify SM that would lead it to have comparable performance as SC and MSC. Although SC performed best of the investigate algorithms, MSC performed similarly and has several desired properties that SC does not: (1) it can generate hull approximations with small concavities; (2) its input data does not need to be binned or have unsuitable proton histories removed, permitting use in an online-mode pCT system scan; (3) can be performed using online-mode data streams and generate an approximate hull by the time the pCT scan is complete, well before when SC or any other preconditioning task is performed. Hence, MSC is the algorithm with the greatest potential for application, but it must be able to perform at least as well as SC before this can be realized.

An efficient implementation of the MLP formalism was described in this dissertation, accomplished by both mathematical simplification and careful software design considerations. Although the entirety of this work has yet to be published, it corresponds to the most mature of the topics described here. An abridged version of the text on this topic is currently being prepared as part of a publication on pCT image reconstruction, intended to be a companion to the recent *IEEE Access* publication "Particle-Tracking Proton Computed Tomography—Data Acquisition, Preprocessing, and Preconditioning". The computational improvements of the MLP implementation resulted in a more than 70% reduction in compute operations. Careful consideration of memory usage and the incorporation of MLP component lookup tables yield computation times (typically less than 1 minute) that are no longer an obstacle for fast pCT image reconstruction. Given this fact, it may be possible to revisit the possibility of developing an improved MLP formalism, either involving removing small-angle and other approximations or by extending the theory to account for the non-homogenous material compositions observed in practice. Such advancements of MLP were deemed infeasible given the computational burden observed for the MLP formalism

based on a homogenous material composition assumption. In fact, the reduction in computation time is so large that, it may be possible to revisit several other tasks involved in pCT and any such approximations they include be considered for removal.

The modern TVS algorithm was incorporated into the Baylor image reconstruction software and its performance investigated for every permutation of the values tested of several different parameters, yielding over 1000 individual image reconstructions. Given the combinatorics with several parameters, each parameter was investigated with a relatively small number of values. Hence, there weren't enough values tested of some parameters to determine a definitive optimal value (or range). However, these investigations were able to identify small regions of superior performance, so it was possible to present reasonable suggestions for the values of each parameter. Future development can then begin investigations of parameters with the suggested values and finer resolution searches of surrounding values.

APPENDICES

APPENDIX A

Glossary and Notation

The following is a description of the terminology and an outline of the notation that is relevant to the content of the dissertation.

A.1 Glossary

The following is a non-exhaustive list of terms used in the context of this dissertation, or pCT in general, and for which there is no (or an indequate) definition accompanying its use. Note that the glossary items whose specific context is unclear include a parenthetical term that helps to establish the context in which they are defined.

- wobble (beam): a beam line configuration where the beam spot is wobbled in one or more directions using a magnetic deflector.
- voxel: the elements of a three-dimensional image, equivalent to a pixel with a nonzero thickness. Etymologically, it is the combination of *volumetric* and *pixel*.
- WEPL, water-equivalent path length: the unit of measure of energy loss appearing in the measurement vector $\vec{b}$. A WEPL value is assigned to each proton history according to the length of water that the proton would have had to traverse in order for the observed energy loss to occur.
- RSP, relative stopping power: a unitless measure of a material's impact, relative to water, on WEPL; i.e. the WEPL per unit length of the material. Image reconstruction yields this measure for each voxel. For example, for a proton history with WEPL = 1.5 mm that traversed a 1 mm thick material, the RSP = $^{\text{WEPL}}/_{\Delta\ell} = {}^{1.5\text{ mm}}/_{1\text{ mm}} = 1.5$
- proton history: A triggered proton event with registered tracking data ((t,u,v) for all 4 tracker planes) and associated WEPL value (converted from energy loss, calculated as the difference between scanner energy and measured output energy).

- hyperplane: a single isolated row of the linear system $\boldsymbol{A}\vec{x} = \vec{b}$.
- projection methods (iterative image reconstruction): methods based on projections of the image vector $\vec{x}^{(k)}$ onto the m hyperplanes of the linear system $\boldsymbol{A}\vec{x} = \vec{b}$.
- feasibility-seeking: an approach in which a feasible solution, rather than an exact solution, is sought. In the case of inconsistent data, an exact solution does not exit, so a feasible solution is found instead. In other cases, an exact solution may exist, but due to some additional constraint (e.g. computation time), a feasible solution may be the preferred approach.
- fully simultaneous (feasibility-seeking): a feasibility-seeking algorithm performing (typically orthogonal) projections onto all m hyperplanes simultaneously (e.g. Cimmino or FS-DROP algorithms)
- partial iteration (feasibility-seeking): completion of simultaneous (typically orthogonal) projections onto all hyperplanes within a block (block-iterative) or completion of projections onto a single hyperplane within a string.
- full iteration (feasibility-seeking): completion of projection operations onto each of the m hyperplanes (i.e. proton histories), either all at once (fully simultaneous) or in a sequence of partial iterations composed of partitions of hyperplanes (blocks-iterative or string-averaging).
- block-iterative (algorithm): a feasibility-seeking algorithm that partitions the m hyperplanes into groups, called blocks, and performs a sequence of partial iterations; each partial iteration is composed of simultaneous orthogonal projections onto each hyperplane within the corresponding block and updates the solution, with a full iteration complete once partial iterations have been performed for each block.
- string-averaging (algorithm): a feasibility-seeking algorithm that partitions the m hyperplanes into groups, called strings, and each string performs an independent sequence of partial iterations; each partial iteration corresponds to a (typically orthogonal) projection onto a single hyperplane, with a full iteration complete once

each string has performed partial iterations for each hyperplane in the corresponding string.

- index set: a mathematical set is a set whose members label (or index) members of another set. For the purposes of this dissertation, an index set dictates the order in which the elements of an unordered source set (w/ N elements) are to be processed, i.e. defining how the source set should be reordered. The index set is an ordered set composed of the integers $(1..N)$, where the position of each integer i indicates the desired position of the ith element of the source set within the reordered source set. For example, index set $(3, 2, 1)$ indicates the reversal of the source set.
- helper function: a subfunction that performs a portion, typically a particular task, of the parent function.
- kernel (GPU): a function that is executed (on a GPU) as a set, or array, of computationally independent threads performing identical code instructions.
- (proton) pile-up: refers to the scenario where the time between consecutive protons entering the calorimeter is smaller than the time it takes for the calorimeter measurement signal to decay back down to baseline, i.e. there is a pile-up of residual proton energies in the detector signal. This scenario results in the residual energy signal from the earlier proton being added to that of the next proton, which yields and unsuitable proton history.
- sequential gap parallelization: an approach where a sequential section of code preventing full parallelization is executed on a single thread rather than transferred to/from the host CPU, thereby reaching full parallelization at a minor GPU efficiency cost.
- dependency distribution parallelization: an approach where inter-thread data dependencies are eliminated by generating all the necessary data on each thread, thereby reaching full parallelization at a minor cost in thread-local memory and computation time.

- m : total # of proton histories
- n : total # of image vector voxels
- k : full iteration # of iterative projection (feasibility-seeking) algorithms
- k_p : partial iteration # within the kth full iteration of iterative projection (feasibility-seeking) algorithms
- K : # of full iterations that an iterative projection (feasibility-seeking) algorithm is to perform
- $\lambda^{(k)}$: relaxation parameter at the kth iteration
- $\boldsymbol{D} = \text{diag}_{1\leq j\leq n}\left(\frac{1}{d_j}\right)$: the diagonal matrix where d_j is the # of nonzero elements of the $j$thth column of $\boldsymbol{A}$, i.e., the # of times the jth voxel was intersected by a proton path
- $P^{(k)}$: total # of BIP blocks at iteration k
- $S^{(k)}$: total # of SAP strings at iteration k
- $\boldsymbol{A}_{*,j}$: Column vector composed of all $* = 1\!:\!m$ rows of the jth column of the matrix A
- $\boldsymbol{A}_{i,*}$: Row vector composed of all $* = 1\!:\!n$ columns of the ith row of the matrix A
- $\vec{x}^{(k)}$: Image vector $\vec{x}$ at iteration k
- $\mathcal{P} = \left\{\mathcal{P}_p \mid 1 \leq p \leq P\right\}$: is the ordered set of blocks/strings
- $\mathcal{B} = \left\{\mathcal{B}_p \mid 1 \leq p \leq P\right\}$: is the unordered set of blocks
- $\mathcal{S} = \left\{\mathcal{S}_p \mid 1 \leq p \leq P\right\}$: is the unordered set of strings
- $\mathcal{I} = \{1, 2, 3, \cdots, m\}$: the (sequentially ordered) index set of all proton history indices
- $f_{(k)} : \mathcal{I} \rightarrow \mathcal{B}_{(k)} = \{\, f_{(k)}(1), f_{(k)}(2), f_{(k)}(3), \cdots, f_{(k)}(m) \,\big|\, f_{(k)}(i) = b \in \mathcal{B}_{(k)}, i \in \mathcal{I}\}$: surjective function $f_{(k)}$ mapping each of the m proton histories to one of the $B_{(k)}$ BIP blocks, which may vary as a function of k, thereby establishing the # of histories in each block and the order they are processed.
- $\mathcal{J}_{(k)} = \{\, 1, 2, 3, \cdots, B_{(k)}\}$: the (sequentially ordered) index set of all BIP block indices, where the # of BIP blocks $B_{(k)}$ may vary as a function of iteration k.

- $f_{(k)} : \mathcal{I} \rightarrow \mathcal{J}_{(k)} = \{ f_{(k)}(1), f_{(k)}(2), f_{(k)}(3), \cdots, f_{(k)}(m) \mid f_{(k)}(i) = j \in \mathcal{J}_{(k)}, i \in \mathcal{I} \}$: surjective function $f_{(k)}$ mapping each of the m proton histories to one of the $B_{(k)}$ BIP blocks, which may vary as a function of k, thereby establishing the # of histories in each block and the order they are processed.
- $\mathcal{M}_{j(k)} = \{ i \in \mathcal{I} \mid f_{(k)}(i) = j \in \mathcal{J}_{(k)} \}$: the ordered set of proton history indices assigned to the j-th BIP block by the function $f_{(k)}$
- $\mathcal{M}_{(k)} = \bigcup_{j \in \mathcal{J}_{(k)}} \mathcal{M}_{j(k)} = \{ \mathcal{M}_{1(k)}, \mathcal{M}_{2(k)}, \cdots, \mathcal{M}_{B(k)} \}$: the family of sets (BIP blocks) of proton history indices for iteration k, assigned according to the function $f_{(k)}$
- $g_{(k)} : \mathcal{J}_{(k)} \rightarrow \mathcal{J}_{(k)} = \{ g_{(k)}(1), g_{(k)}(2), g_{(k)}(3), \cdots, g_{(k)}(B_{(k)}) \mid g_{(k)}(i) = j \wedge i, j \in \mathcal{J}_{(k)} \}$: bijective function $g_{(k)}$ imposing an order on the sets (BIP blocks) of proton histories to define the sequence that the BIP blocks are procecessed at iteration k.
- $\mathcal{B}_{(k)} = \left\{ \left(\mathcal{M}_{j(k)} \right)_{j \in \mathcal{J}_{(k)}} \mid \mathcal{M}_{j(k)} < \mathcal{M}_{i(k)} \Leftrightarrow g(j) < g(i) \right\}$: the ordered family of sets (BIP blocks) of proton histories generated by the bijective mapping function $g_{(k)}$.
- $H_i = \{ x \in \mathbb{R}^n \mid \langle A_{i,*}, x \rangle = b_i, i \in \mathcal{I} \}$: the hyperplane corresponding to the ith row of the $m \times n$ linear system $\boldsymbol{A}\vec{x} = \vec{b}$ upon which the image vector $x^{(k)}$ is projected
- $\mathcal{H} = \{ H_i \mid i \in \mathcal{I} \}$: the set of hyperplanes corresponding to the $m \times n$ linear system $\boldsymbol{A}\vec{x} = \vec{b}$ upon which the image vector $x^{(k)}$ is iteratively projected

APPENDIX B

Voxel Walk Algorithm

The history of the design and implementation of the voxel walk algorithm, initially developed for use in hull-detection, is presented in the following sections.

B.1 Three-Dimensional Voxel Walk Algorithm

In order to carve voxels out of the object along the path of a proton that missed the object, the voxels intersected by the path must be accurately identified. Therefore, hull-detection had to include a procedure for sequentially stepping along the path from one voxel intersection to the next, herein referred to as a *voxel walk algorithm*, in order to accomplish this task. A similar task is common in the context of computer graphics, where a line $\mathcal{L}$ passes through a discretized 2D rectangular grid (typically an image $\mathcal{I}$) with an orientation defined as the angle θ between $\mathcal{L}$ and a line normal to either the image's $x-$ or $y-$axis. The geometry for the case where θ is defined with respect to the line normal to the $x-$axis is illustrated in Figure B.1. The coordinates of the points where the line $\mathcal{L}$ intersects a particular pixel $\mathcal{P}_n$, often called (pixel) intersection points, and the length λ of the chord connecting these points are typically calculated as well.

A common application of this is determining which pixels to fill in an image, or partially fill in the case of anti-aliasing, to render an approximation of a straight line. A number of effective and efficient algorithms exist for performing such tasks with 2D images, such as the digital differential analyzer (DDA) and Bresenham's [47] and Xiaolin Wu's [48, 49] line algorithms, but these do not extend well to 3D images. In fact, there is no obvious extension of these algorithms to three or more dimensions because of their reliance on the binary nature of pixel intersections: if the next pixel intersection is not in the x-direction, then it must be in the y-direction. In addition, these algorithms are specifically

designed for computer graphics, which has an entirely different objective than that of its intended use in hull-detection.

There is another type of computer graphics algorithms, called ray marching, that performs essentially the same task as the one sought for hull-detection. A similar type of algorithm is called ray tracing (or ray casting). They are both methods for determining the intersection of a ray (i.e. SLP) with an object, but the former performs an iterative search along the ray for an intersection and the latter seeks to do solve for the intersection algebraically. Each of these methods is conceptually sufficient for the purposes of hull-detection, but detailed descriptions of efficient implementations of these algorithms are difficult to obtain. This is partly owing to the fact that the field of computer graphics is highly competitive and efficient algorithms are developed for proprietary, rather than public, use.

A complicating factor is the variation in terminology used to refer to these algorithms and the lack of literature that can be used as a resource for identifying the algorithm name variations. In fact, the terms ray marching and, subsequently, ray tracing were only discovered by happenstance in a short paper (more like an advertisement) touting DreamWorks graphical tools, though it provided no implementation details. For the examples of ray marching and ray tracing I could find freely available, their methods were quite crude and presented at a rudimentary conceptual level. Nearly all of the ray marching algorithms I found performed iterative steps of constant length along the ray and included a concept of "close enough" in reference to identifying intersected voxels. In the case of ray tracing, it was only presented in the context of rendering scene illumination. In both cases, there was little to no mathematical prescriptions nor even a passing reference to implementation design considerations. None of the resources I found on these two algorithms were adequately detailed for them to be implemented, nor did discussions include consideration for the parallelization and numerical stability of the algorithms that would be expected in a computer graphics context. Hence, the search for resources on these algorithms was halted.

The task of determining the voxels intersected by a line (SLP) and the coordinates of the corresponding intersection points is a relatively simple concept. It was inconceivable that there wouldn't be an existing 3D algorithm for such a basic procedure. Although there is no doubt that these algorithms exist, the lack of fruitful searches for these led to the independent development of a voxel walk algorithm that can efficiently accomplish the desired task. This algorithm is primarily based on an implementation of the DDA method, though it incorporates concepts from Bresenham's line algorithm, and is herein referred to as the 3D-DDA method. Unlike the ray marching and ray tracing methods, the DDA method has a plethora of resources, though it has the same lack of resources with formal definitions, descriptions, and potential implementations.

B.1.1 Foundational Algorithm

Despite the challenges, a logical framework for the development of the 3D-DDA was established based on Figure B.1.

This figure was created as a visual representation of the version of the DDA algorithm that was found having a form that is most easily extended to 3D. Whether this truly is an implementation of the DDA algorithm or is some combination of algorithms is inconsequential to the present discussion. Initially the plan was to present the 3D-DDA algorithm at a conference so that it would become available to others in the conference record, but this never came to pass. The intent of presenting it here is merely to have it recorded in a formal setting so others can find it and use it without having to develop their own algorithm from scratch. Hence, for the purpose of the following discussions, it will be assumed that Figure B.1 indeed represents some implementation of the DDA algorithm. The notation used in the figure is defined as:

- $\mathcal{I}$ = represents the image
- $\mathcal{L}$ = the line passing through image $\mathcal{I}$
- θ = the angle of line $\mathcal{L}$ with respect to the image $\mathcal{I}$

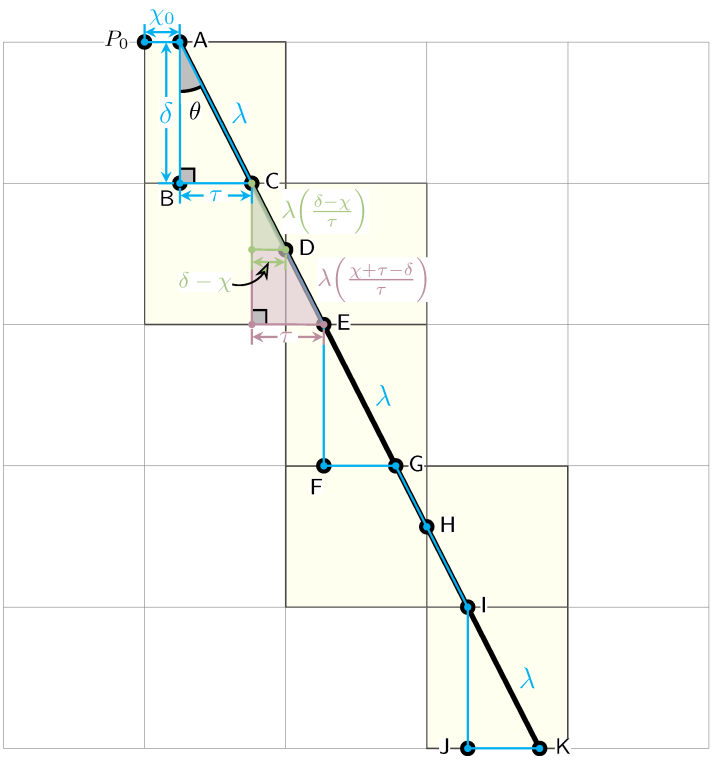


Figure B.1: Diagram showing a line passing through an image annotated with the intersection points/lenths and intersected voxels which a line algorithm must determine.

- δ = the maximum vertical distance Δy the line $\mathcal{L}$ can travel within a pixel before intersecting another pixel.
- τ = the maximum horizontal distance Δx the line $\mathcal{L}$ can travel within a pixel before intersecting another pixel.
- P_0 = the coordinates (x, y) corresponding to the leftmost edge of the first pixel intersected by line $\mathcal{L}$.
- A = the coordinates (x_0, y_0) corresponding to the point where the line $\mathcal{L}$ first intersects the image $\mathcal{I}$.
- $B, C, D, \ldots$ = the coordinates $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots$ corresponding to the first, second, third, etc. pixels intersected after successive steps along $\mathcal{L}$.
- p = the function that maps the tuple (x_0, y_0, θ_0) onto the corresponding pixel $\mathcal{P}$.
- $\mathcal{P}_n = \left(\mathcal{P}_{n,x}, \mathcal{P}_{n,y}\right)$ the pixel
- $\ell\left(\mathcal{P}_n\right)$ = the length of the chord connecting the points where $\mathcal{L}$ intersected pixel $\mathcal{P}_n$.

The following is a pseudocode definition, in terms of the preceding definitions, of a common implementation of the DDA algorithm illustrated in Figure B.1:

Algorithm 4 Standard DDA

procedure DDA($\mathcal{I}, \delta, \mathcal{L}, \theta$)
  $\tau := \delta \tan\theta$
  $\lambda := \delta / \cos\theta = \delta \sec\theta$
  $A = (x_0, y_0) = \overrightarrow{\mathcal{L}} \cap \mathcal{I}$
  $\mathcal{P}_0 = p(x_0, y_0, \theta)$ ▷ $p : (x, y, \theta) \rightarrow$ pixel $\mathcal{P}$
  $\chi_0 = A - P_0 = x_0 \bmod \delta$
  while $(x_n, y_n) \in \mathcal{I}$ **do**
    if $\chi_n + \tau < \delta$ **then**
      $\ell(\mathcal{P}_n) = \lambda$ ▷ chord length $\overrightarrow{\mathcal{L}} \cap \mathcal{P}_n$
      $\chi_{n+1} = \chi_n + \tau$
      $\mathcal{P}_{n+1} = \left(\mathcal{P}_{n,x}, \mathcal{P}_{n,y} + 1\right)$ ▷ down 1 pixel
    else
      $\ell(\mathcal{P}_n) = \lambda\left(\frac{\delta - \chi_n}{\tau}\right)$
      $\mathcal{P}_{n+1} = \left(\mathcal{P}_{n,x} + 1, \mathcal{P}_{n,y}\right)$ ▷ right 1 pixel
      if $\chi_n + \tau = \delta$ **and** $\mathcal{P}_n \in \mathcal{I}$ **then**
        $n = n + 1$
        $l(\mathcal{P}_n) = \lambda\left(\frac{\chi_n + \tau - \delta}{\tau}\right)$
        $\chi_{n+1} = \chi_n + \tau - \delta$
        $\mathcal{P}_{n+1} = \left(\mathcal{P}_{n,x}, \mathcal{P}_{n,y} + 1\right)$ ▷ down 1 pixel
      end if
    end if
    $x_{n+1} = x_n + \tau$
    $y_{n+1} = y_n + \delta$
    $n = n + 1$
  end while
end procedure

Note that in the case illustrated in the figure, unit increments are taken in the y dimension since the slop of the line is skewed towards the $y-$axis, but the procedure with the roles of x and y reversed is mathematically equivalent. For convenience, let the direction in which unit increments are taken be referred to as the control dimension, which is y in the present case. The algorithm begins by calculating the change in x (y), τ, associated with moving one pixel ($w \times h = \delta \times \delta$) in the vertical (horizontal) direction. At each step, the value of τ is compared to the remaining distance in the x (y) direction before

the next pixel, $\Delta = \delta - \chi$, to determine if the next pixel is in the horizontal or vertical direction. When the remaining distance $\Delta > \tau$, the next pixel is in the vertical direction and the length of the chord connecting the current and next intersection points is the constant $\ell = \delta / \cos\theta = \delta \sec\theta$. On the other hand, when the remaining distance $\Delta \geq \tau$, the next pixel intersected is in the horizontal direction. The case where $\Delta = \tau$, i.e. the line passes through the corner of a pixel, is handled as a subcondition of $\Delta \geq \tau$. Since $\Delta = \tau$ is a relatively rare case, it makes more sense to handle it as a subcondition of the `else` clause, rather than implementing the three possible relations as an `if`/`else if`/`else` conditional.

B.1.2 3D-DDA (Voxel Walk) Algorithm

B.1.2.1 Design Considerations and Decisions.

The binary decision of $x-$ or $y-$direction pixel steps is reflected in the test condition of the conditional `if` branch: $\chi_n + \tau < \delta$. In an extension of the notation to 3D, this test condition could be represented as $\chi_{n,x} + \tau_x < \delta_x$, where $\delta_x = \delta_y$ are still equal but referenced independently. For demonstration purposes, assume (1) that $\delta_z = \delta_x = \delta_y = \delta$ and (2) that the $y-$direction still has the greatest slope. Then there would be $\chi_{n,x} + \tau_x < \delta$ and $\chi_{n,z} + \tau_z < \delta$ test conditions, as well as comparisons of $\chi_{n,x} + \tau_x$ and $\chi_{n,z} + \tau_z$ in the case where both x and z satisfy their associated test condition. This can be formulated as a ternary logic decision (e.g. `if-else if-else` conditional) with test conditions having carefully chosen binary relations. However, situations where $\mathcal{L}$ intersects the corner of a voxel do not immediately fit into the ternary branching scheme without modifications and a moderate increase in complexity. Further complicating matters is the fact that in pCT, $\delta_x = \delta_y < \delta_z$, making it impossible to perform direct comparisons of these test conditions. Therefore, this adaptation of notation and test conditions for a 3D implementation is not well suited for the task.

A more suitable approach involves the rewriting of tests conditions as, e.g., $\rho = \delta - \chi_n > \tau$. This seemingly innocuous change actually represents a foundational shift

in perspective; rather than maintaining a value χ_n for the current depth within the nth voxel, the distance remaining before reaching the next voxel, ρ, is maintained. In pCT, the $z-$dimension of the image space, representing axial slices of the tomographic image space, has a relatively large thickness compared to the (square) $xy-$dimensions of each slice. Hence, the 3D-DDA uses the $z-$dimension in the role that the $y-$dimension has in Figure B.1 and the corresponding DDA algorithm (Algorithm 4. The reason the control dimension is not simply chosen as the direction with the largest slope, which is slightly more efficient, is that this would require separate routines for each direction and conditional branches for selecting the appropriate routine to execute. Note that the stream multiprocessors (SMs) of a GPU can only be utilized simultaneously if they are executing the same exact sequence of operations. In the case of conditional `if-else` branches, only the threads for which the condition is evaluated TRUE are executed simultaneously, followed by simultaneous execution of all the threads for which the condition is FALSE after all TRUE thread operations are complete. Thus, branching on a GPU should be avoided whenever possible, particularly if the number of operations within each branch is more than just a handful of basic access/write operations. Any efficiency loss due to atypical cases where z is not the largest slope are more than made up for by the more efficient and reliable computations.

B.1.2.2 Implementation Details.

The 3D-DDA (voxel walk) algorithm iteratively steps from one intersected voxel to the next along a line $\mathcal{L}$. In the case of pCT, the line $\mathcal{L}$ does not explicitly exist, it is merely an implied structure based on the position and angle of a proton as it enters and exits the reconstruction volume. Note that for the purposes of pCT, the chord lengths for each intersected voxel are not calculated, but they easily could be if the need arose. The axial direction z remains the control dimension regardless of whether or not the slope if largest in this direction. The test condition data relates to a new parameter ρ_X representing the remaining distance before the next voxel in the $X-$direction.

The notation used in the following description and implementation of the 3D-DDA (voxel walk) algorithm is given by:

- $\Delta_{YX} = {}^{\Delta Y}/_{\Delta X} = \frac{Y_{\text{exit}} - Y_{\text{entry}}}{X_{\text{exit}} - X_{\text{entry}}}$: the slope in Y relative to X, where $X, Y \in \{ x, y, z\}$
- $\Delta = \left\{ \Delta_{xy}, \Delta_{yx}, \Delta_{xz}, \Delta_{zx}, \Delta_{yz}, \Delta_{zy} \right\}$: set of the slopes associated with each possible pairing of (x, y, z)
- $X_{\pm} = \pm 1$: an integer variable with value +1 representing a proton heading in the direction of increasing X (i.e. $(+X)-$direction) and -1 representing a proton heading in the direction of decreasing X. (i.e. $(-X)-$direction).
- $X_{+} \pm 1$: indicates whether voxel numbers increase in the positive (+1) or negative (-1) $X-$direction.
- $\rho = \left\{ \rho_x, \rho_y, \rho_z \right\}$: the set composed of distances remaining until the next voxel edge in each direction.
- $\mathcal{V} = \left\{ \mathcal{V}_x, \mathcal{V}_y, \mathcal{V}_z \right\}$: the current voxel, represented by a tuple of voxel indices associated with each direction.
- $X_{\mathcal{V}_0}$: the $X-$coordinate on the edge of the voxel with index 0 with the point $p_0 = \{ x_0, y_0, z_0\}$ on the edge of the first intersected voxel.
- $\mathfrak{P}_X$: operator for projecting the current point $p = \{ x, y, z\}$ onto the edge of the next voxel in the $X-$direction for the purpose of determining the corresponding distance traveled in the $z-$direction.

The basic concept of the algorithm is the comparison of the distance ρ_z to the edge of the next voxel in the $z-$direction with the distances travelled in z by stepping to the edges of the next voxels in the x and y directions. In other words, the current point is projected onto the next x and y voxel edges according to the distances ρ_x and ρ_y, respectively, and multiplied by their associated slopes ${}^{dz}/_{dx}$ and ${}^{dz}/_{dy}$ to convert all three into measures of distances in z. Direct comparisons of the three directions can then be performed to determine which direction the next intersected voxel is in. This is a ternary decision implemented as an `if-else if-else` conditional branch, representing the entirety of the high-level structure of the voxel walk algorithm.

The remaining tasks are performed within the branches and amount to nearly the same operations, differing only in the roles that x, y, and z have in each branch. These tasks represent a basic logical scheme, but including their implementations directly within the branches makes it too cluttered to follow and understand the governing logic. Hence, there are three subroutines (i.e. helper functions) that encapsulate the basic tasks performed in each branch: project onto the next voxel edge in the direction associated with the branch (Algorithm 5, determine the corresponding coordinates of this edge point in the other two dimensions (Algorithm 6, then update the ρ variables with the new distance to edge values (Algorithm 7. These subroutines are defined, in this order and in terms of pseudocode, as follows:

Algorithm 5 Edge Projection Helper Function

function PROJ($\mathcal{V}_X, X_{\mathcal{V}_0}, X_+, X_\pm, \delta_X$)
    if $X_\pm = X_+$ **then**
        $E_X \leftarrow \mathcal{V}_X$
    else
        $E_X \leftarrow \mathcal{V}_X + 1$
    end if
    return $X_{\mathcal{V}_0} + \{ X_+ \cdot E_X \cdot \delta_X \}$
end function

Algorithm 6 Edge Component Helper Function

function EDGE(m, X, X_0, Y_0)
    return $m \cdot (X - X_0) + Y_0$
end function

Algorithm 7 Distance Remaining Helper Function

function δTOGO($\mathcal{V}_X, X_{\mathcal{V}_0}, X_0, X_+, X_\pm, \delta_X$)
    $\mathcal{V}_X \leftarrow \mathcal{V}_X + X_+ \cdot X_\pm$
    $E_{X+1} \leftarrow$ PROJ($X_{\mathcal{V}_0}, \mathcal{V}_X, \delta_X, X_+, X_\pm$)
    return $|E_{X+1} - X_0|$
end function

With the encapsulation of these helper function tasks, the logic of the 3D-DDA algorithm is considerably easier to recognize and understand. Algorithm 8 (shown on the next page) is a pseudocode definition of the 3D-DDA (voxel walk) algorithm as it is currently implemented in the BU pCT software. Note that this algorithm is itself designed as a subroutine, requiring initialization data to be provided as input arguments to the function. This, as well as the use of its own subroutines, is not purely for the sake of code readability. It is primarily a subdivision of independent tasks such that other steps in pCT reconstruction can make use of one or more of these subroutines without requiring other tasks to be executed as well. This will also make it easier to add support for chord length calculations since the requisite information is already present in a routine that called `3D-DDA`.

Algorithm 8 3D-DDA (Voxel Walk) Algorithm

Require: $p_0 = \{ x_0, y_0, z_0 \}$, $p_\pm = \{ x_\pm, y_\pm, z_\pm \}$, $\rho = \left\{ \rho_x, \rho_y, \rho_z \right\}$, $\mathcal{V} = \left\{ \mathcal{V}_x, \mathcal{V}_y, \mathcal{V}_z \right\}$

Require: $\Delta = \left\{ \Delta_{xy}, \Delta_{yx}, \Delta_{xz}, \Delta_{zx}, \Delta_{yz}, \Delta_{zy} \right\}$

procedure 3D-DDA($\mathcal{I}, \mathcal{V}, \Delta, \rho, p_0, p_\pm$)
  while STOP = `FALSE` **do**
    $\mathfrak{P}_x = \Delta_{zx} \cdot \rho_x$
    $\mathfrak{P}_y = \Delta_{zy} \cdot \rho_y$
    if $\rho_z \leq \mathfrak{P}_x$ and $\rho_z \leq \mathfrak{P}_y$ **then**
      $\mathcal{V}_z \leftarrow \mathcal{V}_z - z_\pm$ and $\rho_z \leftarrow \delta_z$ ▷ up/down 1 slice
      $z \leftarrow \textsc{Proj}(z_{\mathcal{V}_0}, \mathcal{V}_z, \delta_z, z_+, z_\pm)$
      $x \leftarrow \textsc{Edge}(\Delta_{xz}, z, z_0, x_0)$
      $y \leftarrow \textsc{Edge}(\Delta_{yz}, z, z_0, y_0)$
      $\rho_x \leftarrow \delta\text{ToGo}(x_{\mathcal{V}_0}, x, x_+, x_\pm, \delta_x, \mathcal{V}_x)$
      $\rho_y \leftarrow \delta\text{ToGo}(y_{\mathcal{V}_0}, y, y_+, y_\pm, \delta_y, \mathcal{V}_y)$
    else if $\mathfrak{P}_x \leq \mathfrak{P}_y$ **then**
      $\mathcal{V}_x \leftarrow \mathcal{V}_x + x_\pm$ and $\rho_x \leftarrow \delta_x$ ▷ right/left 1 voxel
      $x \leftarrow \textsc{Proj}(x_{\mathcal{V}_0}, \mathcal{V}_x, \delta_x, x_+, x_\pm)$
      $y \leftarrow \textsc{Edge}(\Delta_{yx}, x, x_0, y_0)$
      $z \leftarrow \textsc{Edge}(\Delta_{zx}, x, x_0, z_0)$
      $\rho_y \leftarrow \delta\text{ToGo}(y_{\mathcal{V}_0}, y, y_+, y_\pm, \delta_y, \mathcal{V}_y)$
      $\rho_z \leftarrow \delta\text{ToGo}(z_{\mathcal{V}_0}, z, z_+, z_\pm, \delta_z, \mathcal{V}_z)$
    else
      $\mathcal{V}_y \leftarrow \mathcal{V}_y - y_\pm$ and $\rho_y \leftarrow \delta_y$ ▷ up/down 1 voxel
      $y \leftarrow \textsc{Proj}(y_{\mathcal{V}_0}, \mathcal{V}_y, \delta_y, y_+, y_\pm)$
      $x \leftarrow \textsc{Edge}(\Delta_{xy}, y, y_0, x_0)$
      $z \leftarrow \textsc{Edge}(\Delta_{zy}, y, y_0, z_0)$
      $\rho_x \leftarrow \delta\text{ToGo}(x_{\mathcal{V}_0}, x, x_+, x_\pm, \delta_x, \mathcal{V}_x)$
      $\rho_z \leftarrow \delta\text{ToGo}(z_{\mathcal{V}_0}, z, z_+, z_\pm, \delta_z, \mathcal{V}_z)$
    end if
    if $\rho_x = 0$ **then**
      $\rho_x \leftarrow \delta_x$ and $\mathcal{V}_x \leftarrow \mathcal{V}_x + x_\pm$
    end if
    if $\rho_y = 0$ **then**
      $\rho_y \leftarrow \delta_y$ and $\mathcal{V}_y \leftarrow \mathcal{V}_y - y_\pm$
    end if
    if $\rho_z = 0$ **then**
      $\rho_z \leftarrow \delta_z$ and $\mathcal{V}_z \leftarrow \mathcal{V}_z - z_\pm$
    end if
    $\mathcal{V}_z \leftarrow \max(\mathcal{V}_z, 0)$ and $\mathcal{V} \leftarrow \mathcal{V}_x + \mathcal{V}_y \cdot$ `COLUMNS` $+ \mathcal{V}_z \cdot$ `COLUMNS` $\cdot$ `ROWS`
    if ⟨`STOP CONDITION`⟩ **then** STOP = `TRUE`
    end if
  end while
end procedure

APPENDIX C

Derivation Of Simplified MLP

C.1 Integration and Algebraic Expansion of Prior Likelihood Scattering Elements

The algebraic expansion and definite integration of the prior likelihood scattering elements $\sigma^2_{t_1}(u_0,u_1)$, $\sigma^2_{\theta_1}(u_0,u_1)$, and $\sigma^2_{t_1\theta_1}(u_0,u_1)$ is shown step-by-step in the following:

$$
\begin{aligned}
\sigma^2_{t_1}(u_0,u_1) &= C(u_1-u_0)\int_{u_0}^{u_1}\frac{(u_1-u)^2}{\beta^2(u)p^2(u)}\frac{du}{X_0}\\
&= \frac{C(u_1-u_0)}{X_0}\int_{u_0}^{u_1}(u_1-u)^2\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)du\\
&= \frac{C(u_1-u_0)}{X_0}\int_{u_0}^{u_1}\left(u_1^2-2u_1u+u^2\right)\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)du\\
&= \frac{C(u_1-u_0)}{X_0}\int_{u_0}^{u_1}\Big[u_1^2\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)\\
&\quad -2u_1u\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)\\
&\quad +u^2\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)\Big]du\\
&= \frac{C(u_1-u_0)}{X_0}\int_{u_0}^{u_1}\Big[u_1^2\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)\\
&\quad -2u_1\left(a_0u+a_1u^2+a_2u^3+a_3u^4+a_4u^5+a_5u^6\right)\\
&\quad +\left(a_0u^2+a_1u^3+a_2u^4+a_3u^5+a_4u^6+a_5u^7\right)\Big]du\\
&= \frac{C(u_1-u_0)}{X_0}\left[u_1^2\left(a_0u+\frac{a_1u^2}{2}+\frac{a_2u^3}{3}+\frac{a_3u^4}{4}+\frac{a_4u^5}{5}+\frac{a_5u^6}{6}\right)\right.\\
&\quad -2u_1\left(\frac{a_0u^2}{2}+\frac{a_1u^3}{3}+\frac{a_2u^4}{4}+\frac{a_3u^5}{5}+\frac{a_4u^6}{6}+\frac{a_5u^7}{7}\right)\\
&\quad \left.+\left(\frac{a_0u^3}{3}+\frac{a_1u^4}{4}+\frac{a_2u^5}{5}+\frac{a_3u^6}{6}+\frac{a_4u^7}{7}+\frac{a_5u^8}{8}\right)\right]_{u=u_0}^{u_1}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{C(u_1-u_0)}{X_0}\left[\left(a_0u_1^3+\frac{a_1u_1^4}{2}+\frac{a_2u_1^5}{3}+\frac{a_3u_1^6}{4}+\frac{a_4u_1^7}{5}+\frac{a_5u_1^8}{6}\right)\right.\\
&\quad -u_1^2\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\\
&\quad -2\left(\frac{a_0u_1^3}{2}+\frac{a_1u_1^4}{3}+\frac{a_2u_1^5}{4}+\frac{a_3u_1^6}{5}+\frac{a_4u_1^7}{6}+\frac{a_5u_1^8}{7}\right)\\
&\quad +2u_1\left(\frac{a_0u_0^2}{2}+\frac{a_1u_0^3}{3}+\frac{a_2u_0^4}{4}+\frac{a_3u_0^5}{5}+\frac{a_4u_0^6}{6}+\frac{a_5u_0^7}{7}\right)\\
&\quad +\left(\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{4}+\frac{a_2u_1^5}{5}+\frac{a_3u_1^6}{6}+\frac{a_4u_1^7}{7}+\frac{a_5u_1^8}{8}\right)\\
&\quad \left.-\left(\frac{a_0u_0^3}{3}+\frac{a_1u_0^4}{4}+\frac{a_2u_0^5}{5}+\frac{a_3u_0^6}{6}+\frac{a_4u_0^7}{7}+\frac{a_5u_0^8}{8}\right)\right]\\
&= \frac{C(u_1-u_0)}{X_0}\left\{a_0u_1^3\left(1-1+\frac{1}{3}\right)+a_1u_1^4\left(\frac{1}{2}-\frac{2}{3}+\frac{1}{4}\right)+a_2u_1^5\left(\frac{1}{3}-\frac{2}{4}+\frac{1}{5}\right)\right.\\
&\quad +a_3u_1^6\left(\frac{1}{4}-\frac{2}{5}+\frac{1}{6}\right)+a_4u_1^7\left(\frac{1}{5}-\frac{2}{6}+\frac{1}{7}\right)+a_5u_1^8\left(\frac{1}{6}-\frac{2}{7}+\frac{1}{8}\right)\\
&\quad -u_1^2\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\\
&\quad +2u_1\left(\frac{a_0u_0^2}{2}+\frac{a_1u_0^3}{3}+\frac{a_2u_0^4}{4}+\frac{a_3u_0^5}{5}+\frac{a_4u_0^6}{6}+\frac{a_5u_0^7}{7}\right)\\
&\quad \left.-\left(\frac{a_0u_0^3}{3}+\frac{a_1u_0^4}{4}+\frac{a_2u_0^5}{5}+\frac{a_3u_0^6}{6}+\frac{a_4u_0^7}{7}+\frac{a_5u_0^8}{8}\right)\right\}
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
\sigma^2_{t_1}(u_0,u_1) = &\ \frac{C(u_1-u_0)}{X_0}\left\{\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{12}+\frac{a_2u_1^5}{30}+\frac{a_3u_1^6}{60}+\frac{a_4u_1^7}{105}+\frac{a_5u_1^8}{168}\right.\\
&-u_1^2\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\\
&+2u_1\left(\frac{a_0u_0^2}{2}+\frac{a_1u_0^3}{3}+\frac{a_2u_0^4}{4}+\frac{a_3u_0^5}{5}+\frac{a_4u_0^6}{6}+\frac{a_5u_0^7}{7}\right)\\
&\left.-\left(\frac{a_0u_0^3}{3}+\frac{a_1u_0^4}{4}+\frac{a_2u_0^5}{5}+\frac{a_3u_0^6}{6}+\frac{a_4u_0^7}{7}+\frac{a_5u_0^8}{8}\right)\right\}
\end{aligned}
}
$$

$$
\begin{aligned}
\sigma^2_{\theta_1}(u_0,u_1) &= C(u_1-u_0)\int_{u_0}^{u_1}\frac{u_1-u}{\beta^2(u)p^2(u)}\frac{du}{X_0}\\
&= \frac{C(u_1-u_0)}{X_0}\int_{u_0}^{u_1}(u_1-u)\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)du
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{C(u_1-u_0)}{X_0}\int_{u_0}^{u_1}\Big[u_1\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)\\
&\quad -u\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)\Big]du\\
&= \frac{C(u_1-u_0)}{X_0}\int_{u_0}^{u_1}\Big[u_1\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)\\
&\quad -\left(a_0u+a_1u^2+a_2u^3+a_3u^4+a_4u^5+a_5u^6\right)\Big]du\\
&= \frac{C(u_1-u_0)}{X_0}\left[u_1\left(a_0u+\frac{a_1u^2}{2}+\frac{a_2u^3}{3}+\frac{a_3u^4}{4}+\frac{a_4u^5}{5}+\frac{a_5u^6}{6}\right)\right.\\
&\quad \left.-\left(\frac{a_0u^2}{2}+\frac{a_1u^3}{3}+\frac{a_2u^4}{4}+\frac{a_3u^5}{5}+\frac{a_4u^6}{6}+\frac{a_5u^7}{7}\right)\right]_{u=u_0}^{u_1}\\
&= \frac{C(u_1-u_0)}{X_0}\left[\left(a_0u_1^2+\frac{a_1u_1^3}{2}+\frac{a_2u_1^4}{3}+\frac{a_3u_1^5}{4}+\frac{a_4u_1^6}{5}+\frac{a_5u_1^7}{6}\right)\right.\\
&\quad -u_1\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\\
&\quad -\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\\
&\quad \left.+\left(\frac{a_0u_0^2}{2}+\frac{a_1u_0^3}{3}+\frac{a_2u_0^4}{4}+\frac{a_3u_0^5}{5}+\frac{a_4u_0^6}{6}+\frac{a_5u_0^7}{7}\right)\right]\\
&= \frac{C(u_1-u_0)}{X_0}\left\{\left[a_0u_1^2\left(\frac{1}{1}-\frac{1}{2}\right)+a_1u_1^3\left(\frac{1}{2}-\frac{1}{3}\right)+a_2u_1^4\left(\frac{1}{3}-\frac{1}{4}\right)\right.\right.\\
&\quad \left.+a_3u_1^5\left(\frac{1}{4}-\frac{1}{5}\right)+a_4u_1^6\left(\frac{1}{5}-\frac{1}{6}\right)+a_5u_1^7\left(\frac{1}{6}-\frac{1}{7}\right)\right]\\
&\quad -u_1\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\\
&\quad \left.+\left(\frac{a_0u_0^2}{2}+\frac{a_1u_0^3}{3}+\frac{a_2u_0^4}{4}+\frac{a_3u_0^5}{5}+\frac{a_4u_0^6}{6}+\frac{a_5u_0^7}{7}\right)\right\}
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
\sigma^2_{\theta_1}(u_0,u_1) = &\ \frac{C(u_1-u_0)}{X_0}\left[\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{6}+\frac{a_2u_1^4}{12}+\frac{a_3u_1^5}{20}+\frac{a_4u_1^6}{30}+\frac{a_5u_1^7}{42}\right)\right.\\
&-u_1\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\\
&\left.+\left(\frac{a_0u_0^2}{2}+\frac{a_1u_0^3}{3}+\frac{a_2u_0^4}{4}+\frac{a_3u_0^5}{5}+\frac{a_4u_0^6}{6}+\frac{a_5u_0^7}{7}\right)\right]
\end{aligned}
}
$$

$$\sigma^2_{t_1\theta_1}(u_0,u_1) = C(u_1-u_0)\int_{u_0}^{u_1}\frac{1}{\beta^2(u)p^2(u)}\frac{du}{X_0}$$
$$= \frac{C(u_1-u_0)}{X_0}\int_{u_0}^{u_1}\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)du$$
$$= \frac{C(u_1-u_0)}{X_0}\left[a_0u+\frac{a_1u^2}{2}+\frac{a_2u^3}{3}+\frac{a_3u^4}{4}+\frac{a_4u^5}{5}+\frac{a_5u^6}{6}\right]_{u=u_0}^{u_1}$$

$$\boxed{\sigma^2_{t_1\theta_1}(u_0,u_1) = \frac{C(u_1-u_0)}{X_0}\left[\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)-\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\right]}$$

C.2 Integration and Algebraic Expansion of Posterior Likelihood Scattering Elements

The algebraic expansion and definite integration of the prior likelihood scattering elements $\sigma^2_{t_2}(u_1,u_2)$, $\sigma^2_{\theta_2}(u_1,u_2)$, and $\sigma^2_{t_2\theta_2}(u_1,u_2)$ is shown step-by-step in the following:

$$\sigma^2_{t_2}(u_1,u_2) = C(u_2-u_1)\int_{u_1}^{u_2}\frac{(u_2-u)^2}{\beta^2(u)p^2(u)}\frac{du}{X_0}$$
$$= \frac{C(u_2-u_1)}{X_0}\int_{u_1}^{u_2}(u_2-u)^2\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)du$$
$$= \frac{C(u_2-u_1)}{X_0}\int_{u_1}^{u_2}\left(u_2^2-2u_2u+u^2\right)\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)du$$
$$= \frac{C(u_2-u_1)}{X_0}\left[u_2^2\left(a_0u+\frac{a_1u^2}{2}+\frac{a_2u^3}{3}+\frac{a_3u^4}{4}+\frac{a_4u^5}{5}+\frac{a_5u^6}{6}\right)\right.$$
$$-2u_2\left(\frac{a_0u^2}{2}+\frac{a_1u^3}{3}+\frac{a_2u^4}{4}+\frac{a_3u^5}{5}+\frac{a_4u^6}{6}+\frac{a_5u^7}{7}\right)$$
$$\left.+\left(\frac{a_0u^3}{3}+\frac{a_1u^4}{4}+\frac{a_2u^5}{5}+\frac{a_3u^6}{6}+\frac{a_4u^7}{7}+\frac{a_5u^8}{8}\right)\right]_{u=u_1}^{u_2}$$

$$
\begin{aligned}
&= \frac{C(u_2-u_1)}{X_0}\left[\left(a_0u_2^3+\frac{a_1u_2^4}{2}+\frac{a_2u_2^5}{3}+\frac{a_3u_2^6}{4}+\frac{a_4u_2^7}{5}+\frac{a_5u_2^8}{6}\right)\right.\\
&\quad -u_2^2\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\\
&\quad -2\left(\frac{a_0u_2^3}{2}+\frac{a_1u_2^4}{3}+\frac{a_2u_2^5}{4}+\frac{a_3u_2^6}{5}+\frac{a_4u_2^7}{6}+\frac{a_5u_2^8}{7}\right)\\
&\quad +2u_2\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\\
&\quad +\left(\frac{a_0u_2^3}{3}+\frac{a_1u_2^4}{4}+\frac{a_2u_2^5}{5}+\frac{a_3u_2^6}{6}+\frac{a_4u_2^7}{7}+\frac{a_5u_2^8}{8}\right)\\
&\quad \left.-\left(\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{4}+\frac{a_2u_1^5}{5}+\frac{a_3u_1^6}{6}+\frac{a_4u_1^7}{7}+\frac{a_5u_1^8}{8}\right)\right]\\
&= \frac{C(u_2-u_1)}{X_0}\left\{a_0u_2^3\left(1-1+\frac{1}{3}\right)+a_1u_2^4\left(\frac{1}{2}-\frac{2}{3}+\frac{1}{4}\right)+a_2u_2^5\left(\frac{1}{3}-\frac{2}{4}+\frac{1}{5}\right)\right.\\
&\quad +a_3u_2^6\left(\frac{1}{4}-\frac{2}{5}+\frac{1}{6}\right)+a_4u_2^7\left(\frac{1}{5}-\frac{2}{6}+\frac{1}{7}\right)+a_5u_2^8\left(\frac{1}{6}-\frac{2}{7}+\frac{1}{8}\right)\\
&\quad -u_2^2\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\\
&\quad +2u_2\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\\
&\quad \left.-\left(\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{4}+\frac{a_2u_1^5}{5}+\frac{a_3u_1^6}{6}+\frac{a_4u_1^7}{7}+\frac{a_5u_1^8}{8}\right)\right\}
\end{aligned}
$$

$$
\begin{aligned}
\sigma^2_{t_2}(u_1,u_2) &= \frac{C(u_2-u_1)}{X_0}\left\{\left[\frac{a_0u_2^3}{3}+\frac{a_1u_2^4}{12}+\frac{a_2u_2^5}{30}+\frac{a_3u_2^6}{60}+\frac{a_4u_2^7}{105}+\frac{a_5u_2^8}{168}\right]\right.\\
&\quad -u_2^2\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\\
&\quad +2u_2\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\\
&\quad \left.-\left(\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{4}+\frac{a_2u_1^5}{5}+\frac{a_3u_1^6}{6}+\frac{a_4u_1^7}{7}+\frac{a_5u_1^8}{8}\right)\right\}
\end{aligned}
$$

$$
\begin{aligned}
\sigma^2_{\theta_2}(u_1,u_2) &= C(u_2-u_1)\int_{u_1}^{u_2}\frac{u_2-u}{\beta^2(u)p^2(u)}\frac{du}{X_0}\\
&= \frac{C(u_2-u_1)}{X_0}\int_{u_1}^{u_2}(u_2-u)\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)du\\
&= \frac{C(u_2-u_1)}{X_0}\left[u_2\left(a_0u+\frac{a_1u^2}{2}+\frac{a_2u^3}{3}+\frac{a_3u^4}{4}+\frac{a_4u^5}{5}+\frac{a_5u^6}{6}\right)\right.
\end{aligned}
$$

$$
\begin{aligned}
&-\left(\frac{a_0u^2}{2}+\frac{a_1u^3}{3}+\frac{a_2u^4}{4}+\frac{a_3u^5}{5}+\frac{a_4u^6}{6}+\frac{a_5u^7}{7}\right)\Bigg]_{u=u_1}^{u_2}\\
&=\frac{C(u_2-u_1)}{X_0}\left[\left(a_0u_2^2+\frac{a_1u_2^3}{2}+\frac{a_2u_2^4}{3}+\frac{a_3u_2^5}{4}+\frac{a_4u_2^6}{5}+\frac{a_5u_2^7}{6}\right)\right.\\
&-u_2\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\\
&-\left(\frac{a_0u_2^2}{2}+\frac{a_1u_2^3}{3}+\frac{a_2u_2^4}{4}+\frac{a_3u_2^5}{5}+\frac{a_4u_2^6}{6}+\frac{a_5u_2^7}{7}\right)\\
&\left.+\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\right]\\
&=\frac{C(u_2-u_1)}{X_0}\left\{\left[a_0u_2^2\left(\frac{1}{1}-\frac{1}{2}\right)+a_1u_2^3\left(\frac{1}{2}-\frac{1}{3}\right)+a_2u_2^4\left(\frac{1}{3}-\frac{1}{4}\right)\right.\right.\\
&+a_3u_2^5\left(\frac{1}{4}-\frac{1}{5}\right)+a_4u_2^6\left(\frac{1}{5}-\frac{1}{6}\right)+a_5u_2^7\left(\frac{1}{6}-\frac{1}{7}\right)\Big]\\
&-u_2\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\\
&\left.\left.+\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\right\}\right.
\end{aligned}
$$

$$
\begin{aligned}
\sigma^2_{\theta_2}(u_1,u_2)=\;&\frac{C(u_2-u_1)}{X_0}\left[\left(\frac{a_0u_2^2}{2}+\frac{a_1u_2^3}{6}+\frac{a_2u_2^4}{12}+\frac{a_3u_2^5}{20}+\frac{a_4u_2^6}{30}+\frac{a_5u_2^7}{42}\right)\right.\\
&-u_2\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\\
&\left.+\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
\sigma^2_{t_2\theta_2}(u_1,u_2)&=C(u_2-u_1)\int_{u_1}^{u_2}\frac{1}{\beta^2(u)p^2(u)}\frac{du}{X_0}\\
&=\frac{C(u_2-u_1)}{X_0}\int_{u_1}^{u_2}\left(a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5\right)du\\
&=\frac{C(u_2-u_1)}{X_0}\left[a_0u+\frac{a_1u^2}{2}+\frac{a_2u^3}{3}+\frac{a_3u^4}{4}+\frac{a_4u^5}{5}+\frac{a_5u^6}{6}\right]_{u=u_1}^{u_2}
\end{aligned}
$$

$$
\begin{aligned}
\sigma^2_{t_2\theta_2}(u_1,u_2)=\;&\frac{C(u_2-u_1)}{X_0}\left[\left(a_0u_2+\frac{a_1u_2^2}{2}+\frac{a_2u_2^3}{3}+\frac{a_3u_2^4}{4}+\frac{a_4u_2^5}{5}+\frac{a_5u_2^6}{6}\right)\right.\\
&\left.-\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\right]
\end{aligned}
$$

C.3.1 Evaluation of Terms Under Linear Coordinate Translation

The path of a proton is independent of the coordinate system under which its behavior is analyzed. Inspection of the equations defining the prior and posterior likelihood scattering elements reveals that they are both translationally and rotationally invariant, indicating that the theory coincides with intuition of the underlying physical processes a proton undergoes while traversing an object. Considering the translational invariance first, the immediately obvious choice is to translate the coordinate system such that either the object entry or exit points lie at the origin, which is arbitrarily chosen to be the object entry point such that $t_0 = u_0 = 0$. Note that the posterior likelihood scattering elements have no u_0 or t_0 dependency, so the linear translation does not affect them. Hence, evaluating only the prior likelihood scattering elements under this coordinate system translation yields the following simplified equations:

$$\begin{aligned}
\sigma^2_{t_1}(u_0,u_1)\Big|_{u_0=0} &= \frac{C(u_1-u_0)}{X_0}\left[\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{12}+\frac{a_2u_1^5}{30}+\frac{a_3u_1^6}{60}+\frac{a_4u_1^7}{105}+\frac{a_5u_1^8}{168}\right.\\
&\quad -u_1^2\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\\
&\quad +2u_1\left(\frac{a_0u_0^2}{2}+\frac{a_1u_0^3}{3}+\frac{a_2u_0^4}{4}+\frac{a_3u_0^5}{5}+\frac{a_4u_0^6}{6}+\frac{a_5u_0^7}{7}\right)\\
&\quad \left.-\left(\frac{a_0u_0^3}{3}+\frac{a_1u_0^4}{4}+\frac{a_2u_0^5}{5}+\frac{a_3u_0^6}{6}+\frac{a_4u_0^7}{7}+\frac{a_5u_0^8}{8}\right)\right]\Bigg|_{u_0=0}\\
&= \frac{C(u_1-0)}{X_0}\left[\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{12}+\frac{a_2u_1^5}{30}+\frac{a_3u_1^6}{60}+\frac{a_4u_1^7}{105}+\frac{a_5u_1^8}{168}\right.\\
&\quad -u_1^2\left(a_0(0)+\frac{a_1(0)^2}{2}+\frac{a_2(0)^3}{3}+\frac{a_3(0)^4}{4}+\frac{a_4(0)^5}{5}+\frac{a_5(0)^6}{6}\right)\\
&\quad +2u_1\left(\frac{a_0(0)^2}{2}+\frac{a_1(0)^3}{3}+\frac{a_2(0)^4}{4}+\frac{a_3(0)^5}{5}+\frac{a_4(0)^6}{6}+\frac{a_5(0)^7}{7}\right)\\
&\quad \left.-\left(\frac{a_0(0)^3}{3}+\frac{a_1(0)^4}{4}+\frac{a_2(0)^5}{5}+\frac{a_3(0)^6}{6}+\frac{a_4(0)^7}{7}+\frac{a_5(0)^8}{8}\right)\right]\\
&= \frac{C(u_1)}{X_0}\left[\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{12}+\frac{a_2u_1^5}{30}+\frac{a_3u_1^6}{60}+\frac{a_4u_1^7}{105}+\frac{a_5u_1^8}{168}\right.
\end{aligned}$$

$$
\begin{aligned}
&- u_1^2\left(\cancelto{0}{0 + \frac{0}{2} + \frac{0}{3} + \frac{0}{4} + \frac{0}{5} + \frac{0}{6}}\right) + 2u_1\left(\cancelto{0}{\frac{0}{2} + \frac{0}{3} + \frac{0}{4} + \frac{0}{5} + \frac{0}{6} + \frac{0}{7}}\right) \\
&- \left(\cancelto{0}{\frac{0}{3} + \frac{0}{4} + \frac{0}{5} + \frac{0}{6} + \frac{0}{7} + \frac{0}{8}}\right)\Bigg] \\
&= \frac{C(u_1)}{X_0}\left\{\left[\frac{a_0u_1^3}{3} + \frac{a_1u_1^4}{12} + \frac{a_2u_1^5}{30} + \frac{a_3u_1^6}{60} + \frac{a_4u_1^7}{105} + \frac{a_5u_1^8}{168}\right]\right. \\
&\left.\cancelto{0}{-u_1^2(0) + 2u_1(0) - (0)}\right\}
\end{aligned}
$$

$$
\boxed{\sigma_{t_1}^2(u_0,u_1)\big|_{u_0=0} = \sigma_{t_1}^2(0,u_1) = \frac{C(u_1)}{X_0}\left[\frac{a_0u_1^3}{3} + \frac{a_1u_1^4}{12} + \frac{a_2u_1^5}{30} + \frac{a_3u_1^6}{60} + \frac{a_4u_1^7}{105} + \frac{a_5u_1^8}{168}\right]}
$$

$$
\begin{aligned}
\sigma_{\theta_1}^2(u_0,u_1)\big|_{u_0=0} &= \frac{C(u_1-u_0)}{X_0}\left[\left(\frac{a_0u_1^2}{2} + \frac{a_1u_1^3}{6} + \frac{a_2u_1^4}{12} + \frac{a_3u_1^5}{20} + \frac{a_4u_1^6}{30} + \frac{a_5u_1^7}{42}\right)\right. \\
&\quad - u_1\left(a_0u_0 + \frac{a_1u_0^2}{2} + \frac{a_2u_0^3}{3} + \frac{a_3u_0^4}{4} + \frac{a_4u_0^5}{5} + \frac{a_5u_0^6}{6}\right) \\
&\quad \left.+\left(\frac{a_0u_0^2}{2} + \frac{a_1u_0^3}{3} + \frac{a_2u_0^4}{4} + \frac{a_3u_0^5}{5} + \frac{a_4u_0^6}{6} + \frac{a_5u_0^7}{7}\right)\right]\Bigg|_{u_0=0} \\
&= \frac{C(u_1-0)}{X_0}\left[\left(\frac{a_0u_1^2}{2} + \frac{a_1u_1^3}{6} + \frac{a_2u_1^4}{12} + \frac{a_3u_1^5}{20} + \frac{a_4u_1^6}{30} + \frac{a_5u_1^7}{42}\right)\right. \\
&\quad - u_1\left(a_0(0) + \frac{a_1(0)^2}{2} + \frac{a_2(0)^3}{3} + \frac{a_3(0)^4}{4} + \frac{a_4(0)^5}{5} + \frac{a_5(0)^6}{6}\right) \\
&\quad \left.+\left(\frac{a_0(0)^2}{2} + \frac{a_1(0)^3}{3} + \frac{a_2(0)^4}{4} + \frac{a_3(0)^5}{5} + \frac{a_4(0)^6}{6} + \frac{a_5(0)^7}{7}\right)\right] \\
&= \frac{C(u_1)}{X_0}\left[\left(\frac{a_0u_1^2}{2} + \frac{a_1u_1^3}{6} + \frac{a_2u_1^4}{12} + \frac{a_3u_1^5}{20} + \frac{a_4u_1^6}{30} + \frac{a_5u_1^7}{42}\right)\right. \\
&\quad -u_1\left(\cancelto{0}{a_0(0) + \frac{0}{2} + \frac{0}{3} + \frac{0}{4} + \frac{0}{5} + \frac{0}{6}}\right) + \left(\cancelto{0}{\frac{0}{2} + \frac{0}{3} + \frac{0}{4} + \frac{0}{5} + \frac{0}{6} + \frac{0}{7}}\right)\Bigg] \\
&= \frac{C(u_1)}{X_0}\left[\left(\frac{a_0u_1^2}{2} + \frac{a_1u_1^3}{6} + \frac{a_2u_1^4}{12} + \frac{a_3u_1^5}{20} + \frac{a_4u_1^6}{30} + \frac{a_5u_1^7}{42}\right) - \cancelto{0}{u_1(0) + (0)}\right.
\end{aligned}
$$

$$
\boxed{\sigma_{\theta_1}^2(u_0,u_1)\big|_{u_0=0} = \sigma_{\theta_1}^2(0,u_1) = \frac{C(u_1)}{X_0}\left[\frac{a_0u_1^2}{2} + \frac{a_1u_1^3}{6} + \frac{a_2u_1^4}{12} + \frac{a_3u_1^5}{20} + \frac{a_4u_1^6}{30} + \frac{a_5u_1^7}{42}\right]}
$$

$$\begin{aligned}
\sigma^2_{t_1\theta_1}(u_0,u_1)\big|_{u_0=0} &= \frac{C(u_1-u_0)}{X_0}\left[\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\right.\\
&\quad\left.-\left(a_0u_0+\frac{a_1u_0^2}{2}+\frac{a_2u_0^3}{3}+\frac{a_3u_0^4}{4}+\frac{a_4u_0^5}{5}+\frac{a_5u_0^6}{6}\right)\right]\Bigg|_{u_0=0}\\
&= \frac{C(u_1-0)}{X_0}\left[\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\right.\\
&\quad\left.-\left(a_0(0)+\frac{a_1(0)^2}{2}+\frac{a_2(0)^3}{3}+\frac{a_3(0)^4}{4}+\frac{a_4(0)^5}{5}+\frac{a_5(0)^6}{6}\right)\right]\\
&= \frac{C(u_1)}{X_0}\left[\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\right.\\
&\quad\left.-\left(\cancel{0+\frac{0}{2}+\frac{0}{3}+\frac{0}{4}+\frac{0}{5}+\frac{0}{6}}\right)0\right]
\end{aligned}$$

$$\boxed{\sigma^2_{t_1\theta_1}(u_0,u_1)\big|_{u_0=0} = \sigma^2_{t_1\theta_1}(0,u_1) = \frac{C(u_1)}{X_0}\left[a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right]}$$

The simplified polynomial equations for each of the prior likelihood scattering elements, as well as the previously simplified posterior likelihood scattering elements, are summarized as follows:

$$\boxed{\sigma^2_{t_1}(u_0,u_1)\big|_{u_0=0} = \sigma^2_{t_1}(0,u_1) = \frac{C(u_1)}{X_0}\left[\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{12}+\frac{a_2u_1^5}{30}+\frac{a_3u_1^6}{60}+\frac{a_4u_1^7}{105}+\frac{a_5u_1^8}{168}\right]} \tag{C.1}$$

$$\boxed{\sigma^2_{\theta_1}(u_0,u_1)\big|_{u_0=0} = \sigma^2_{\theta_1}(0,u_1) = \frac{C(u_1)}{X_0}\left[\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{6}+\frac{a_2u_1^4}{12}+\frac{a_3u_1^5}{20}+\frac{a_4u_1^6}{30}+\frac{a_5u_1^7}{42}\right]} \tag{C.2}$$

$$\boxed{\sigma^2_{t_1\theta_1}(u_0,u_1)\big|_{u_0=0} = \sigma^2_{t_1\theta_1}(0,u_1) = \frac{C(u_1)}{X_0}\left[a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right]} \tag{C.3}$$

$$\boxed{\begin{aligned}
\sigma^2_{t_2}(u_1,u_2) = \quad & \frac{C(u_2-u_1)}{X_0}\left\{\left[\frac{a_0u_2^3}{3}+\frac{a_1u_2^4}{12}+\frac{a_2u_2^5}{30}+\frac{a_3u_2^6}{60}+\frac{a_4u_2^7}{105}+\frac{a_5u_2^8}{168}\right]\right.\\
& -u_2^2\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\\
& +2u_2\left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\\
& \left.-\left(\frac{a_0u_1^3}{3}+\frac{a_1u_1^4}{4}+\frac{a_2u_1^5}{5}+\frac{a_3u_1^6}{6}+\frac{a_4u_1^7}{7}+\frac{a_5u_1^8}{8}\right)\right\}
\end{aligned}} \tag{C.4}$$

$$\sigma^2_{\theta_2}(u_1,u_2) = \frac{C(u_2-u_1)}{X_0}\left[\left(\frac{a_0u_2^2}{2}+\frac{a_1u_2^3}{6}+\frac{a_2u_2^4}{12}+\frac{a_3u_2^5}{20}+\frac{a_4u_2^6}{30}+\frac{a_5u_2^7}{42}\right) - u_2\left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right) + \left(\frac{a_0u_1^2}{2}+\frac{a_1u_1^3}{3}+\frac{a_2u_1^4}{4}+\frac{a_3u_1^5}{5}+\frac{a_4u_1^6}{6}+\frac{a_5u_1^7}{7}\right)\right] \tag{C.5}$$

$$\sigma^2_{t_2\theta_2}(u_1,u_2) = \frac{C(u_2-u_1)}{X_0}\left[\left(a_0u_2+\frac{a_1u_2^2}{2}+\frac{a_2u_2^3}{3}+\frac{a_3u_2^4}{4}+\frac{a_4u_2^5}{5}+\frac{a_5u_2^6}{6}\right) - \left(a_0u_1+\frac{a_1u_1^2}{2}+\frac{a_2u_1^3}{3}+\frac{a_3u_1^4}{4}+\frac{a_4u_1^5}{5}+\frac{a_5u_1^6}{6}\right)\right] \tag{C.6}$$

A careful inspection of the polynomial terms comprising Equations C.1–C.6 reveals several common polynomial terms; some of the polynomials appear identically in multiple equations, whereas others have the same structure (i.e. same coefficients appearing with the same polynomial degrees) but are a function of a different depth parameter (e.g. u_1 or u_2). The scattering elements terms can be simplified considerably by exploiting this fact to identify and define common polynomial equations of a generic depth u, then rewriting the equations for the scattering elements in terms of these common polynomials, where the generic depth parameter is defined as $u = u_1$ or $u = u_2$ as the case dictates.

These common polynomial equations, $P_N(u)$, of generic proton depth u, are defined according to the following system of equations:

$$\begin{bmatrix}P_1\\P_2\\P_3\\P_4\\P_5\end{bmatrix} = \begin{bmatrix}0 & 0 & \frac{a_0}{3} & \frac{a_1}{12} & \frac{a_2}{30} & \frac{a_3}{60} & \frac{a_4}{105} & \frac{a_5}{168}\\ 0 & \frac{a_0}{2} & \frac{a_1}{6} & \frac{a_2}{12} & \frac{a_3}{20} & \frac{a_4}{30} & \frac{a_5}{42} & 0\\ a_0 & \frac{a_1}{2} & \frac{a_2}{3} & \frac{a_3}{4} & \frac{a_4}{5} & \frac{a_5}{6} & 0 & 0\\ 0 & \frac{a_0}{2} & \frac{a_1}{3} & \frac{a_2}{4} & \frac{a_3}{5} & \frac{a_4}{6} & \frac{a_5}{7} & 0\\ 0 & 0 & \frac{a_0}{3} & \frac{a_1}{4} & \frac{a_2}{5} & \frac{a_3}{6} & \frac{a_4}{7} & \frac{a_5}{8}\end{bmatrix}\begin{bmatrix}u\\u^2\\u^3\\u^4\\u^5\\u^6\\u^7\\u^8\end{bmatrix} \tag{C.7}$$

Rewriting Equations C.1–C.3 and Equations C.4–C.6 in terms of these $P_N(u)$ yields the following simplified equations for the prior and posterior likelihood scattering elements, respectively:

$$\sigma^2_{t_1}(u_0, u_1) = C(u_1)P_1(u_1) \tag{C.8}$$

$$\sigma^2_{\theta_1}(u_0, u_1) = C(u_1)P_2(u_1) \tag{C.9}$$

$$\sigma^2_{t_1\theta_1}(u_0, u_1) = C(u_1)P_3(u_1) \tag{C.10}$$

and

$$\sigma^2_{t_2}(u_1, u_2) = C(u_2 - u_1)\left[P_1(u_2) - u_2^2 P_3(u_1) + 2u_2 P_4(u_1) - P_5(u_1)\right] \tag{C.11}$$

$$\sigma^2_{\theta_2}(u_1, u_2) = C(u_2 - u_1)\left[P_2(u_2) - u_2 P_3(u_1) + P_4(u_1)\right] \tag{C.12}$$

$$\sigma^2_{t_2\theta_2}(u_1, u_2) = C(u_2 - u_1)\left[P_3(u_2) - P_3(u_1)\right] \tag{C.13}$$

Equations C.8 – C.10 and Equations C.11 – C.13 are the final forms of the prior and posterior likelihood scattering elements and are provided without derivation in Equations 5.12 – 5.17 of the main text. The substitution matrix above in Equation C.7 is also provided in the main text in Equation 5.18. The vectors $\vec{y}_0$ and $\vec{y}_2$, which characterize the path of a proton as it enters and exits the object, respectively, are also transformed in their t components by the linear translation, but the derivation of the resulting equations is deferred until the coordinate system rotation transforming the θ components is also evaluated. The combined transformation of the vectors is described next.

C.3.2 Evaluation of Terms Under Coordinate Rotation

The scattering matrices Σ_1 and Σ_2 are unaffected by the coordinate rotation since none of their elements contain any angle dependent terms. On the other hand, the vectors $\vec{y}_0$ and $\vec{y}_2$ characterizing the path of a proton at the object entry and exit points are affected by both the linear translation and rotation transformations. Since the vector $\vec{y}_0$ only appears

in y_{MLP} in conjunction with the (small-angle approximated) rotation matrix $\boldsymbol{R}_0$, this is also evaluated:

$$\vec{y}_0 = \begin{bmatrix} t_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{C.14}$$

$$\boldsymbol{R}_0 = \begin{bmatrix} 1 & u_1 - \cancelto{0}{u_0} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & u_1 \\ 0 & 1 \end{bmatrix} \tag{C.15}$$

$$\begin{aligned} \implies \boldsymbol{R}_0 \vec{y}_0 &= \begin{bmatrix} 1 & u_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cancelto{0}{(1)(0)} + \cancelto{0}{(u_1)(0)} \\ \cancelto{0}{(0)(0)} + \cancelto{0}{(1)(0)} \end{bmatrix} = \begin{bmatrix} 0 + 0 \\ 0 + 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned} \tag{C.16}$$

and

$$\begin{aligned} \begin{bmatrix} u_2 \\ t_2 \end{bmatrix} = \boldsymbol{R}_{\Theta_0} \begin{bmatrix} U_2 - U_0 \\ T_2 - T_0 \end{bmatrix} &= \begin{bmatrix} \cos\Theta_0 & -\sin\Theta_0 \\ \sin\Theta_0 & \cos\Theta_0 \end{bmatrix} \begin{bmatrix} U_2 - U_0 \\ T_2 - T_0 \end{bmatrix} \\ &= \begin{bmatrix} \cos\Theta_0 (U_2 - U_0) - \sin\Theta_0 (T_2 - T_0) \\ \sin\Theta_0 (U_2 - U_0) + \cos\Theta_0 (T_2 - T_0) \end{bmatrix} \end{aligned} \tag{C.17}$$

$$\implies \vec{y}_2 = \begin{bmatrix} t_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \sin\Theta_0 (U_2 - U_0) + \cos\Theta_0 (T_2 - T_0) \\ \Theta_2 - \Theta_0 \end{bmatrix} \tag{C.18}$$

APPENDIX D

TVS Notation, Algorithm, and Procedure Definitions

D.1 Definition of Terms

The list below defines the terms and mathematical notation used in describing the OTVS and NTVS algorithms:

- k : full iteration #, i.e., k-th iteration of superiorized FS.
- K : total # of full iterations, i.e., total # of iterations of superiorized FS.
- r : TV perturbation step #, $1 \leq r \leq N$.
- $\vec{x}^{(k)}$: image vector $\vec{x}$ at cycle k.
- $\bar{x}$: initial iterate $\vec{x}^{(0)}$ of image reconstruction.
- N: # of TV perturbation steps per FS iteration.
- α : perturbation kernel, $0 < \alpha < 1$.
- $\ell^{(k)}$: perturbation kernel exponent.
- $\beta^{(k,r)}$: perturbation (magnitude) coefficient $\beta^{(k,r)} = \alpha^{\ell^{(k)}}$ at TV perturbation step r of FS iteration k.
- ϕ : the cost function to which superiorization is applied; here, ϕ = TV, the TV of the image vector.
- $\phi(\vec{x}^{(k,r)})$: TV of image vector $\vec{x}^{(k,r)}$ at TV perturbation step r and FS iteration k.
- $v^{(k,r)}$: normalized non-ascending perturbation vector for ϕ at $\vec{x}^{(k,r)}$, i.e.,

$$v^{(k,r)} = -\frac{\nabla\phi(\vec{x}^{(k,r)})}{\left\|\nabla\phi(\vec{x}^{(k,r)})\right\|} = \phi'(\vec{x}^{(k,r)})$$

- P_T : projection operator representative of an iterative FS algorithm.

A pseudocode definition of the NTVS algorithm, written in terms of the notation defined in Appendix D.1, is as follows:

Algorithm 9 NTVS Algorithm

set $k = 0$
set $\ell_{-1} = 0$
set $\vec{x}^{(k)} = \bar{x}$
while $k < K$ **do**
    set $r = 0$
    set $\ell^{(k)} = \text{rand}(k, \ell_{k-1})$
    set $\vec{x}^{(k,r)} = \vec{x}^{(k)}$
    while $r < N$ **do**
        set $\vec{v}^{(k,r)} = \phi'(\vec{x}^{(k,r)})$
        set $\beta^{(k,r)} = \alpha^{\ell^{(k)}}$
        set $\vec{x}^{(k,r+1)} = \vec{x}^{(k,r)} + \beta^{(k,r)}\vec{v}^{(k,r)}$
        set $r = r + 1$
        set $\ell^{(k)} = \ell^{(k)} + 1$
    end while
    set $\vec{x}^{(k+1)} = P_T(\vec{x}^{(k,N)})$
    set $k = k + 1$
end while

FOR REFERENCE ONLY: A pseudocode definition of the NTVS algorithm with the TVRVS (TV reduction verification step) included:

Algorithm 10 NTVS Algorithm

set $k = 0$
set $\ell_{-1} = 0$
set $\vec{x}^{(k)} = \bar{x}$
while $k < K$ **do**
    set $r = 0$
    set $\ell^{(k)} = \text{rand}(k, \ell_{k-1})$
    set $\vec{x}^{(k,r)} = \vec{x}^{(k)}$
    while $r < N$ **do**
        set $\vec{v}^{(k,r)} = \phi'(\vec{x}^{(k,r)})$
        set $\beta^{(k,r)} = \alpha^{\ell^{(k)}}$
        set $loop = true$
        while *loop* **do**
            set $\vec{z}^{(k,r)} = \vec{x}^{(k,r)} + \beta^{(k,r)}\vec{v}^{(k,r)}$
            if $\phi(\vec{z}^{(k,r)}) \leq \phi(\vec{x}^{(k,r)})$ **then**
                set $\vec{x}^{(k,r)} = \vec{z}^{(k,r)}$
                set $loop = false$
            end if
            set $\ell^{(k)} = \ell^{(k)} + 1$
        end while
        set $r = r + 1$
    end while
    set $\vec{x}^{(k+1)} = P_T(\vec{x}^{(k,N)})$
    set $k = k + 1$
end while

Note that this version of the NTVS algorithm was only used to isolate and investigate the impact that including/excluding the TVRVS constraint has on image quality as a function of the other NTVS parameter values. Algorithm 10 above is not intended for use, it is only defined here to provide readers with the precise definition of the algorithm used to compare NTVS performance with and without the TVRVS constraint.

D.3 OTVS Algorithm

A pseudocode definition of the OTVS algorithm, translated from the notational form presented in previous publications and rewritten in terms of the same notation used to define the NTVS algorithm (Appendix D.1), is as follows:

Algorithm 11 OTVS Algorithm

set $k = 0$
set $\ell = 0$
set $\beta = 1$
set $\vec{x}^{(k)} = \bar{x}$
while $k < K$ **do**
    set $\vec{v}^{(k)} = \phi'(\vec{x}^{(k)})$
    set $loop = true$
    while $loop$ **do**
        set $\vec{z} = \vec{x}^{(k)} + \beta\vec{v}^{(k)}$
        if $\phi(\vec{z}) \leq \phi(\vec{x}^{(k)})$ **then**
            set $\vec{x}^{(k)} = \vec{z}$
            set $loop = false$
        end if
        set $\ell = \ell + 1$
        set $\beta = (\frac{1}{2})^{\ell}$ (originally $\beta \leftarrow \beta/2$)
    end while
    set $\vec{x}^{(k+1)} = P_T(\vec{x}^{(k)})$
    set $k = k + 1$
end while

D.4 Stepwise NTVS Procedure and Calculations

The TV-superiorized feasibility-seeking algorithms used in pCT are implemented by adding the following steps between each feasibility-seeking step:

(1) For each axial slice of the current iterate of the pCT solution vector $\vec{x}^{(k)}$, calculate the RSP differences, $\Delta_x(X, Y)$ and $\Delta_y(X, Y)$, of each voxel (X, Y) relative to its immediate neighbors in the positive orthogonal directions, respectively, as:

$$\Delta_x(X, Y) = \mathrm{RSP}(X + 1, Y) - \mathrm{RSP}(X, Y) \tag{D.1}$$

$$\Delta_y(X, Y) = \mathrm{RSP}(X, Y + 1) - \mathrm{RSP}(X, Y) \tag{D.2}$$

These are illustrated as green arrows in the following figure (Figure D.1).:

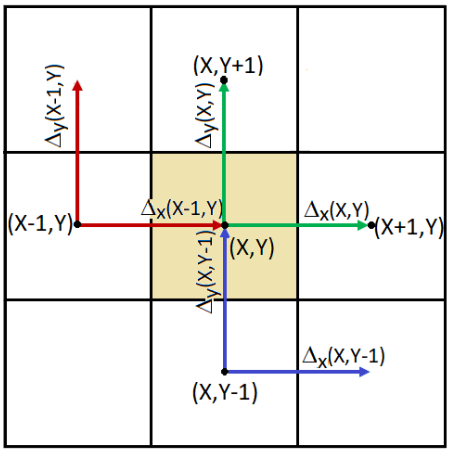


Figure D.1: Illustration of the ΔRSP differences, $\mathrm{RSP}_{\mathrm{tip}} - \mathrm{RSP}_{\mathrm{tail}}$, in the horizontal and vertical direction. The differences are normalized in Eqs. D.3 and D.4 and then used in Eq. D.5 to calculate the RSP variation of the central (shaded) pixel.

(2) For each voxel (X, Y), calculate the normalized differences δ_x and δ_y as:

$$\delta_x(X, Y) = \frac{\Delta_x(X, Y)}{\sqrt{\Delta_x^2(X, Y) + \Delta_y^2(X, Y)}} \tag{D.3}$$

$$\delta_y(X, Y) = \frac{\Delta_y(X, Y)}{\sqrt{\Delta_x^2(X, Y) + \Delta_y^2(X, Y)}} \tag{D.4}$$

(3) For each voxel (X, Y), calculate the RSP variation $V(\mathrm{X}, \mathrm{Y})$ as:

$$V(\mathrm{X}, \mathrm{Y}) = \delta_x(\mathrm{X\text{-}1}, \mathrm{Y}) + \delta_y(\mathrm{X}, \mathrm{Y\text{-}1}) - \left(\delta_x(\mathrm{X}, \mathrm{Y}) + \delta_y(\mathrm{X}, \mathrm{Y})\right) \tag{D.5}$$

(4) The variations of all pixels in the image set form components of the variation vector $\vec{V}^{(k)}$.

(5) The n-dimensional perturbation vector $\vec{v}^{(k)}$ is calculated by normalizing the RSP variation vector $\vec{V}^{(k)}$:

$$\vec{v}^{(k)} = \frac{\vec{V}^{(k)}}{\left\|\vec{V}^{(k)}\right\|} \tag{D.6}$$

(6) The N perturbations to be applied to the image are then calculated by multiplying the perturbation vector $\vec{v}^{(k)}$ by a scalar multiplier $\beta^{(k,r)}$ that determines the magnitude of each perturbation, yielding the following for each perturbation repetition r, $0 \leq r \leq N - 1$:

$$\vec{x}^{(k,r+1)} = \vec{x}^{(k,r)} + \beta^{(k,r)}\vec{v}^{(k)} \tag{D.7}$$

$$\text{where } \beta^{(k,r)} = \alpha^{\ell^{(k,r)}}, \tag{D.8}$$

$$\ell^{(k,r+1)} = \ell^{(k,r)} + 1 \tag{D.9}$$

and α is the perturbation kernel $0 < \alpha < 1$, and $\ell^{(k,r)}$ is the perturbation kernel exponent, where $\ell^{(0,0)} = 1$. Note that the perturbation vector $v^{(k)}$ remains constant for each repetition r and is only calculated once for each FS-DROP step k.

(7) Randomly decrease $\ell^{(k,N)}$ to an integer from the closed interval $[k, \ell^{(k,N)}]$ to generate the starting value of the next iteration $\ell^{(k+1,0)}$, i.e. $\ell^{(k+1,0)} = \mathrm{rand}\left(k, \ell^{(k,N)}\right)$

BIBLIOGRAPHY

[1] R. P. Johnson, "Review of medical radiography and tomography with proton beams," *Reports on Progress in Physics*, vol. 81, no. 1, p. 016701, 2018. [Online]. Available: http://stacks.iop.org/0034-4885/81/i=1/a=016701

[2] R. I. MacKay, "Image guidance for proton therapy," *Clin. Oncol. (R. Coll. Radiol.)*, vol. 30, no. 5, pp. 293–298, 2018. [Online]. Available: http://iopscience.iop.org/article/10.1088/1361-6633/aa8b1d/meta

[3] H. Paganetti, "Range uncertainties in proton therapy and the role of Monte Carlo simulations," *Physics in Medicine & Biology*, vol. 57, no. 11, pp. R99—-R117, May 2012.

[4] G. Coutrakon, J. Hubbard, J. Johanning, G. Maudsley, T. Slaton, and P. Morton, "A performance study of the loma linda proton medical accelerator," *Medical Physics*, vol. 21, pp. 1691–1701, 1994.

[5] J. Missaghian, R. F. Hurley, V. A. Bashkirov, B. Colby, V. Rykalin, S. Kachigiun, D. Fusi, R. W. Schulte, M. F. Martinez, H. F.-W. Sadrozinski, and S. N. Penfold, "Beam test results of a csi calorimeter matrix element," *JINST*, vol. 5, p. P06001, 2010.

[6] R. F. Hurley, R. W. Schulte, V. A. Bashkirov, G. Coutrakon, H. F.-W. Sadrozinski, and B. Patyal, "The phase i proton CT scanner and test beam results at llumc," *Transactions of the American Nuclear Society*, vol. 106, pp. 63–66, 2012.

[7] Y. Chen, E. Gomez, R. F. Hurley, Y. Nie, K. E. Schubert, and R. W. Schulte, "Accurate proton beam localization," in *Proceedings of The 2012 International Conference on Bioinformatics and Computational Biology (BIOCOMP'12)*, 2012, pp. 213–217.

[8] R. W. Schulte, V. A. Bashkirov, R. P. Johnson, H. F.-W. Sadrozinski, and K. E. Schubert, "Overview of the llumc/ucsc/csusb phase 2 proton CT project," in *Transactions of the American Nuclear Society*, vol. 106, 2012, pp. 59–62.

[9] H. F.-W. Sadrozinski, R. P. Johnson, S. Macafee, A. Plumb, D. Steinberg, A. Zatserklyaniy, V. A. Bashkirov, R. F. Hurley, and R. W. Schulte, "Development of a head scanner for proton CT," *Nuclear Instruments & Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 699, pp. 205–210, 2013.

[10] R. R. Wilson, "Radiological use of fast protons," *Radiology*, vol. 47, no. 5, pp. 487–491, 1946, pMID: 20274616.

[11] A. M. Cormack, "Representation of a function by its line integrals, with some radiological applications," *Journal of Applied Physics*, vol. 34, no. 9, pp. 2722–7, September 1963.

[12] ——, "Representation of a function by its line integrals, with some radiological applications. ii," *Journal of Applied Physics*, vol. 35, no. 10, pp. 2908–13, October 1964.

[13] K. M. Crowe, T. F. Budinger, J. L. Cahoon, V. P. Elischer, R. H. Huesman, and L. L. Kanstein, "Axial scanning with 900 mev alpha particles," *IEEE Transactions on Nuclear Science*, vol. 22, no. 3, pp. 1752–1754, June 1975.

[14] A. Cormack and A. Koehler, "Quantitative proton tomography: preliminary experiments," *Physics in Medicine & Biology*, vol. 21, no. 4, pp. 560–569, 1976. [Online]. Available: http://stacks.iop.org/0031-9155/21/i=4/a=007

[15] K. M. Hanson, J. N. Bradbury, T. M. Cannon, R. L. Hutson, D. B. Laubacher, R. Macek, M. A. Paciotti, and C. A. Taylor, "The application of protons to computed tomography," *IEEE Transactions on Nuclear Science*, vol. 25, no. 1, pp. 657–660, Feb 1978.

[16] K. M. Hanson, "Proton computed tomography," *IEEE Transactions on Nuclear Science*, vol. 26, no. 1, pp. 1635–1640, Feb 1979.

[17] K. M. Hanson, J. N. Bradbury, T. M. Cannon, R. L. Hutson, D. B. Laubacher, R. J. Macek, M. A. Paciotti, and C. A. Taylor, "Computed tomography using proton energy loss," *Physics in Medicine & Biology*, vol. 26, no. 6, pp. 965–983, 1981. [Online]. Available: http://stacks.iop.org/0031-9155/26/i=6/a=001

[18] U. Schneider and E. Pedroni, "Proton radiography as a tool for quality control in proton therapy," *Medical Physics*, vol. 22, no. 4, pp. 353–363, 04 1995. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.597470

[19] R. W. Schulte, S. N. Penfold, J. Tafas, and K. E. Schubert, "A maximum likelihood proton path formalism for application in proton computed tomography," *Medical Physics*, vol. 35, pp. 4849–4856, Nov 2008.

[20] V. A. Bashkirov, R. P. Johnson, H. F.-W. Sadrozinski, and R. W. Schulte, "Development of proton computed tomography detectors for applications in hadron therapy," *Nuclear Instruments & Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 809, pp. 120–129, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168900215009274

[21] R. P. Johnson, V. A. Bashkirov, L. DeWitt, V. Giacometti, R. F. Hurley, P. Piersimoni, T. E. Plautz, H. F.-W. Sadrozinski, K. E. Schubert, R. W. Schulte, B. E. Schultze, and A. Zatserklyaniy, "A fast experimental scanner for proton CT: Technical performance and first experience with phantom scans," *IEEE Transactions on Nuclear Science*, vol. 63, pp. 52–60, 2016.

[22] S. Rit, G. Dedes, N. Freud, D. Sarrut, and J. M. Létang, "Filtered backprojection proton CT reconstruction along most likely paths," *Medical Physics*, vol. 40, no. 3, p. 031103, 2013. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.4789589

[23] M. Bruzzi, C. Civinini, M. Scaringella, D. Bonanno, M. Brianzi, M. Carpinelli, G. Cirrone, G. Cuttone, D. L. Presti, G. Maccioni, S. Pallotta, N. Randazzo, F. Romano, V. Sipala, C. Talamonti, and E. Vanzi, "Proton computed tomography images with algebraic reconstruction," *Nuclear Instruments & Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 845, pp. 652–655, May 2017, Proceedings of the Vienna Conference on Instrumentation 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168900216304454

[24] G. Poludniowski, N. M. Allinson, and P. M. Evans, "Proton radiography and tomography with application to proton therapy," *The British Journal of Radiology*, vol. 88, p. 20150134, 06 2015. [Online]. Available: https://doi.org/10.1259/bjr.20150134

[25] B. E. Schultze, P. Karbasi, C. Sarosiek, G. Coutrakon, C. E. Ordoñez, N. T. Karonis, K. L. Duffin, V. A. Bashkirov, R. P. Johnson, K. E. Schubert, and R. W. Schulte, "Particle-tracking proton computed tomography—data acquisition, preprocessing, and preconditioning," *IEEE Access*, vol. 9, pp. 25 946–25 958, 2021.

[26] G. Cimmino, "Cacolo approssimato per le soluzioni dei systemi di equazioni lineari," *La Ricerca Scientifica (Roma)*, vol. 1, pp. 326–333, 1938.

[27] Y. A. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms and Applications*. USA: Oxford University Press, Inc., 1997.

[28] A. Smith, "Vision 20/20: proton therapy," *Medical Physics*, vol. 36, pp. 556–568, 2009.

[29] S. N. Penfold, "Image Reconstruction and Monte Carlo Simulations in the Development of Proton Computed Tomography for Applications in Proton Radiation Therapy," Ph.D. dissertation, University of Wollongong, Australia, 2010.

[30] S. N. Penfold, R. W. Schulte, Y. Censor, and A. B. Rosenfeld, "Total variation superiorization schemes in proton computed tomography image reconstruction," *Medical Physics*, vol. 37, pp. 5887–5895, 2010.

[31] M. Witt, B. E. Schultze, R. W. Schulte, K. E. Schubert, and E. Gomez, "A proton simulator for testing implementations of proton CT reconstruction algorithms on GPGPU clusters," in *Proceedings of the IEEE Nuclear Science Symposium & Medical Imaging Conference (NSS/MIC) 2012*, 2012, pp. 4329–4334.

[32] M. Bruzzi, N. Blumenkrantz, J. Feldt, J. Heimann, H. F.-W. Sadrozinski, A. Seiden, D. Williams, V. A. Bashkirov, R. W. Schulte, D. Menichelli, M. Scaringella, G. Cirrone, G. Cuttone, N. Randazzo, V. Sipala, and D. L. Presti, "Prototype tracking studies for proton CT," *IEEE Transactions on Nuclear Science*, vol. 54, pp. 140–145, Feb 2007.

[33] G. Cuttone, G. Cirrone, G. Candiano, F. D. Rosa, G. Russo, N. Randazzo, V. Sipala, S. L. Nigro, D. L. Presti, J. Feldt, J. Heimann, H. F.-W. Sadrozinski, A. Seiden, D. Williams, V. A. Bashkirov, R. W. Schulte, M. Bruzzi, and D. Menichelli, "Monte Carlo studies of a proton computed tomography system," *IEEE Transactions on Nuclear Science*, vol. 54, pp. 1487–1491, Oct 2007.

[34] C. Tschalär, "Straggling distributions of extremely large energy losses," *Nuclear Instruments & Methods*, vol. 61, pp. 141–156, 1968.

[35] International Commission on Radiation Units and Measurements, "Stopping powers and ranges for protons and alpha particles," *ICRU Report*, vol. 49, 1993.

[36] B. E. Schultze, M. Witt, K. E. Schubert, and R. W. Schulte, "Space carving and filtered back projection as preconditioners for proton computed tomography reconstruction," in *Proceedings of the IEEE Nuclear Science Symposium & Medical Imaging Conference (NSS/MIC) 2012*, 2012, pp. 4335–4340.

[37] G. Ramanchandran and A. Lakshminarayanan, "Three dimensional reconstructions from radiographs and electron micrographs: Application of convolution instead of Fourier transforms," *Proceedings of the National Academy of Sciences, USA*, vol. 68, pp. 2236–2240, 1971.

[38] L. Feldkamp, L. Davis, and J. Kress, "Practical cone-beam algorithms," *Journal of the Optical Society of America*, vol. A1, pp. 612–619, 1984.

[39] L. Shepp and B. Logan, "The Fourier reconstruction of a head section," *IEEE Transactions on Nuclear Science*, vol. NS-21, pp. 21–43, 1974.

[40] W. Niem, "Robust and fast modelling of 3D natural objects from multiple views," in *SPIE Proceedings Image and Video Processing*, vol. 2182, no. II, 1994, pp. 388–397.

[41] ——, "Error analysis for silhouette-based 3D shape estimation from multiple views," in *Proceedings of International Workshop on Synthetic-Natural Hybrid Coding and Three-Dimensional Imaging*, 1997, pp. 143–146.

[42] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," in *Proceedings of the Seventh International Conference on Computer Vision (ICCV)*, 1999, pp. 307–314.

[43] ——, "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, Marr Prize Special Issue 2000.

[44] S. Vedula, S. Baker, S. Seitz, and T. Kanade, "Shape and motion carving in 6d," in *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*, 2000.

[45] R. F. Hurley, R. W. Schulte, V. A. Bashkirov, A. Wroe, A. Ghebremedhin, H. F.-W. Sadrozinski, V. Rykalin, G. Coutrakon, P. Koss, and B. Patyal, "Water-equivalent path length calibration of a prototype proton CT scanner," *Medical Physics*, vol. 39, pp. 2438–2446, 2012.

[46] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, Nov 1986.

[47] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[48] X. Wu, "An efficient antialiasing technique," *Computer Graphics*, vol. 25, no. 4, p. 143–152, Jul. 1991. [Online]. Available: https://doi.org/10.1145/127719.122734

[49] M. Abrash, "Fast antialiasing," *Dr. Dobb's J.*, vol. 17, no. 6, p. 139–143, Jun. 1992.

[50] T. Li, Z. Liang, J. V. Singanallur, T. J. Satogata, D. C. Williams, and R. W. Schulte, "Reconstruction for proton computed tomography by tracing proton trajectories: A monte carlo study," *Medical Physics*, vol. 33, no. 3, pp. 699–706, 2006. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.2171507

[51] S. Penfold, R. Schulte, Y. Censor, V. Bashkirov, and A. Rosenfeld, "Characteristics of proton CT images reconstructed with filtered backprojection and iterative projection algorithms," in *IEEE Nuclear Science Symposium Conference Record*, 12 2009, pp. 4176 – 4180.

[52] U. Schneider and E. Pedroni, "Multiple coulomb scattering and spatial resolution in proton radiography," *Medical Physics*, vol. 21, no. 11, pp. 1657–1663, 1994. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.597212

[53] B. Gottschalk, A. Koehler, R. Schneider, J. Sisterson, and M. Wagner, "Multiple coulomb scattering of 160 mev protons," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 74, no. 4, pp. 467–490, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0168583X9395944Z

[54] D. C. Williams, "The most likely path of an energetic charged particle through a uniform medium," *Physics in Medicine and Biology*, vol. 49, no. 13, pp. 2899–2911, jun 2004. [Online]. Available: https://doi.org/10.1088/0031-9155/49/13/010

[55] S. N. Penfold, A. B. Rosenfeld, R. W. Schulte, and K. E. Schubert, "A more accurate reconstruction system matrix for quantitative proton computed tomography," *Medical Physics*, vol. 36, no. 10, pp. 4511–4518, Oct 2009.

[56] W. G. Horner, "A new method of solving numerical equations of all orders, by continuous approximation," *Philosophical Transactions of the Royal Society of London*, vol. 109, pp. 308–335, 1819. [Online]. Available: http://www.jstor.org/stable/107508

[57] B. E. Schultze, Y. Censor, P. Karbasi, K. E. Schubert, and R. W. Schulte, "An Improved Method of Total Variation Superiorization Applied to Reconstruction in Proton Computed Tomography," *IEEE Transactions on Medical Imaging*, vol. 39, no. 2, pp. 294–307, Apr 2019.

[58] B. E. Schultze, P. Karbasi, V. Giacometti, T. E. Plautz, K. E. Schubert, and R. W. Schulte, "Reconstructing highly accurate relative stopping powers in proton computed tomography," in *Proceedings of the IEEE Nuclear Science Symposium & Medical Imaging Conference (NSS/MIC) 2015*, Oct 2015, pp. 1–3.

[59] G. T. Herman, E. Garduño, R. Davidi, and Y. Censor, "Superiorization: An optimization heuristic for medical physics," *Medical Physics*, vol. 39, no. 9, pp. 5532–5546, 2012. [Online]. Available: http://dx.doi.org/10.1118/1.4745566

[60] Y. Censor, R. Davidi, and G. T. Herman, "Perturbation resilience and superiorization of iterative algorithms," *Inverse problems*, vol. 26, p. 65008, Jun 2010.

[61] Y. Censor, "Weak and strong superiorization: Between feasibility-seeking and minimization," *Analele Stiintifice ale Universitatii Ovidius Constanta, Seria Matematica*, vol. 23, pp. 41–54, Oct 2014.

[62] Y. Censor, R. Davidi, G. T. Herman, R. W. Schulte, and L. Tetruashvili, "Projected subgradient minimization versus superiorization," *Journal of Optimization Theory and Applications*, vol. 160, no. 3, pp. 730–747, Mar 2014. [Online]. Available: https://doi.org/10.1007/s10957-013-0408-3

[63] Y. Censor, G. T. Herman, and M. Jiang, "Superiorization: Theory and applications," Special Issue of *Inverse Problems*, vol. 33, no. 4, p. 040301, 2017. [Online]. Available: http://stacks.iop.org/0266-5611/33/i=4/a=040301

[64] Y. Censor and R. Mansour, "Convergence analysis of processes with valiant projection operators in hilbert space," *Journal of Optimization Theory and Applications*, vol. 176, no. 1, pp. 35–56, 2018.

[65] T. Humphries, J. Winn, and A. Faridani, "Superiorized algorithm for reconstruction of CT images from sparse-view and limited-angle polyenergetic data," *Physics in Medicine & Biology*, vol. 62, no. 16, pp. 6762–6783, 2017. [Online]. Available: http://stacks.iop.org/0031-9155/62/i=16/a=6762

[66] E. Helou, M. Zibetti, and E. Miqueles, "Superiorization of incremental optimization algorithms for statistical tomographic image reconstruction," *Inverse Problems*, vol. 33, no. 4, p. 044010, 2017. [Online]. Available: http://stacks.iop.org/0266-5611/33/i=4/a=044010

[67] E. Garduño and G. T. Herman, "Computerized tomography with total variation and with shearlets," *Inverse Problems*, vol. 33, no. 4, p. 044011, 2017. [Online]. Available: http://stacks.iop.org/0266-5611/33/i=4/a=044011

[68] Q. Yang, W. Cong, and G. Wang, "Superiorization-based multi-energy CT image reconstruction," *Inverse Problems*, vol. 33, no. 4, p. 044014, 2017. [Online]. Available: http://stacks.iop.org/0266-5611/33/i=4/a=044014

[69] Y. Censor, "Superiorization and perturbation resilience of algorithms: A bibliography compiled and continuously updated," http://math.haifa.ac.il/yair/bib-superiorization-censor.html.

[70] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock, "An introduction to total variation for image analysis," in *Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter*, 2010.

[71] S. N. Penfold and Y. Censor, "Techniques in iterative proton CT image reconstruction," *Sensing and Imaging*, vol. 16, no. 1, Oct 2015. [Online]. Available: https://doi.org/10.1007/s11220-015-0122-3

[72] S. N. Penfold, R. W. Schulte, Y. Censor, V. A. Bashkirov, S. A. McAllister, K. E. Schubert, and A. B. Rosenfeld, "Block-iterative and string-averaging projection algorithms in proton computed tomography image reconstruction," in *Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning and Inverse Problems*, Y. Censor, M. Jiang, and G. Wang, Eds., The Huangguoshu International Interdisciplinary Conference. Madison, WI, USA: Medical Physics, 2010, pp. 347–367.

[73] R. Bracewell and A. Riddle, "Inversion of fan beam scawns in radio astronomy," *Astrophysics Journal*, vol. 150, pp. 427–434, 1967.

[74] Y. Censor, "Can linear superiorization be useful for linear optimization problems?" *Inverse Problems*, vol. 33, no. 4, p. 044006, 2017. [Online]. Available: http://stacks.iop.org/0266-5611/33/i=4/a=044006

[75] G. T. Herman and R. Davidi, "Image reconstruction from a small number of projections," *Inverse Problems*, vol. 24, no. 4, p. 045011, 2008. [Online]. Available: http://stacks.iop.org/0266-5611/24/i=4/a=045011

[76] R. Davidi, G. T. Herman, and Y. Censor, "Perturbation-resilient block-iterative projection methods with application to image reconstruction from projections," *International Transactions in Operational Research*, vol. 16, no. 4, pp. 505–524, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-3995.2009.00695.x

[77] D. Butnariu, R. Davidi, G. T. Herman, and I. G. Kazantsev, "Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 540–547, Dec 2007.

[78] O. Langthaler, "Incorporation of the superiorization methodology into biomedical imaging software," Salzburg University of Applied Sciences, Salzburg, Austria, and the Graduate Center of the City University of New York, NY, USA, Marshall Plan Scholarship Report, September 2014, 76 pages.

[79] B. Prommegger, "Verification and evaluation of superiorized algorithms used in. biomedical imaging: Comparison of iterative algorithms with and without superiorization for image reconstruction from projections," Salzburg University of Applied Sciences, Salzburg, Austria, and the Graduate Center of the City University of New York, NY, USA, Marshall Plan Scholarship Report, October 2014, 84 pages.

[80] C. Havas, "Revised implementation and empirical study of maximum likelihood expectation maximization algorithms with and without superiorization in image reconstruction," Salzburg University of Applied Sciences, Salzburg, Austria, and the Graduate Center of the City University of New York, NY, USA, Marshall Plan Scholarship Report, October 2016, 49 pages.

[81] S. Agostinelli, J. Allison, K. Amako *et al.*, "Geant4 - a simulation toolkit," *Nuclear Instruments & Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250–303, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168900203013688

[82] V. Giacometti, V. A. Bashkirov, P. Piersimoni, S. Guatelli, T. E. Plautz, H. F.-W. Sadrozinski, R. P. Johnson, A. Zatserklyaniy, T. Tessonnier, K. Parodi, A. B. Rosenfeld, and R. W. Schulte, "Software platform for simulation of a prototype proton CT scanner," *Medical Physics*, vol. 44, no. 3, pp. 1002–1016, 2017. [Online]. Available: http://dx.doi.org/10.1002/mp.12107

[83] R. P. Johnson, V. A. Bashkirov, G. Coutrakon, V. Giacometti, P. Karbasi, N. T. Karonis, C. Ordoñez, M. Pankuch, H. F.-W. Sadrozinski, K. E. Schubert, and R. W. Schulte, "Results from a prototype proton-CT head scanner," *Conference on the Application of Accelerators in Research and Industry, CAARI 2016, 30 October - 4 November 2016, Ft. Worth, TX, USA*, Jul 2017. [Online]. Available: https://arxiv.org/pdf/1707.01580

[84] C. T. Rueden, J. Schindelin, M. C. Hiner, B. E. DeZonia, A. E. Walter, E. T. Arena, and K. W. Eliceiri, "Imagej2: Imagej for the next generation of scientific image data," *BMC Bioinformatics*, vol. 18, no. 1, p. 529, Nov 2017. [Online]. Available: https://doi.org/10.1186/s12859-017-1934-z

[85] G. Dedes, L. D. Angelis, S. Rit, D. Hansen, C. Belka, V. A. Bashkirov, R. P. Johnson, G. Coutrakon, K. E. Schubert, R. W. Schulte, K. Parodi, and G. Landry, "Application of fluence field modulation to proton computed tomography for proton therapy imaging," *Physics in Medicine & Biology*, vol. 62, no. 15, pp. 6026–6043, 2017. [Online]. Available: http://stacks.iop.org/0031-9155/62/i=15/a=6026

[86] G. Dedes, R. P. Johnson, M. Pankuch, N. Detrich, W. M. A. Pols, S. Rit, R. W. Schulte, K. Parodi, and G. Landry, "Experimental fluence modulated proton computed tomography by pencil beam scanning," *Medical Physics*, vol. 45, pp. 3287–3296, May 2018.

[87] B. E. Schultze, M. Witt, Y. Censor, K. E. Schubert, and R. W. Schulte, "Performance of hull-detection algorithms for proton computed tomography reconstruction," in *Infinite Products of Operators and Their Applications*, ser. Contemporary Mathematics, S. Reich and A. Zaslavski, Eds., vol. 636. American Mathematical Society, 2015, pp. 211–224.