Block-Iterative and String-Averaging Projection Algorithms in Proton Computed Tomography Image Reconstruction

S. N. Penfold R. W. Schulte Y. Censor V. Bashkirov
S. Macallister K. E. Schubert A. B. Rozenfeld

August 29, 2008

Abstract

Proton computed tomography (pCT) is an imaging modality that is based on tracking individual protons as they traverse the object to be imaged. Proton-by-proton tracking is necessary due to the effects of multiple Coulomb scattering (MCS), a process that deviates the proton path from a striaght line. If optimal spatial resolution is to be achieved, the path of each proton must be predicted with a maximum likelihood formalism that models MCS. Further, image reconstruction methods are required that are able to handle these non-linear paths. This had led to the exploration of algebraic reconstruction techniques (ART) in pCT. However, because iterative algebraic methods are computationally expensive, parallel compatible versions of the ART class, executed simultaneously over multiple processing units are required if pCT is to be realistic for a clinical environment. In this study we investigate the image quality achieveable with block-iterative and string-averaging projection algorithms in application to simulated pCT data. From the results we make a recommendation as to which algorithms should be used in future studies with pCT image reconstruction.

1 Introduction

Currently, proton therapy treatment plans are carried out using data from X-ray CT scans, an imaging modality that generates tomographical maps of scaled γ linear attenuation coefficients, commonly known as Hounsfield units. However, to perform the treatment planning, one requires knowledge of the spatial distribution of *electron density* within the patient. In clinical practice Hounsfield units are converted to electron densities through an empirically derived relationship generated from measurements with tissue equivalent materials [1]. The end result of this conversion is an error ranging from several millimetres up to more that 1 cm between the proton range calculated by the treatment planning software and the true proton range within the patient, depending on the anatomical region treated. Thus, becuse the Bragg peak depth cannot be accurately predicted, the inherent advantages of proton therapy are partially negated in such an approach.

Proton computed tomography (pCT) is an imaging modality that has been suggested as a means for reducing the uncertainty of Bragg peak location in proton radiation treatments. In pCT, the spatial location of individual protons pre and post patient, as well as the energy lost along the path is recorded [2]. The spatial measurements are employed in a maximum likelihood proton path formalism that models multiple Coulomb scattering within the patient [3], maximizing the spatial resolution. The corresponding energy loss measurements are converted to the integral relative electron density along this predicted path with the use of the Bethe-Bloch equation. By reconstructing many such events with an algebraic reconstruction technique capable of handling these non-linear paths, 3D electron density maps can be generated without the need for any empirical conversion. These images can then used in the treatment planning system to better predict proton energy loss within the patient at treatment time.

It has been demonstrated by previous pCT studies that superior spatial resolution can be achieved by employing algebraic reconstruction techniques in comparison to transform methods, such as filtered back-projection [4]. This is primarily because transform methods must assume the proton traveled along a straight path in the reconstruction volume. Algebraic techniques, however, are much more flexible, not only allowing proton paths to be non-linear but also permitting the inclusion of *a priori* knowledge about the object to be reconstructed. This flexibility comes at the expense of computation time, however, which is far greater for iterative techniques than that for transform methods.

If pCT is to be implemented in a clinical environment, fast image reconstruction times are required. It has been suggested that the image reconstruction process should take less than 15 minutes for treatment planning images and less than 5 minutes for pre-treatment patient position verification images [2]. Kaczmarz's algebraic reconstruction technique (ART) [5] has been implemented in previous pCT studies that have made use of an algebraic method of image reconstruction, displaying promising results [4]. However, the ART carries out image updates after each proton history and is therefore fundamentally serial, meaning that the speed of the reconstruction is completely dependent on the speed of the computer processing unit. As an example of the infeasibility of using ART in clinical practice we have recently observed that, using general purpose processing units, three dimensional images made up of a 256x256x48 voxel reconstruction volume, reconstructed with 10 million proton histories will take approximately 1.5 hours to complete a single cycle, with the optimal image often being reached after 3-4 cycles [6].

With the development of parallel computing, work has been dedicated to developing algebraic reconstruction algorithms that can be executed in parallel over multiple processors to enable fast algebraic reconstructions. This paper compares the performance, in terms of image quality, of a number of parallel compatible block-iterative and string-averaging algebraic reconstruction algorithms with simulated pCT projection data. Quantitative assessment of image quality is based on the *normalized mean absolute distance measure* described by Herman [17] and a qualitative note is made about image appearance. From these results recommendations are made on which image reconstruction algorithm should be used in future studies with pCT.

2 Reconstruction Algorithms

All of the algorithms discussed in this paper belong to the class of *projection methods.* These are iterative algorithms that use projections onto sets while relying on the general principle that when a family of (usually closed and convex) sets is present then projections onto the given individual sets are easier to perform than projections onto other sets (intersections, image sets under some transformation, etc.) that are derived from the given individual sets. This is definitely the case here where the sets to be projected on in the iterative process are the hyperplanes H_i defined by the i-th row of the $m \times n$ linear system $Ax = b$, namely,

$$H_i = \{x \in \Re^n \mid \langle a^i, x \rangle = b_i, \text{ for } \ i = 1, 2, \ldots, m. \tag{1}$$

Here $\Re^n$ is the Euclidean n-dimensional space and a^i is the i-th column vector of A^T (the transpose of A), i.e., its components a^i_j occupy the i-th row of A. The right-hand side vector is $b = (b_i)_{i=1}^m$. In pCT, the a^i_j correspond to the length of intersection of the ith proton history with the jth picture element, x is the relative electron density image vector we are attempting to find and b_i is the integral relative electron denisty corresponding the energy lost by the ith proton along its path.

2.1 Fully Sequential Algebraic Reconstruction Technique

The ART is a sequential projections method for the solution of linear systems of equations of the form $Ax = b$ originally published by Kaczmarz in 1937 [5]. It is obtained also by applying to the hyperplanes, described by each equation of the linear system, the method of successive projections onto convex sets. The latter is called in the literature POCS (for "projections onto convex sets") or SOP (for "successive orthogonal projections") and was originally published by Bregman [7] and further studied by Gubin, Polyak and Raik [8].

Given the control sequence $\{i\,(k)\}_{k=0}^{\infty}$ where $i\,(k) = mod\,(k, m) + 1$ and m is the number of proton histories used in the algorithm, the general scheme for the ART is as follows.

Algorithm 1 *Kaczmarz's Algebraic Reconstruction Technique*

Initialization: $x^0 \in \Re^n$ *is arbitrary.*

Iterative Step: Given x^k, *compute the next iterate* x^{k+1} *by*

$$x^{k+1} = x^k + \lambda_k \frac{b_{i(k)} - \langle a^{i(k)}, x^k \rangle}{\| a^{i(k)} \|^2} a^{i(k)} \tag{2}$$

where $\{\lambda_k\}_{k=0}^{\infty}$ *is a sequence of user-determined relaxation parameters.*

As mentioned the ART is a fully sequential method, and thus cannot be executed in parallel. It is used in this investigation as a standard for comparison.

2.2 Block-Iterative Algorithms

All block-iterative algorithms to be analysed in this paper can be derived from the generalized block-iterative projections (BIP) method of Aharoni and Censor [9]. The BIP method allows the processing of blocks (i.e., groups of hyperplanes H_i) which need not be fixed in advance, but could change dynamically throughout the cycles. The number of blocks, their sizes, and the assignments of the hyperplanes H_i to the blocks may all vary, provided that the weights attached to the hyperplanes fulfill a simple technical condition, which can be described as follows.

Let $I = \{1, 2, ..., m\}$, and let $\{H_i | i \in I\}$ be a finite family of hyperplanes with nonempty intersection $H = \cap_{i \in I} H_i$. Denoting the nonnegative ray of the real line by R_+, introduce a mapping $w : I \to R_+$, called a weight vector, with the property $\sum_{i \in I} w\,(i) = 1$. A sequence $\{w^k\}_{k=0}^{\infty}$ of weight vectors is called *fair* if, for every $i \in I$, there exists infinitely many values for k for which $w^k\,(i) > 0$.

2.2.1 Block-Iterative Projections Algorithm

Given a fair weight vector w, define the convex combination $P_w(x) = \sum_{i \in I} w(i) P_i(x)$, where $P_i(x)$ is the orthogonal projection of x onto the hyperplane H_i. The general scheme for the BIP technique is as follows.

Algorithm 2 *Block-Iterative Projections (BIP)*

Initialization: $x^0 \in \Re^n$ *is arbitrary.*

Iterative Step: Given x^k, compute the next iterate x^{k+1} by

$$x^{k+1} = x^k + \lambda_k \left(P_{w^k}(x^k) - x^k \right), \tag{3}$$

where $\{w^k\}_{k=0}^{\infty}$ is a fair sequence of weight vectors and $\{\lambda_k\}_{k=0}^{\infty}$ is a sequence of user-determined relaxation parameters.

The block-iterative algorithmic structure stems from the possibility to have at each iteration k some (but of course not all) of the components $w^k(i)$, for some of the indices i, of the weight vector w^k equal to 0. A block-iterative version with fixed blocks is obtained from Algorithm 2 by partitioning the indices of I as $I = I_1 \cup I_2 \cup \cdots \cup I_M$ into M blocks and using weight vectors of the form

$$w^k = \sum_{i \in I_{t(k)}} w^k(i) e^i, \tag{4}$$

where e^q is the q-th standard basis vector (with 1 in its q-th component and zeros elsewhere) and $\{t(k)\}_{k=0}^{\infty}$ is a control sequence over the set $\{1, 2, \ldots, M\}$ of block indices. In this case, and incorporating the expressions for the orthogonal projections P_i onto the hyperplanes H_i into the formula, the iterative step (3) of Algorithm 2 takes the form

$$x^{k+1} = x^k + \lambda_k \left(\sum_{i \in I_{t(k)}} w^k(i) \, \frac{b_i - \langle a^i, x^k \rangle}{\parallel a^i \parallel^2} a^i \right), \tag{5}$$

where $\{t(k)\}_{k=0}^{\infty}$ is a cyclic (or almost cyclic) control sequence on $\{1, 2, \ldots, M\}$. While the generality of the definition of a fair sequence of weight vectors permits variable block sizes and variable assignments of hyperplanes into the blocks that can be used, equal hyperplane weighting and constant block sizes were used in the implementation of BIP in this investigation.

2.2.2 Block-Iterative Component Averaging Algorithm

The block-iterative component averaging (BICAV) algorithm, presented by Censor *et al.* in 2001 [10], is a special case of Algorithm 2 in that it incorporates *component related* weighting in the vector w^k. BICAV also differs in the method of projection onto the indivdual hyperplanes, making use of generalized *oblique projections* as opposed to orthogonal projections. For a detailed discussion of the consequences of this on the projection algorithm see [10]. The iterative step in BICAV is defined in (6).

Algorithm 3 *Block-Iterative Component Averaging (BICAV)*

Initialization: $x^0 \in \Re^n$ *is arbitrary.*

Iterative Step: Given x^k*, compute the next iterate* x^{k+1} *by using for* $j = 1, 2, ..., n$

$$x_j^{k+1} = x_j^k + \lambda_k \sum_{i \in I_{t(k)}} \frac{b_i - \langle a^i, x^k \rangle}{\sum_{l=1}^n s_l^{t(k)} (a_l^i)^2} a_j^i, \tag{6}$$

where $\{s_l^t\}_{l=1}^n$ *is the number of non-zero elements in the lth column of all the* a^i*'s in the current data block and* $\{\lambda_k\}_{k=0}^\infty$ *is a sequence of user-determined relaxation parameters.*

2.2.3 Diagonally Relaxed Orthogonal Projections Algorithm

Recently, Censor *et al.* [11] derived a component averaging technique that made use of orthogonal projections onto hyperplanes rather than the generalized oblique projections employed in the BICAV algorithm. This new algorithm was called diagonally relaxed orthogonal projections (DROP). The structure of DROP is outlined in Algorithm 4.

Algorithm 4 *Diagonally Relaxed Orthogonal Projections (DROP)*

Initialization: $x^0 \in \Re^n$ *is arbitrary.*

Iterative Step: Given x^k*, compute the next iterate* x^{k+1}

$$x^{k+1} = x^k + \lambda_k U \sum_{i \in I_{t(k)}} \frac{b_i - \langle a^i, x^k \rangle}{\|(a^i)\|^2} a^i, \tag{7}$$

where $U = diag(1/\tau_j)$ *where* $\tau_j = \max\{s_j^t | t = 1, 2, ..., T\}$ *with* s_j^t *as defined in Algorithm 3, and* $\{\lambda_k\}_{k=0}^\infty$ *is a sequence of user-determined relaxation parameters.*

Both the DROP and BICAV algorithms are more computationally expensive than the BIP method because of the need to calculate the s_j^t's prior to any image updates. However, it is the goal of component dependent weighting to markedly improve the convergence rate of the algorithm, compensating for time spent on extra calculations.

2.2.4 Ordered Subsets Simultaneous Algebraic Reconstruction Technique

At around the same time as Kaczmarz published his fully sequential algebraic reconstruction technique, Cimmino introduced a fully *simultaneous* algebraic alternative [12]. Anderson and Kak went on to further develop this fully simultaneous technique and published the simultaneous algebraic reconstruction technique (SART) [13]. Although the fully simultaneous SART was found to deal well with noisy images, it was also found to demonstrate slow convergence rates. Fortunately, the algorithm was developed in such a way that it was equally applicable to subsets, or blocks, of data as it was to the complete data set. This block-iterative form was first called ordered subsets simultaneous algebraic reconstruction technique (OS-SART) in [20]. The algorithmic scheme for OS-SART is as follows.

Algorithm 5 *Ordered Subsets Simulataneous Algebraic Reconstruction Technique (OS-SART)*

Initialization: $x^0 \in \Re^n$ *is arbitrary.*

Iterative Step: Given x^k*, compute the next iterate* x^{k+1} *by using for* $j = 1, 2, ..., n$

$$x_j^{k+1} = x_j^k + \lambda_k \frac{1}{\sum_{i \in I_{t(k)}} a_j^i} \sum_{i \in I_{t(k)}} \frac{b_i - \langle a^i, x^k \rangle}{\sum_{j=1}^n a_j^i} a_j^i, \tag{8}$$

where $\{\lambda_k\}_{k=0}^{\infty}$ *is a sequence of user-determined relaxation parameters.*

2.3 String-Averaging Algorithms

In contrast to the block-iterative algorithmic scheme, the string-averaging scheme (see [14]) dictates that, from the current iterate x^k, sequential successive projections be performed along the strings and then the end-points of all strings be combined with a weighted convex combination. In other words, each operation *within* a string must be executed serially, but each string end-point can be calculated in parallel. Firstly, let us intoroduce the string notation. For $t = 1, 2, ..., M$, let the *string* I_t be an ordered subset of $1, 2, ..., m$ of the form

$$I_t = (i_1^t, i_2^t, ..., i_{m(t)}^t), \tag{9}$$

with $m(t)$ the number of elements in I_t. Suppose that there is a set $S \subseteq \Re^n$ such that there are operators $R_1, R_2, ..., R_m$ mapping S into S and an operator R which maps S^m into S.

Algorithm 6 *String-Averaging Algorithmic Scheme*

Initialization: $x^0 \in S$ *is arbitrary.*

Iterative Step: Given x^k*, calculate, for all* $t = 1, 2, ..., M$*,*

$$T_t x^{(k)} = R_{i_{m(t)}^t} ... R_{i_2^t} R_{i_1^t} x^{(k)}, \tag{10}$$

and then calculate

$$x^{(k+1)} = R(T_1 x^{(k)}, T_2 x^{(k)}, ..., T_M x^{(k)}) \tag{11}$$

For every $t = 1, 2, ..., M$ this algorithmic scheme applies to $x^{(k)}$ successively the operators whose indices belong to the tth string. This can be done in parallel for all strings and then the operator R maps all end-points onto the next iterate $x^{(k+1)}$.

2.3.1 String-Averaging Projections Algorithm

In order to arrive at the iterative algorithmic structure for the string-averaging orthogonal projection algorithm, we must define the following. For $i = 1, 2, ..., m(t)$, the operation $R_i x = x + \lambda_i(P_i x - x)$, where P_i is the orthogonal projections onto the hyperplane H_i. Then, to combine the strings we use $R(x^1, x^2, ..., xM) = \sum_{t=1}^{M} w_t x^t$, with $w^k > 0$ for all $t = 1, 2, ..., M$, and $\sum_{t=1}^{M} w_t = 1$. The iterative structure thus becomes,

Algorithm 7 *String-Averaging Projections (SAP)*

Initialization: $x^0 \in \Re^n$ *is arbitrary.*

Iterative Step: Given x^k*, for each* $t = 1, 2, ..., M$ *set* $y_0 = x^k$ *and calculate, for* $i = 0, 1, ..., m(t) - 1$,

$$y_{i+1} = y_i + \lambda_i \frac{b_i - \langle a^i, y_i \rangle}{\| a^i \|^2} a^i \tag{12}$$

and let $y^t = y_{m(t)}$ *for each* $t = 1, 2, ..., M$*. Then, calculate the next iterate by*

$$x^{(k+1)} = \sum_{t=1}^{M} w_t y^t \tag{13}$$

Similarly to the block-iterative algorithms, each string end-point was assigned equal weighting in this investigation.

2.3.2 Component-Averaged Row Projections Algorithm

In a similar vain to the introduction of component related weighting into block-iterative projection algorithms, so to was the component-averaged row projections (CARP) method developed for string-averaging algorithms [15]. Now, we calculate in how many strings the jth pixel was found to be intersected. In other words, s_j^t is now the number of strings which contain at least one equation with a nonzero coefficient of x_j. The algorithmic scheme for the CARP algorithm can be given as,

Algorithm 8 *Component-Averaged Row-Action Projections (CARP)*

Initialization: $x^0 \in \Re^n$ *is arbitrary.*

Iterative Step: Given x^k, for each $t = 1, 2, ..., M$ set $y_0 = x^k$ and calculate, for $i = 0, 1, ..., m(t) - 1$,

$$y_{i+1} = y_i + \lambda_i \frac{b_i - \langle a^i, y_i \rangle}{\parallel a^i \parallel^2} a^i \tag{14}$$

and let $y^t = y_{m(t)}$ for each $t = 1, 2, ..., M$. Then, calculate the next iterate by

$$x_j^{(k+1)} = \frac{1}{s_j^t} \sum_{t=1}^{M} y_j^t \tag{15}$$

Although all the algorithms above have mentioned that the associated relaxation parameter (λ) may be a sequence of user chosen values, in this study we will only consider the case of constant relaxation parameter. The data will be subdivided into 180, 60 and 12 blocks (or strings) of equal size and the optimal relaxation parameter for each block size found. The optimal relaxation parameter is defined to be the value that returns the best image quality within ten complete cycles. Note that an iteration refers to the update of the image while a cycle is a complete run through of m proton histories.

Image quality is not a well defined concept. The quality of an image depends on the purpose for which the image is generated. In the case of pCT images, visual appearance is important so that structures can easily be identified in the treatment planning process. Also, the actual values of the digitized picture are of equal importance as it is these values that are used to calculate dose deposition by the treatment planning software. Since it is difficult to quantitatively evaluate image appearance, we will base our analysis of image quality on how close the values of the reconstructed images are to the test phantom. Herman has described two such measures for this purpose; the *normalized root mean squared distance measure* and *normalized mean absolute distance measure* [17]. Only the results of the latter, hereby referred to as the relative error, will be presented in this paper as both measures were found to undergo very similar trends.

3 Proton CT Reconstruction Process

While the section above outlines the algorithmic structure of the iterative steps to be investigated in the various algebraic methods of reconstruction, these algorithms are but one section of the overall pCT reconstruction process. The overall process can be broken into the sub-routines listed below.

1. Loading the measured data in list mode.

2. Binning the individual proton histories based on their exit location for each projection angle.

3. Analysing the exit angle and exit energy of protons within each bin. This allows us to determine which protons should be excluded by the 3σ exit angle and exit energy cuts.

4. Determining where the object boundary is. This information is then passed to the most likely path (MLP) formalism to allow for a more accurate calculation of proton path approximation. In this work, the object boundary was calculated by performing an intitial run through of the data with the direct summation method described by Herman and Rowland [16], and by simplifying the proton path to a straight line. This initial image is used for the object boundary only, as the actual pixel values calculated with this method are quite erroneous. See Fig. 2(b) for an example of how well the object boundary is defined with the direct summation method.

5. Begin the iterative process. We proceed through each proton history in list mode. Only the paths of those histories found to have suitable exit angle and exit energy are calculated. If the proton was found to pass through the object, the MLP formalism is employed, if not, a straight line is used.

4 Proton CT Simulations

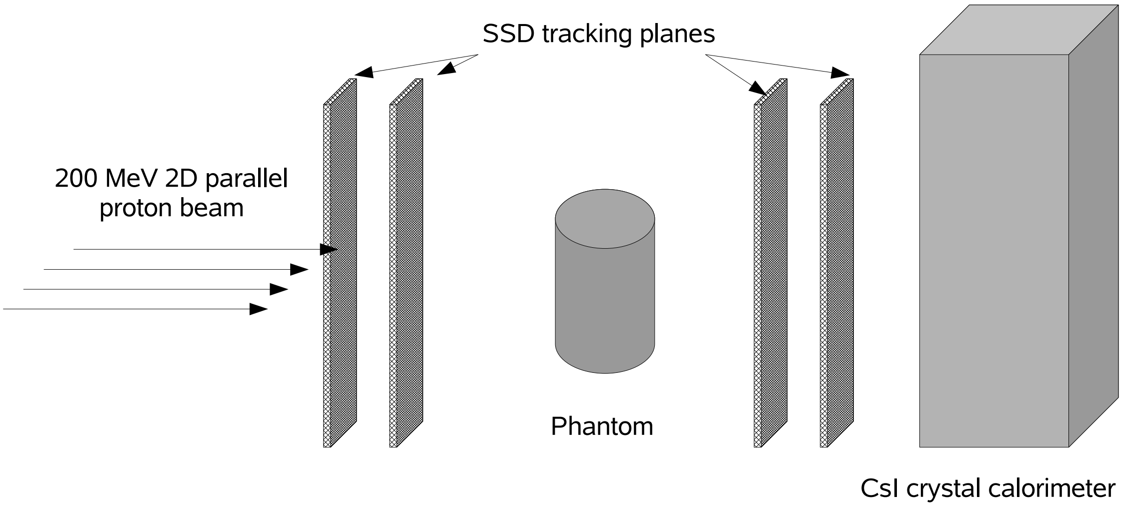


Figure 1: Schematic of the proton CT system modelled by the GEANT4 simulation.

A GEANT4 [18] application was created to model an ideal pCT parallel beam system. The proton beam consisted of a 200 MeV monoenergetic 2D parallel geometry. Two upstream and two downstream 2D sensitive silicon tracking planes 30cm x 30cm x 0.04 cm in size were located at -30 cm, -25 cm, 25 cm and 30 cm along the axis of the beam, relative to the centre of the phantom. All tracking sensitive volumes were allocated a pitch of 0.2 mm. To accurately record proton exit energy a 32 cm x 32 cm x 12 cm CsI crystal calorimeter was placed downstream of the tracking modules. The face of the crystal was positioned 5 cm behind the second exiting tracking module (Fig. 1). An ellipsoidal cylindrical phantom, based on the head phantom design of Herman [17], was located at the centre of the imaging system. A cross-section of the phantom can be seen in Fig. 2(a).

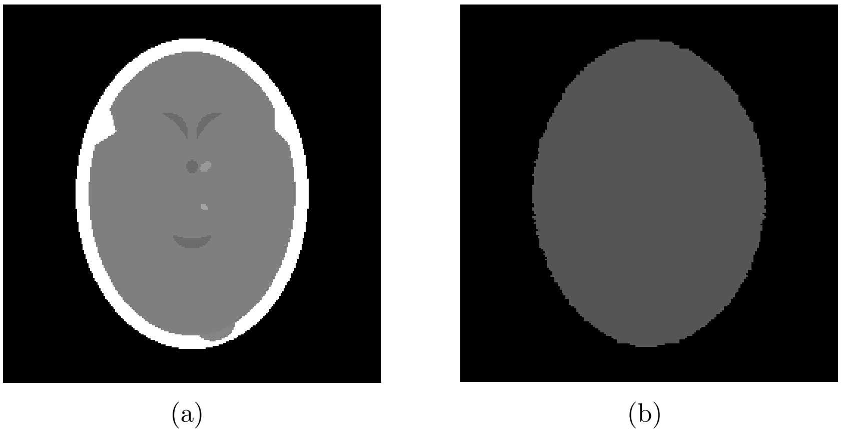


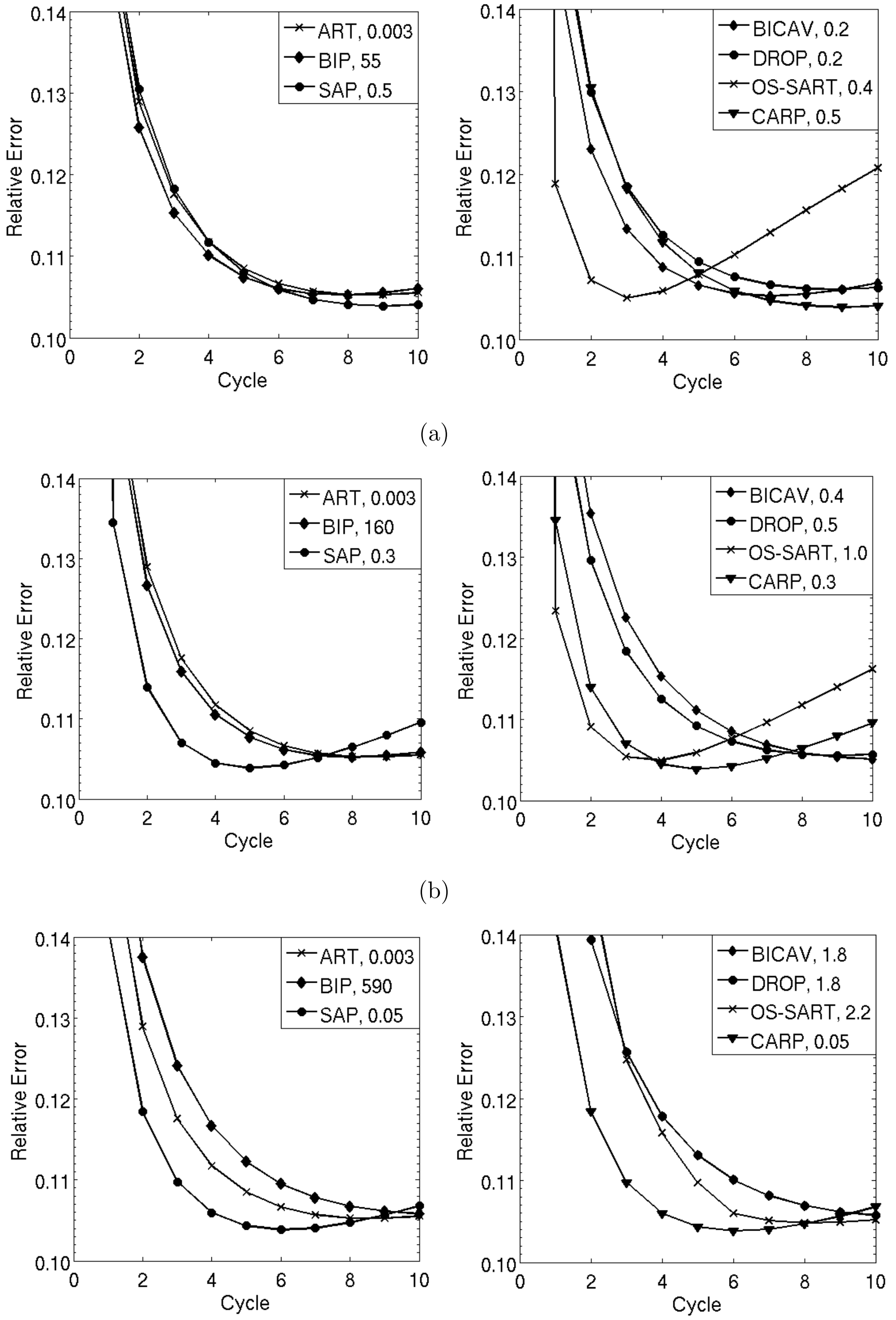
Figure 2: a) Cross-section of the phantom used in the GEANT4 simulation. The different regions all have the same chemical composition but varying physical density. b) Object boundary definition by the direct summation method.

A total of 180 projections were carried out at two degree intervals with the first 20000 protons that were found to traverse the geometry and deposit energy in the CsI scintillator being recorded in each projection angle. Three σ cuts on the exit angle and exit energy were also implemented, the motivation for which is described elsewhere [3]. The low energy electromagnetic and low energy hadronic physics process were used as the basis for the interactions to be considered in the simulation.

5 Results

In this section we will illustrate how image quality is affected by not only the choice of reconstruction algorithm, but also the number of blocks or strings into which the raw data is subdivided. We only analyse images up until the completion of the tenth cycle, as any more than this will likely result in an image reconstruction time too large for clinical practicality.

In Figure 3 the relative error as a function of cycle number is plotted for each algorithm with the data subdivided into 180, 60 and 12 blocks of equal size (with the exception of ART

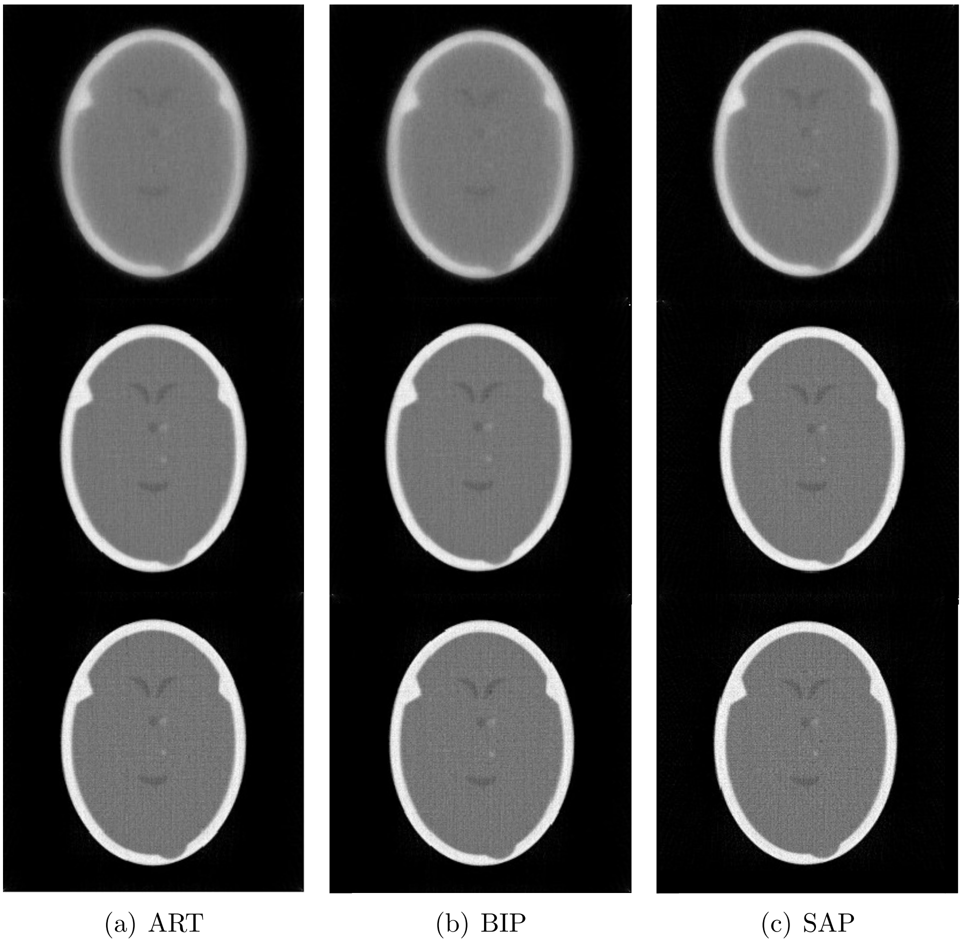


(a)

(b)

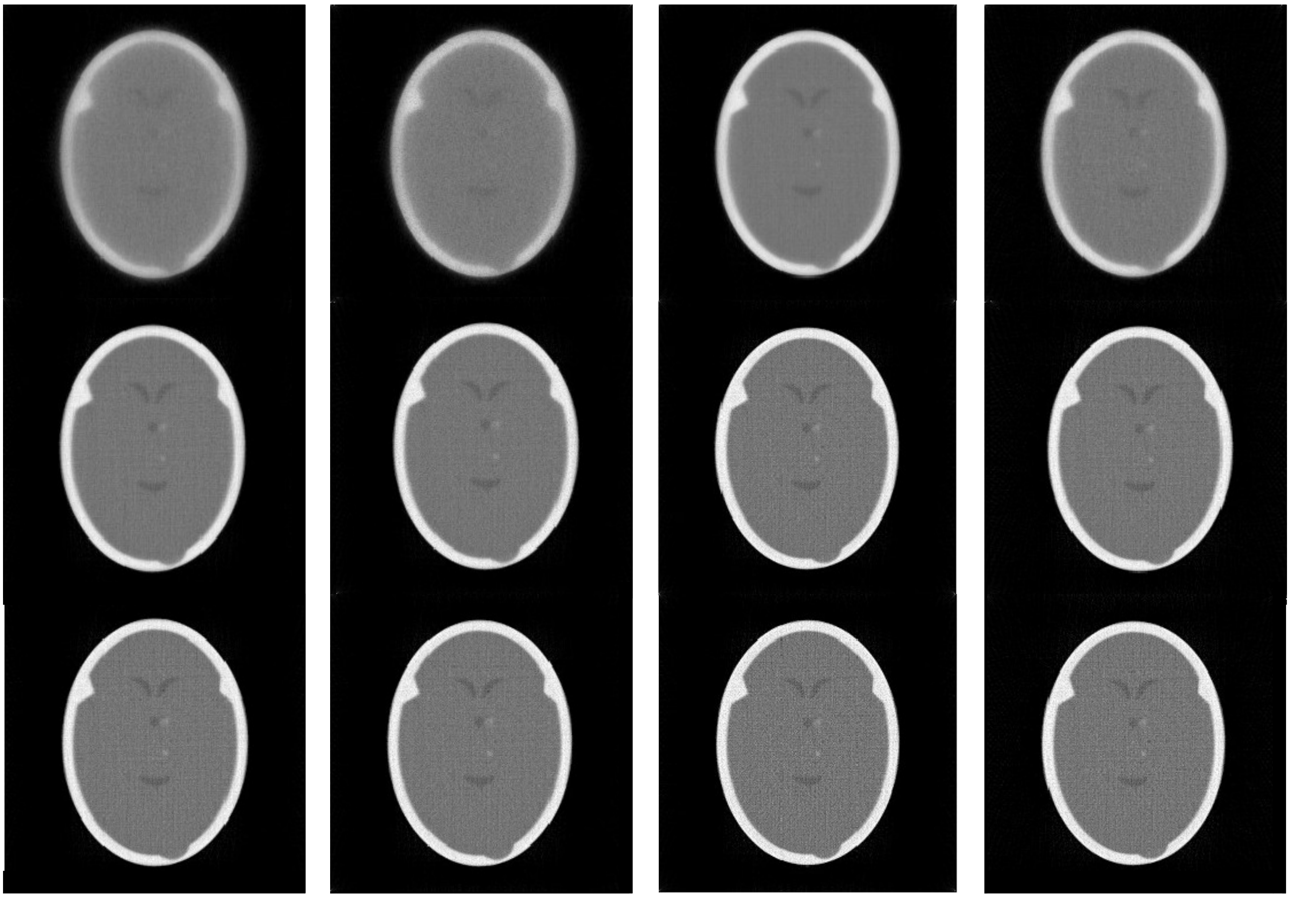
(c)

Figure 3: Relative error as a function of cycle number for all algorithms tested. The left hand column contains the ART and component independent algorithms BIP and SAP, while the right hand column contains the component dependent algorithms BICAV, DROP, OS-SART and CARP. The data was divided into a) 180 blocks, b) 60 blocks and c) 12 blocks. In each case ART is plotted for comparative purposes and was not divided into the aforementioned blocks. The number next to each algorithm in the legends correspond the relaxation parameter that resulted in the smallest relative error within ten cycles.



(a) ART (b) BIP (c) SAP

Figure 4: Reconstructed images after 1 (top row), 5 (middle row) and 10 (bottom row) cycles with ART and component independent parallel compatible algorithms with the data divided into 60 blocks



(a) BICAV (b) DROP (c) OS-SART (d) CARP

Figure 5: Reconstructed images after 1 (top row), 5 (middle row) and 10 (bottom row) cycles with component dependent parallel compatible algorithms with the data divided into 60 blocks

which is fully sequential). The left column contains the fully sequential ART and component *independent* block-iterative and string-averaging algorithms (Algorithms 1, 2 and 7), while the right-hand column contains the component *dependent* algorithms (Algorithms 3, 4, 5 and 8).

It can be seen that with 180 blocks ART, BIP and SAP all perform similarly. Of these BIP displays a slight advantage in terms of convergence speed but the SAP algorithm ultimately reaches the lowest relative error. If the component dependent algorithms are considered, we see that OS-SART with 180 blocks has the most rapid convergence of any algorithm tested, reaching a minimum relative error after only 3 cycles. However, the string-averaging component dependent algorithm CARP reaches a lower relative error than any block-iterative algorithm.

Considering the relative error calculated with 60 blocks, of the component independent algorithms, SAP displays both the minimum relative error and the most rapid convergence. Indeed, while the minimum relative error for SAP has remained constant between reconstructions with 180 and 60 blocks, the minimum is now observed to occur after 5 cycles as opposed to 9 with 180 blocks. Notice also that extreme over-relaxation is required for the BIP algorithm to achieve a competetive convergence rate. This is due to the fact that the weighting factor in Algorithm 2 is $<<1$ when equal weighting is assigned to each proton history. Once again, OS-SART displays the most rapid convergence of all the algorithms tested but now only reaches its minimum relative error one cycle before the CARP algorithm. The latter is again found to produce the minimum relative error of the algorithms tested.

With the data divided into 12 blocks, the string-averaging algorithms SAP and CARP once again display a smaller minimum relative error than any of the block-iterative algorithms. It can be seen that the relationship between number of blocks and convergence rate is the inverse for string-averaging and block-iterative algorithms. Block-iterative algorithms converge faster when the data is divided into a larger number of blocks, meaning that there are more sequential steps. String-averaging algorithms converge more rapidly with a smaller number of blocks, once again, when there are more sequential steps. This may suggest that ART should display the fastest convergence since every proton history is operated on in sequence. However, it appears that the parallel operations of the block-iterative and string-averaging algorithms are superior at dealing with noise, allowing for better image quality.

The images produced by each reconstruction algorithm with 60 blocks and optimal relaxation parameter after 1 (top), 5 (middle) and 10 (bottom) cycles are shown in Fig's. 4 and 5. In Fig. 4 it can be seen that while the smallest relative error may occur in images closer to ten cycles, the phantom appears “smoother”at around five cycles, perhaps providing better visualization. The images in Fig. 4, however, display little visual difference between algorithms.

The component dependent algorithms in Fig. 5 display a somewhat more varied visual appearance. While BICAV and DROP produce very similar images, the more rapid convergence to minimum relative error exhibited by OS-SART and CARP with 60 data blocks is reflected in the appearance of the images. OS-SART in particular is able to produce a smooth image with distinguishable features after only a single cycle.

6 Discussion

Using data acquired with a GEANT4 program simulating the current generation pCT system design, the application of various parallel compatible projection algorithms to pCT image reconstruction was tested. The block-iterative and string-averaging algorithms investigated not only provide the possibility of greatly improving the image reconstruction time through parallel execution, but the results in this study suggest that a choice of any one of these would enable images of superior quality to be produced, in comparison to the currently used ART algorithm.

A major reason for this superior performance is that pCT data is inherently noisy. It has been observed that simultaneous algorithms are able to cope better with noisy data than sequential methods [19], but have the disadvantage of slow convergence rates. Therefore, the block-iterative and string-averaging algorithms that contain both simultaneous and sequential operations are ideal for pCT image reconstruction.

The results from this study suggest that the string-averaging algorithms are able to produce superior image quality, in terms of matching the digitized phantom, than block-iterative methods. The results also show that the choice of block size is extremely important to obtain the best possible image quality in the smallest number of cycles. We have demonstrated that when subdividing the data for the string-averaging algorithm, one should choose a block size that is not so large that the number of sequential operations is so numerous that noise becomes an issue, but not so small that the convergence rate suffers. This actual choice of M of course depends on how many histories are to be used in the reconstruction process.

It can also be seen from the results that component dependent weighting has little effect on the string averaging algorithm. Indeed, SAP and CARP display identical results. This is because the method of weighting suggested in [15] and implemented here is based on the number of blocks in which the particular pixel was intersected by a proton history. Since there are a huge number of proton histories in each block, all corresponding to an equation in the $Ax = b$ imaging problem (far more equations in pCT than xCT), nearly all pixels are intersected in each block. This reduces the weighting system of SAP and CARP to be approximately equal.

One draw-back of all the projection algorithms discussed in this study is the need to "guess"an optimal relaxation parameter. For all the algorithms, with the exception of the BIP method which required extreme over-relaxation, the optimal relaxation parameter was found anywhere between 0 and 2 (slightly more for OS-SART with 12 data blocks), depending on the number of blocks chosen and method used. It has also been observed elsewhere that the optimal relaxation parameter is dependent on the size of the reconstruction volume, and the quality and amount of acquired data [10]. In future work we hope to present a method that takes the guess work out of choosing the relaxation parameter for string-averaging projection algorithms, also leading to rapid convergence.

Furthermore, we believe that these parallel compatible algorithms can be modified to further improve the handling of noisy pCT data. The primary factors that contribute to the noise in pCT data are

1. The statistical nature of energy loss when traversing an object.

2. The inability to calculate the exact path of the proton through the object.

3. The noise associated with the detector system itself.

We have begun an investigation into incorporating the method of projections onto *hyperslabs* as opposed to hyperplanes for string-averaging and block-iterative projection algorithms. Hyperslabs allow for the modelling of the uncertainties in the measured data described above, which is of particular relevance to pCT reconstructions.

It was the purpose of this study to examine the performance of the various parallel compatible projection algorithms in terms of image quality. The time saving advantage of the algorithms was not demonstrated here. A joint investigation will go on to examine the computational efficiency of the algorithms that were found to provide the best image quality in this study. The algorithms will be executed on general purpose graphical processing units (GPGPU's), an exciting new direction in image reconstruction hardware.

7 Conclusion

Image reconstruction in proton CT has two major issues to overcome before it can become a clinical reality; time and image noise. The block-iterative and string-averaging projection algorithms investigated in this paper provide a method for combatting both. The parallel compatible nature means that execution on a local area network or parallel GPGPU's would speed up the image reconstruction process considerably, producing images in clinically pratical amounts of time. Also, the combination of simultaneous and sequential operations means that convergence rates are superior to fully simultaneous algorithms and can handle noisy data better than fully sequential methods. The results of this paper suggest that string-averaging methods can achieve superior image quality in comparison to block-iterative algorithms. Further, component dependent weighting was found to have minimal effect in the string-averaging approach meaning there is little advantage in using the computationally more expensive CARP algorithm in comparison to SAP. The block-iterative OS-SART algorithm displayed the most rapid convergence, however, and should not be overlooked.

References

[1] A. A. Mustafa and D. F. Jackson, "The relation between x-ray CT numbers and charged particle stopping powers and its significance for radiotherapy treatment planning", Phys. Med. Biol., **28**, 169-176 (1983).

[2] R. Schulte *et al.*, "Conceptual design of a proton computed tomography system for applications in proton radiation therapy", IEEE Trans. Nuc. Sci., **51**, 866-872 (2004).

[3] R. W. Schulte, S. N. Penfold, J. T. Tafas, K. E. Schubert, "A maximum likelihood proton path formalism for application in proton computed tomography", Med. Phys., *accepted.*

[4] T. Li *et al.*, "Reconstruction for proton computed tomography by tracing proton trajectories: A Monte Carlo study", Med. Phys., **33**, 699-706 (2006).

[5] S. Kaczmarz, "Angenäherte Auflösung von Systemen linearer Gleichungen", *Bulletin de l'Académie Polonaise des Sciences et Lettres,* **A35**, 355-357 (1937).

[6] V. Bashkirov, R. Schulte, G. Coutrakon, B. Erdelyi, K. Wong, H. Sadrozinski, S. Penfold, A. Rosenfeld, S. McAllister, K. Schubert, "Development of proton computed tomography for applications in proton therapy", CAARI Conf. Rec. (2008).

[7] L. M. Bregman, "The method of successive projections for finding a common point of convex sets", Soviet Mathematics Doklady, **6**, 688-692 (1965).

[8] L. Gubin, B. Polyak and E. Raik, "The method of projections for finding the common point of convex sets", USSR Computational Mathematics and Mathematical Physics, **7**, 1-24 (1967).

[9] R. Aharoni and Y. Censor, "Block-iterative projection methods for parallel computation of solutions to convex feasibility problems", Linear Algebra and Its Applications, **120**, 165-175 (1989).

[10] Y. Censor, D. Gordon and R. Gordon, "BICAV: A block-iterative parallel algorithm for sparse systems with pixel-related weighting", IEEE Trans. Med. Imag., **20**, 1050-1059 (2001).

[11] Y. Censor, T. Elfving, G. T. Herman and T. Nikazad, "On diagonally-relaxed orthogonal projection methods", SIAM Journal on Scientific Computing, **30**, 473-504 (2008).

[12] G. Cimmino, "Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari", La Ricerca Scientifica, **XVI**, Series II, Anno IX, 326-333 (1938).

[13] A. H. Andersen and A. C. Kak, "Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm", Ultrasonic Imaging, **6**, 81-94 (1984).

[14] Y. Censor, T. Elfving and G. T. Herman, "Averaging strings of sequential iterations for convex feasibility problems", Inherently Parallel Algorithms in Feasibility and Optimization nd Their Applications, D. Butnariu, Y Censor and S Reich (Ed), Elsevier Science Publications, Amsterdam, The Netherlands, 101-114 (2001).

[15] D. Gordon and R. Gordon, "Component-averaged row projections: A robust block-parallel scheme for sparse linear systems", SIAM J. Sci. Comput., **27**, 1092-117 (2005).

[16] G. T. Herman, S. Rowland, "Three methods for reconstructing objects from x-rays: a comparative study", Computer Graphics and Image Processing, **2**, 151-178 (1973).

[17] G. T. Herman, "Image reconstruction from projections: The fundamentals of computerized tomography", Academic Press, New York, NY, USA (1980).

[18] S. Agostinelli *et al.*, "GEANT4-a simulation toolkit", Nuc. Instrum. Meth. Phys. Res. A, **506**, 250-303 (2003).

[19] G. T. Herman, "A relaxation method for reconstructing objects from noisy X-rays", Mathematical Programming, **8**, 1-19 (1975).

[20] M. Jiang and G. Wang, "Development of iterative algorithms for image reconstruction", J. X-ray Science Technology, **10**, 77-86 (2001).