Generic iterative subset algorithms for discrete tomography

K.J. Batenburg * J. Sijbers [1]

University of Antwerp, Vision Lab,Universiteitsplein 1, B-2610 Wilrijk, Belgium

Abstract

Discrete tomography deals with the reconstruction of images from their projections where the images are assumed to contain only a small number of grey values. In particular, there is a strong focus on the reconstruction of binary images (binary tomography). A variety of binary tomography problems have been considered in the literature, each using different projection models or additional constraints. In this paper, we propose a generic iterative reconstruction algorithm that can be used for many different binary reconstruction problems. In every iteration, a subproblem is solved based on at most two of the available projections. Each of the subproblems can be solved efficiently using network flow methods. We report experimental results for various reconstruction problems. Our results demonstrate that the algorithm is capable of reconstructing complex objects from a small number of projections.

Key words: Discrete tomography, Network flows

1 Introduction

Tomography deals with the reconstruction of images from their projections. In practice, the goal of tomography is usually to reconstruct some physical object, of which projection images are acquired by a tomographic scanner. Several efficient algorithms exist for computing reconstructions, such as Filtered Backprojection (FBP) and Algebraic Reconstruction Techniques (ART,

* Send correspondence to Joost Batenburg
Email addresses: joost.batenburg@ua.ac.be (K.J. Batenburg), jan.sijbers@ua.ac.be (J. Sijbers).
URLs: http:\\visielab.ua.ac.be\staff\batenburg (K.J. Batenburg), http:\\visielab.ua.ac.be\staff\sijbers (J. Sijbers).

[1] K.J. Batenburg and J. Sijbers are both postdoctoral fellows of the F.W.O. (Fund for Scientific Research - Flanders, Belgium)

see, e.g., (1)). These algorithms are capable of computing accurate reconstructions, but require a large number of projections. The field of *discrete tomography* focuses on the reconstruction of images that consist of only a few different grey levels (2; 3). In particular, there is a strong focus on the reconstruction of binary images. By using the prior knowledge of the grey levels that can occur in the reconstructed image, it is often possible to vastly reduce the number of projections required for an accurate reconstruction.

In this paper, we consider the problem of reconstructing binary images from a small number of projections. The exact definition of the term *projection* varies among different reconstruction problems. Many authors have studied the problem of reconstructing binary images from *discrete X-rays* (4; 5; 6). This problem can be solved in polynomial time for any set of two projections, but is NP-hard for any set of three or more projections (4). In (7), Frosini and Nivat consider the reconstruction of binary images from projections measured by sliding a fixed window over the image, counting the total number of ones within the window for each window position. Gardner et al. have recently investigated the case of discrete point X-rays (8). In all these reconstruction problems, the term *projection* refers to a partition of the pixels in the image, such that for each subset in the partition the total number of 1s in the unknown binary image is given. This general projection concept was also considered in (9).

Projection models based on counting the number of elements in discrete sets have important practical applications in electron microscopy. It is now possible to count the number of atoms in projected columns of a nanocrystal, viewed from different directions (10). Since atoms are discrete objects, the projection measurements are also discrete. In many other applications of tomography, it can be assumed that the image should consist of only two grey values, but the reconstructed object does not have an intrinsic lattice structure. For example, if one scans a slice of a homogeneous plastic object in a tomographic X-ray scanner, its reconstruction should be a binary image, indicating the presence or absence of plastic in each pixel. The measured projection data can now be considered as line integrals of the density of the object, through the reconstructed volume. Such reconstruction problems are studied in the field of discrete tomography (11; 12; 13) as well as in *geometric tomography*, see (14) for an overview. Although much of the work on geometric tomography is concerned with the reconstruction of rather specific objects, such as convex or star-shaped objects, the domain of geometric objects can also be as general as the collection of measurable sets. The problem of reconstructing a binary image from its projections can be considered as reconstructing a subset of the plane, corresponding to the interior of the object. In the present paper, we do not aim at *exact* reconstructions: the reconstruction is computed on a pixel grid, at a finite resolution.

The problem of reconstructing a binary image from two of its projections, taken along discrete lines, can be solved efficiently using methods from the field of Network Flows. The reconstruction problem can be modeled as an integral max flow problem in a certain graph, which can be solved in polynomial time (15). Extending the network flow approach to the case of more than two projections is a nontrivial task. For any set of more than two projection directions, the problem of reconstructing a binary image from its discrete X-rays along these directions is NP-hard (4). Recently, an iterative network flow algorithm was proposed for this problem in (15). The algorithm is very successful in reconstructing images that are relatively smooth, containing large areas of connected 0s or 1s. For such images, prior knowledge about the fact that the reconstruction is smooth can efficiently be incorporated into the iterative algorithm, using a local weight function.

In this paper we describe an iterative algorithm that can be used for any of the reconstruction problems mentioned above. Similar to the network flow algorithm for discrete X-rays, our generalized algorithm works best for images that are relatively smooth. Any arbitrary (possibly partial) partition of the pixels can be used to define a projection. It is also possible to mix projections from different projection models, such as a combination of discrete X-rays with projections from a sliding window. We restrict ourselves to the reconstruction of two-dimensional images. All methods described in this paper can be easily generalized to a three- or higher-dimensional setting.

In each iteration of the algorithm, a subproblem is solved, corresponding to a subset of at most two of the available projections. These subproblems are typically highly underdetermined and a large number of solutions may exist. Therefore, a solution is sought that closely resembles the reconstruction computed in the previous iteration, in addition to satisfying the current subset of projections. The comparison between the current reconstruction and the previous reconstruction is performed pixel-by-pixel, assigning a weight to each pixel that represents the desirability of retaining the value from the previous reconstruction. A preference for smooth regions is incorporated by assigning higher weights to pixels that are surrounded by other pixels having the same value.

In Section 2, we introduce some basic concepts and notation and propose a general model for describing projections of binary images. In Section 3, we describe how three different binary reconstruction problems can be formulated within our general projection model. Section 4 describes two subproblems of the reconstruction problem and their solution by network flow methods. The first subproblem uses only one of the available projections; the second problem uses a pair of projections. These subproblems are used as building blocks in our iterative algorithm. In Section 5, an iterative algorithm is described for reconstructing binary images from more than two projections. We performed

simulation experiments for various reconstruction problems, using different types of projections. Section 6 provides reconstruction results for a variety of test images, based on the three reconstruction problems from Section 3. Section 7 concludes.

2 Preliminaries

We consider the problem of reconstructing a set $F \subset \mathbb{Z}^2$ from a number of *projections*, which we will define more precisely below. Let $A \subset \mathbb{Z}^2$ be a finite set, called the *reconstruction area*. For any instance of the reconstruction problem, A is a known fixed set and we assume that $F \subset A$. In the discrete tomography literature, A is typically chosen to be the interior of an $m \times n$ rectangle, such that any $F \subset A$ can be represented as a binary matrix $\bar{F} \in \{0,1\}^{m \times n}$, with $\bar{F}_{ij} = 1$ iff $(i,j) \in A$. Alternatively, F can be represented as a function $A \to \{0,1\}$. Throughout this paper, we will use these three representations interchangeably.

Put $\mathbb{N}_0 = \{k \in \mathbb{Z} : k \geq 0\}$. For any finite set S, we denote the number of elements of S by $|S|$.

Let $k \in \mathbb{N}_0$, $\mathbf{S} = \{S_1, \ldots, S_k\} \subset 2^F$. We call $\mathbf{S}$ a *partition* of A if all elements of $\mathbf{S}$ are pairwise disjoint and $\cup_{S \in \mathbf{S}} S = A$.

Define

$$\mathbf{P}^{\mathbf{S}} = \{P : \mathbf{S} \to \mathbb{N}_0 \mid P(S) \leq |S| \text{ for all } S \in \mathbf{S}\}.$$

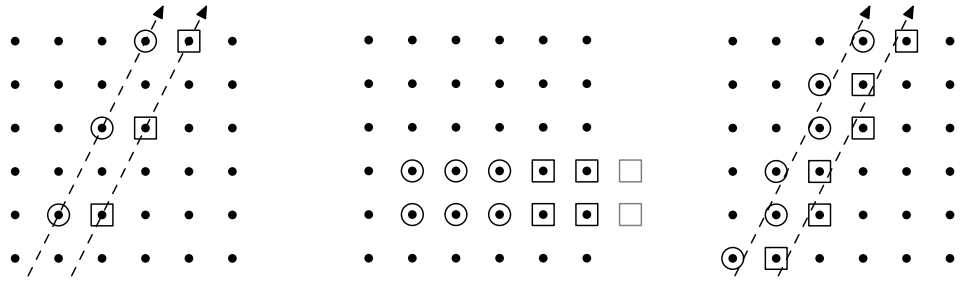
For a binary image $F \subset A$ and partition $\mathbf{S}$, define the *projection* $P_F^{\mathbf{S}} \in \mathbf{P}^{\mathbf{S}}$ of F along $\mathbf{S}$ by

$$P_F^{\mathbf{S}}(S) = |F \cap S|. \tag{1}$$

We refer to the values $P_F^{\mathbf{S}}(S)$ as *set sums*. The set $\mathbf{P}^{\mathbf{S}}$ contains exactly the projections along $\mathbf{S}$ of all binary images $F \subset A$. The *reconstruction problem* consists of finding a binary image that has prescribed projections along several partitions:

Problem 1 *Let $A \subset \mathbb{Z}^2$. Let $d > 0$ and let $\mathbf{S}_1, \ldots, \mathbf{S}_d$ be partitions of A. Let $Q_1 \in \mathbf{P}^{\mathbf{S}_1}, \ldots, Q_d \in \mathbf{P}^{\mathbf{S}_d}$ be given functions. Find $F \subset A$ such that $P_F^{\mathbf{S}_i} = Q_i$ for $i = 1, \ldots, d$.*

Problem 1 is not guaranteed to have a solution. For example, the same set S can occur in two different partitions, each with a different set sum. In practical applications of tomography, the projections will typically contain noise or other errors, leading to an inconsistent reconstruction problem. For



(a) discrete X-rays (b) rectangular scan (c) continuous X-rays

Fig. 1. Three different projection models. (a): discrete X-rays; (b): rectangular scan; (c): continuous X-rays along strips, discretized as discrete segments.

$i = 1, \ldots, d$, define the *projection distance* $D_i(F)$ by

$$D_i(F) = \sum_{S \in \mathbf{S}_i} |P_F^{\mathbf{S}_i}(S) - Q_i(S)|. \tag{2}$$

Define the *total projection distance* $D_{tot}(F)$ by $D_{tot}(F) = \sum_{i=1}^{d} D_i(F)$. We reformulate the reconstruction problem as an optimization problem:

Problem 2 *Let $A, \mathbf{S}_1, \ldots, \mathbf{S}_d, Q_1, \ldots, Q_d$ be as in Problem 1. Find $F \subset A$ such that $D_{tot}(F)$ is minimal.*

Clearly, any solution of Problem 1 is also a solution of Problem 2. Conversely, if Problem 1 has a solution, then any solution of Problem 2 is also a solution of Problem 1.

For each given projection Q_i, define $U(Q_i) = \sum_{S \in \mathbf{S}_i} Q_i(S)$. Note that if Problem 1 has a solution, then $U(Q_i) = |F|$ for $i = 1, \ldots, d$. If the given projections have been obtained by measuring some physical object F, but have been perturbed by noise (without a bias), we can therefore estimate that $|F| \approx \frac{1}{d} \sum_{i=1}^{d} U(Q_i)$.

3 Three different reconstruction problems

We will now relate our general tomography model, based on arbitrary partitions, to three reconstruction problems from the literature, each corresponding to a different choice of the partitions $\mathbf{S}_1, \ldots, \mathbf{S}_d$. This list is by no means exhaustive, but serves as a basis for the experimental results in Section 6.

3.1 Discrete X-rays

The problem of reconstructing binary images from discrete X-rays has been studied extensively in the context of discrete tomography. Using the model of partitions to define the projections of an image, each partition $\mathbf{S}_i$ consists of sets of points that correspond to parallel discrete lines, defined by a direction $(a, b) \in \mathbb{Z}^2$ (see Figure 1a). Put $T = \{t : ay - bx = t, (x, y) \in A\}$. The partition $\mathbf{S} = \{S_t : t \in T\}$ is now given by

$$S_t = \{(x, y) \in A : ay - bx = t\}.$$

The reconstruction problem for discrete X-rays can be solved in polynomial time for any set of at most two projections (see, e.g., (15)), while it is NP-hard for any set of three or more distinct projections (4).

3.2 Rectangular scan

Another kind of binary tomography problem is obtained if the reconstruction area is scanned by moving a fixed rectangular window across it, each time obtaining the total number of 1's within the window. This problem was studied in (7). Put $A = \{(i, j) \in \mathbb{Z}^2 : 0 \leq i < n, 0 \leq j < m\}$. Let $1 \leq p \leq n$, $1 \leq q \leq m$. For $0 \leq i < n$, $0 \leq j < m$, define

$$S_{i,j}^{p,q} = \{(i + c, j + r) : 0 \leq c < p, 0 \leq r < q\} \cap A.$$

Note that this definition is slightly different from the one used in (7). We allow the window to be partially outside the boundary of the reconstruction area (see Figure 1b).

For $0 \leq a < p$, $0 \leq b < q$, define

$$\mathbf{S}^{a,b} = \{S_{a+ip,b+jq}^{p,q} : a + ip < n, b + jq < m\}$$

Each set $\mathbf{S}^{a,b}$ is a partition of A. In (7), it is assumed that the total number of 1s is given for each position of the window. We consider a slightly different version of the problem, where the projections are only given for certain values of (a, b), but not all. In Section 6, we show that if the image is sufficiently smooth, an accurate reconstruction can often be obtained from a number of projections that is much smaller than the total number pq of possible projections.

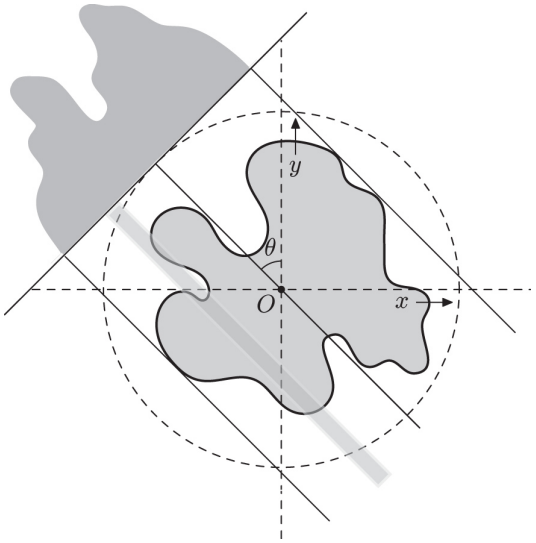


Fig. 2. Basic setting of transmission tomography with strips of fixed width.

3.3 Continuous X-rays

The third projection model is directly related to the practical application of X-ray tomography. Let $f : \mathbb{R}^2 \to \{0, 1\}$ be an unknown function with bounded support. Let $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2})$. For $l, r \in \mathbb{R}$, $l < r$, define

$$P_\theta^f(l, r) = \iint_{l \le x \cos\theta + y \sin\theta < r} f(x, y)\, \mathrm{d}x\, \mathrm{d}y \tag{3}$$

In X-ray scanners, the value $P_\theta^f(l, r)$ is typically measured for consecutive strips of fixed width by an array of adjacent detectors, see Figure 2.

To model the reconstruction problem from continuous X-rays in the context of Problem 2, we approximate the continuous strips by discrete segments. Suppose that $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. For $u \in \mathbb{Z}$, put

$$S_\theta^u = \{(x, y) \in A : u \le x + y \tan\theta < u + 1\}$$

Note that S_θ^u is nonempty for only finitely many values of u. Define

$$\mathbf{S}^\theta = \{S_\theta^u : u \in \mathbb{Z}, S_\theta^u \neq \emptyset\}$$

The partition $\mathbf{S}^\theta$ partitions A into discrete segments such that any two adjacent horizontal pixels are contained in adjacent segments, see Figure 1c. Note that the width of each discrete segment corresponds to a strip of width $u \cos\theta$ on the detector, whereas the width of each physical detector cell is constant for all angles θ. Therefore, we measured projection data must be interpolated to obtain the measured projection for each discrete segment. In the experiments

from Section 6, we used linear interpolation based on projections measured using a fixed strip width of 1 for all projection angles.

For projection angles $\theta \in [-\frac{\pi}{2}, -\frac{\pi}{4}) \cup (\frac{\pi}{4}, \frac{\pi}{2})$, a corresponding partition can be defined similarly, exchanging the role of x and y.

4 Basic subproblems

In the next section, we will describe a generic iterative algorithm that can be used for any of the reconstruction problems described in the previous section. In each iteration, a subproblem is solved, based on either one or two of the available projections. In this section, we introduce both subproblems and describe how they can be solved efficiently.

4.1 Reconstruction from one projection

We now define a reconstruction problem similar to Problem 2, but using only one projection. Clearly, Problem 2 can have a huge number of solutions in that case. As we intend to use the reconstruction problem from one projection as a building block in an iterative algorithm that uses information from all projections, we augment the projection difference with an additional term, that allows for the incorporation of prior knowledge to select a particular reconstruction among all solutions.

Problem 3 *Let $A \subset \mathbb{Z}^2$. Let $\mathbf{S}$ be a partition of A. Let $Q \in \mathbf{P}^{\mathbf{S}}$ and $W : A \to \mathbb{Z}$ be given functions. Let $\alpha, t \in \mathbb{Z}$, $0 \leq t \leq |A|$. Find $F \subset A$ such that $|F| = t$ and*

$$\alpha \sum_{S \in \mathbf{S}} |P_F^{\mathbf{S}}(S) - Q(S)| - \sum_{(i,j) \in A} W(i,j) F(i,j)$$

is minimal.

For a pixel $(i, j) \in A$, we refer to the value $W(i, j)$ as its *pixel weight*. The pixel weights allow for the incorporation of prior knowledge in the reconstruction problem.

We first consider a special case of Problem 3, where $\alpha > \sum_{(i,j) \in A} |W(i,j)|$. In this case, for any solution $F \subset A$ of Problem 3, the projection distance $D(F) = |P_F^{\mathbf{S}}(S) - Q(S)|$ must be minimal. For any $z \in \mathbb{Z}$, define $|z|^+ = \max(z, 0)$. For any $F \subset A$ with $|F| = t$ we have

$$t = U(Q) + \sum_{S \in \mathbf{S}} (|P_F^{\mathbf{S}}(S) - Q(S)|^+ - |Q(S) - P_F^{\mathbf{S}}(S)|^+). \tag{4}$$

Hence,

$$D(F) = \sum_{S \in \mathbf{S}} (|P_F^{\mathbf{S}}(S) - Q(S)|^+ + |Q(S) - P_F^{\mathbf{S}}(S)|^+) = \\ 2 \sum_{S \in \mathbf{S}} |P_F^{\mathbf{S}}(S) - P(S)|^+ + U(Q) - t \tag{5}$$

Therefore, only the sets $F \subset A$ with $|F| = t$ that minimize $\sum_{S \in \mathbf{S}} |P_F^{\mathbf{S}}(S) - P(S)|^+$ can be a solution of Problem 3.

Note that if $t \leq U(Q)$, D(F) is minimal iff $P_F^{\mathbf{S}}(S) \leq P(S)$ for all $S \in \mathbf{S}$. In this case we can use a simple greedy algorithm to solve Problem 3: order the elements (i, j) of A in decreasing order of $W(i, j)$. Starting with an empty set F, iteratively add new elements to F, each time choosing the $(i, j) \in A$ with heighest weight $W(i, j)$ such that the set sum of its corresponding set S does not become larger than $Q(S)$.

If $t > U(Q)$, $D(F)$ is minimal iff $P_F^{\mathbf{S}}(S) \geq P(S)$ for all $S \in \mathbf{S}$. Again, Problem 3 can be solved using a greedy algorithm, first filling up all sets $S \in \mathbf{S}$ with elements using the heighest total weight, and then adding $t - U(Q)$ additional elements to F that have the heighest pixel weight.

Sorting the elements of $|A|$ in decreasing order of $W(i, j)$ takes time $O(n \log n)$. Therefore, Problem 3 can be solved in time $O(n \log n)$ if $\alpha > \sum_{(i,j) \in A} |W(i, j)|$.

We will now describe how Problem 3 can be solved as a minimum cost flow problem in a graph, also if $\alpha \leq \sum_{(i,j) \in A} |W(i, j)|$. For an introduction to the theory of network flows, we refer to the book (16).

With a given instance of Problem 3, we associate a directed graph $G = (V, E)$, called the *associated graph*. The basic structure of this graph is shown in Figure 3a. It contains a node S (the *source*), a node T (the *sink*) and one layer of nodes for each set in $S_a \in \mathbf{S}$, called *set nodes*. Put $k = |\mathbf{S}|$. We label the set nodes as $S_1, \ldots, S_k$.

To each edge $e \in E$, we assign a *cost* $c(e)$ and a *capacity* $u(e)$. There are two edges from the source node S to each set node S_a, labeled by $(S, S_a)_r$ and $(S, S_a)_e$ and having capacities $Q(S_a)$ and $|S_a| - Q(S_a)$ respectively.

The edge $(S, S_a)_r$ is called a *regular set edge*, whereas the edge $(S, S_a)_e$ is called an *excess set edge*. Each regular set edge is assigned a cost of 0; each excess set edge is assigned a cost of 2α.

For every $(i, j) \in A$, there is exactly one edge $e_{ij} = (S_a, T) \in E$ connecting a set node to the sink node, corresponding to the set S_a that contains (i, j).

These edges are called the *pixel edges* (depicted as squares in Figure 3). All pixel edges e_{ij} have a capacity of 1 and a cost of $-W(i,j)$.

A mapping $Y : E \to \mathbb{Z}$ is called an *integral flow* in G if $\sum_{(v,w)\in E} Y(v,w) = \sum_{(w,v)\in E} Y(w,v)$ for all $w \in V\backslash\{S,T\}$. The flow Y is called *admissable* if $0 \leq Y(e) \leq u(e)$ for all $e \in E$ and *set optimal* if

$$Y((S,S_a)_r) = \min(Q(S), \sum_{(i,j)\in S_a} Y(i,j)) \qquad \text{for all } S_a \in \mathbf{S}.$$

A set optimal flow uses the excess set edges only in case the corresponding regular set edge is completely saturated. Let $\mathbf{Y}$ be the set of all set optimal flows in G. We remark that any $Y \in \mathbf{Y}$ is completely determined by its value $Y(i,j)$ in the pixel edges of G. There is a 1-1 correspondence between the set $\mathbf{A} = 2^A$ of all subsets of A and the set optimal flows in G, induced by the mapping $\Phi : 2^{\mathbf{A}} \to \mathbf{Y}$:

$$\Phi(F)(e_{ij}) = F(i,j).$$

Let $F \subset A$. The total cost of the flow $Y_F = \Phi(F)$ is given by

$$C(Y_F) = \sum_{e\in E} c(e) = \; - \sum_{(i,j)\in F} W(i,j) + 2\alpha \sum_{S\in\mathbf{S}} |P_F^{\mathbf{S}}(S) - Q(S)|^+. \qquad (6)$$

Using Eq. (5), we obtain

$$C(Y_F) = \; - \sum_{(i,j)\in F} W(i,j) + \alpha \sum_{S\in\mathbf{S}} |P_F^{\mathbf{S}}(S) - Q(S)| - \alpha U(Q) + \alpha t. \qquad (7)$$

Since both $U(Q)$ and t are constant, Problem 3 is equivalent to the problem of finding a minimum cost integral flow of size t in the associated graph. The problem of finding a minimum cost flow of a predefined size in a graph can be solved in polynomial time. A variety of algorithms are available for computing minimum cost flows, see (16) for an overview.

4.2 Reconstruction from two projections

Similar to Problem 3, we define a reconstruction problem using two of the available projections.

Problem 4 *Let $A \subset \mathbb{Z}^2$. Let $\mathbf{S}_1, \mathbf{S}_2$ be partitions of A. Let $Q_1 \in \mathbf{P}^{\mathbf{S}_1}, Q_2 \in \mathbf{P}^{\mathbf{S}_2}$ and $W : A \to \mathbb{Z}$ be given functions. Let $\alpha, t \in \mathbb{Z}$, $0 \leq t \leq |A|$. Find $F \subset A$ such that $|F| = t$ and*

$$\sum_{S\in\mathbf{S}_1} |P_F^{\mathbf{S}_1}(S) - Q_1(S)| + \sum_{S\in\mathbf{S}_2} |P_F^{\mathbf{S}_2}(S) - Q_2(S)| - \sum_{(i,j)\in A} W_{ij}F(i,j)$$

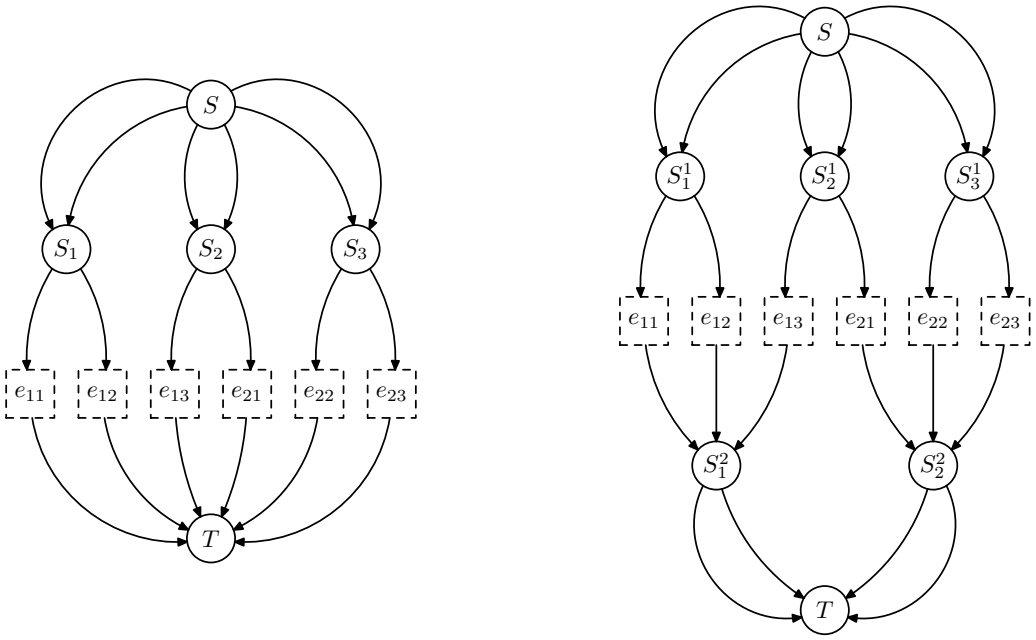


Fig. 3. Left: Structure of the associated graph for Problem 3 (single projection); Right: Structure of the associated graph for Problem 4 (two projections).

is minimal.

Problem 4 can be solved similarly to Problem 3, using an associated graph that contains an extra layer of set nodes (see Figure 3b). The basic idea of using network flow methods for the reconstruction of binary images from two projections was first described by Gale in 1957 (17), in the context of reconstructing binary *matrices* from their row and column sums. Contrary to Problem 3, a greedy approach cannot be used here, even if α is very large. The associated graph for Problem 4 contains an additional layer of set nodes, corresponding to the sets in the second partition. Each set node S_b in the second layer contains a regular set edge and an excess set edge to the sink node T, having capacities $Q_2(S_b)$ and $|S_b| - Q_2(S_b)$ respectively. Again, any set optimal flow is completely determined by the flow through the pixel edges.

The cost of a set optimal flow $Y_F = \Phi(F)$ is given by

$$C(Y_F) = -\sum_{(i,j)\in F} W(i,j) + \alpha\Big(\sum_{S\in\mathbf{S}_1} |P_F^{\mathbf{S_1}}(S) - Q_1(S)| + \sum_{S\in\mathbf{S}_2} |P_F^{\mathbf{S_2}}(S) - Q_2(S)|\Big) - \alpha(U(Q_1) + U(Q_2) - 2t) \tag{8}$$

Hence, Problem 4 can be solved by computing a minimum cost flow in the associated graph.

Compute $t := \text{round}(\frac{1}{d}\sum_{i=1}^{d} U(Q_i))$;
$F^0 := \emptyset$; $k := 0$;

while (stop criterion is not met) **do**
begin

$k := k + 1$;

Compute a pixel weight $W^k(i,j)$ for all $(i,j) \in A$, based on F^{k-1};
Select an instance of Problem 3 or Problem 4;
Compute the solution F^k with $|F^k| = t$ of the subproblem;

end
$F := F^k$;

Fig. 4. Basic steps of the generic iterative subset algorithm.

5 Reconstruction from more than two projections

As described in the previous section, the reconstruction problem for one and two projections can be solved efficiently. Unfortunately, the reconstruction problem for more than two projections is often NP-hard, depending on the particular projection model used. The reconstruction problems for one and two projections are usually highly underdetermined and a large number of solutions may exist. In (15), an iterative algorithm was proposed that uses a network flow algorithm for the two-projection problem as a subroutine, solving a new problem in each iteration. The algorithm was specifically aimed at discrete X-ray projections. We will now describe a generalization of this iterative approach, that can be used for arbitrary projection models (i.e., induced by arbitrary partitions), and uses either an algorithm for the single projection problem or the two-projection problem to solve the subproblems. Figure 4 shows the basic steps of our algorithm. In the next subsections, we describe each of the algorithmic steps in more detail.

5.1 Selection of projections

In every iteration of the algorithm a new subset of the projections (either one or two) is selected to be used as a new subproblem. There are several sensible ways for choosing this subset. In (15), experiments were performed using a fixed, predefined order for choosing the projections. For the experiments in Section 6, we repeatedly select the subset of projections that has the largest projection distance. In this way, projections that have been used in recent iterations are unlikely to be selected again.

5.2 Computing the pixel weights

In each iteration k, a one- or two-projection problem is solved, corresponding to a subset of the available projections. The solution from the previous iteration is used to determine the pixel weights $W^k(i,j)$ for the new reconstruction problem.

In the computation of the pixel weights, a preference for smooth local regions of pixels is incorporated. The weight of a pixel depends not only on the corresponding pixel value in the previous solution F^{k-1}, but also on the values of pixels in a neighbourhood, assigning higher weights to pixels having a homogeneous neighbourhood. In our experiments, we used the 4-neighbourhood of each pixel for computing the pixel weights.
For any $(i,j) \notin A$, define $F^{k-1}(i,j) = 0$. Let $\beta > 0$. Put

$$f(i,j) = \frac{1}{1+4\beta}(F^{k-1}(i,j) + \beta(F^{k-1}(i-1,j) + F^{k-1}(i+1,j) \\ +F^{k-1}(i,j-1) + F^{k-1}(i,j+1))). \tag{9}$$

We used $\beta = 1$ for all experiments in Section 6, which corresponds to all pixels in the 4-neighbourhood (including the central pixel) having the same influence on the pixel weight.

Put $\mathbb{R}_{>0} = \{x \in \mathbb{R} : x > 0\}$. Let $g : [0,1] \to \mathbb{R}_{>0}$ be a non-decreasing odd function (i.e., $g(x) = -g(-x)$), which we call the *local weight function*. This function determines the preference for locally smooth regions. The pixel weight $W(i,j)$ is computed as an integer-valued approximation of the local weight function:

$$W^k(i,j) = \text{round}(K \cdot g(f(i,j) - \frac{1}{2})), \tag{10}$$

where K is a large integer.

When we take $g(f) = 1$ for all $f \in [0,1]$, there is no preference for local smoothness. In that case, the solution of the new network flow problem will simply have the same value as F in as many pixels as possible. If no prior knowledge is used at all in the reconstruction (i.e., no preference for smooth regions), the iterative algorithm generally produces very bad reconstructions and does not converge.

If we choose g to be an increasing function, pixels surrounded by other pixels having the same value are preferred to retain their value in the next reconstruction. We will show in Section 6 that incorporating a preference for local

smoothness in the local weight function results in (empirically) good convergence for images that are relatively smooth. We used the linear local weight function $g(x) = x$ for all experiments in Section 6.

5.3 Termination

As we cannot evaluate how close the current reconstructed image is to the unknown original image, the only measure for the distance between those two images is the total projection distance. Unfortunately, there may be different images having the same, or almost the same projections, yet being very different. In all experiments described in Section 6, we terminated the iterative algorithm if the minimal projection distance found so far has not decreased for 300 iterations. Admittedly, this is quite a large number. However, as demonstrated in Section 6, the algorithm sometimes tends to remain at an almost constant projection distance for a large number of iterations, before making substantial improvements again. If the projection distance reaches 0, the algorithm always terminates immediately.

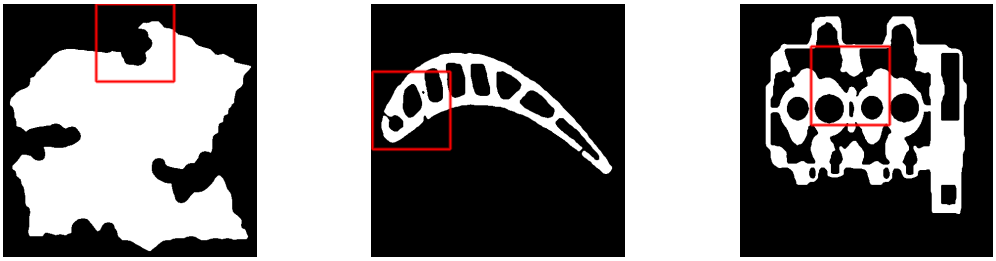
6 Experimental results

In this section, we present reconstruction results of our algorithm for the three reconstruction problems described in Section 3.

In all subproblems, both for one and two projections, we use a value of α for which $\alpha > \sum_{(i,j)\in A} |W(i,j)|$. For the iterative algorithm that uses a single projection in each iteration, this means that we can use the fast greedy approach described in Section 4.1. All experiments were performed using two different versions of the generic iterative algorithm. The first version solves an instance of Problem 3 (i.e., using one projection) in each iteration. The second version repeatedly solves instances of Problem 4 (i.e., using two projections).

Our experiments were performed using three phantom images, shown in Figure 5a,b,c. The first image (*simple*) contains a single homogeneous object which is not convex and has a rather complex boundary. The second image (*turbine*) represents a turbine blade from a jet engine, which contains several minor defects. It can be used to assess the ability of the reconstruction algorithm to reconstruct small details. The third image (*cylinder*) represents a cross-section of a cylinder head from an engine. All three phantom images have a size of 512×512 pixels.

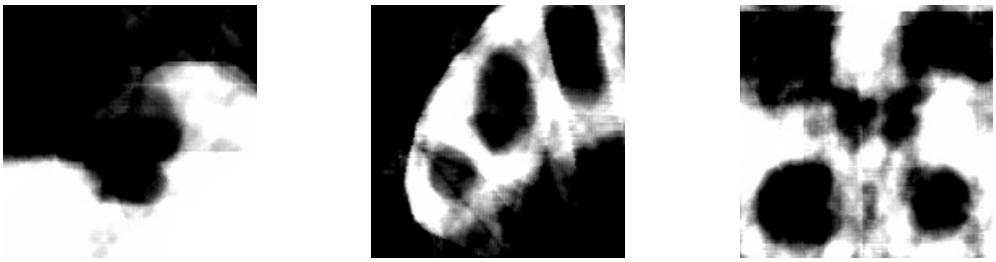
To compare our approach with a common method from continuous tomog-



a. simple phantom b. turbine phantom c. cylinder phantom

d. Rectangular scan, Kaczmarz e. Rectangular scan, Kaczmarz f. Rectangular scan, Kaczmarz

g. Rectangular scan, GIS h. Rectangular scan, GIS i. Rectangular scan, GIS



j. Cont. X-rays, Kaczmarz k. Cont. X-rays, Kaczmarz l. Cont. X-rays, Kaczmarz

m. Cont. X-rays, GIS n. Cont. X-rays, GIS o. Cont. X-rays, GIS

Fig. 5. Phantom images and magnified selections of their reconstructions. a,b,c: Three phantom images. d,e,f: Reconstructions computed using the Kaczmarz algorithm from 16 sliding window projections. g,h,i: Reconstructions computed using our generic iterative subset algorithm (GIS) from 16 sliding window projections. j,k,l: Reconstructions computed using the Kaczmarz algorithm from 5, 7 and 10 continuous X-ray projections respectively. m,n,o: Reconstructions computed using GIS from continuous X-rays.

raphy, we implemented the Kaczmarz algorithm as described in (1). In this algorithm, the reconstruction problem is considered as a large system of real-valued linear equations. Let $\tilde{F} \in \mathbb{R}^{|A|}$, where the elements of the vector $\tilde{F}$ correspond 1-to-1 with the elements of A. We denote the element of $\tilde{F}$ that corresponds to $(i, j) \in A$ by $\tilde{F}(i, j)$.

For each partition $\mathbf{S}_i$ and each set $S \in \mathbf{S}_i$, the system contains the equation

$$\sum_{(i,j)\in S} \tilde{F}(i, j) = Q_i(S)$$

The Kaczmarz algorithm iteratively solves the system of equations, applying sequential update steps for all equations. When solving a binary tomography problem as a real-valued problem, we may assume that $\tilde{F}(i, j) \in [0, 1]$ for all $(i, j) \in A$. Therefore, we set $\tilde{F}(i, j) = 0$ whenever the value for pixel (i, j) becomes less than 0 during the algorithm, and set $\tilde{F}(i, j) = 1$ whenever it becomes greater than 1. For binary images, this operation results in far better reconstructions compared to the original Kaczmarz algorithm, where no such bounds are used. We define one *iteration* of the Kaczmarz algorithm as a series of sequential updates corresponding to all sets in a single partition. In our experiments, the ordering of the sequence of partitions is chosen randomly and the algorithm is terminated after 10000 iterations. The Kaczmarz algorithm has been used in numerous variations in continuous tomography. Chapter 7 of (1) provides an overview of the more general class of *algebraic reconstruction algorithms*. The most prominent example, which is widely used in practice, is the Algebraic Reconstruction Technique (ART) (18).

For comparing the Kaczmarz algorithm with our iterative subset algorithms, the continuous reconstruction computed by the Kaczmarz algorithm needs to be segmented to obtain a binary image. We used thresholding with a fixed threshold of 0.5 in all experiments. One could argue that setting the threshold at 0.5 can introduce a bias in the segmentation results, but we did not obtain better results using different values for the global threshold.

We implemented the iterative subset algorithm and the Kaczmarz algorithm in C++. For solving the network flow problems we used the CS2-library by Andrew Goldberg; see (19) for a description of the network flow algorithm. All experiments were performed on an Intel Core Duo E6700 PC using a single core (i.e., no parallelism).

In the next three subsections, we describe the experimental results for discrete X-rays, sliding window projections and continuous X-rays, respectively.

Table 1 shows the reconstruction results for the three phantom images, using discrete X-ray projections, as described in Section 3.1. For each of the phantoms, a certain minimum number of projections is required to obtain a reasonably accurate reconstruction. For each phantom and each algorithm, three experiments were performed using an increasing number of projections.

For every experiment using k projections, the directions $(a_1, b_1), \ldots, (a_k, b_k)$ from the set

$$\begin{aligned}\{(a_1, b_1), \ldots, (a_{12}, b_{12})\} = \{\ & (1,0), (0,1), (1,1), (1,-1), (1,2), (2,1) \\ & (1,-2), (2,-1), (1,3), (3,1), (1,-3), (3,-1)\}\end{aligned}$$

were used.

The table shows results for the Kaczmarz algorithm and for our iterative subset algorithm, both using one and two projections per iteration. For each experiment, the total number of iterations, the total number of pixel errors in the reconstruction (i.e., different from the phantom image) and the projection distance of the reconstruction are shown.

For each phantom, the number k of projections that is needed for an accurate reconstruction is different. We determined this number experimentally and performed experiments using $k-1$, k and $k+1$ projections.

When a sufficient number of projections are used, the iterative algorithm computes highly accurate reconstructions, both using one and two projections. The algorithm that uses two projections in each iteration is consistently capable of finding a perfect reconstruction, identical to the phantom. The algorithm that uses only one projection in each iteration only computes a perfect reconstruction for the "simple" phantom. If too few projections are used for the "simple" phantom (i.e., 4 projections), the pixel error of the Kaczmarz algorithm is even smaller than the pixel error for the iterative subset algorithms. However, the projection distance for the Kaczmarz algorithm is much larger than for the iterative subset algorithms.

Figure 6 shows the total projection error for the three methods, as a function of the iteration number. The graph indicates that the iterative subset algorithm does not converge at a regular rate. It sometimes remains at an almost constant projection distance for a large number of iterations, after which improvement is again achieved. We observed similar behaviour in many of the experiments.

To compare the running times of the Kaczmarz algorithm and both iterative

subset algorithms experimentally, additional experiments were performed for the cylinder phantom, using scaled versions of the original phantom of various sizes. Figure 7 shows the average time per iteration, as a function of the (square) phantom size. The iterative subset algorithm clearly does not scale as well as the other two approaches, yet it can still be used effectively for images of sizes up to 1000×1000 on a modern computer. We remark that for larger images, the number of iterations may also increase.

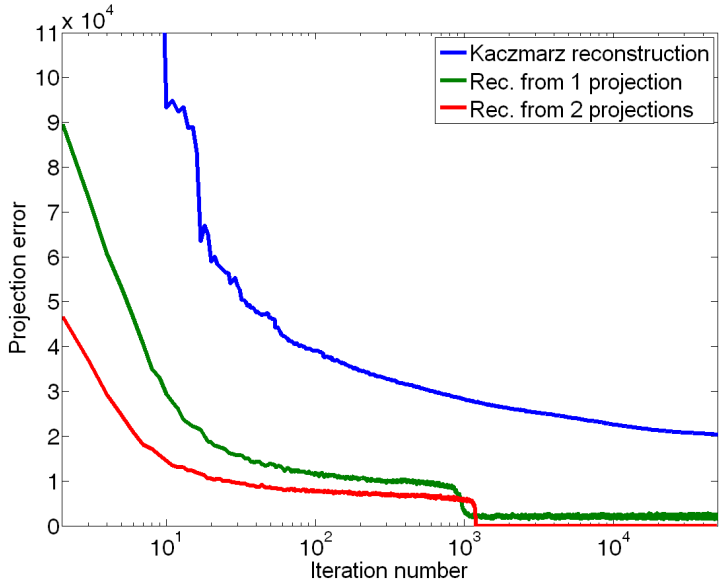


Fig. 6. Projection distance for the cylinder phantom using 9 discrete X-ray projections, as function of the iteration number.

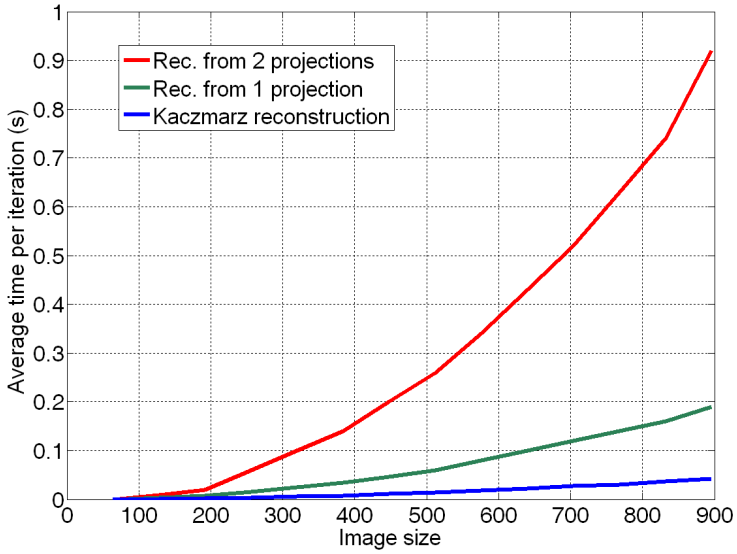


Fig. 7. Average time per iteration of the cylinder phantom using 12 discrete X-ray projections, as function of the phantom size n. For each size n, the original 512×512 phantom is scaled to size $n \times n$.

phantom	#projections	subproblem	#iterations	pixel error	proj. dist.
simple	4	Kaczmarz	10000	16138	10319
	4	1 proj.	1481	1476	11918
	4	2 proj.	826	808	17528
	5	Kaczmarz	10000	5863	2637
	5	1 proj.	220	0	0
	5	2 proj.	104	0	0
	6	Kaczmarz	10000	3668	1113
	6	1 proj.	102	0	0
	6	2 proj.	37	0	0
cylinder	8	Kaczmarz	10000	28212	16462
	8	1 proj.	905	7334	19914
	8	2 proj.	1115	4856	18868
	9	Kaczmarz	10000	22604	11420
	9	1 proj.	1861	1584	272
	9	2 proj.	1198	0	0
	10	Kaczmarz	10000	22442	10051
	10	1 proj.	1146	1594	224
	10	2 proj.	571	0	0
turbine	5	Kaczmarz	10000	8941	6239
	5	1 proj.	1133	1752	5756
	5	2 proj.	2042	938	7476
	6	Kaczmarz	10000	5964	2977
	6	1 proj.	894	464	130
	6	2 proj.	284	0	0
	7	Kaczmarz	10000	4900	1644
	7	1 proj.	922	318	60
	7	2 proj.	76	0	0

Table 1
Discrete X-ray projections. Quantitative comparison between the threshold result of the Kaczmarz algorithm and the proposed reconstruction methods with subproblems based on 1 and 2 projections. The table shows the number of projections used, the number of iterations, the total number of pixel differences between the reconstruction and the original phantom (*pixel error*), and the projection distance of the reconstructed image (*proj. dist.*).

6.2 Rectangular scan

Table 2 shows the reconstruction results for the three phantom images, using rectangular scan projections, as described in Section 3.2.

All experiments were performed using a window of size $(p, q) = (32, 32)$. The partitions $\mathbf{S}^{a,b}$ were selected randomly from the 32×32 possible projections, using the same random sequence in all experiments. For every experiment using k projections, the window positions $(a_1, b_1), \ldots, (a_k, b_k)$ from the set

$$\begin{aligned}\{(a_1, b_1), \ldots, (a_{16}, b_{16})\} = \{\ & (6,7), (19,9), (31,17), (12,10), (13,9), (11,26), \\ & (27,18), (6,3), (2,28), (24,20), (8,27), (13,7), \\ & (26,22), (3,14), (31,19), (26,9)\}\end{aligned}$$

were used.

Figure 5d,e,f shows the reconstruction computed from 16 projections by the Kaczmarz algorithm, before thresholding is applied. Figure 5g,h,i, shows reconstructions computed from the same 16 projections by our iterative algorithm, using two projections in each iteration. It is quite remarkable that high quality reconstructions can be obtained based using only 16 out of the 1024 possible projections. For the turbine phantom, some of the fine details are reconstructed accurately, while other details do not show up properly in the reconstruction.

The results show that the iterative subset algorithm, both for one and two projections per iteration, consistently produce much better reconstructions compared to the Kaczmarz algorithm. In all experiments, the algorithm that uses two projections in each iteration finds a reconstruction with fewer pixel errors than the algorithm based on a single projection per iteration.

6.3 Continuous X-rays

Table 3 shows the reconstruction results for the three phantom images, using continuous X-ray projections, as described in Section 3.3.

For each experiment, continuous X-ray projections were computed for an array of equally spaced detectors (having a width equal to the pixel size), based on Eq. 3. The projections were computed by numerical integration, where the phantoms are assumed to have a constant value (either 0 or 1) within each pixel. Note that the projections measured by the detector array are real-valued. The integer-valued projections that are required for Problem 2 were computed from the real-valued detector data by linear interpolation and rounding of the result. The resulting sequence of projections is very likely to be inconsistent, due to rounding and interpolation errors when converting from continuous to discrete projection data.

For every experiment using k projections, the projection angles $\{\theta_1, \ldots, \theta_k\}$

phantom	#projections	subproblem	#iterations	proj. dist.	pixel error
simple	8	Kaczmarz	10000	14926	3769
	8	1 proj.	554	652	992
	8	2 proj.	1095	18	712
	12	Kaczmarz	10000	12108	2562
	12	1 proj.	1230	1010	838
	12	2 proj.	540	172	418
	16	Kaczmarz	10000	13296	2133
	16	1 proj.	805	1432	702
	16	2 proj.	469	380	378
cylinder	8	Kaczmarz	10000	33812	10129
	8	1 proj.	2043	1544	3248
	8	2 proj.	1025	368	2810
	12	Kaczmarz	10000	31004	7286
	12	1 proj.	1163	2446	2454
	12	2 proj.	1092	804	1496
	16	Kaczmarz	10000	31718	5898
	16	1 proj.	697	3244	2082
	16	2 proj.	601	1360	1478
turbine	8	Kaczmarz	10000	17032	6038
	8	1 proj.	988	974	2286
	8	2 proj.	848	258	1744
	12	Kaczmarz	10000	14972	4494
	12	1 proj.	654	1480	1512
	12	2 proj.	862	554	1032
	16	Kaczmarz	10000	14934	3596
	16	1 proj.	729	2158	1320
	16	2 proj.	852	848	774

Table 2
Rectangular scan (32×32). Quantitative comparison between the threshold result of the Kaczmarz algorithm and the proposed reconstruction methods with subproblems based on 1 and 2 projections. The table shows the number of projections used, the number of iterations, the total number of pixel differences between the reconstruction and the original phantom (*pixel error*), and the projection distance of the reconstructed image (*proj. dist.*).

are spaced equally within the interval $[0, \pi]$: $\theta_i := \frac{(i-1)\pi}{k}$.

Similar to the case of discrete X-rays, each phantom requires a certain minimum number k of projections to be reconstructed with reasonable accuracy. We determined this number experimentally and performed experiments using

$k-1$, k and $k+1$ projections.

Figure 5j,k,l shows the reconstruction of the simple, turbine and cylinder phantom, computed from 5, 7 and 10 projections, respectively by the Kaczmarz algorithm, before thresholding is applied. Figure 5m,n,o shows reconstructions computed from the same projections by our iterative algorithm, using two projections in each iteration.

When a sufficient number of projections are used, the iterative algorithm computes highly accurate reconstructions, both using one and two projections. In most cases, the algorithm that uses two projections in each iteration computes a slightly better reconstruction compared to the algorithm that uses only one projection per iteration. For the turbine phantom, the difference is even substantial.

7 Conclusions

We have described an iterative algorithm for reconstructing binary images from their projections. The algorithm is based on the approach from (15), but is much more general. A variety of tomographic reconstruction problems fit within the proposed projection model, based on arbitrary partitions of the image pixels. For three of these problems, dealing with discrete X-rays, continuous X-rays and sliding window projections, we performed experiments based on three phantom images.

The results consistently show that a certain minimum number of projections is required to obtain an accurate reconstruction using our iterative algorithm. The number of required projections depends on the image being reconstructed. If sufficient projections are used, the iterative algorithm produces highly accurate reconstruction results for each of the phantom images. The boundary between "too few" projections and "enough" projections appears to be fairly sharp in all cases.

A natural question in these experiments concerns the relation between certain properties of the original image and the number of projections that are required for an accurate reconstruction. So far, our research on this question has not resulted in a simple model for predicting the required set of projections based on features of the original image.

When sufficiently many projections are used, the version of the iterative algorithm that uses two projections in each iteration consistently produces more accurate reconstructions than the version that uses only a single projection. On the other hand, the single projection problem can be solved much more

phantom	#projections	subproblem	#iterations	proj. dist.	pixel error
simple	4	Kaczmarz	10000	16324	10370
	4	1 proj.	1064	1424	15063
	4	2 proj.	902	758	16109
	5	Kaczmarz	10000	3436	1399
	5	1 proj.	450	989	272
	5	2 proj.	733	731	378
	6	Kaczmarz	10000	2734	762
	6	1 proj.	497	1158	268
	6	2 proj.	338	780	38
cylinder	9	Kaczmarz	10000	32375	16902
	9	1 proj.	1141	7930	20985
	9	2 proj.	1032	4838	20895
	10	Kaczmarz	10000	16988	7191
	10	1 proj.	953	4282	1164
	10	2 proj.	608	3156	788
	11	Kaczmarz	10000	16993	6495
	11	1 proj.	789	4942	1186
	11	2 proj.	629	3816	1024
turbine	5	Kaczmarz	10000	11837	6632
	5	1 proj.	1381	1707	9618
	5	2 proj.	1550	955	8244
	6	Kaczmarz	10000	9500	4972
	6	1 proj.	1847	1964	4278
	6	2 proj.	1639	862	316
	7	Kaczmarz	10000	8911	3590
	7	1 proj.	746	1775	640
	7	2 proj.	584	1285	564

Table 3
Continuous X-ray projections. Quantitative comparison between the threshold result of the Kaczmarz algorithm and the proposed reconstruction methods with subproblems based on 1 and 2 projections. The table shows the number of projections used, the number of iterations, the total number of pixel differences between the reconstruction and the original phantom (*pixel error*), and the projection distance of the reconstructed image (*proj. dist.*).

efficiently than the two-projection problem and is easier to implement. Both versions are capable of computing reconstructions that are far more accurate than the Kaczmarz algorithm.

In this paper we have focused on the reconstruction of two-dimensional images.

Our proposed model for iterative network flow algorithms deals with general partitions of the set of image elements (i.e., pixels) and does not require the images to be two-dimensional: it can be used for three-dimensional reconstruction as well. In (20) it was already demonstrated that for discrete X-rays, iterative network flow algorithms can be used effectively for the reconstruction of three-dimensional binary images.

References

[1] A. C. Kak, M. Slaney, Principles of Computerized Tomographic Imaging, Vol. Algorithms for reconstruction with non-diffracting sources, IEEP Press, New York, NY, 1988.

[2] G. T. Herman, A. Kuba (Eds.), Discrete Tomography: Foundations, Algorithms and Applications, Birkhäuser, Boston, 1999.

[3] G. T. Herman, A. Kuba (Eds.), Advances in discrete tomography and its applications, Birkhäuser, Boston, 2007.

[4] R. J. Gardner, P. Gritzmann, D. Prangenberg, On the computational complexity of reconstructing lattice sets from their X-rays, Discrete Math. 202 (1999) 45–71.

[5] P. Gritzmann, S. de Vries, M. Wiegelmann, Approximating binary images from discrete X-rays, SIAM J. Optimization 11 (2000) 522–546.

[6] L. Hajdu, R. Tijdeman, Algebraic aspects of discrete tomography, J. reine angew. Math. 534 (2001) 119–128.

[7] A. Frosini, M. Nivat, Binary matrices under the microscope: a tomographical problem, Theoret. Comp. Sci. 370 (2007) 201–217.

[8] P. Dulio, R. J. Gardner, C. Peri, Discrete point x-rays, SIAM J. Discrete Math. 20(1) (2006) 171–188.

[9] Y. Gerard, Additive subsets, Lect. Notes Comp. Sci. 4040 (2006) 347–353.

[10] J. R. Jinschek, K. J. Batenburg, H. Calderon, D. Van Dyck, F. R. Chen, C. Kisielowski, Prospects for bright field and dark field electron tomography on a discrete grid, Microscopy and Microanalysis, Vol. 10 Suppl. 3, Cambridge Journals Online (2004).

[11] T. Schüle, C. Schnörr, S. Weber, J. Hornegger, Discrete tomography by convex-concave regularization and d.c. programming, Discrete Appl. Math. 151.

[12] S. Weber, A. Nagy, T. Schüle, C. Schnörr, A. Kuba, A benchmark evaluation of large-scale optimization approaches to binary tomography, in: A. Kuba, L. Nyúl, K. Palágyi (Eds.), Proceedings of the 13th International Conference on Digital Geometry and Computer Imagery. Lecture Notes Comp. Sci., Vol. 4245, Springer, 2006, pp. 146–156.

[13] K. J. Batenburg, A network flow algorithm for binary image reconstruction from few projections, in: A. Kuba, L. Nyúl, K. Palágyi (Eds.), Proceedings of the 13th International Conference on Digital Geometry and

Computer Imagery. Lecture Notes Comp. Sci., Vol. 4245, Springer, 2006, pp. 86–97.
[14] R. J. Gardner, Geometric tomograpy, 2nd edition, Cambridge University Press, 2006.
[15] K. J. Batenburg, A network flow algorithm for reconstructing binary images from discrete X-rays, J. Math. Imaging Vision 27(2) (2007) 175–191.
[16] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network flows: theory, algorithms, and applications, Prentice-Hall, Inc., 1993.
[17] D. Gale, A Theorem on Flows in Networks, Pacific J. Math. 7 (1957) 1073–1082.
[18] R. Gordon, R. Bender, G. T. Herman, Algebraic Reconstruction Techniques (ART) for three dimensional electron microscopy and X-ray photography, J. Theor. Biol. 29 (1970) 471–481.
[19] A. V. Goldberg, An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm, J. Algorithms 22 (1997) 1–29.
[20] K. J. Batenburg, A new algorithm for 3D binary tomography, Electron. Notes Discrete Math. 20 (2005) 247–261.