DOI: 10.1784/insi.2023.65.9.XXX

Volume integral model for algebraic image reconstruction and computed tomography

R Hanna, M Sutcliffe, D Carswell, P Charlton and S Mosey

Industrial computed tomography (CT) has seen widespread adoption as an inspection technique due to its ability to resolve small defects and perform high-resolution measurements on complex structures. The reconstruction of CT data is usually performed using filtered back-projection (FBP) methods, such as the Feldkamp-Davis-Kress (FDK) method, and are selected as they offer a good compromise between reconstruction time and quality. More recently, iterative reconstruction algorithms have seen a resurgence in research interest as computing power has increased. Iterative reconstruction algorithms, such as the algebraic reconstruction technique (ART), use a reconstruction approach based on linear algebra to determine voxel attenuation coefficients based on the measured attenuation of the sample at the detector and calculation of the ray paths traversing the voxel grid. This offers a more precise model for CT reconstruction but at the cost of computational complexity and reconstruction time. Existing ART implementations are based on the 2D weighting models of the binary integral method (BIM), line integral method (LIM) and area integral method (AIM). For full 3D reconstruction, BIM and LIM only offer approximations leading to numerical inaccuracies. AIM for 2D reconstruction is mathematically exact but considers the divergent nature of a fan beam for 2D only. For a full 3D volumetric reconstruction, the X-ray cone beam is divergent in all directions and therefore AIM cannot be applied in its current form. A novel voxel weighting method for 3D volumetric image reconstruction using ART and providing a mathematically exact fractional volume weighting is introduced in this paper and referred to as the volume integral method (VIM). A set of algorithms is provided based on computer graphics techniques to determine ray/voxel intersections with volume reconstruction computed based on the divergence theorem. A set of experimental configurations is developed to provide a comparison against existing methods and conclusions are provided. Optimisation is achieved through graphic acceleration.

1. Introduction

The use of X-ray computed tomography (CT) has led to image reconstruction challenges as the applications have become more varied and data sizes have increased. The innovation of cone-beam CT allows data to be acquired faster but presents a challenge for the reconstruction process due to the divergence of the cone beam. Analytic reconstruction techniques such as filtered back-projection (FBP) and the 3D Feldkamp-Davis-Kress (FDK) algorithm[1] are limited by geometric transformations used to approximate a parallel-beam acquisition[2]. This is due to the use of the Fourier slice theorem to perform the reconstruction. The FDK algorithm is less computationally intensive than other methods and is well suited to scenarios with a sufficient number of equally spaced projections over a large rotational range.

Alternative methods of cone-beam reconstruction include iterative methods based on the algebraic reconstruction technique (ART) originally proposed by Gordon, Bender and Herman[3]. ART has been demonstrated for image reconstruction by Hounsfield[4] during the first CT scans. While iterative techniques offer accurate solutions to the image reconstruction problem without geometric transformations and approximations, they come at greater computational cost. Recent technological advancements have allowed for general-purpose computing on graphics processing units (GPUs)[5,6], enabling high levels of parallelisation to occur during code execution, utilising a brute-force approach to achieve computational speed.

As iterative techniques such as ART utilise a discretised method for calculations in the reconstruction, voxel weighting coefficients are implemented to increase the accuracy. The simplest of these is the binary integral method (BIM)[7], in which any voxel that is in the path of the X-ray beam contributes its full attenuation coefficient to the ray sum (a value of 1). A more accurate technique is the use of the line integral method (LIM)[8], which assigns a weight to a voxel attenuation coefficient based on the line length of the X-ray beam as it traverses a given voxel (a normalised value between 0 and 1). Both of these methods provide a high level of reconstruction but do not accurately model the system. In 2D reconstruction environments, the area integral method (AIM) has been demonstrated to further improve accuracy for fan-beam reconstruction. AIM considers the X-ray source as a point source and the detector pixel as an object with physical dimensions, meaning the X-ray beam is treated as a divergent entity. This allows a weighting model to be determined based on the X-ray beam area intersection through a given voxel (again normalised to lie between 0 and 1)[9-11].

Submitted 29.09.22 / Accepted 14.06.23

Ross Hanna is with the Wales Institute for Science and Art (WISA), University of Wales Trinity Saint David.

Mark Sutcliffe and David Carswell are with TWI Ltd, VAL.

Peter Charlton is with the School of Engineering, Manufacturing & Logistics, University of Wales Trinity St David.

Stephen Mosey is with the School of Applied Computing, University of Wales Trinity St David.

AIM is limited to 2D reconstructions and to obtain a full model of the system the volume intersection through a given voxel for the X-ray beam should be considered. However, this introduces computational complexity and has not been widely researched within the literature. This paper introduces a full volume integral method (VIM) based on determining X-ray beam intersections by exploiting 2D and 3D computer graphics techniques. The X-ray source is considered a point in 3D space and the detector pixel as an object in 3D space. Connecting X-ray source and pixel detector coordinates leads to a region in 3D space bounded by a square-based pyramid, referred to as a frustum[12]. The intersection of a given voxel and the frustum provides a fractional volume for a voxel, which is used as the weighting coefficient. Methods are provided within this paper for computing intersections and voxel volume for efficient reconstruction.

2. Algebraic reconstruction technique

ART was initially developed for electron microscopy by Gordon[3] and has since been used for CT reconstruction[13]. ART offers higher numerical accuracy with the ability to perform accurate reconstructions with sub-optimal acquisitions, such as limited-angle and limited-projection datasets. The ART algorithm attempts to provide a solution that satisfies a sum of voxels in the path of an X-ray beam, such that the computed value of the sum of voxels equals the value at the detector pixel. This is repeated for all X-ray beam paths and voxel intersections. Previously computed individual voxel values are exploited by iteratively applying the algorithm until the reconstruction volume has converged.

Implementation is a two-step process: forward projection and correction. The forward projection involves a ray traversing the grid and a summation of every voxel that is intersected by the ray to calculate the ray sum. The ray sum is compared to the detector pixel value and the difference is applied to each of the intersected voxels based on a weighted correction modelling a single voxel's contribution to the ray sum. This is illustrated in Figure 1 for LIM.

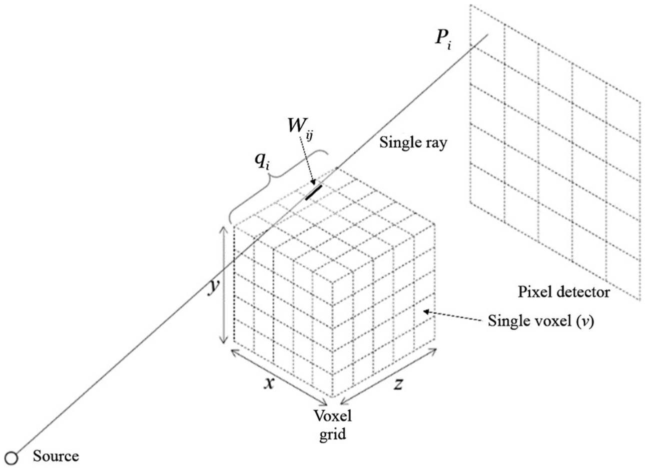


Figure 1. ART coordinate system used in this work

The precise definition for ART is given in Equation (1):

$$v^{(k+1)}{}_j = v^{(k)}{}_j + \lambda \frac{p_i - q_i}{\sum_n w_{in}{}^2} w_{ij} 0 \text{..................} (1)$$

where v is the voxel attenuation coefficient, w is the weight coefficient, k is the iteration number, j is the voxel index value, i is the ray index value and n is the voxel index value for a specific ray. The weighting coefficient assigned to a particular voxel is denoted by w_{ij}, w_{in} denotes an individual voxel weight along a ray, λ is the relaxation factor (described later), p_i is the value of a detector pixel and q_i is the calculated ray sum, based on the previous iteration's calculation of voxel values. This is expressed in Equation (2):

$$q_i = \sum_n w_{in} \,.\, v^{(k)}{}_n \text{..............................} (2)$$

An alternative to the ART algorithm is the simultaneous ART (SART) algorithm presented by Kak and Andersen[14]. In contrast to the update of the cell value on a beam-by-beam basis in ART, the SART algorithm updates the cells based on the average correction value for a projection. The advantage of the SART algorithm is a reduction in the reconstructed image noise when compared to the ART algorithm, at the expense of speed of convergence. The precise formula for SART is:

$$v_j^{(k+1)} = v_j^k + \frac{\sum_{i=1}^N \lambda \frac{P_i - q_i}{\sum_{n=1} w_{in}}}{\sum_{i=1}^N w_{ij}} w_{ij} \text{.................} (3)$$

While the presented method can be applied to both ART and SART, the ART algorithm was chosen for its higher convergence speed.

3. Volume integral method

As discussed, VIM provides a weighting coefficient technique based on the volume intersection between the X-ray beam and the voxel. In computer graphics, the frustum is a four-sided pyramid representing the visible region of the screen[15], with its use in determining what portions of the 3D world should be rendered. The apex of the pyramid represents the camera, the base the projection plane and the sides the limits of the viewing angle. For X-ray CT, this is analogous to the region occupied from the X-ray source to a pixel within the detector of the X-ray CT system. Representing the problem in this way allows for common computer graphics techniques to be exploited. This system is therefore limited to only two object types: axis-aligned bounding boxes (AABBs) representing the voxels and the frustum representing the projection beam. The objective is to determine the volume occupied by a beam intersecting a voxel. To achieve this, it is first necessary to compute a polyhedron representing the intersected region of the voxel.

There are three possible ways in which the voxel and beam may interact: any of the four frustum edges may intersect a voxel face (eight intersection points); any voxel edge (12 in total) may intersect with one of the four frustum planes; and any voxel corner (eight in total) may exist within the body of the frustum. From these three scenarios a maximum of 64 points are evaluated representing the geometry of the newly intersected volume. This is illustrated in Figure 2.

Computing the points of beam/voxel intersections allows for a new polyhedral shape to be constructed. The VIM weight is the normalised volume of this newly constructed object. Calculation of the volume weight is described in Section 3.4. First, methods to compute the intersections are given.

3.1 Ray/voxel face intersection

One of the fastest and computationally efficient methods for performing ray/AABB line intersection is the slab method[16]. Given a bounding box as illustrated in Figure 3 (in this example in 2D), a box is positioned in 2D space with known corner coordinates (shown here as b^0 to b^3) as well as the intersecting ray start and end positions (r^0 and r^1). The ray intersection points are not yet known (r^2 and r^3).

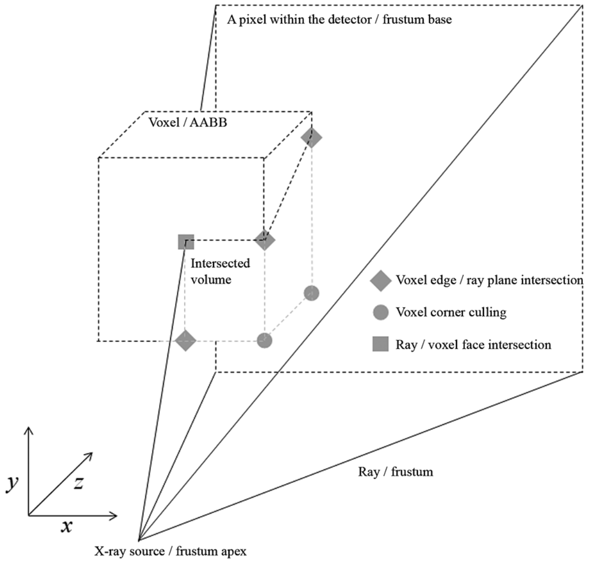

A pixel within the detector / frustum base
Voxel / AABB
Intersected volume
Voxel edge / ray plane intersection
Voxel corner culling
Ray / voxel face intersection
Ray / frustum
y
z
x
X-ray source / frustum apex

Figure 2. Illustration of the coordinate system used in this work and the types of ray/voxel interaction that may occur

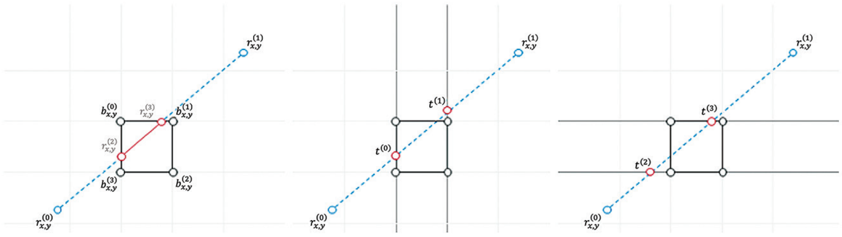


Figure 3. Illustration of the slab line intersection method. Left: box and line; middle: solve for left and right; right: solve for top and bottom

Two vector positions for t are evaluated for a change in the x-coordinate with respect the left and right bounding box positions. This is repeated for a change in y with respect to the top and bottom box positions.

This provides four vector positions (t), which are evaluated as:

$$t^{(0)} = \frac{b_x^{(0)} - r_x^{(0)}}{r_x^{(1)} - r_x^{(0)}}, \quad t^{(1)} = \frac{b_x^{(1)} - r_x^{(0)}}{r_x^{(1)} - r_x^{(0)}}$$
$$t^{(2)} = \frac{b_y^{(3)} - r_y^{(0)}}{r_y^{(1)} - r_y^{(0)}}, \quad t^{(3)} = \frac{b_y^{(0)} - r_y^{(0)}}{r_y^{(1)} - r_y^{(0)}} \quad \text{.............. (4)}$$

Values of t that satisfy the intersection points of the box are evaluated from each minimum and maximum pair:

$$t^{(4)} = \max\left(t^{(0)}, t^{(2)}\right) \text{ .... (5)}$$

$$t^{(5)} = \min\left(t^{(1)}, t^{(3)}\right) \text{............................ (6)}$$

Intersection values for the ray in Cartesian coordinates are calculated as:

$$r_x^{(2)} = r_x^{(0)} + \left(r_x^{(1)} - r_x^{(0)}\right)t^{(4)}, \quad r_y^{(2)} = r_y^{(0)} + \left(r_y^{(1)} - r_y^{(0)}\right)t^{(4)}$$
$$r_x^{(3)} = r_x^{(0)} + \left(r_x^{(1)} - r_x^{(0)}\right)t^{(5)}, \quad r_y^{(3)} = r_y^{(0)} + \left(r_y^{(1)} - r_y^{(0)}\right)t^{(5)} \text{ .... (7)}$$

Extending this algorithm to 3D is trivial and in the case of a frustum instead of a single ray the voxel is represented by the bounding box and the frustum edge by the ray intersection.

For each edge of the frustum the slab method is computed and intersection values obtained as 3D coordinates. Each intersection yields two sets of coordinates: the entry point and the exit point. In the case of a non-intersection a false or null value is returned. This provides a maximum of eight intersection points (two intersection points for each of the four frustum edges).

3.2 Voxel edge/ray plane intersection

The next way in which voxel intersection may occur is where one or more of the four frustum planes intersects with one or more edges of the voxel. There are four planes to check against the twelve edges of the voxel. Figure 4 illustrates a voxel/frustum plane intersection, where a, b and c are the points representing the frustum plane coordinates and d and e are start and end points for one edge of a voxel. To compute the intersection point p, the surface normal of the plane n (computed from the cross product of b-a and c-a) and any single point on the plane surface f are used with the plane intersection algorithm:

$$p = d - (e - d)\frac{(d - f) \cdot n}{(e - d) \cdot n} \text{ .... (8)}$$

The output is a point along a voxel edge (represented here as d and e) where the intersection occurs. Intersection is guaranteed for anything other than edges parallel to the plane. Voxel edge/ray plane intersection provides a maximum of 48 intersection points (12 voxel edges and four frustum planes). Again, the algorithm yields a false or null if no intersection point is evaluated.

3.3 Voxel corner culling

Having computed the intersections, one further check is needed before a new polyhedron can be constructed. Frustum culling is required for all eight corners of the voxel, where any voxel vertex identified as being inside the body of the frustum is added to the list of intersection points while all others are discarded (culled). This can be achieved by examining each plane of the frustum, computing its plane normal and evaluating if the corner point being checked is in the direction of the plane normal, as shown in Figure 5. This is repeated for all planes of the frustum. If all planes return true and given that the frustum is a convex polygon, the corner point must lie within the body of the frustum. All other corner points are discarded. This provides a further eight possible points for consideration.

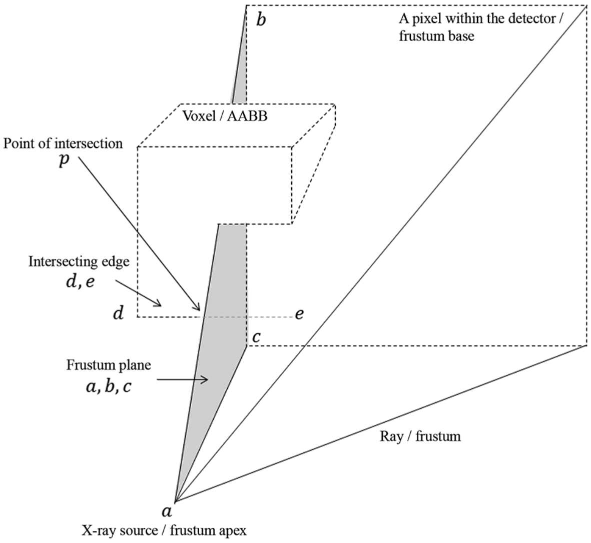


Figure 4. Illustration of the voxel edge/frustum plane intersection

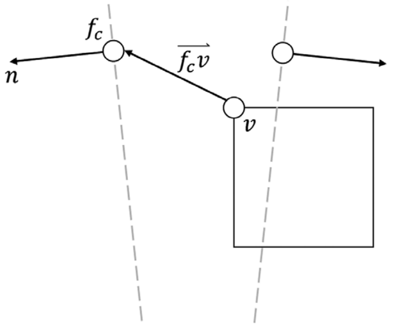


Figure 5. Diagram of frustum corner culling

To determine if a voxel corner is in the direction of the plane normal, the dot product of the normal with the vector from the voxel corner v to the face centre f_c is computed:

$$n \cdot \overrightarrow{f_c v} \leq 0 \ldots\ldots\ldots\ldots (9)$$

If this value is negative, then the angle between n and $\overrightarrow{f_c v}$ is ≥90° so the corner is in the direction of the normal. If this check is true for all five frustum faces, then it is included as part of the polyhedron to be constructed.

3.4 Voxel volume construction

The output of the previous steps provides a possible 64 points with which a new shape (a polyhedron) may be constructed. This new shape will consist of up to ten faces (six based on the existing voxel faces and four based on the plane intersections). Prior to shape construction it is first necessary to sort the vertices on each face, which is carried out in an anticlockwise order. This sorts the vertices in a radial order and the face may then be reconstructed using the triangle strip method[17]. Repeating this for each face allows for generation of a polyhedron representing the volume intersection.

Since a voxel has been intersected with a frustum, the volume region is a polyhedron and its volume is calculated using Equation (10):

$$\text{Volume} = \frac{1}{3}\sum_{F} A(F)\, x_F \,.\, n_F \ldots\ldots\ldots\ldots (10)$$

where x_F is an arbitrary point on face F and n_F is the outward unit normal to face F, x_F. n_F is the dot product and A is the area of a face. Note that this formula is a result of applying the divergence theorem to a polyhedron[18].

In order to apply this equation, the area of each face, the outward unit normal to each face and one point on each face must be known. Determining one point on the face is straightforward, since all of the vertices are known. The normals are also trivial to calculate since six of the potential faces were the faces of the voxel. The directions of the normals to the other four potential faces can be calculated as the cross-product of the vectors from the source to the two ends of the pixel edge (note that it is necessary to take the cross product in the correct sense to give the outward normal not the inward normal). This is then scaled to make it unit length. The formula for the area (A) of a polygon in the x,y plane is:

$$A = \frac{1}{2}\sum_{j=1}^{n}(x_j y_{j+1} - x_{j+1} y_j) \ldots\ldots\ldots\ldots (11)$$

where (x_j, y_j) are the x, y coordinates, taken in order, of the corners of the polygon and j are interpreted modulo n (the number of corners on the face). Note that there must be at least three corners to define a proper face. This formula can be derived by applying Green's theorem[19] in the plane (*ie* the divergence theorem in 2D). Using Equation (10), the volume of the region of intersection between the voxel and the X-ray beam can be computed. The final stage is to normalise it to the maximum possible value by dividing by three (where voxel dimensions are 1 unit cubed in size). This gives the normalised volume weight (a value between 0 and 1), as shown in Figure 6.

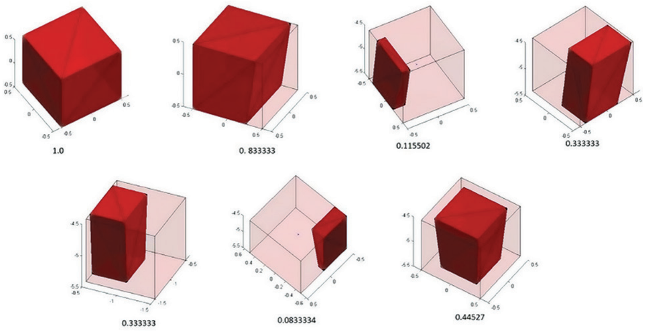


Figure 6. Examples of VIM weights in 3D, where the weight corresponds to a normalised volume intersection measurement

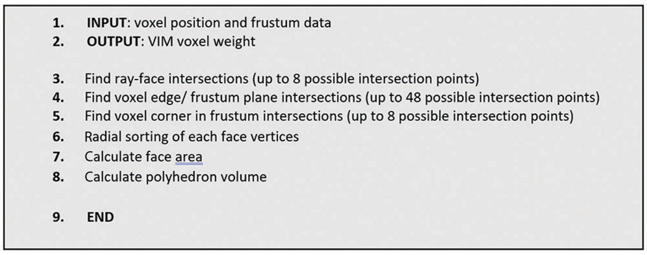
3.5 GPU computation

The use of a general-purpose graphics processing unit (GPGPU) for use with non-destructive testing (NDT) imaging algorithms[20] has shown much success and parallel processing of ART has been demonstrated for biomedical imaging[21]. While computationally efficient algorithms have been provided here for VIM calculation, the number of instructions required to determine the weight coefficients for all voxels and projections is still large. To assist with this, GPGPU acceleration was utilised to further improve computational time. Due to the iterative nature of the ART algorithm, parallelisation is carried out for each iterative step, as data dependencies prohibit multiple iterations being executed in parallel.

The fast voxel traversal algorithm[22] was implemented alongside VIM to determine which voxels to use to compute VIM weight coefficients. These algorithms combined with the ART summation were carried out in parallel and executed on a GPGPU. Floating-point memory is assigned to accommodate three sets of voxel data on the GPU: the current iteration answer; the previous iteration answer; and the weight correction coefficients. Each detector pixel (voxel intersection ray) is assigned its own compute unified device architecture (CUDA) processing thread. To minimise memory requirements, each ray intersection is evaluated dynamically.

3.6 Computational complexity

For a reconstruction algorithm using the well-known LIM approach, the proposed VIM can be substituted without modification to the underlying algorithm. The implication of this is that the number of calculations of the voxel weight is equal between the two weighting methods. The number of calculations required for the VIM is higher than the lightweight VIM technique; however, with 64 possible intersections to determine compared to two for the LIM. In Figure 7, the pseudocode for the implementation of the VIM can be seen, where each ray-face intersection would be called only once before finding the length of the line segment for the LIM. The impact on the reconstruction time is described in Section 5.



```
INPUT: voxel position and frustum data
OUTPUT: VIM voxel weight

Find ray-face intersections (up to 8 possible intersection points)
Find voxel edge/ frustum plane intersections (up to 48 possible intersection points)
Find voxel corner in frustum intersections (up to 8 possible intersection points)
Radial sorting of each face vertices
Calculate face area
Calculate polyhedron volume

END
```

Figure 7. VIM pseudocode

4. Experimental configuration

To evaluate the VIM compared to alternative reconstruction approaches, a study was conducted with different reconstruction algorithms, angular ranges, projection numbers and weighting methods for the iterative techniques. The parameters and their values can be seen in Table 1.

Table 1. Parameters used for experimental configurations

Parameter	Parameter value			
Algorithm	ART			
Weighting	LIM		VIM	
Angular range	360	180	135	90
Angular increment	1	2	5	10

In total, 32 experiments were conducted by combining each of the parameters in Table 1. The ART algorithm was used for all 32 experiments, with a varying number of projections over the angular ranges specified. This was then repeated for both LIM and VIM weighting.

To evaluate and compare the quality of the reconstructions, acquisition and reconstruction were simulated for scenarios with full rotation and limited-angle scenarios. The simulated projections were modelled as noiseless and with a point X-ray source to allow investigation of the reconstruction algorithms in an ideal set-up. The object being reconstructed was the Shepp-Logan[23] phantom with a side length of 300 voxels. This was chosen due to its use as a common point of reference within the literature[24]. The simulated projections were generated by forward-projecting through the volume using the Siddon forward-projection model[25] to create a 400 × 400 detector pixel projection at a 16-bit data resolution. The geometry used within the system consisted of a source-to-object distance of 573.8 mm, a source-to-detector distance of 1147.8 mm and a detector pixel pitch of 0.2 mm. This is consistent with the Nikon XT H 225 computed tomography cabinet system used for the physical acquisition.

The number of projections required for reconstruction using the FDK algorithm can be determined using the Nyquist limit, shown in Equation (12) to be approximately 480 projections for a full rotation with an angular spacing $\Delta\theta$ of 0.75°:

$$N_p = \frac{\pi N_x}{2} \quad \text{......................................} (12)$$

where N_x is the number of voxels in the x and y axes. For the volume of side length 300 (as used in the later experimental configurations), the minimum number of projections N_p is 472 over a range of 0 to 360°. The Nyquist limit was used as the baseline for the number of projections used for the experiments conducted. An alternative definition for algebraic reconstruction methods can be seen in Equation (13), using the same nomenclature as in Equation (12). While algebraic reconstruction methods are more tolerant of lower projection numbers, Equation (13) approximates a fully determined system of equations:

$$N_p = \frac{\pi N_x}{4} \quad \text{......................................} (13)$$

For a side length of 300, Equation (13) yields a requirement of 236 projections. In the experiments conducted this will be represented using half the Nyquist limit.

To evaluate the effectiveness of the VIM technique compared to the LIM technique, a parametric study was conducted based on the parameters specified in Table 1. For the iterative techniques, 50 iterations were used to ensure the mean difference between voxel values was less than 5×10^{-5} between iterations. All reconstructions were performed on an Intel i9-9900KF central processing unit (CPU) running at 3.6 GHz running Windows 10 64-bit OS as the host computer for the GPU. The GPU was an NVIDIA RTX 3090 with 10496 CUDA cores and 24 GB of GDDR6X memory. In each scenario, the voxel volume was exported to file after each iteration for later analysis. The relaxation factor was set to 1 as a starting value and decreased by halving the current relaxation factor on each iteration until the value reached 0.01, where it was kept static for subsequent iterations. This technique was chosen based on the research conducted in[26]. Therefore, the eighth iteration and onwards were reconstructed with a relaxation factor of 0.01, to avoid over-convergence. Results have been provided for 50 iterations although convergence was observed sooner, as shown in Figure 8.

To evaluate the performance of the new technique on a physical acquisition, a CT scan of a pencil sharpener was performed using the same acquisition parameters as the simulated experiment on a Nikon XT H 225 system with an accelerating potential of 125 kV, a tube current of 120 μA and 370 projections. The object was reconstructed using all available projections. The convergence curves for the physical acquisition can be seen in Figure 8.

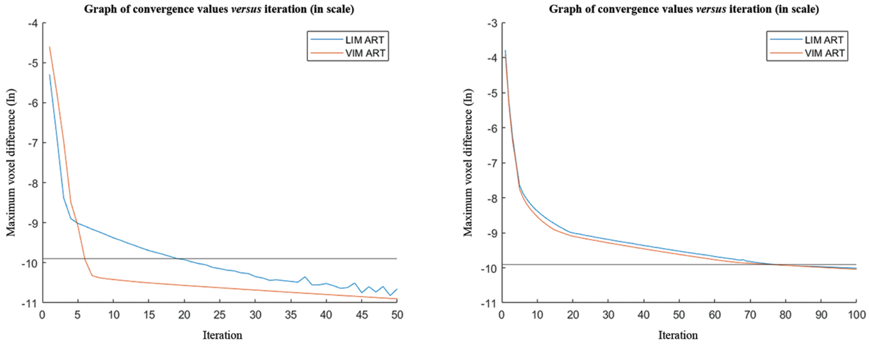


Figure 8. Convergence curves for simulated data (left) and physical acquisition (right)

5. Results and discussion

5.1 Simulated acquisition

All data presented has been normalised between 0 and 1 using min-max normalisation (Equation (14)) to allow direct comparison against the ground truth Shepp-Logan phantom shown in Figure 9. Reconstructions were completed for the entire volume with data extracted for each central slice, as shown in Figure 10:

$$V_{\text{norm}} = \frac{v - \min}{\max - \min} \quad \text{.............................} (14)$$

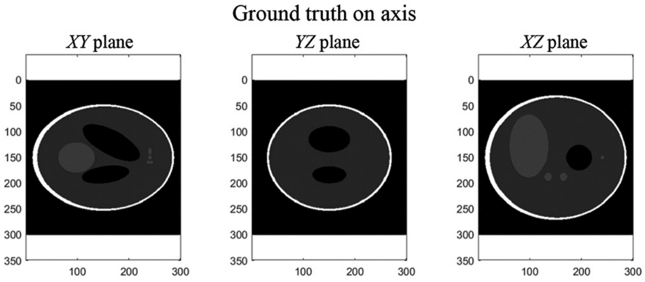


Figure 9. On-axis planes of the Shepp-Logan phantom

Two metrics were chosen to evaluate the accuracy of the reconstructions. These metrics are the root mean square error (RMSE) and the contrast-to-noise ratio (CNR). The use of RMSE allows the error in a plane of the reconstructed volume to be evaluated as a single value. An RMSE of 0 is a perfect fit for the data compared to the ground truth and the larger the value, the larger the error. The equation for RMSE is shown below (Equation (15)), where n is the number of samples in the data, Y_i are the reconstructed voxel values and $\hat{Y}_i$ are the voxel values of the ground truth:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2} \quad \text{......................} (15)$$

The CNR was chosen as a metric due to its use as a common image quality measure, including standards for CT such as ISO-15708[27]. With CNR, the higher the value, the greater the detectability and resolvability of a structure. The CNR was calculated for the reconstructions based on a 2D image from the central plane of the 3D volume, shown in Figure 9. The CNR can be found using Equation (16), where S_A is the average of the signal region of interest (ROI), S_B is the average of the background ROI and σ_B is the standard deviation of the background:

$$\text{CNR} = \frac{|S_A - S_B|}{\sigma_B} \quad \text{...............................} (16)$$

The reconstruction results for the 180° rotation range with a number of projections equal to the Nyquist limit are shown in Figure 10. Through visual inspection, the differences between the reconstruction methods are not significant. The results for the quantitative analysis of the reconstructed volume are shown in Figure 11. The X-axis labels are in the form of the angular range covered and the step in the number of images from the ideal case. Using this format, '135-05' would be a scan range of 135° using every fifth projection from the full dataset, as defined in Equation (12).

Using CNR as a metric to indicate the detectability of a defect, the results show that for all configurations and reconstruction planes, the VIM weighting improves the contrast in all configurations for the XZ plane that is normal to the rotation axis and specific configurations for the YZ and XY planes. The smallest improvement for the XZ plane was 13.6% for the 360° rotation with every projection compared to LIM ART with the same configuration. The largest improvement was 34.2% for the 90° rotation with every tenth projection compared to LIM ART with the same configuration. For the YZ and XY planes the results do not show a global improvement and are comparable to the results for the LIM weighted techniques.

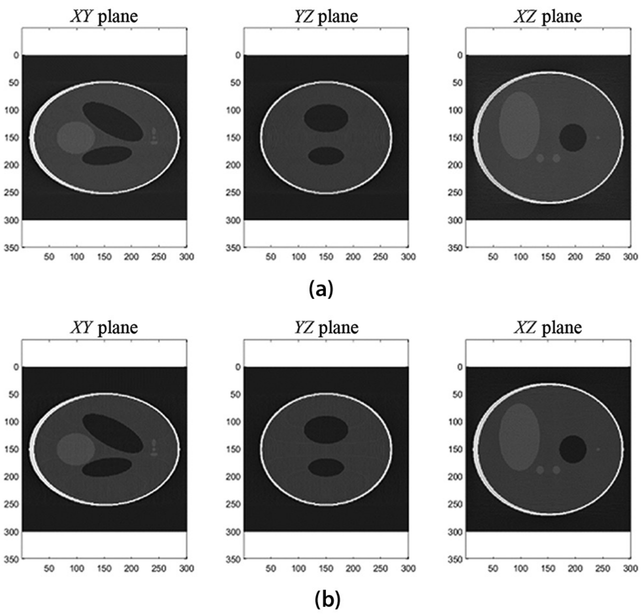


Figure 10. Reconstruction planes for: (a) LIM ART; and (b) VIM ART

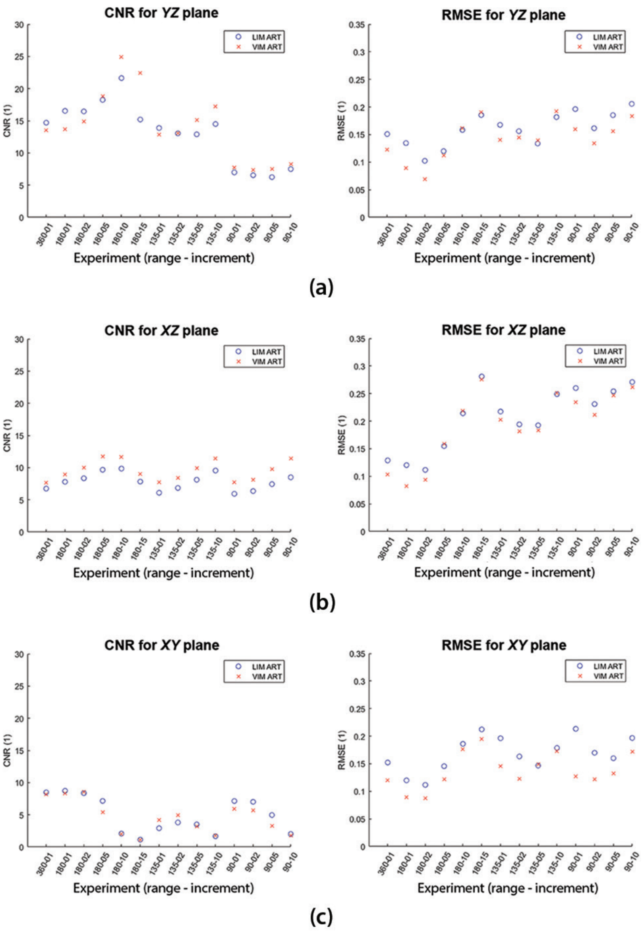


Figure 11. Results for CNR (column 1) and RMSE (column 2) for planes: (a) *YZ*; (b) *XZ*; and (c) *XY*

The RMSE metric that was used to evaluate numerical accuracy indicates an improvement with the VIM weighting on all midplanes for nine out of the 14 experimental configurations, compared with the LIM technique. For the remaining five configurations, the lowest RMSE technique was dependent on the plane orientation. From the results it can be observed that with several projections similar to the Nyquist limit, a higher numerical accuracy is observed at the cost of a longer computational time.

5.2 Physical acquisition

To evaluate the performance of the VIM weighting for a physical acquisition the analysis conducted for the simulated projections was repeated with the exception of the RMSE due to the lack of a ground truth. The LIM technique took an average of 8 s per iteration, while the VIM technique took 113 s on average, with these times being comparable to the simulated result. The reconstructions of the physical acquisition can be seen in Figure 12.

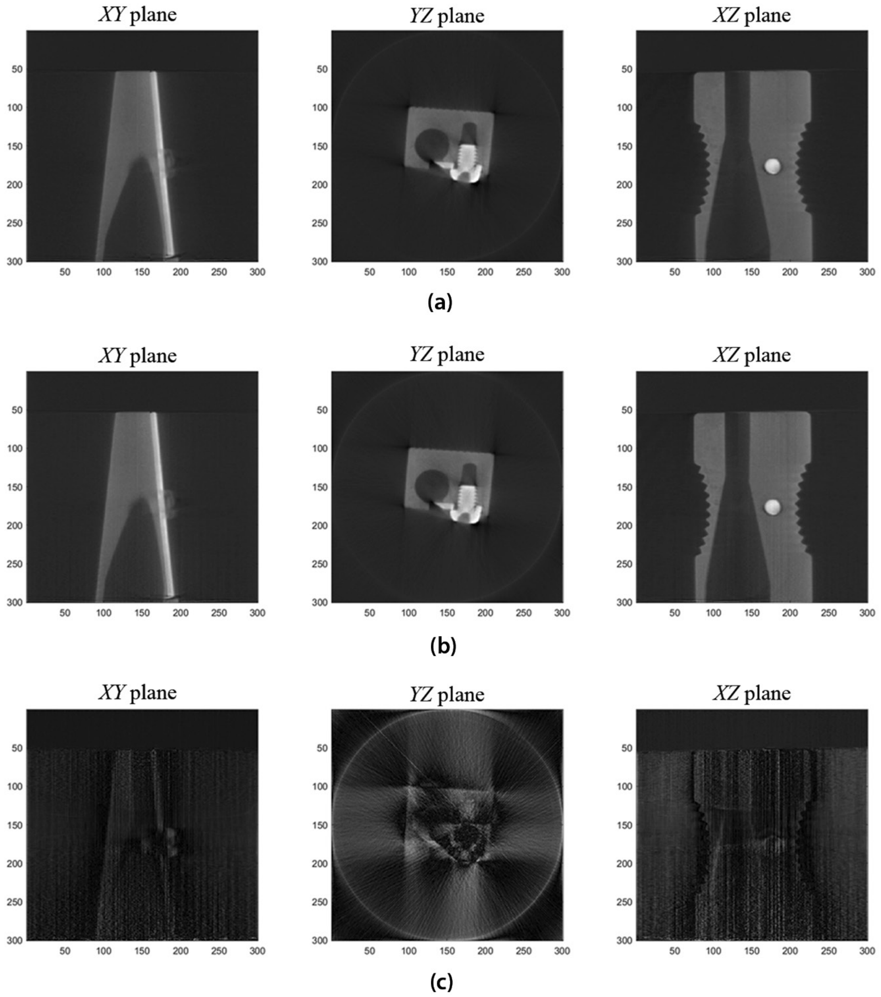


Figure 12. Reconstruction planes of: (a) LIM ART; (b) VIM ART; and (c) absolute difference between LIM ART and VIM ART

Visual inspection of the reconstruction results shows little difference between the two methods. Figure 12(c) shows the absolute difference between the two reconstructions with a maximum of 4% between voxels for the two weighting techniques. The quantitative analysis using CNR shows a higher contrast for the *YZ* reconstruction planes at 13% higher, while the *XZ* and *XY* planes have a lower CNR (16% and 9%, respectively).

Table 2. CNR of physical acquisition

	CNR (*YZ*)	CNR (*XZ*)	CNR (*XY*)
LIM ART	32.41	36.52	53.39
VIM ART	36.59	30.51	48.69

In this experiment, the *YZ* plane was parallel to the detector for the first projection and the *XZ* plane was perpendicular to the rotation. The orientation of the planes with respect to the acquisition system may have an influence on the measured CNR after reconstruction. The object for physical acquisition was reconstructed in a volume larger than required in the *X* and *Z* axes to minimise parameter changes compared to the simulated experiments. This configuration is non-optimal for iterative CT due to the increased number of voxels that represent the region outside the object.

The VIM reconstruction showed a slightly higher amount of streak artefacts in the physical acquisition that were not present in the simulated data. This suggests that the quality of the acquired projections has an impact on the reconstruction with VIM and further work will be needed to assess if this is the cause of the small increase in artefacts.

6. Conclusions

In this paper, the volume integral method has been presented for the algebraic reconstruction technique for full tomographic 3D reconstruction with experimental data demonstrating numerical accuracy. The performance of the technique has been shown to be generally comparable with the line integral method, which is consistent with the literature (2D area integral method *versus* line integral method). In high-precision inspection where an increased reconstruction time is acceptable, the proposed method offers an improved numerical accuracy, which has applications for metrology purposes.

The new technique has been compared to traditional LIM-based techniques on simulated noiseless projections of the Shepp-Logan phantom and using data acquired with a physical system. Reconstructions were performed using the algebraic reconstruction technique for LIM and VIM over a range of scanning parameters to validate the technique in terms of numerical accuracy and contrast, using the root mean square error and contrast-to-noise ratio as metrics. The experiments performed show a lower construction error using the RMSE in the majority of the experiments and comparable RMSE in the remaining experiments. The CNR shows an improvement in the contrast ratio of the on-axis reconstruction plane. The datasets used for this study were simulated noiseless projections of a volume size 300 × 300 × 300 voxels. Further work will aim to investigate the effect of noise and larger dataset sizes and incorporate larger cone angles to evaluate the impact on the reconstruction quality.

Acknowledgements

The authors thank their colleagues Dr Georgios Liaptsis and Alan L Clarke, from TWI Ltd, who provided insight and expertise that greatly assisted the research.

This project is part of an initiative known as the Advanced Engineering Materials Research Institute (AEMRI), which is funded by the Welsh European Funding Office (WEFO) using European Regional Development Funds (ERDF).

The authors gratefully acknowledge funding from the Knowledge Economy Skills Scholarships (KESS 2) and TWI Ltd. KESS 2 is a pan-Wales higher-level skills initiative led by Bangor University on behalf of the higher education sector in Wales. It is part-funded by the Welsh government's European Social Fund (ESF) convergence programme for East Wales.

References

1. L A Feldkamp, L C Davis and J W Kress, 'Practical cone-beam algorithm', Journal of the Optical Society of America A, Vol 1, No 6, pp 612-619, 1984.
2. C Shaw, Cone Beam Computed Tomography, CRC Press, 2014.
3. R Gordon, R Bender and G T Herman, 'Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography', Journal of Theoretical Biology, Vol 29, No 3, pp 471-476, 1970. DOI: 10.1016/0022-5193(70)90109-8
4. G N Hounsfield, 'Computerised transverse axial scanning (tomography): 1. Description of system', The British Journal of Radiology, Vol 46, No 552, pp 1016-1022, 1973. DOI: 10.1259/0007-1285-46-552-1016
5. M Harakal, 'Compute unified device architecture (CUDA) GPU programming model and possible integration to the parallel environment', Science & Military, Vol 3, No 2, pp 64-68, 2008.
6. R Hanna, M Sutcliffe, P Charlton and S Mosey, 'Efficient algebraic image reconstruction technique for computed tomography', Insight: Non-Destructive Testing and Condition Monitoring, Vol 64, No 6, pp 326-333, 2022. DOI: 10.1784/insi.2022.64.6.326
7. A C Kak and M Slaney, 'Algorithms for reconstruction with non-diffracting sources', In: Principles of Computerized Tomographic Imaging, IEEE, 1988.
8. J R Mitchell, P Dickof and A G Law, 'A comparison of line integral algorithms', Computers in Physics, Vol 4, No 2, pp 166-172, 1990. DOI: 10.1063/1.168381
9. G Liaptsis, 'Advancing the techniques of non-standard computed tomography for complex material analysis', Dissertation, Swansea University, Wales, 2019.
10. H Yu and G Wang, 'Finite detector-based projection model for high spatial resolution', Journal of X-ray Science and Technology, Vol 20, No 2, pp 229-238, 2012. DOI: 10.3233/XST-2012-0331.Finite
11. S Zhang, D Zhang, H Gong, O Ghasemalizadeh, G Wang and G Cao, 'Fast and accurate computation of system matrix for area integral model-based algebraic reconstruction technique', Optical Engineering, Vol 53, No 11, 113101, 2014. DOI: 10.1117/1.oe.53.11.113101
12. D J Eck, Introduction to Computer Graphics, Hobart and William Smith College, 2021.
13. G T Herman and L B Meyer, 'Algebraic reconstruction techniques can be made computationally efficient', IEEE Transactions on Medical Imaging, Vol 12 No 3, pp 600-609, 1993.
14. A Kak and A Andersen, 'Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm', Ultrasonic Imaging, Vol 6, No 1, pp 81-94, 1984. DOI: 10.1016/0161-7346(84)90008-7
15. J Han, Introduction to Computer Graphics with OpenGL ES, first edition, CRC Press, 2018. DOI: 10.1201/9780429443145
16. T L Kay and J T Kajiya, 'Ray tracing complex scenes', Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86), Dallas, Texas, USA, pp 269-278, 18-22 August 1986. DOI: 10.1145/15922.15916
17. E Angel, OpenGL: A Primer, Pearson, 2007.
18. S Janke, 'Chapter 6: Polygons and Polyhedra', In: Mathematical Structures for Computer Graphics, pp 196-200, Wiley-Blackwell, 2014.
19. G Green, 'An essay on the application of mathematical analysis to the theories of electricity and magnetism', Journal für die reine und angewandte Mathematik, Vol 1852, No 44, pp 356-374. DOI: 10.1515/crll.1852.44.356
20. M Sutcliffe, M Weston, B Dutton, P Charlton and K Donne, 'Real-time full matrix capture for ultrasonic non-destructive testing with acceleration of post-processing through graphic hardware', NDT&E International, Vol 51, pp 16-23, 2012. DOI: 10.1016/j.ndteint.2012.06.005
21. M A Al-masni, M A Al-antari, M K Metwally, Y M Kadah, S M Han and T S Kim, 'A rapid algebraic 3D volume image reconstruction technique for cone-beam computed tomography', Biocybernetics and Biomedical Engineering, Vol 37, No 4, pp 619-629, 2017. DOI: 10.1016/j.bbe.2017.07.001
22. J Amanatides and A Woo, 'A fast voxel traversal algorithm for ray tracing', Eurographics, Vol 87, pp 3-10, 1987.
23. L A Shepp, B F Logan and M Hill, 'The Fourier reconstruction of a head section', IEEE Transactions on Nuclear Science, Vol 21, No 3, pp 21-43, 1974.
24. A V Narasimhadhan and K Rajgopal, 'FDK-type algorithms with no back-projection weight for circular and helical scan CT', International Journal of Biomedical Imaging, Vol 2012, 969432, 2012. DOI: 10.1155/2012/969432
25. H Turbell, 'Cone-beam reconstruction using filtered back-projection', Dissertation, Linköping Studies in Science and Technology, 2000.
26. A C Kak and M Slaney, Principles of Computerised Tomographic Imaging', Society for Industrial and Applied Mathematics, 2001. DOI: 10.1137/1.9780898719277
27. ISO 15708-1, 'Non-destructive testing – Radiation methods for computed tomography – Part 1: Terminology', 2017. Available at: www.iso.org/standard/72254.html (Accessed: 20 March 2021).