

Approximate Discrete Reconstruction Algorithm

K. Joost Batenburg^{*†}

Centrum Wiskunde & Informatica

Science Park 123, Amsterdam, The Netherlands

joost.batenburg@cwi.nl

Wagner Fortes[‡]

Centrum Wiskunde & Informatica

Science Park 123, Amsterdam, The Netherlands

wagner.fortes@cwi.nl

Robert Tijdeman

Mathematical Institute, Leiden University, The Netherlands

tijdeman@math.leidenuniv.nl

Abstract. Discrete tomography deals with tomographic reconstruction of greyscale images for which the set of possible grey levels is discrete and small. Here, we develop a discrete approximate reconstruction algorithm. Our algorithm computes an image that has only grey values belonging to a given finite set. It also guarantees that the difference between the given projections and the projections of the reconstructed discrete image is bounded. The bound, which is computable, is independent of the image size. We present reconstruction experiments for a range of phantom images and a varying number of grey values.

Keywords: Iterative algorithm, Discrete tomography, Image reconstruction

1. Introduction

Discrete tomography deals with tomographic reconstruction of greyscale images for which the set of possible grey levels is discrete and small [11, 12]. An image of an unknown object is reconstructed

^{*}Address for correspondence: Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, The Netherlands

[†]Acknowledges financial support from the Netherlands Fund for Scientific Research (NWO) (Project number 639.072.005)

Also works: Vision Lab, University of Antwerp, Belgium and Math. Institute, Leiden University, The Netherlands

[‡]Acknowledges support from Leiden University

from a series of *projections* of the object, taken along a range of angles. Contrary to general computed tomography (CT) [10, 15], which requires a large number of projections to obtain an accurate reconstruction, the constraint on the set of grey levels in discrete tomography enables accurate reconstruction from a relatively small number of projections, depending on the properties of the object [2, 11, 12]. A projection for a single angle can be modeled as a collection of *line sums* along parallel lines through the object, where the exact definition of "line sum" depends on the particular model used for the object and the projection process. The *grid model*, where the object is modeled as an image defined on a discrete set of points and line sums are taken along lattice lines through these grid points, can be used to model the position of atoms in crystalline nanostructures [19, 13]. Other projection models, such as the *line* and *strip* models [15, section 7.4.1],[21], are formed by discrete approximation of integral operators that compute integrals of a function defined on the plane. Such an integral model can be used to model the projection process in tomography at lower magnifications than the atomic level, such as X-ray tomography. A range of reconstruction algorithms for discrete tomography have been proposed in the literature [1, 2, 12, 18]. Most of these algorithms are specifically designed for the case of *binary* tomography, where only two grey levels are allowed in the reconstruction.

None of these algorithms comes with a guarantee that an exact solution of the discrete tomography problem is always found, which is not surprising considering the fact that the reconstruction problem is NP-hard for the grid model case [7]. At the same time, the results of computational experiments suggest that in many cases, a solution is found that is near-optimal, or even completely identical to the original object from which the projections were taken, even if the number of projections is very small (less than 10, say). A principal problem with these algorithms is the fact that the error made in the reconstruction depends on the particular problem instance and cannot be bounded sharply. One notable exception is the algorithm proposed in [7], where the ratio between the *number* of 1's (i.e. grid points having a value of 1) in the unknown ground truth object and the reconstructed object is bounded in the optimal solution of certain relaxed variants of the binary reconstruction problem. We are not aware of any algorithm for which nontrivial bounds have been described for the difference between the projections of the reconstructed (discrete) image and the given projections.

In this article, we propose a discrete approximate reconstruction algorithm that comes with such guarantees, based on bounds derived in [9], which are in turn based on the Beck-Fiala Theorem [4]. Our algorithm computes an image that has only grey values belonging to the given finite set. It also guarantees that the difference between the given projections and the projections of the reconstructed discrete image is bounded. The bound, which is explicitly computable, is independent of the image size and scales linearly with the number of projection angles. The algorithm combines techniques from combinatorics with algebraic methods for the solution of linear equation systems.

This paper is structured as follows. Section 2 introduces notation and describes the main concepts and mathematical objects used throughout the paper. In Section 3, the basic algorithm is introduced. A proof for the upper bound on the projection error of the reconstructed image is given in Section 4. Section 5 presents several variations of the algorithm, which can improve its computational performance in practice. Computational and numerical aspects are discussed in Section 6. In Section 7, results are presented for a series of simulation experiments based on three phantom images, including both binary images and an image with three grey levels. Finally, Section 8 provides an outlook and conclusions.

2. Notation and concepts

Tomography concerns the reconstruction of an object from its *projections*. The exact definition of the term "projection" can vary, but in general it is related to the set of line integrals (or surface integrals along a tight strip) through the object in a particular direction. We refer to the specific model used to determine such integrals as the *imaging model*. Throughout the discrete tomography literature, several imaging models have been considered, such as the line, strip and Joseph's model [14]. Since the algorithm developed here can be used for any imaging model, we do not specify any model for the development of the theory, only for the experiments.

The *unknown object* that one wants to reconstruct is typically approximated by an image defined on a discrete pixel grid. The image that we want to reconstruct is unknown but we do know its grey values, which belong to a discrete set. Let $D = \{d_1, \dots, d_s\} \subset \mathbb{R}^s$ be a given set of grey values such that $d_1 < d_2 < \dots < d_s$ with s small. Also, let $d = \max_{i=1, \dots, s-1} (d_{i+1} - d_i)$. An *image* is represented by a vector $\mathbf{x} = (x_i) \in \mathbb{R}^n$. We refer to the entries of \mathbf{x} as *pixels*, corresponding to elements in the discrete pixel grid. A *discrete image*, in this paper, corresponds with a vector $\bar{\mathbf{x}} \in D^n$.

For given set of k projection directions, each consisting of w measured values, the *projection map* maps an image \mathbf{x} to a vector $\mathbf{p} \in \mathbb{R}^m$ of projection data, where $m = kw$ denotes the total number of measurements. As the projection map is a linear transformation, it can be represented by a matrix $\mathbf{W} = (w_{ij}) \in \mathbb{R}^{m \times n}$, called the *projection matrix*. The entry w_{ij} represents the weight of the contribution of pixel x_j to the projection value p_i , which typically satisfies $0 \leq w_{ij} \leq 1$. For two images $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we refer to $\mathbf{W}(\mathbf{x} - \mathbf{y})$ as the *projection difference* of \mathbf{x} and \mathbf{y} , and to its largest component in absolute value $\|\mathbf{W}(\mathbf{x} - \mathbf{y})\|_\infty$ as the *projection distance*.

The *general reconstruction problem* consists of finding a solution of the system

$$\mathbf{W}\mathbf{x} = \mathbf{p} \quad (1)$$

for given projection data $\mathbf{p} = (p_i)$, i.e., to find an image that has the given projections. In *discrete tomography*, one seeks a solution of the general reconstruction problem which belongs to D .

Throughout the article, we use the symbol $\mathbf{0}$ to denote a column vector $(0, \dots, 0)^T$ consisting of 0's, where its dimension is clear from the context. Also, we often use $\kappa = \|\mathbf{W}\|_1 = \max_{j=1, \dots, n} \sum_{i=1}^m |w_{ij}|$.

For the following sections, consider the problem of finding a solution, belonging to the set D^n , of a fixed linear system $\mathbf{W}\mathbf{x} = \mathbf{p}$ called the *discrete reconstruction problem*. We call $\bar{\mathbf{x}} \in D^n$ an *approximate discrete solution* if its projection difference with respect to a solution of the general reconstruction problem is small.

3. Algorithm description

The algorithm presented here aims to reconstruct an image that only contains grey values belonging to the given set D and ensures that its projections have a projection distance from the given projections smaller than κd . The parameter $\kappa = \|\mathbf{W}\|_1$ depends on the projection model and the number of projection angles, while $d = \max_{i=1, \dots, s-1} (d_{i+1} - d_i)$ is the largest difference between two consecutive values in D . The bounds on the projection distance do not depend on the image size.

The algorithm requires an initial grey scale image $\mathbf{x}^{(0)} \in [d_1, d_s]^n$ and uses the projections $\mathbf{p} = \mathbf{W}\mathbf{x}^{(0)}$ of the initial image as the given projections for the reconstruction. If a pixel $x_j^{(0)}$ already belongs

to the set D , it is left unchanged by the algorithm. For every pixel j such that $x_j^{(0)} \notin D$, there is a unique i such that $d_i < x_j^{(0)} < d_{i+1}$. In the reconstructed image $\bar{x} \in D^n$, this pixel j satisfies $\bar{x}_j \in \{d_i, d_{i+1}\}$, so a pixel can either be assigned the first grey level larger than its initial value, or the first grey level smaller than its initial value.

In each iteration, a new image $\mathbf{x}^{(t)}$ is computed that has a projection distance smaller than κd with respect to the initial image $\mathbf{x}^{(0)}$. Whenever the current image $\mathbf{x}^{(t)}$ contains pixels in D , the next image $\mathbf{x}^{(t+1)}$ will contain these pixels with the same values as the current image. Furthermore, the current image $\mathbf{x}^{(t)}$ contains more pixels belonging to D than the previous image $\mathbf{x}^{(t-1)}$.

The iteration step adds a specific *ghost image* to the current image $\mathbf{x}^{(t)}$. A ghost image for a given projection matrix is an image in the null space of that matrix. The ghost image may change the value of pixels not in D but not the ones in D , and is constructed in such a way that the number of pixels for which the value is in D increases. Therefore, the number of such iteration steps is at most n , the total number of pixels.

In each iteration, some line sums remain unchanged. Such line sums are associated with equations in the system (1) for which the sum of the coefficients, of the pixels currently not in D , is not smaller than κ . The remaining line sums may have an error of at most κd with respect to the projections of the initial image.

In order to specify the ghost image, define the following sets

Definition 3.1. For $i = 1, \dots, m$, define $L_i = \{1 \leq j \leq n : w_{ij} \neq 0\}$, the set of indices of pixels that occur in the i th equation of the linear system (1). Based on the current image $\mathbf{x}^{(t-1)}$ in iteration $t - 1$, we can define $\bar{I}_D^{(t)} = \{1 \leq j \leq n : x_j^{(t-1)} \notin D\}$, the set of indices of pixels with values not in D in $\mathbf{x}^{(t-1)}$ and its complement $I_D^{(t)} = \{1, \dots, n\} \setminus \bar{I}_D^{(t)}$. Another important set for the algorithm is $G^{(t)} = \{L_i : \sum_{j \in L_i \cap \bar{I}_D^{(t)}} w_{ij} \geq \kappa, \text{ for } i = 1, \dots, m\}$, the family of sets L_i of pixel indices, such that the sum of the corresponding coefficients is at least κ for pixels not in D in $\mathbf{x}^{(t-1)}$.

Definition 3.2. Let $\mathbf{x}^{(t-1)}$ be the reconstructed image that resulted from iteration $t - 1$. Consider the following homogeneous linear system of equations

$$\sum_{j \in L_i \cap \bar{I}_D^{(t)}} w_{ij} y_j^{(t)} = 0 \quad \text{for all } L_i \in G^{(t)} \quad (2)$$

$$y_j^{(t)} = 0 \quad \text{for all } j \notin (\cup_{L_i \in G^{(t)}} L_i) \cap \bar{I}_D^{(t)}. \quad (3)$$

Let $\mathbf{A}^{(t)}$ be the matrix associated with the linear system corresponding to the equations (2) and (3). An image $\mathbf{y}^{(t)} \in \mathcal{N}(\mathbf{A}^{(t)}) = \{\mathbf{x} : \mathbf{A}^{(t)} \mathbf{x} = \mathbf{0}\}$ is called a *ghost image* associated with $\mathbf{A}^{(t)} \mathbf{y}^{(t)} = \mathbf{0}$.

In iteration t of the algorithm, a ghost image associated with $\mathbf{A}^{(t)} \mathbf{y}^{(t)} = \mathbf{0}$ is computed, which is subsequently used to form the new reconstruction $\mathbf{x}^{(t)}$. Pseudocode for the procedure that uses this ghost image is shown in Procedure 1. As a result of applying Procedure 1, a new reconstruction is formed for which at least one new pixel belongs to D .

If $\mathcal{N}(\mathbf{A}^{(t)}) = \{\mathbf{0}\}$, hence $\mathbf{y}^{(t)} = \mathbf{0}$, Procedure 1 has no effect. Another technique must be employed to change the current value of pixels not yet in D . In that case, all pixel values not in D are rounded to

Procedure 1 Calculate $\mathbf{x}^{(t)}$

```

 $\lambda_{\min} \leftarrow \infty$ 
for  $j \in \bar{I}_D^{(t)}$  do
    Compute the smallest  $\lambda > 0$  s.t.  $x_j^{(t-1)} + \lambda y_j^{(t)} \in D$ 
    if  $\lambda < \lambda_{\min}$  then
         $\lambda_{\min} \leftarrow \lambda$ 
    end if
end for
 $\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t-1)} + \lambda_{\min} \mathbf{y}^{(t)}$ 
    
```

Procedure 2 Calculate $\bar{\mathbf{x}}$

```

for  $j \in \{1, \dots, n\}$  do
    if  $x_j^{(t)} \in D$  then
         $\bar{x}_j \leftarrow x_j^{(t)}$ 
    else
        Assign  $\bar{x}_j$  the element of  $D$  which is nearest to  $x_j^{(t)}$ 
    end if
end for
    
```

their nearest value in D , after which the algorithm terminates. The algorithm ends with a final vector $\bar{\mathbf{x}} \in D^n$ satisfying the given bounds. This part of the algorithm is shown in Procedure 2.

The iterative algorithm which obtains a vector in D^n with projection error at most κd is given in the flowchart represented in Fig. 1.

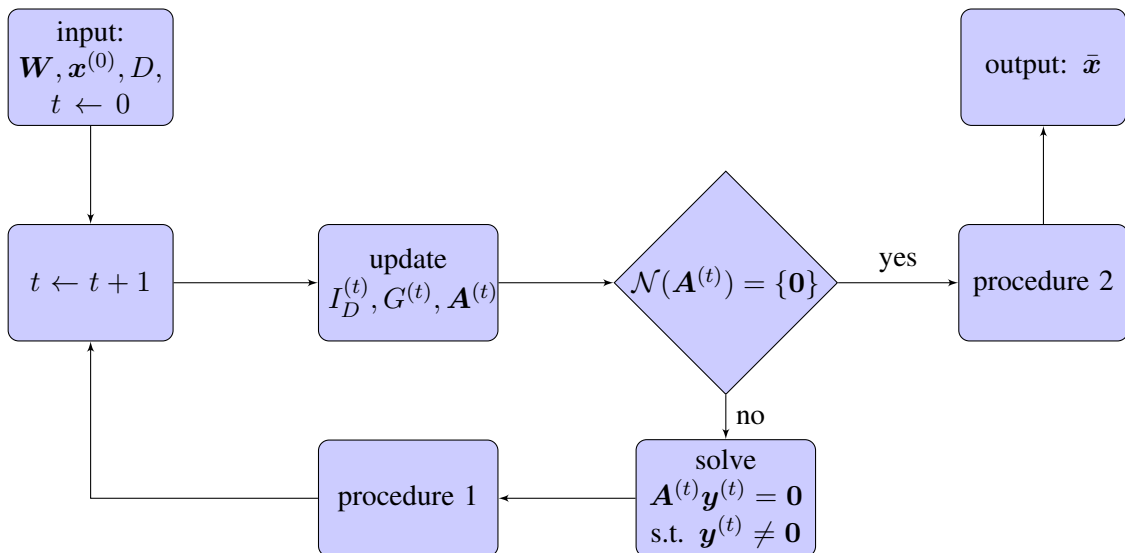


Figure 1: This flowchart describes the discrete reconstruction algorithm which obtains a discrete image $\bar{\mathbf{x}}$ in D with projection distance of at most κd with respect to a given image $\mathbf{x}^{(0)}$.

4. Walkthrough example

We have built a simple example to illustrate each step of Algorithm 1 (Fig. 1). In this example, the size of the image to be reconstructed is 3×3 and $D = \{0, 1\}$. We use 3 projection angles: horizontal, vertical and diagonal from top left to bottom right. The projection model is defined below setting $\kappa = 3$.

The initial approximate image $\mathbf{x}^{(0)}$ is shown in Fig. 2(a) and the 11 line sums which form the projection data are: horizontal: $h_1 = x_1 + x_2 + x_3$, $h_2 = x_4 + x_5 + x_6$ and $h_3 = x_7 + x_8 + x_9$; vertical: $v_1 = x_1 + x_4 + x_7$, $v_2 = x_2 + x_5 + x_8$ and $v_3 = x_3 + x_6 + x_9$ and diagonal: $d_1 = x_7$, $d_2 = x_4 + x_8$, $d_3 = x_1 + x_5 + x_9$, $d_4 = x_2 + x_6$ and $d_5 = x_3$. We leave the computation of each line sum for each iteration for the reader.

$\mathbf{x}_1^{(0)} = 0.5$	$\mathbf{x}_2^{(0)} = 0.8$	$\mathbf{x}_3^{(0)} = 0.5$	1	-1	0	0.8	0.5	0.5
$\mathbf{x}_4^{(0)} = 0.5$	$\mathbf{x}_5^{(0)} = 0.6$	$\mathbf{x}_6^{(0)} = 0.7$	-1	0	1	0.2	0.6	1
$\mathbf{x}_7^{(0)} = 0.5$	$\mathbf{x}_8^{(0)} = 0.4$	$\mathbf{x}_9^{(0)} = 0.5$	0	1	-1	0.5	0.7	0.2

(a) initial approx. solution $\mathbf{x}^{(0)}$ (b) ghost image $\mathbf{y}^{(1)}$ (c) approximate solution $\mathbf{x}^{(1)}$

Figure 2: Ghost image $\mathbf{y}^{(1)}$ together with $\mathbf{x}^{(0)}$ produces $\lambda^{(1)} = 0.3$, generating $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda^{(1)}\mathbf{y}^{(1)}$.

In order to simplify the notation, we use the same symbols of the line sums to represent the sets L_i which form $G^{(t)}$, where $G^{(1)} = \{h_1, h_2, h_3, v_1, v_2, v_3, d_3\}$. A ghost image $\mathbf{y}^{(1)}$ is shown in Fig. 2(b), which has zero line sums for each $L_i \in G^{(1)}$. Therefore, for each $L_i \in G^{(1)}$ the respective line sums of $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$ are equal.

For iteration 2, we have $G^{(2)} = \{h_1, h_3, v_1, v_2, d_3\}$ from Def. 3.1 and $\mathbf{y}_6^{(2)} = 0$ from Def. 3.2. As a consequence, the line sums h_2 and v_3 of the ghost image $\mathbf{y}^{(2)}$ are no longer restricted to zero. A possible ghost image $\mathbf{y}^{(2)}$ and the resulting $\mathbf{x}^{(2)}$ are shown in Fig. 3.

1	-1	0	1	0.3	0.5
-1	0	0	0	0.6	1
0	1	-1	0.5	0.9	0

(a) ghost image $\mathbf{y}^{(2)}$ (b) approximate solution $\mathbf{x}^{(2)}$

Figure 3: Ghost image $\mathbf{y}^{(2)}$ together with $\mathbf{x}^{(1)}$ produces $\lambda^{(2)} = 0.2$, generating $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda^{(2)}\mathbf{y}^{(2)}$.

Notice that $I_D^{(3)} = \{1, 4, 6, 9\}$, based on $\mathbf{x}^{(2)}$. For iteration 3, we have $G^{(3)} = \{v_2\}$ and from Def. 3.2, the pixels $\mathbf{y}_2^{(3)}$, $\mathbf{y}_5^{(3)}$ and $\mathbf{y}_8^{(3)}$ are the only ones allowed to be non-zero. A ghost image $\mathbf{y}^{(3)}$ and the resulting image $\mathbf{x}^{(3)}$ are shown in Fig. 4.

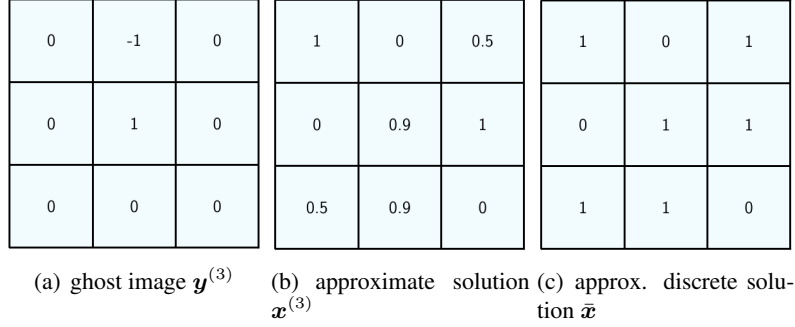


Figure 4: Ghost image $\mathbf{y}^{(3)}$ together with $\mathbf{x}^{(2)}$ produces $\lambda^{(3)} = 0.2$, generating $\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + \lambda^{(3)}\mathbf{y}^{(3)}$.

As $G^{(4)} = \emptyset$, we have $\mathcal{N}(\mathbf{A}^{(4)}) = \{\mathbf{0}\}$ and Procedure 2 is used, obtaining $\bar{\mathbf{x}}$ by rounding the entries of $\mathbf{x}^{(3)}$ to the nearest element of D , after which the algorithm terminates. In this case, pixels having a value of 0.5 are rounded to 1. The final approximate solution $\bar{\mathbf{x}}$ is shown in Fig. 4(c).

5. The algorithm’s proof

In this section we prove that the algorithm represented in the flowchart of Fig. 1 finds an image $\bar{\mathbf{x}} \in D^n$ such that the projection distance between the initial image $\mathbf{x}^{(0)}$ and $\bar{\mathbf{x}}$ is smaller than κd . First, we prove the requirements and implications of one iteration of Procedure 1.

Lemma 5.1. Suppose that

- (H1) $\mathbf{x}^{(t-1)} \in [d_1, d_s]^n$ is the reconstructed image that resulted from iteration $t - 1$;
- (H2) $\mathbf{y}^{(t)} \neq \{\mathbf{0}\}$ is a solution of $\mathbf{A}^{(t)}\mathbf{y} = \mathbf{0}$, for $\mathbf{A}^{(t)}$ given by Definition 3.2.

Then after applying Procedure 1, the following statements are true:

- (T1) $|I_D^{(t)}| > |I_D^{(t-1)}|$;
- (T2) $x_j^{(t)} = x_j^{(t-1)}$ for all $x_j^{(t-1)} \in D$;
- (T3) there is no $1 \leq j \leq n$ such that there exists a $d_i \in D$ for which $x_j^{(t-1)} < d_i < x_j^{(t)}$;
- (T4) $\sum_{j \in L_i} w_{ij}(x_j^{(t)} - x_j^{(t-1)}) = 0$ for all $L_i \in G^{(t)}$;
- (T5) $\sum_{j \in L_i} w_{ij} |x_j^{(t)} - x_j^{(t-1)}| < \kappa d$ for all $L_i \notin G^{(t)}$.

Proof:

Let $\mathbf{y}^{(t)} \neq \mathbf{0}$ such that $\mathbf{A}^{(t)}\mathbf{y}^{(t)} = \mathbf{0}$. Define $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} + \lambda^{(t)}\mathbf{y}^{(t)}$ with $\lambda^{(t)}$ the smallest positive scalar λ for which $x_j^{(t-1)} + \lambda y_j^{(t)} \in D$ for some $\hat{j} \in \bar{I}_D^{(t)}$. This definition of $\mathbf{x}^{(t)}$ follows exactly the computation of Procedure 1. As $x_{\hat{j}}^{(t)} \in D$ and none of the entries that were already in D are modified, statements (T1) and (T2) are true. By definition of $\lambda^{(t)}$, statement (T3) is also true.

Statement (T4) follows from the definition of $\mathbf{A}^{(t)}$ and statement (T2):

$$\sum_{j \in L_i} w_{ij} (x_j^{(t)} - x_j^{(t-1)}) = \lambda \sum_{j \in L_i \cap \bar{I}_D^{(t)}} w_{ij} y_j^{(t)} + 0 \sum_{j \in L_i \cap I_D^{(t)}} w_{ij} = 0 \quad \text{for all } L_i \in G^{(t)}.$$

Also, (T3) implies $|x_j^{(t)} - x_j^{(t-1)}| < d$, for all $1 \leq j \leq n$, and since $\sum_{j \in L_i \cap \bar{I}_D^{(t)}} w_{ij} < \kappa$ for $L_i \notin G^{(t)}$ we have

$$\sum_{j \in L_i} w_{ij} |x_j^{(t)} - x_j^{(t-1)}| < d \sum_{j \in L_i \cap \bar{I}_D^{(t)}} w_{ij} + 0 \sum_{j \in L_i \cap I_D^{(t)}} w_{ij} < \kappa d \quad \text{for } L_i \notin G^{(t)}$$

for all $i = 1, \dots, m$. □

Whenever $\mathbf{y}^{(t)} = \mathbf{0}$ is the unique ghost image of iteration t , Procedure 1 has no effect. The next Lemma is an intermediate result for the case that Procedure 2 is required.

Lemma 5.2. Suppose that $\mathcal{N}(\mathbf{A}^{(t)}) = \{\mathbf{0}\}$ for $\mathbf{A}^{(t)}$ given by Definition 3.2. Then

$$\sum_{j \in L_i \cap \bar{I}_D^{(t)}} w_{ij} \leq \kappa \quad \text{for } i = 1, \dots, m.$$

Proof:

As equations (3) already define $y_j^{(t)} = 0$ for a set of j 's, let $\hat{\mathbf{A}}^{(t)} = (\hat{a}_{ij}^{(t)})$ be the matrix associated with equations (2). Since $\mathcal{N}(\hat{\mathbf{A}}^{(t)}) = \{\mathbf{0}\}$, the number of linear independent rows of $\hat{\mathbf{A}}^{(t)}$ is equal to the number of columns of $\hat{\mathbf{A}}^{(t)}$. We can eliminate rows which are linear combinations of others obtaining a square matrix, say of size $r \times r$. Supposing $r > 0$, the definition of $\kappa = \|\mathbf{W}\|_1$ yields

$$\sum_{i=1}^r \hat{a}_{ij}^{(t)} \leq \kappa \quad \text{for } j = 1, \dots, r, \quad \text{implying} \quad \sum_{j=1}^r \sum_{i=1}^r \hat{a}_{ij}^{(t)} \leq r\kappa \quad (4)$$

and the definition of $G^{(t)}$ gives

$$\sum_{j=1}^r \hat{a}_{ij}^{(t)} \geq \kappa \quad \text{for } i = 1, \dots, r, \quad \text{implying} \quad \sum_{i=1}^r \sum_{j=1}^r \hat{a}_{ij}^{(t)} \geq r\kappa. \quad (5)$$

From statements (4) and (5) we conclude that $\sum_{i=1}^r \sum_{j=1}^r \hat{a}_{ij}^{(t)} = r\kappa$ and $\sum_{j=1}^r \hat{a}_{ij}^{(t)} = \kappa$. Then

$$\sum_{j \in L_i \cap \bar{I}_D^{(t)}} w_{ij} = \sum_{j \in L_i \cap \bar{I}_D^{(t)}} \hat{a}_{ij}^{(t)} = \kappa \quad \text{for } L_i \in G^{(t)}$$

for $i = 1, \dots, m$.

If $r = 0$, we have $G^{(t)} = \emptyset$ and it follows from the definition of $G^{(t)}$ that $\sum_{j \in L_i \cap \bar{I}_D^{(t)}} w_{ij} < \kappa$ for $L_i \notin G^{(t)}$. □

Theorem 5.3. Let $\mathbf{x}^{(0)} \in [d_1, d_s]^n$ be a given image such that $\mathbf{W}\mathbf{x}^{(0)} = \mathbf{p}$. Then, after termination of Algorithm 1 (Fig. 1), we obtain a vector $\bar{\mathbf{x}} \in D^n$ such that

$$\|\mathbf{W}\bar{\mathbf{x}} - \mathbf{p}\|_\infty < \kappa d.$$

Proof:

Procedure 2 is used as the last iteration of the algorithm generating $\bar{\mathbf{x}} \in D^n$. Hence we have to prove that the projection error accumulated during the iterations satisfies $\|\mathbf{W}\bar{\mathbf{x}} - \mathbf{p}\|_\infty < \kappa d$.

Lemma 5.1 ensures that between any two iterations $\hat{t} - 1$ and \hat{t} of Procedure 1 there is no $1 \leq j \leq n$ such that there exists a $d_i \in D$ for which $x_j^{(\hat{t}-1)} < d_i < x_j^{(\hat{t})}$, implying that there is no $1 \leq j \leq n$ such that there exists a $d_i \in D$ for which $x_j^{(0)} < d_i < x_j^{(t)}$.

Suppose that iteration $t + 1$ uses Procedure 2: set \bar{x}_j as the element from D which is nearest to $x_j^{(t)}$ for all $j \in \bar{I}_D^{(t+1)}$ and set $\bar{x}_j = x_j^{(t)}$ for all $j \in I_D^{(t+1)}$. Then we can conclude that $|\bar{x}_j - x_j^{(0)}| < d$ for $j = 1, \dots, n$.

From Lemma 5.1, it is easy to check that after t iterations of Procedure 1 we have

$$\sum_{L_i} w_{ij}(x_j^{(t)} - x_j^{(0)}) = 0 \quad \text{for } L_i \in G^{(t)}.$$

After applying Procedure 2, Lemma 5.2 provides $\sum_{j \in L_i \cap \bar{I}_D^{(t+1)}} w_{ij} \leq \kappa$ for $i = 1, \dots, m$, which yields

$$\left| \sum_{L_i} w_{ij}(\bar{x}_j - x_j^{(0)}) \right| < \kappa d \quad \text{for } L_i \in G^{(t)}. \tag{6}$$

Also from Lemma 5.1, for each $i = 1, \dots, m$ such that, $L_i \notin G^{(t)}$, there exist disjoint sets $A, B \notin G^{(t)}$ such that $L_i = A \cup B$ with $\sum_{j \in A} w_{ij}(x_j^{(t)} - x_j^{(0)}) = 0$ and $\sum_{j \in B} w_{ij} < \kappa$. These properties are still valid after applying Procedure 2 yielding

$$\left| \sum_{L_i} w_{ij}(\bar{x} - x_j^{(0)}) \right| = \left| \sum_{j \in B} w_{ij}(\bar{x} - x_j^{(0)}) \right| < \kappa d \quad \text{for } L_i \notin G^{(t)}. \tag{7}$$

Therefore, from Eqs. (6) and (7), it follows that $\bar{\mathbf{x}} \in D^n$ and $\|\mathbf{W}\bar{\mathbf{x}} - \mathbf{p}\|_\infty < \kappa d$.

□

6. A threshold variation

In some cases, after an iteration step, there are pixels with current values very close to a value in the set D . The use of a threshold parameter $\tau < d$ for rounding some pixels can speed up the algorithm, as it would need fewer iterations to make $\mathbf{x}^{(0)}$ converge to $\bar{\mathbf{x}} \in D^n$. As a drawback, the use of threshold may increase the difference in the projections. Moreover, the bound on the projection distance becomes dependent on the image size. Our variant of the reconstruction algorithm that uses such a threshold only requires a modification in Procedure 1. We denote the modified procedure as Procedure 3.

Procedure 3 Calculate $\mathbf{x}^{(t)}$

```

 $\lambda_{\min} \leftarrow \infty$ 
for  $j \in \bar{I}_D^{(t)}$  do
  Compute the smallest  $\lambda > 0$  s.t.  $x_j^{(t-1)} + \lambda y_j^{(t)} \in D$ 
  if  $\lambda < \lambda_{\min}$  then
     $\lambda_{\min} \leftarrow \lambda$ 
  end if
end for
 $\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t-1)} + \lambda_{\min} \mathbf{y}^{(t)}$ 
for  $j \in \{1, \dots, n\}$  do
  if there exist  $d_i \in D$  such that  $|x_j^{(t)} - d_i| < \tau$  for  $i = 1, \dots, s$  then
     $x_j^{(t)} \leftarrow d_i$ 
  end if
end for

```

Theorem 6.1. Let $\mathbf{x}^{(0)} \in [d_1, d_s]^n$ be a given image such that $\mathbf{W}\mathbf{x}^{(0)} = \mathbf{p}$ and $\tau < d$. Then, after termination of the Algorithm 1 (Fig. 1) using Procedure 3 instead of Procedure 1, we obtain a vector $\bar{\mathbf{x}} \in D^n$ such that

$$\|\mathbf{W}\bar{\mathbf{x}} - \mathbf{p}\|_{\infty} < \kappa d + (\|\mathbf{W}\|_{\infty} - \kappa)\tau.$$

Proof:

If $|x_j^{(t)} - d_l| \leq \tau$ for some $l = 1, \dots, s$, then $|x_j^{(t+1)} - x_j^{(t)}| \leq \tau$. This implies that the projection difference of the i -th component increases by, at most, τ for each thresholded pixel in equation i , in one iteration step. Since $\tau < d$, the worst case happens when for an equation i , there exist sets of indices A and B such that $\sum_{j \in A} w_{ij} = \kappa$ and $\sum_{j \in B} w_{ij} = \|\mathbf{W}\|_{\infty} - \kappa$, and all the pixels defined by B are thresholded, but not the ones defined by A . After applying Procedure 2, we have $\sum_{j=1}^n w_{ij} |\bar{x}_j - x_j^{(0)}| < \kappa d + (\|\mathbf{W}\|_{\infty} - \kappa)\tau$. \square

7. Computations

In this section, we will cover two important computational aspects of our approach: how to compute a suitable initial solution and how to compute a ghost image for the iteration steps. In both cases, we outline a particular method for computing an image. Many alternative methods exist for solving the corresponding mathematical problems, some of which may yield a more efficient algorithm.

7.1. Initial solution $\mathbf{x}^{(0)}$

The discrete reconstruction algorithm presented in this paper needs an initial image $\mathbf{x}^{(0)} \in [d_1, d_s]^n$ which can be achieved, e.g., by an algorithm presented as a *Norm minimization over an interval* algorithm in [6]. Algorithm 4 is the version of the algorithm that we have used in the experiments. It minimizes $\frac{1}{2}\|\mathbf{z}\|^2$ subject to $\mathbf{W}\mathbf{z} = \mathbf{p}$ and $d_1 \leq z_j \leq d_s$ for all $j = 1, \dots, n$, which guarantees $\mathbf{x}^{(0)} \in [d_1, d_s]^n$. This algorithm is a variation of ART [8] (ART is also known as Kaczmarz method [10, 16]).

Algorithm 4 Compute $\mathbf{x}^{(0)}$

```

 $t \leftarrow 0, \mathbf{z} \leftarrow \mathbf{0}_n, \hat{\mathbf{z}} \leftarrow \mathbf{0}_n$ 
while  $\|\mathbf{W}\mathbf{z}^{(t)} - \mathbf{p}\|_\infty > \varepsilon$  do
   $i \leftarrow (t \bmod m) + 1$ 
   $z_j \leftarrow z_j + \frac{p_i - \sum_{j=1}^n w_{ij}\hat{z}_j}{\sqrt{\sum_{j=1}^n w_{ij}^2}} w_{ij}$  for  $j = 1, \dots, n$ 
  for  $j \in \{1, \dots, n\}$  do
     $\hat{z}_j \leftarrow \begin{cases} d_1 & \text{if } z_j < d_1 \\ z_j & \text{if } d_1 \leq z_j \leq d_s \\ d_s & \text{if } z_j > d_s \end{cases}$ 
  end for
   $t \leftarrow t + 1$ 
end while
 $\mathbf{x}^{(0)} \leftarrow \mathbf{z}$ 

```

Since we want to guarantee a small projection distance between $\bar{\mathbf{x}}$ and the unknown original object from which \mathbf{p} was formed, it is necessary to have a small distance between $\mathbf{W}\mathbf{x}^{(0)}$ and \mathbf{p} . In order to do so, the stopping criteria of Algorithm 4 is given by $\|\mathbf{W}\mathbf{z} - \mathbf{p}\|_\infty < \varepsilon$, where ε is a positive constant close to zero.

7.2. Non-null ghost images

If $\mathcal{N}(\mathbf{A}^{(t)}) \neq \{\mathbf{0}\}$, Procedure 1 is executed. To this end, we must find a non-null solution of a homogeneous linear system

$$\mathbf{A}\mathbf{z} = \mathbf{0}. \quad (8)$$

To solve this problem we have used an iterative method called CGLS (Conjugate Gradient Least Squares) [17]. The CGLS algorithm requires an initial guess $\mathbf{z}^{(0)}$. Apart from numerical errors, applying the CGLS algorithm to the system (8) results, after convergence, in the computation of $\mathbf{z}_{LS} = (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{z}^{(0)}$, where \mathbf{A}^\dagger is the *Moore-Penrose pseudo inverse* of \mathbf{A} , see [3, 5, 20]. The CGLS algorithm computes \mathbf{z}_{LS} without explicitly computing \mathbf{A}^\dagger .

The matrix $(\mathbf{I} - \mathbf{A}^\dagger \mathbf{A})$ orthogonally projects $\mathbf{z}^{(0)}$ onto $\mathcal{N}(\mathbf{A})$. As a result, $\mathbf{z}_{LS} = \mathbf{0}$ if and only if $\mathbf{z}^{(0)} \perp \mathcal{N}(\mathbf{A})$. Hence, randomly selecting $\mathbf{z}^{(0)}$ will almost certainly yield a $\mathbf{z}_{LS} \neq \mathbf{0}$.

In practice, several algorithms use the relative residual as a parameter for the stopping criteria. The relative residual is given by the current iteration residual norm divided by the norm of the right-hand side. When the right-hand side is the null-vector, the relative residual is computed as infinity and this algorithm can never identify convergence. To overcome this problem, instead of applying the CGLS algorithm to (8), we apply it to the linear system

$$\mathbf{A}\mathbf{z} = \mathbf{b}, \quad (9)$$

with $\mathbf{b} = -\mathbf{A}\mathbf{z}^{(0)}$. When applying CGLS to Eqs. (9), we select the null-vector as initial guess obtaining, therefore, $\mathbf{z}^* = \mathbf{A}^\dagger \mathbf{b}$, the minimum norm solution of system (9) [3, 5, 20]. Defining $\mathbf{y} = \mathbf{z}^{(0)} + \mathbf{z}^*$, it satisfies Eqs. (8): $\mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{z}^{(0)} + \mathbf{A}\mathbf{z}^* = \mathbf{A}\mathbf{z}^{(0)} + \mathbf{b} = \mathbf{0}$.

8. Numerical experiments

A series of experiments was performed to investigate the quality of images reconstructed by the algorithm presented and the resulting bounds on the projection distance, for a range of projection angles and images. The experiments are all based on simulated projection data obtained by computing the projections of the test images (so-called *phantoms*) in Fig. 5:

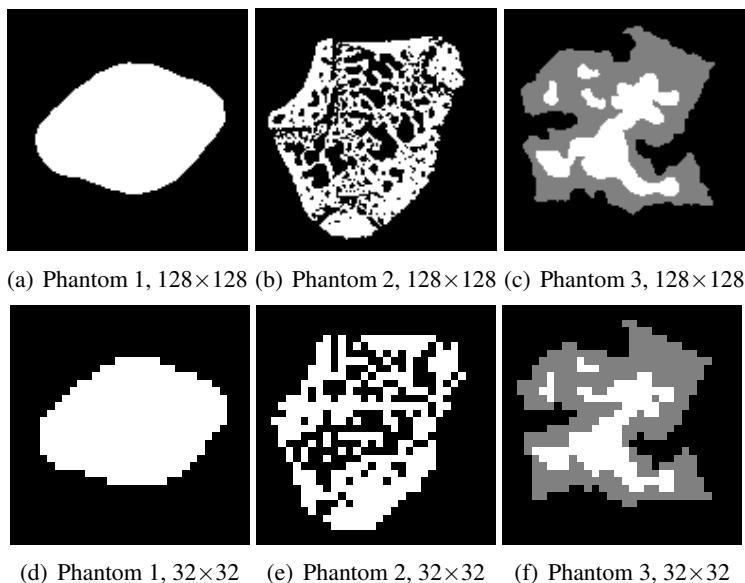


Figure 5: Original phantom images used for the experiments.

Phantom 1 represents a very simple, nearly convex shaped object;

Phantom 2 was constructed from a micro-CT image of a rat bone acquired with a SkyScan 1072 cone-beam micro-CT scanner.

Phantom 3 represents an object with three grey levels and fairly complex boundaries.

As a projection model, we focus on a model for a *continuous* representation of the object, the *strip model* [15, section 7.4.1]. In the strip model, a projection is computed by considering a set of parallel strips in a given direction. For each strip we compute the weighted sum of all the pixels which intersect that strip. Each weight w_{ij} equals the intersection area of the strip and the pixel, see Fig. 6.

8.1. Reconstruction comparison

In this section we compare two variations of the reconstruction algorithm: (i) Algorithm 1 (Fig. 1), (ii) the thresholded version of Alg. 1 presented in Section 6. For the two algorithms mentioned above, we compare quality of the images reconstructed as well as reconstruction time required for each one of them. Also, the quality of the initial solution $\mathbf{x}^{(0)}$ directly affects the final reconstruction. We tested different approximate solutions $\mathbf{x}^{(0)}$.

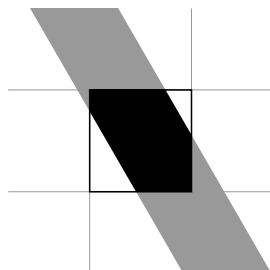
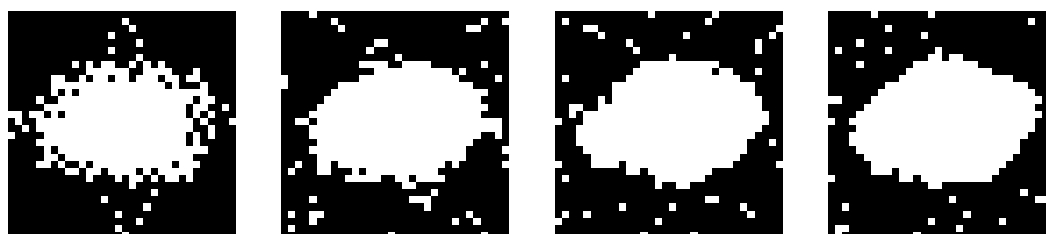


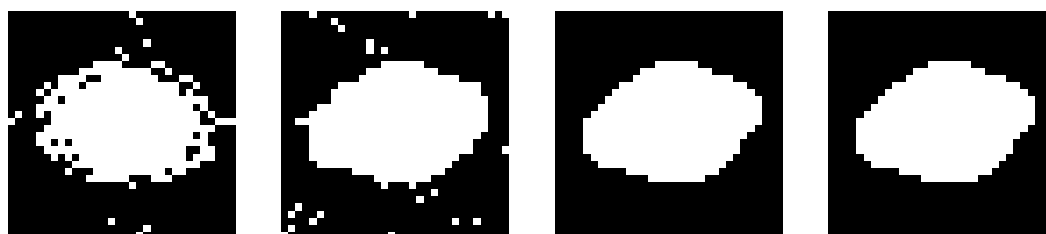
Figure 6: The contribution weight of each pixel to each projection component is given by the intersection area between the pixel and the respective strip.

8.1.1. The quality of the initial approximate solution $x^{(0)}$

The quality of the image reconstructed by the presented algorithms depends on the quality of the initial solution $x^{(0)}$ which was computed by the algorithm presented in section 7.1. Fig. 7 presents reconstructions of phantom image 5(d), using the standard algorithm (Fig. 1), for different number of projection angles. For \mathbf{p} being the vector of projections of phantom 5(d), Fig. 7 shows reconstructions in which the initial approximate solution satisfies $\|\mathbf{W}x^{(0)} - \mathbf{p}\|_{\infty} \leq \varepsilon$ for $\varepsilon \in \{10^{-1}, 5 \times 10^{-1}\}$. Since $\mathbf{W}x^{(0)} \neq \mathbf{p}$, the reconstruction algorithm 1 can only guarantee $\|\mathbf{W}\bar{x} - \mathbf{p}\|_{\infty} < \kappa d + \varepsilon$. If a threshold parameter τ is included, then $\|\mathbf{W}\bar{x} - \mathbf{p}\|_{\infty} < \kappa d + (\|\mathbf{W}\|_{\infty} - \kappa)\tau + \varepsilon$.



(a) $\varepsilon = 5 \times 10^{-1}$ and number of projection angles: left: 2; middle-left: 6; middle-right: 8; right: 12.



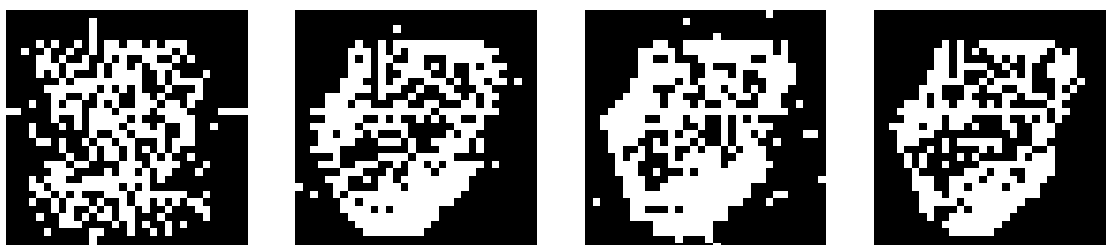
(b) $\varepsilon = 10^{-1}$ and number of projection angles: left: 2; middle-left: 6; middle-right: 8; right: 12.

Figure 7: Reconstructions of phantom 5(d) of dimensions 32×32 using Algorithm 1.

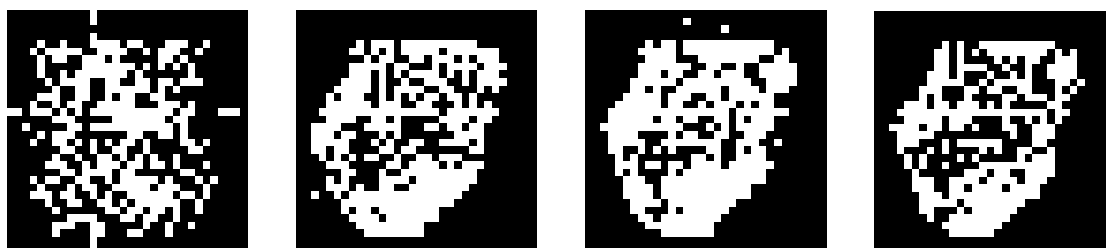
In Fig. 7, it is possible to see white pixels (pixels assigned with a value larger than zero) in regions for which a projected strip that intersects these pixels has a projection value of 0. These line sums accumulate detectable errors and since the error per projection is limited to κd , the error in a region where it is not detectable will diminish. After the reconstruction algorithm, post-processing of the reconstructed image may correct these easily identifiable wrong pixels.

8.1.2. The quality of the reconstruction \bar{x}

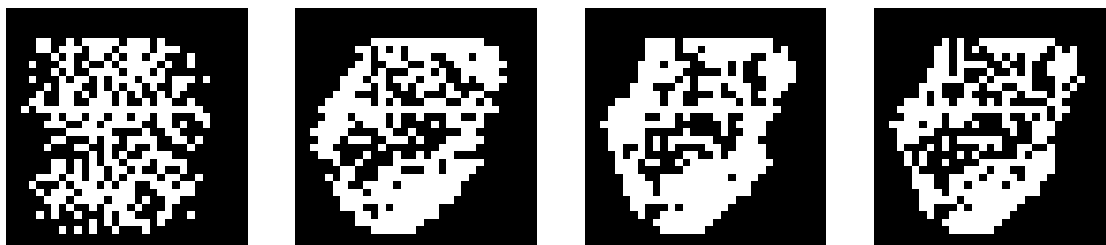
Reconstructed images from the thresholded version of Algorithm 1 for phantoms 5(e) and 5(f) are displayed in Figs. 8 and 9 for $\varepsilon = 10^{-1}$ and $\tau \in \{0, \frac{1}{32}, \frac{1}{\sqrt{32}}\}$. For phantoms 5(b) and 5(c), the reconstructed images from the thresholded version of Algorithm 1 with $\tau \in \{\frac{1}{128}, \frac{1}{\sqrt{128}}\}$, are displayed in Fig. 10.



(a) Reconstructions using Alg. 1. Number of projection angles: left: 2; middle-left: 6; middle-right: 10; right: 14.

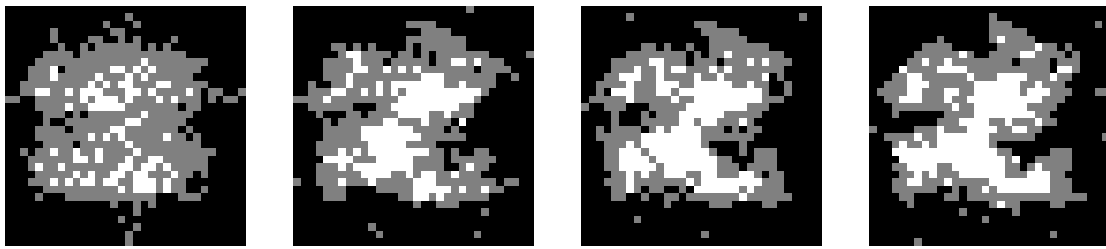


(b) Algorithm 1 with $\tau = \frac{1}{32}$. Number of projection angles: left: 2; middle-left: 6; middle-right: 10; right: 14.

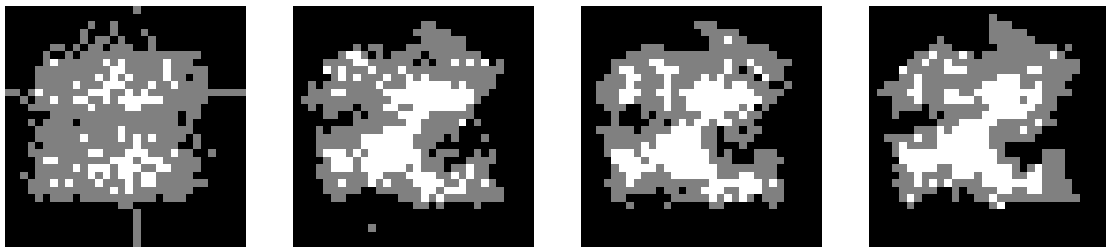


(c) Algorithm 1 with $\tau = \frac{1}{\sqrt{32}}$. Number of projection angles: left: 2; middle-left: 6; middle-right: 10; right: 14.

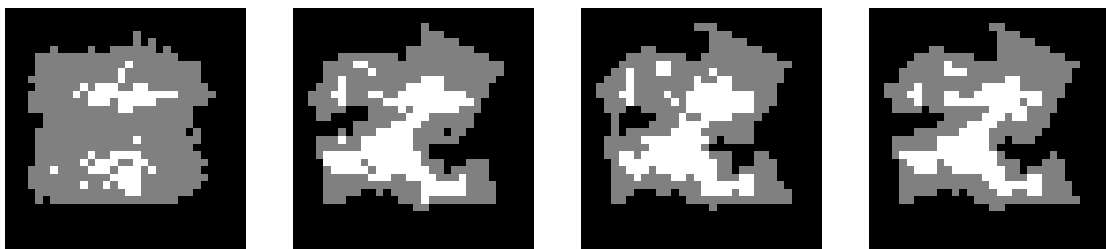
Figure 8: Reconstructions of phantom 5(e) of dimensions 32×32 and $\varepsilon = 10^{-1}$.



(a) Reconstructions using Alg. 1. Number of projection angles: left: 2; middle-left: 6; middle-right: 8; right: 12.

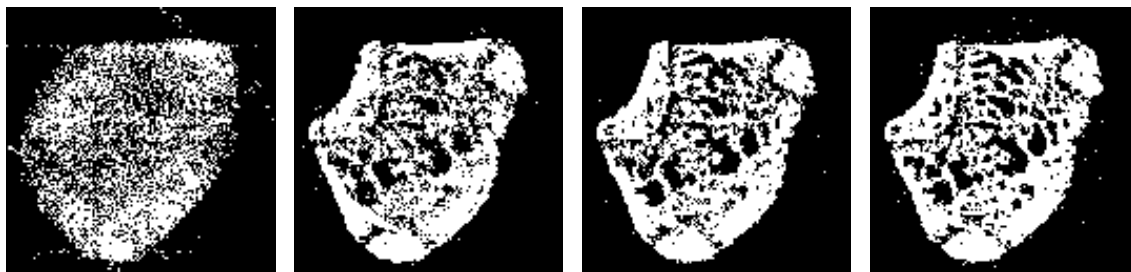


(b) Algorithm 1 with $\tau = \frac{1}{32}$. Number of projection angles: left: 2; middle-left: 6; middle-right: 8; right: 12.

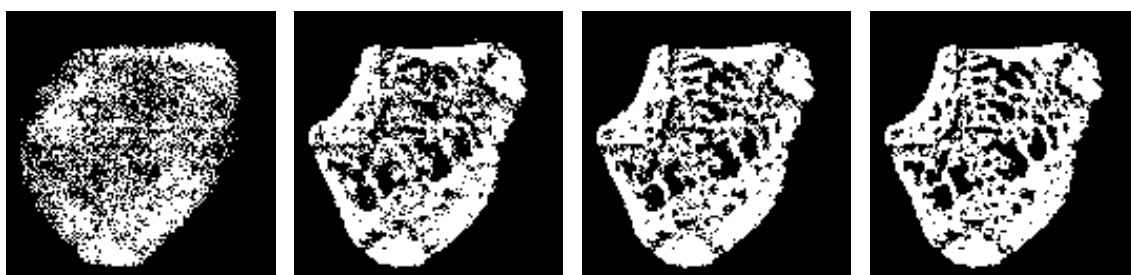


(c) Algorithm 1 with $\tau = \frac{1}{\sqrt{32}}$. Number of projection angles: left: 2; middle-left: 6; middle-right: 8; right: 12.

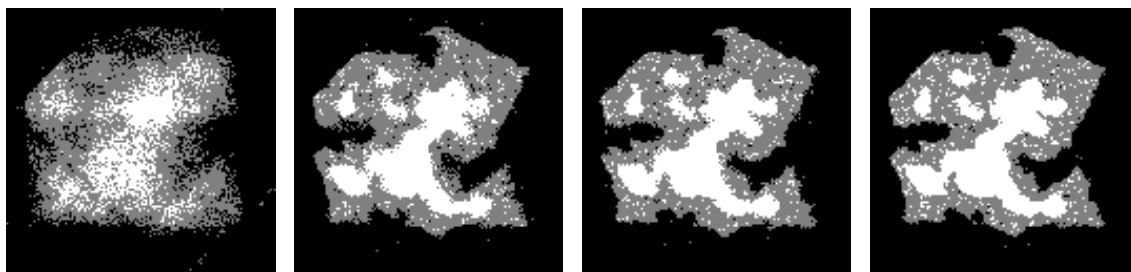
Figure 9: Reconstructions of phantom 5(f) of dimensions 32×32 and $\varepsilon = 10^{-1}$.



(a) Phantom 5(b). Alg. 1 with $\tau = \frac{1}{128}$. N. of projection angles: left: 4; middle-left: 16; middle-right: 20; right: 24.



(b) Phantom 5(b). Alg. 1 with $\tau = \frac{1}{\sqrt{128}}$. N. of projection angles: left: 4; middle-left: 16; middle-right: 20; right: 24.



(c) Phantom 5(c). Alg. 1 with $\tau = \frac{1}{128}$. N. of projection angles: left: 4; middle-left: 16; middle-right: 20; right: 24.



(d) Phantom 5(c). Alg. 1 with $\tau = \frac{1}{\sqrt{128}}$. N. of projection angles: left: 4; middle-left: 16; middle-right: 20; right: 24.

Figure 10: Reconstructions of phantoms 5(b) and 5(c) of dimensions 128×128 and $\varepsilon = 10^{-1}$.

We have measured the projection distance (Pd) of the previously presented reconstructions and compared these with their respective bound (B) as can be seen in Figs. 11 and 12. Also, Figs. 11 and 12 present the *Image distance* (Id) defined as the Euclidean distance between the reconstructed image and the respective phantom image. Notice that despite the bound $\kappa d + (\|\mathbf{W}\|_\infty - \kappa)\tau + \varepsilon$ might increase with the number of projection angles, the quality of the reconstructed images improves, in general. It also means that the projection distance of the reconstructed images decreases with increasing number of projections angles.

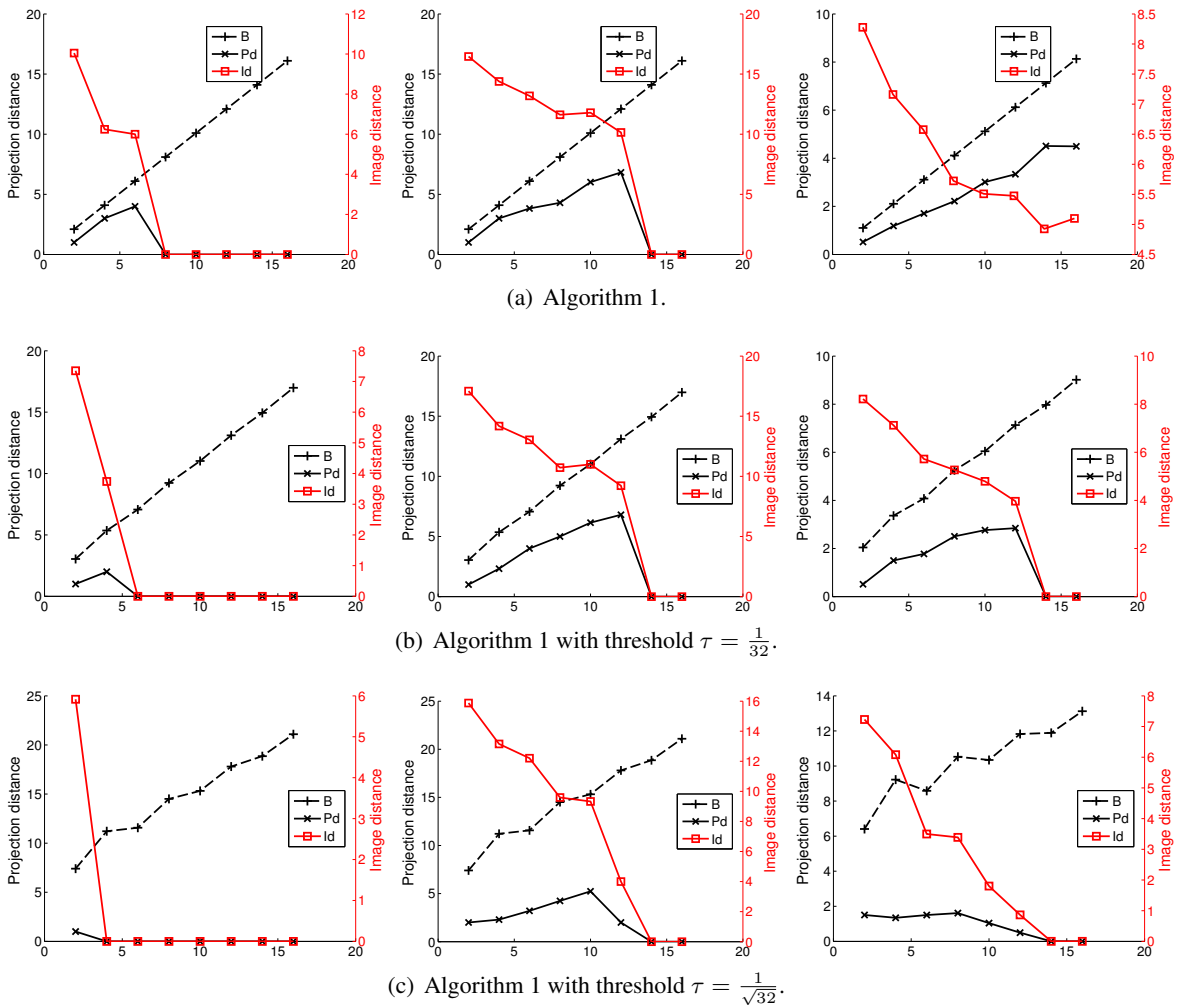


Figure 11: Comparison between the projection distance bound (B) and actual projection distance (Pd) measured in the left vertical axis for increasing number of projection angles. The Image distance (Id) uses the right vertical axis. From left to right: Phantom images 5(d), 5(e) and 5(f).

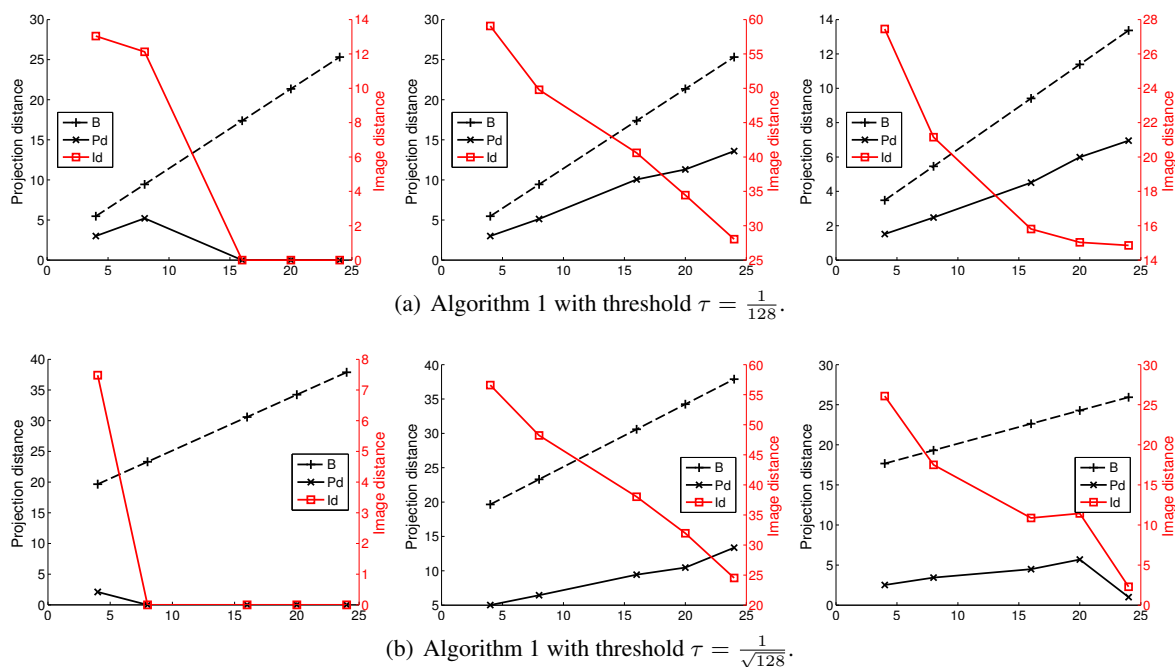


Figure 12: Comparison between the projection distance bound (B) and actual projection distance (Pd) measured in the left vertical axis for increasing number of projection angles. The Image distance (Id) uses the right vertical axis. From left to right: Phantom images 5(a), 5(b) and 5(c).

We remark that the key advantage of the proposed algorithm is the guaranteed error bound of the resulting reconstruction. Alternative algorithms, which do not yield such a bound, can yield more accurate reconstructions in practice, using less computation time. As an example, we refer to the DART algorithm in [2], where similar phantoms are used in the experiments. It can be seen in Fig. 7 of [2] that Phantom 2 and 3 (corresponding to Phantom 5 and 7 in [2]) are reconstructed by DART far more accurately from few projections compared to our proposed method, yet without any guarantee on the reconstruction error.

8.1.3. Running time

The algorithm variants with different threshold parameters compared in this section have different running times, which can vary from 0.1 seconds for the fastest run up to 10's of minutes for the slowest runs. All experiments were run on a workstation PC using a single core of an Intel Core-i5 CPU at 2.8GHz. As the computation time is machine dependent, we display a *relative time*, which is computed by dividing the time needed for a reconstruction by the time of the fastest reconstruction among all of the runs presented in the experiments. The relative time comparison between the reconstruction algorithms can be seen in Table 1. For the phantoms in Fig. 5(a), 5(b), and 5(c), the running time was prohibitively large for the basic algorithm, which does not use a threshold. The timing results for these cases are not included in the table.

N. proj.	4	16	24	N. proj.	4	16	24
$\tau = \frac{1}{128}$	5387	34.8	34.7	$\tau = \frac{1}{128}$	5×10^5	2×10^5	7×10^4
$\tau = \frac{1}{\sqrt{128}}$	817	35.6	35.6	$\tau = \frac{1}{\sqrt{128}}$	3765	1881	423
(a) Phantom 5(a).				(b) Phantom 5(b)			
N. proj.	4	16	24	N. projections	2	10	16
$\tau = \frac{1}{128}$	3×10^5	1×10^5	1×10^5	$\tau = 0$	37.7	3.25	1
$\tau = \frac{1}{\sqrt{128}}$	1361	428	79.1	$\tau = \frac{1}{32}$	12.2	1	1
(c) Phantom 5(c).				(d) Phantom 5(d).			
N. projections	2	10	16	N. projections	2	10	16
$\tau = 0$	83.7	59.8	9.5	$\tau = 0$	82.7	91	61.1
$\tau = \frac{1}{32}$	36.8	15.7	1	$\tau = \frac{1}{32}$	18.7	10	1.6
$\tau = \frac{1}{\sqrt{32}}$	5.2	1.8	1	$\tau = \frac{1}{\sqrt{32}}$	1.8	1	1
(e) Phantom 5(e).				(f) Phantom 5(f).			

Table 1: Relative reconstruction time for Alg. 1 with different thresholds. $\varepsilon = 10^{-1}$

We have measured the number of iterations of the algorithm in each experiment and, in general, it decreases with increasing number of projection angles. In the experiments, the maximum number of iterations required was less than 60% of the number of image pixels.

The key bottleneck in the computation complexity of the algorithm is the fact that a large system of equations (up to the same order as the tomography problem itself) must be solved to obtain a ghost image, and that this computation will have to be performed many times (at most once for each pixel). As there is a stochastic component in the computation of the ghost images, and as the particular computational route followed by the algorithm depends strongly on the input projections, a meaningful (somewhat sharp) complexity analysis of the algorithm is highly challenging. We consider the algorithm proposed here as a proof-of-concept method that requires further optimization and analysis to be useful for reconstructing large images.

9. Outlook and conclusions

In this article, we have presented a reconstruction algorithm which computes an image that has only pixel values from a given finite set, and for which the projection distance to the unknown ground truth object is bounded. Contrary to alternative methods, which often perform well in practice but do not come with guarantees, our approach is specifically designed to yield a reconstruction for which the projections are provably close to the given projection data. Our experimental results for simulated phantom images demonstrate that the algorithm not only computes images that approximately match the given projections, but also yields reconstructions that resemble the unknown original image, even if only a small number of

projections are used. The techniques of using ghost images and thresholding employed in our algorithm are quite generic and leave a large degree of freedom in parameter selection. For example, the particular switching element considered in a step of Procedure 1 is currently determined randomly, while it could also be chosen depending on features of the current reconstruction.

In its current form, the proposed algorithm is neither optimized for speed of computation, nor for quality of the reconstructed image. To make the step In future work, we intend to optimize the algorithm, and make it more robust, to the extend where it can be applied to larger images, and to experimental data.

References

- [1] Batenburg, K. J.: A network flow algorithm for reconstructing binary images from continuous X-rays, *J. Math. Im. Vision*, **30**(3), 2008, 231–248.
- [2] Batenburg, K. J., Sijbers, J.: DART: a Practical Reconstruction Algorithm for Discrete Tomography, *IEEE Trans. Image Processing*, **20**(9), 2011, 2542–2553.
- [3] Ben-Israel, A., Greville, T. N. E.: *Generalized inverses: Theory and Applications*, Canadian Math. Soc., 2002.
- [4] Beck, J., Fiala, T.: Integer-making theorems, *Discr. Appl. Math.*, (3), 1981, 1–8.
- [5] Björck, Å.: *Numerical Methods for Least Square Problems*, SIAM, Linköping University, Sweden, 1996.
- [6] Censor, Y.: Row-action methods for huge and sparse systems and their applications, *SIAM review*, **23**(4), 1981, 444–466.
- [7] Gardner, R. J., Gritzmann, P., Prangenberg, D.: On the computational complexity of reconstructing lattice sets from their X-rays, *Discrete Math.*, **202**, 1999, 45–71.
- [8] Gordon, R., Bender, R., Herman, G. T.: Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography, *J. Theor. Biol.*, **29**(3), 1970, 471–481.
- [9] Hajdu, L., Tijdeman, R.: Bounds for approximate discrete tomography solutions, *arXiv*, **1207.3933**, 2012.
- [10] Herman, G. T.: *Fundamentals of Computerized Tomography: Image reconstruction from projections*, Springer, 2009.
- [11] Herman, G. T., Kuba, A., Eds.: *Discrete Tomography: Foundations, Algorithms and Applications*, Birkhäuser, Boston, 1999.
- [12] Herman, G. T., Kuba, A., Eds.: *Advances in Discrete Tomography and its Applications*, Birkhäuser, Boston, 2007.
- [13] Jinschek, J. R., Batenburg, K. J., Calderon, H. A., Kilaas, R., Radmilovic, V., Kisielowski, C.: 3-D reconstruction of the atomic positions in a simulated gold nanocrystal based on discrete tomography, *Ultramicroscopy*, **108**(6), 2007, 589–604.
- [14] Joseph, P. M.: An improved algorithm for reprojecting rays through pixel images, *IEEE Trans. Med. Imag.*, **MI-1**(3), 1982, 192–196.
- [15] Kak, A. C., Slaney, M.: *Principles of Computerized Tomographic Imaging*, SIAM, 2001.
- [16] Natterer, F.: *The mathematics of computerized tomography*, John Wiley & Sons, 1986, ISBN 0471909599, 9780471909590.

- [17] Saad, Y.: *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, USA, 2003, ISBN 0898715342.
- [18] Schüle, T., Schnörr, C., Weber, S., Hornegger, J.: Discrete tomography by convex-concave regularization and D.C. programming, *Discr. Appl. Math*, **151**, 2005, 229–243.
- [19] Van Aert, S., Batenburg, K. J., Rossell, M., Erni, R., Van Tendeloo, G.: Three-dimensional atomic imaging of crystalline nanoparticles, *Nature*, **470**, 2011, 374–377.
- [20] van der Sluis, A., van der Vorst, H. A.: SIRT and CG-type methods for the iterative solution of sparse linear least-squares problems, *Linear Algebra Appl.*, **130**, 1990, 257–302.
- [21] Zhua, J., Li, X., Ye, Y., Wang, G.: Analysis on the strip-based projection model for discrete tomography, *Discrete Appl. Math.*, **156**(12), 2008, 2359–2367.