Perturbation-Resilient Block-Iterative Projection Methods with Application to Image Reconstruction from Projections

R. Davidi,[1] G.T. Herman,[1] and Y. Censor[2]

December 25, 2008

[1]Department of Computer Science, Graduate Center, City University of New York, New York, NY 10016, USA

[2]Department of Mathematics, University of Haifa, Mount Carmel, Haifa 31905, Israel

Abstract

A block-iterative projection algorithm for solving the consistent convex feasibility problem in a finite-dimensional Euclidean space that is resilient to bounded and summable perturbations (in the sense that convergence to a feasible point is retained even if such perturbations are introduced in each iterative step of the algorithm) is proposed. This resilience can be used to steer the iterative process towards a feasible point that is superior in the sense of some functional on the points in the Euclidean space having a small value. The potential usefulness of this is illustrated in image reconstruction from projections, using both total variation and negative entropy as the functional.

1 Introduction

We discuss block-iterative projection methods for the *convex feasibility problem* that can be stated as: Find a point in the nonempty intersection of a finite family of closed convex subsets of a Euclidean space. This is a fundamental problem in many areas of mathematics and the physical sciences, see [18, 19] and references therein. It has been used to model significant real-world problems in many fields, including image reconstruction from projections [28], radiation therapy treatment planning [12], and crystallography [35], to name but a few.

Projection methods are iterative algorithms that use projections onto sets while relying on the general principle that when a family of (usually closed and convex) sets is present then projections onto the individual sets are easier to perform than projections onto other sets (e.g., the intersection of the sets) that are derived from the given individual sets. Projection algorithms have various algorithmic structures, of which some are particularly suitable for parallel computing, and they have desirable convergence properties and/or good initial behavior; see, e.g., [5] and [17, Chapter 5].

In this paper we focus our attention on *block-iterative projection methods*. These are algorithms that in an iterative step first project the current iterate simultaneously onto the sets of some subfamily (called a block) of the whole family of sets and then take a convex combination of the resulting points as the next iterate. This block-iterative algorithmic scheme encompasses as special cases the sequential (row-action) algorithmic structure as well as the fully-simultaneous algorithmic structure, but many additional in-between structures are permitted and convergence to a solution of the convex feasibility problem is guaranteed under reasonable conditions.

The specific question that we ask is: Are block-iterative projection methods resilient to perturbations of the iterates occurring during the iterative process? This question is important for two reasons. First, because numerical errors and system model noise may cause at every iterative step deviations from the ideal (mathematical) projection point. Second, because we can use the flexibility that such perturbations allow to create block-iterative algorithms that will not only solve a convex feasibility problem but will steer the iterates toward a feasible point that has some desirable properties.

The answer that we give here is affirmative. We show that the convergence to a solution of a block-iterative projection algorithm for a consistent convex feasibility problem is retained even if in each iterative step the current iterate is perturbed into $\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k$. For this to happen we need the perturbations to be bounded and summable; i.e., they are such that the sequence $\left\{\underline{\boldsymbol{w}}^k\right\}_{k=0}^{\infty}$ is bounded and the real parameters γ_k obey $\sum_{k=0}^{\infty} \gamma_k < \infty$.

Our analysis that proves that block-iterative projection methods are resilient to perturbations is based on the construction of another convex feasibility problem in an appropriately chosen product space, and on using for it an amalgamated projection method. In [9] we recently proved the convergence under summable perturbations for such methods and in this paper we utilize that result for our product space construction that allows us to extend it to block-iterative projection methods.

The proved resilience of our algorithm is used to steer the iterative process towards a feasible point that is superior in the sense of some functional on the points in the Euclidean space having a small value. The potential usefulness of this is demonstrated in image reconstruction from projections, using total variation (TV) as the functional. We confirm, using our algorithm, the validity of previously-reported results obtained using alternative TV-minimizing algorithms: When a physically-unrealistic mathematical idealization of data collection in the image reconstruction from projections problem is employed (one in which consistency is assured by simulating data collection using the same model as used in turning the image reconstruction problem into a convex feasibility problem), then remarkably high quality reconstructions can be obtained in very underdetermined situations by TV-minimization (as opposed to, say, norm-minimization or entropy-maximization). We also illustrate that even when applied to data that have been collected in a physically-realistic manner, the result produced using the TV-minimizing perturbations appears to be superior to that produced without them.

Section 2 provides the necessary background by restating the previously-published perturbation-resilient amalgamated projection method. Our new perturbation-resilient block-iterative algorithm is specified in Section 3, together with a proof of its convergence under perturbations. Our proof is based on reduction, using the methodology of product spaces, to the result presented in Section 2. In Section 4 the notion of superiorization is introduced; it is the method of steering a perturbation-resilient algorithm towards the minimizer of a given functional by an appropriate selection of the perturbations. The potential usefulness of this is illustrated in Section 5, by showing the applicability of our approach to image reconstruction from projections with TV as the given functional. The examples that are given in this section are all linear; more nonlinear problems can probably benefit from the superiorization methodology proposed here, but are outside the scope of this paper. Finally, we present a discussion of our approach in Section 6.

2 Background

Let J be a positive integer. Throughout this paper we use bold italic letters for points in the vector space $\mathbb{R}^J$ and regular italic letters for real numbers in $\mathbb{R}$. For $\boldsymbol{x} \in \mathbb{R}^J$, we use the notation

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_J \end{pmatrix}, \; x_j \in \mathbb{R} \text{ for } 1 \leq j \leq J. \tag{1}$$

For any $\boldsymbol{x} \in \mathbb{R}^J$ and for any nonempty subset M of $\mathbb{R}^J$, the *distance between the point* $\boldsymbol{x}$ *and the set* M is defined by

$$d(\boldsymbol{x}, M) = \inf \{\|\boldsymbol{x} - \boldsymbol{y}\| \mid \boldsymbol{y} \in M\}, \tag{2}$$

where $\|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{\sum_{j=1}^{J}(x_j - y_j)^2}$ denotes the *norm* of $\boldsymbol{x} - \boldsymbol{y}$. If M is closed and convex, then there is one, and only one, $\boldsymbol{y} \in M$ such that

$$\|\boldsymbol{x} - \boldsymbol{y}\| = d(\boldsymbol{x}, M). \tag{3}$$

This $\boldsymbol{y}$ is called the *projection of* $\boldsymbol{x}$ *onto* M and is denoted by $P_M\boldsymbol{x}$.

For the rest of this section we consider the following *consistent convex feasibility problem*: Given I closed and convex subsets $C_1, ..., C_I$ of $\mathbb{R}^J$ such that the set

$$C = \bigcap_{i=1}^{I} C_i, \tag{4}$$

is nonempty, find a point $\boldsymbol{x}$ in C. Such points $\boldsymbol{x}$ are called *feasible.*

An *index vector* is a nonempty ordered set $t = (t_1, \ldots, t_N)$, where N is an arbitrary positive integer, whose elements t_n are in the set $\{1, ..., I\}$. For an index vector t we define the composite operator

$$P[t] := P_{C_{t_N}} \cdots P_{C_{t_1}}. \tag{5}$$

A finite set Ω of index vectors is called *fit* if, for each $i \in \{1, ..., I\}$, there exists $t = (t_1, \ldots, t_N) \in \Omega$ such that $t_n = i$ for some $n \in \{1, ..., N\}$. If Ω is a fit set of index vectors, then a function $\omega : \Omega \to \mathbb{R}_{++} = (0, \infty)$ is called a *fit weight function* if $\sum_{t\in\Omega} \omega(t) = 1$. A pair (Ω, ω) consisting of a fit set of index vectors and a fit weight function defined on it is called an *amalgamator*. For each amalgamator (Ω, ω), we define the operator $\mathbf{P} : \mathbb{R}^J \rightarrow \mathbb{R}^J$ by

$$\mathbf{P}\boldsymbol{x} := \sum_{t \in \Omega} \omega(t) P[t] \boldsymbol{x}. \tag{6}$$

Theorem 1. *If $\{\beta_k\}_{k \in \mathbb{N}}$ is a sequence of positive real numbers such that $\sum_{k=0}^{\infty} \beta_k < \infty$ and $\{\boldsymbol{v}^k\}_{k \in \mathbb{N}}$ is a bounded sequence of points in $\mathbb{R}^J$, then for any amalgamator (Ω, ω) and any $\boldsymbol{x}^0 \in \mathbb{R}^J$, the sequence $\boldsymbol{x}^0$, $\boldsymbol{x}^1$, $\boldsymbol{x}^2, \ldots$ generated by*

$$\boldsymbol{x}^{k+1} = \mathbf{P}\left(\boldsymbol{x}^k + \beta_k \boldsymbol{v}^k\right), \ \forall k \in \mathbb{N}, \tag{7}$$

converges, and its limit is in C.

Proof: See [9, Section II]. □

We refer to a procedure defined by (7) as a *perturbation-resilient amalgamated projection method.* Amalgamated projection methods without perturbations (i.e., with $\beta_k = 0$, for all $k \in \mathbb{N}$ in (7)) are *string averaging methods* as introduced in [13], and further studied in [7, 8, 16, 21, 22].

3 A Perturbation-Resilient Block-Iterative Algorithm

In this section we propose a perturbation-resilient block-iterative algorithm for solving convex feasibility problems. Our convergence analysis shows that the perturbation-resilient block-iterative algorithm converges to a solution of the consistent convex feasibility problem as long as the perturbations are bounded and summable. Since the proof of convergence of the new algorithm will be by reduction to Theorem 1 applied in a different space, we need to introduce extra notation. Let L and Z be positive integers. Throughout this paper we use underlined bold italic letters for points in $\mathbb{R}^L$. Our new convex feasibility problem is stated as: Given Z closed and convex subsets $D_1, \ldots, D_Z$ of $\mathbb{R}^L$, such that the set

$$D = \bigcap_{z=1}^{Z} D_z \tag{8}$$

is nonempty, find a point $\underline{\boldsymbol{x}}$ in D.

Given an ordered set $S = (S_1, \ldots, S_R)$, such that $S_r \subseteq \mathbb{R}^L$, for $1 \leq r \leq R$, we set $J = RL$ and define the *product set* $\mathbb{P}S$ as the set of all

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_L \\ \vdots \\ x_{(R-1)L+1} \\ \vdots \\ x_{RL} \end{pmatrix} \in \mathbb{R}^J, \tag{9}$$

such that, for $1 \leq r \leq R$,

$$\begin{pmatrix} x_{(r-1)L+1} \\ \vdots \\ x_{rL} \end{pmatrix} \in S_r. \tag{10}$$

Under these circumstances, we use the notations

$$\underline{\boldsymbol{x}}_r = \begin{pmatrix} x_{(r-1)L+1} \\ \vdots \\ x_{rL} \end{pmatrix} \text{ and } \boldsymbol{x} = \begin{pmatrix} \underline{\boldsymbol{x}}_1 \\ \vdots \\ \underline{\boldsymbol{x}}_R \end{pmatrix}, \tag{11}$$

and we refer to $\mathbb{R}^J \left(= \mathbb{R}^{RL}\right)$ as the *product space*. Note that for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^J$,

$$\|\boldsymbol{x} - \boldsymbol{y}\|^2 = \sum_{r=1}^{R} \left\|\underline{\boldsymbol{x}}_r - \underline{\boldsymbol{y}}_r\right\|^2. \tag{12}$$

It is well-known (and easy to prove) that if every element S_r of $S = (S_1, \ldots, S_R)$ is a closed convex subset of $\mathbb{R}^L$, then $\mathbb{P}S$ is a closed convex subset of $\mathbb{R}^J$. Furthermore, according to [17, Lemma 5.9.2],

$$P_{\mathbb{P}S} \begin{pmatrix} \underline{\boldsymbol{x}}_1 \\ \vdots \\ \underline{\boldsymbol{x}}_R \end{pmatrix} = \begin{pmatrix} P_{S_1}\underline{\boldsymbol{x}}_1 \\ \vdots \\ P_{S_R}\underline{\boldsymbol{x}}_R \end{pmatrix}. \tag{13}$$

One more piece of terminology that we need is the following. The *canonical mapping* $\boldsymbol{\delta} : \mathbb{R}^L \to \mathbb{R}^J$ is defined by: For any $\underline{\boldsymbol{x}} \in \mathbb{R}^L$, $\boldsymbol{\delta}\left(\underline{\boldsymbol{x}}\right) := \boldsymbol{x}$, with $\underline{\boldsymbol{x}}_1 = \cdots = \underline{\boldsymbol{x}}_R = \underline{\boldsymbol{x}}$ (here we made use of the notation of (11)).

We now return to the convex feasibility problem that is stated in the first paragraph of this section. For $1 \leq u \leq U$, let B_u be an ordered set $\left(b_{u,1}, \ldots, b_{u,|B_u|}\right)$ of elements of $\{1, \ldots, Z\}$ ($|B_u|$ denotes the cardinality of B_u). We call such a B_u a *block*. We define the composite operator $\mathbf{Q} : \mathbb{R}^L \to \mathbb{R}^L$ as

$$\mathbf{Q} := Q_U \cdots Q_1, \tag{14}$$

where, for $\underline{\boldsymbol{x}} \in \mathbb{R}^L$ and $1 \leq u \leq U$,

$$Q_u \underline{\boldsymbol{x}} := \frac{1}{R} \sum_{z \in B_u} P_{D_z} \underline{\boldsymbol{x}} + \frac{R - |B_u|}{R} \underline{\boldsymbol{x}}, \tag{15}$$

and

$$R := \max \{ |B_u| \mid 1 \leq u \leq U \} . \tag{16}$$

An iterative procedure based on $\underline{\boldsymbol{x}}^{k+1} = \mathbf{Q} \underline{\boldsymbol{x}}^k$ is a member of the family of *block-iterative projection (BIP) methods*[1, 2, 14, 15, 24, 25].

Theorem 2. *Let L, Z, U, $\{B_u\}_{u=1}^U$ and $\mathbf{Q}$ be as defined above. Let λ be a real number such that $0 < \lambda \leq 1$, $\{\gamma_k\}_{k \in \mathbb{N}}$ be a sequence of positive real numbers such that $\sum_{k=0}^{\infty} \gamma_k < \infty$, $\left\{\underline{\boldsymbol{w}}^k\right\}_{k \in \mathbb{N}}$ be a bounded sequence of points in $\mathbb{R}^L$, and $\underline{\boldsymbol{x}}^0 \in \mathbb{R}^L$. If $\{1, \ldots, Z\} = \bigcup_{u=1}^U B_u$, then the sequence $\underline{\boldsymbol{x}}^0$, $\underline{\boldsymbol{x}}^1$, $\underline{\boldsymbol{x}}^2, \ldots$ generated by* the iterative *procedure*

$$\underline{\boldsymbol{x}}^{k+1} = \lambda \mathbf{Q} \left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k \right) + (1 - \lambda) \left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k \right), \ \forall k \in \mathbb{N}, \tag{17}$$

converges, and its limit is in D.

Proof: Our proof utilizes Theorem 1 by showing that the procedure in (17) is a perturbation-resilient amalgamated projection method in a product space. (Pierra [37] was the first to use convergence results of sequential algorithms in a product space to prove the convergence of simultaneous algorithms, see also [16] and [17, Section 5.9].) We present the proof for the case $\lambda < 1$, and afterward indicate how it can be altered for the case $\lambda = 1$.

We define a convex feasibility problem with $I = U + 2$ convex sets in $\mathbb{R}^J$ for $J = RL$ and a perturbation-resilient amalgamated projection method. For $1 \leq u \leq U$,

$$C_u := \mathbb{P} \left(D_{b_{u,1}}, \cdots, D_{b_{u,|B_u|}}, \mathbb{R}^L, \cdots, \mathbb{R}^L \right), \tag{18}$$

with $R - |B_u|$ copies of $\mathbb{R}^L$ at the end of (18). Further,

$$\begin{aligned} C_{U+1} &:= \left\{ \boldsymbol{\delta} \left(\underline{\boldsymbol{x}} \right) \mid \underline{\boldsymbol{x}} \in \mathbb{R}^L \right\}, & (19) \\ C_{U+2} &:= \mathbb{R}^J. & (20) \end{aligned}$$

We now specify P_{C_u}, for $1 \leq u \leq U+2$, by making use of (13). For $1 \leq u \leq U$,

$$P_{C_u} \begin{pmatrix} \underline{\boldsymbol{x}}_1 \\ \vdots \\ \underline{\boldsymbol{x}}_{|B_u|} \\ \underline{\boldsymbol{x}}_{|B_u|+1} \\ \vdots \\ \underline{\boldsymbol{x}}_R \end{pmatrix} = \begin{pmatrix} P_{D_{b_{u,1}}} \underline{\boldsymbol{x}}_1 \\ \vdots \\ P_{D_{b_{u,|B_u|}}} \underline{\boldsymbol{x}}_{|B_u|} \\ \underline{\boldsymbol{x}}_{|B_u|+1} \\ \vdots \\ \underline{\boldsymbol{x}}_R \end{pmatrix}. \tag{21}$$

The set C_{U+1} is called the *diagonal convex* [37] or the *diagonal subset* [16] and

$$P_{C_{U+1}} \boldsymbol{x} = \boldsymbol{\delta} \left(\frac{1}{R} \sum_{r=1}^{R} \underline{\boldsymbol{x}}_r \right) \tag{22}$$

(for a proof see [37, Lemma 1.1]). Clearly,

$$P_{C_{U+2}} \boldsymbol{x} = \boldsymbol{x}. \tag{23}$$

Next we show that $C = \bigcap_{u=1}^{U+2} C_u$ is nonempty. By the assumption of the consistent convex feasibility problem of this section, $D = \bigcap_{z=1}^{Z} D_z$ is nonempty. It is easy to see that if $\underline{\boldsymbol{x}} \in D$ then $\boldsymbol{\delta}\left(\underline{\boldsymbol{x}}\right) \in C$. From this follows that $C_1, \ldots, C_I$ satisfy the conditions of the convex feasibility problem of the last section and hence Theorem 1 applies.

To show that Theorem 1 implies Theorem 2 we need to make some specifications. We let

$$\Omega := \left\{ \left(1, U+1, \ldots, U, U+1\right), \left(U+2\right) \right\}. \tag{24}$$

(To be precise, in the first of these two index vectors every second element is $U+1$ and these are preceded by $1, \ldots, U$ respectively.) This Ω is fit because, for each $i \in \{1, \ldots, U, U+1, U+2 = I\}$, there exist $(t_1, \ldots,\ t_N) \in \Omega$ such that $t_n = i$ for some $n \in \{1, \ldots, N\}$. Let the weight of the first of the two index vectors in Ω be λ and the weight of the second one be $1 - \lambda$. The resulting weight function ω is fit since each weight is in $\mathbb{R}_{++}$ (this is where one needs $\lambda < 1$) and $\sum_{t \in \Omega} \omega(t) = 1$. We also set $\beta_k = \gamma_k$ and $\boldsymbol{v}^k = \boldsymbol{\delta}\left(\underline{\boldsymbol{w}}^k\right)$, for all $k \in \mathbb{N}$, and $\boldsymbol{x}^0 = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^0\right)$.

Claim: Given the specifications of the previous paragraph, the sequence $\boldsymbol{x}^0$, $\boldsymbol{x}^1$, $\boldsymbol{x}^2, \ldots$ produced by (7) of Theorem 1 has the property, that for $k \in$

$\mathbb{N}$, $\boldsymbol{x}^k = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k\right)$, where $\underline{\boldsymbol{x}}^0$, $\underline{\boldsymbol{x}}^1$, $\underline{\boldsymbol{x}}^2, \ldots$ is the sequence produced by (17) of Theorem 2.

If this claim is true, then Theorem 2 follows as we now show. By Theorem 1, $\boldsymbol{x}^0$, $\boldsymbol{x}^1$, $\boldsymbol{x}^2, \ldots$ converges to an $\boldsymbol{x}^* \in C$. Since $\boldsymbol{x}^* \in C_{U+1}$, it is of the form $\boldsymbol{x}^* = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^*\right)$, for some $\underline{\boldsymbol{x}}^* \in \mathbb{R}^L$. By the Claim, $\boldsymbol{x}^k = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k\right)$, for $k \in \mathbb{N}$. By (12),

$$\left\|\underline{\boldsymbol{x}}^k - \underline{\boldsymbol{x}}^*\right\|^2 \leq \left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^2, \tag{25}$$

and since the right-hand side converges to zero, so must the left-hand side, and hence $\underline{\boldsymbol{x}}^0$, $\underline{\boldsymbol{x}}^1$, $\underline{\boldsymbol{x}}^2, \ldots$ converges to $\underline{\boldsymbol{x}}^*$. Since $\boldsymbol{x}^* = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^*\right) \in C_{u,}$, for $1 \leq u \leq U$, we have, by (18), that $\underline{\boldsymbol{x}}^* \in D_{b_{u,m}}$ for $1 \leq u \leq U$ and $1 \leq m \leq |B_u|$. Since $\{1, \ldots, Z\} = \bigcup_{u=1}^{U} B_u$ and $D = \bigcap_{z=1}^{Z} D_z$, it must be so that $\underline{\boldsymbol{x}}^* \in D$.

We next prove the Claim by induction on k. The Claim is clearly true for $k = 0$ since $\boldsymbol{x}^0 = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^0\right)$. Let us now assume that $\boldsymbol{x}^k = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k\right)$ is true for some $k \geq 0$ and prove that it follows that $\boldsymbol{x}^{k+1} = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^{k+1}\right)$.

Using (5), (6), (7), (23) and (24) with $0 < \lambda < 1$, we obtain

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \lambda P_{C_{U+1}} P_{C_U} \cdots P_{C_{U+1}} P_{C_1}\left(\boldsymbol{x}^k + \beta_k \boldsymbol{v}^k\right) \\ &\quad + (1-\lambda)\left(\boldsymbol{x}^k + \beta_k \boldsymbol{v}^k\right). \end{aligned} \tag{26}$$

By the induction hypothesis and the assumptions that $\beta_k = \gamma_k$ and $\boldsymbol{v}^k = \boldsymbol{\delta}\left(\underline{\boldsymbol{w}}^k\right)$ we obtain

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \lambda P_{C_{U+1}} P_{C_U} \cdots P_{C_{U+1}} P_{C_1}\left(\boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k\right) + \gamma_k \boldsymbol{\delta}\left(\underline{\boldsymbol{w}}^k\right)\right) \\ &\quad + (1-\lambda)\left(\boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k\right) + \gamma_k \boldsymbol{\delta}\left(\underline{\boldsymbol{w}}^k\right)\right). \end{aligned} \tag{27}$$

By (14) and (17) and the obvious linearity of $\boldsymbol{\delta}$ we obtain

$$\begin{aligned} \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^{k+1}\right) &= \lambda \boldsymbol{\delta}\left(Q_U \cdots Q_1\left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k\right)\right) \\ &\quad + (1-\lambda)\left(\boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k\right) + \gamma_k \boldsymbol{\delta}\left(\underline{\boldsymbol{w}}^k\right)\right). \end{aligned} \tag{28}$$

Since the second terms of (27) and (28) are identical, the proof that $\boldsymbol{x}^{k+1} = \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^{k+1}\right)$ is complete if we can show that

$$P_{C_{U+1}} P_{C_U} \cdots P_{C_{U+1}} P_{C_1} \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k\right) = \boldsymbol{\delta}\left(Q_U \cdots Q_1\left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k\right)\right). \tag{29}$$

In fact we show by induction on u the stronger statement that

$$P_{C_{U+1}} P_{C_u} \cdots P_{C_{U+1}} P_{C_1} \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k\right) = \boldsymbol{\delta}\left(Q_u \cdots Q_1\left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k\right)\right), \tag{30}$$

for $0 \leq u \leq U$. This is clearly true for $u = 0$. Let us now assume that it is true for a u such that $0 \leq u < U$, and show that it is also true for $u + 1$.

Let $\underline{\boldsymbol{y}}$ abbreviate $Q_u \cdots Q_1 \left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k\right)$. By the induction hypothesis, (30), (21), (22) and (15),

$$P_{C_{U+1}} P_{C_{u+1}} P_{C_{U+1}} P_{C_u} \cdots P_{C_{U+1}} P_{C_1} \boldsymbol{\delta}\left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k\right) = P_{C_{U+1}} P_{C_{u+1}} \boldsymbol{\delta}\left(\underline{\boldsymbol{y}}\right)$$

$$= P_{C_{U+1}} \begin{pmatrix} P_{D_{b_{u+1,1}}} \underline{\boldsymbol{y}} \\ \vdots \\ P_{D_{b_{u+1,|B_{u+1}|}}} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{y}} \\ \vdots \\ \underline{\boldsymbol{y}} \end{pmatrix} = \boldsymbol{\delta}\left(\frac{1}{R} \sum_{z \in B_{u+1}} P_{D_z} \underline{\boldsymbol{y}} + \frac{R - |B_{u+1}|}{R} \underline{\boldsymbol{y}}\right)$$

$$= \boldsymbol{\delta}\left(Q_{u+1} \underline{\boldsymbol{y}}\right) = \boldsymbol{\delta}\left(Q_{u+1} Q_u \cdots Q_1 \left(\underline{\boldsymbol{x}}^k + \gamma_k \underline{\boldsymbol{w}}^k\right)\right). \tag{31}$$

This completes the proof for the case $\lambda < 1$.

For the special case where $\lambda = 1$, we alter Ω to include the single index vector

$$(1, U + 1, \ldots, U, U + 1), \tag{32}$$

with weight 1. The rest of the proof is just a simpler version of the proof for the case $\lambda < 1$. □

4 A Heuristic Approach to Superiorization

Theorem 2 guarantees the convergence of the perturbation-resilient block-iterative algorithm (17) to a feasible point when the perturbations are bounded and summable. We make use of this property to steer the iterates towards the minimizer of a given convex function ϕ; i.e., towards the $\underline{\boldsymbol{x}} \in \mathbb{R}^L$ that provides the solution of the problem

$$\text{minimize } \phi(\underline{\boldsymbol{x}}), \text{ subject to } \underline{\boldsymbol{x}} \in D. \tag{33}$$

The heuristic provided below is not guaranteed to achieve actual convergence to the minimizer $\underline{\boldsymbol{x}}$. However, as demonstrated by examples in the next section, it proceeds so that the value of the given function tends to be appropriately reduced and yet convergence to a feasible point is not compromised. This allows us to do *superiorization*, by which we mean the production of a superior solution (just as optimization produces an optimal solution) subject to given (convex) constraints. The idea is that the ability to perturb the original algorithm without losing convergence to a feasible point allows us to steer the algorithm towards a feasible point that is superior, according to some criterion, to the one to which we would get without the perturbations.

Consider a convex function $\phi : \mathbb{R}^L \to \mathbb{R}$ that has a minimizer over the set D. For any $k \in \mathbb{N}$, let $\rho^k \in \partial\phi(\underline{\boldsymbol{x}}^k)$ be a subgradient of ϕ at $\underline{\boldsymbol{x}}^k$, and define

$$\underline{\boldsymbol{w}}^k := \begin{cases} -\dfrac{\rho^k}{\|\rho^k\|}, & \text{if } \rho^k \neq 0, \\ \\ 0, & \text{if } \rho^k = 0. \end{cases} \tag{34}$$

Clearly, the sequence $\left\{\underline{\boldsymbol{w}}^k\right\}_{k\in\mathbb{N}}$ defined by (34) is bounded. Therefore, by Theorem 2, for any summable sequence of positive real numbers $\{\gamma_k\}_{k\in\mathbb{N}}$, the sequence $\left\{\underline{\boldsymbol{x}}^k\right\}_{k\in\mathbb{N}}$ generated according to (17) converges to a point in D.

In our implementation, we use the following methodology for generating the real numbers $\{\gamma_k\}_{k\in\mathbb{N}}$. We define, for any $\underline{\boldsymbol{x}} \in \mathbb{R}^L$, the proximity function

$$\mathrm{Res}(\underline{\boldsymbol{x}}) := \sqrt{\sum_{z=1}^{Z} \left(d(\underline{\boldsymbol{x}}, D_z)\right)^2}. \tag{35}$$

It can be seen that $\underline{\boldsymbol{x}} \in D$ if, and only if, $\mathrm{Res}(\underline{\boldsymbol{x}}) = 0$. Furthermore, if $\mathrm{Res}(\underline{\boldsymbol{x}}) > 0$, then its size indicates how badly $\underline{\boldsymbol{x}}$ violates the given collection $\{D_1, \ldots, D_Z\}$ of constraints. An approximate solution $\underline{\boldsymbol{x}}$ to the convex optimization problem (33) should have a small value of $\mathrm{Res}(\underline{\boldsymbol{x}})$ and should aim at finding, among all $\underline{\boldsymbol{x}}$ with similar (or smaller) value of $\mathrm{Res}(\underline{\boldsymbol{x}})$, an $\underline{\boldsymbol{x}}$ for which $\phi(\underline{\boldsymbol{x}})$ is small relative to the others.

Guided by this principle, we generate $\{\gamma_k\}_{k\in\mathbb{N}}$ as follows. We initialize γ to be an arbitrary positive number, which we denote by γ_{-1}. (We have always used $\gamma_{-1} = 1$.) In the process of the iterative step from $\underline{\boldsymbol{x}}^k$ to $\underline{\boldsymbol{x}}^{k+1}$, we also update the value of γ, which is (in the notation of (17)) γ_{k-1} at the beginning of the iterative step and γ_k at its end. This updating is done according to the pseudocode below (in which $\underline{\boldsymbol{w}}^k$ is defined by (34)). We terminate the iterative process when we find an $\underline{\boldsymbol{x}}^k$ such that $\mathrm{Res}(\underline{\boldsymbol{x}}^k) < \epsilon$, where ϵ is a user-specified small positive number. The complete superiorization algorithm consists of (17) with the $\underline{\boldsymbol{w}}^k$ defined by (34) and the γ_k defined by the following pseudocode that makes use of (35).

```
 1: logic = true
 2: while (logic)
 3:       $\underline{\boldsymbol{y}} = \underline{\boldsymbol{x}}^k + \gamma \underline{\boldsymbol{w}}^k$
 4:       if ( $\phi(\underline{\boldsymbol{y}}) \leq \phi(\underline{\boldsymbol{x}}^k)$ )
 5:           then
 6:               $\underline{\boldsymbol{x}}^{k+1} = \lambda \mathbf{Q} \underline{\boldsymbol{y}} + (1 - \lambda) \underline{\boldsymbol{y}}$
 7:               if ( $\mathrm{Res}(\underline{\boldsymbol{x}}^{k+1}) < \mathrm{Res}(\underline{\boldsymbol{x}}^k)$ )
 8:                   then logic = false
 9:                   else $\gamma = \gamma/2$
10:           else $\gamma = \gamma/2$
```

What this algorithm performs is a steering process towards a small value of ϕ (see Step 4 of the pseudocode), while attempting to maintain the convergence to the feasible region, as guaranteed by Theorem 2 for a proper choice of the sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ (see Step 7 of the pseudocode). The idea of using such a heuristic with a perturbation-resilient iterative method was first introduced in [9] in the context of amalgamated projection methods. The resulting point of the iterative process depends on the starting point $\underline{\boldsymbol{x}}^0$ and the choice of ϵ. In the next section we give several illustrations of the new approach using the above algorithm, in which a functional is reduced and a solution point is obtained with a value of ϕ that is in all but one of the cases nearer to the minimum than the value for the phantom (i.e., the test image).

5 Results

We illustrate our superiorization approach in an area of image processing, namely tomographic reconstruction of images that are not uniquely determined from the available data, with the help of a convex functional ϕ that assigns to each image a number that indicates, in some sense, the "undesirability" of the image.

Figure 1(a) shows a phantom that is a 243×243 digitized image (thus $L = 59,049$), representing a cross-section of a human head; c.f., [28, Section 4.3]. (We use this phantom in three out of the four experiments reported in this section.) The components of $\underline{\boldsymbol{x}}$ represent the average x-ray attenuations within the $59,049$ pixels, each of which is of size 0.0752×0.0752 (the assumed unit of length is 1 cm). The values of these components range from 0 to 0.5639; for display purposes any value below 0.204 is shown as black (gray value 0) and any value above 0.21675 is shown as white (gray value 255), with a linear mapping of the $\underline{\boldsymbol{x}}$-component values into gray values in between (this is true for all the figures throughout this paper).

In the first experiment, data were collected by calculating line integrals through the digitized image for 82 sets of equally spaced parallel lines. Each such line integral gives rise to a linear equation in the components of $\underline{\boldsymbol{x}}$; the set of $\underline{\boldsymbol{x}}$ that is consistent with such a line integral is a hyperplane in $\mathbb{R}^L$. These 82 sets are composed of 60 equally spaced views with 3° increments between consecutive directions and an additional 22 views with angle directions that make the large tumor on the left hand side of the phantom in Figure 1(a) a ghost (i.e., its ray-sums are zero for those 22 directions [30]). The vector $\underline{\boldsymbol{x}}$ that represents the phantom itself lies in the intersection of all the hyperplanes that are associated with these directions. In this experiment, we used measurements for $25,452$ lines, making our problem very much underdetermined. (The intersection of all the hyperplanes is an at least $33,597$-dimensional subspace of $\mathbb{R}^{59,049}$). In the terminology of Section 3, $Z = 25,452$ and, for $1 \leq z \leq Z$, D_z is one of the hyperplanes. It is typical in practice (and this is what we will be doing here) that each block B_u consists of all the indices z associated with the measurements taken in a particular direction (and therefore $\{1, \ldots, Z\} = \bigcup_{u=1}^{U} B_u$ with $U = 82$). We now describe the two convex functions ϕ that we have chosen for the first experiment.

Many researchers in image processing have been advocating the use of total variation, e.g., [9, 20, 34, 36, 38]. For a $G \times H$ image q whose pixels are denoted by $q_{g,h}$ ($1 \leq g \leq G, 1 \leq h \leq H$), the *total variation* (TV) of q is

$$TV(q) := \sum_{g=1}^{G-1} \sum_{h=1}^{H-1} \sqrt{(q_{g+1,h} - q_{g,h})^2 + (q_{g,h+1} - q_{g,h})^2}. \tag{36}$$

By mapping q into a ($G \times H = L$)-dimensional point $\underline{\boldsymbol{x}}$ (by stacking into a single column all the columns of q), this definition gives rise to a functional ϕ that can be used in our superiorization algorithm described in the previous section.

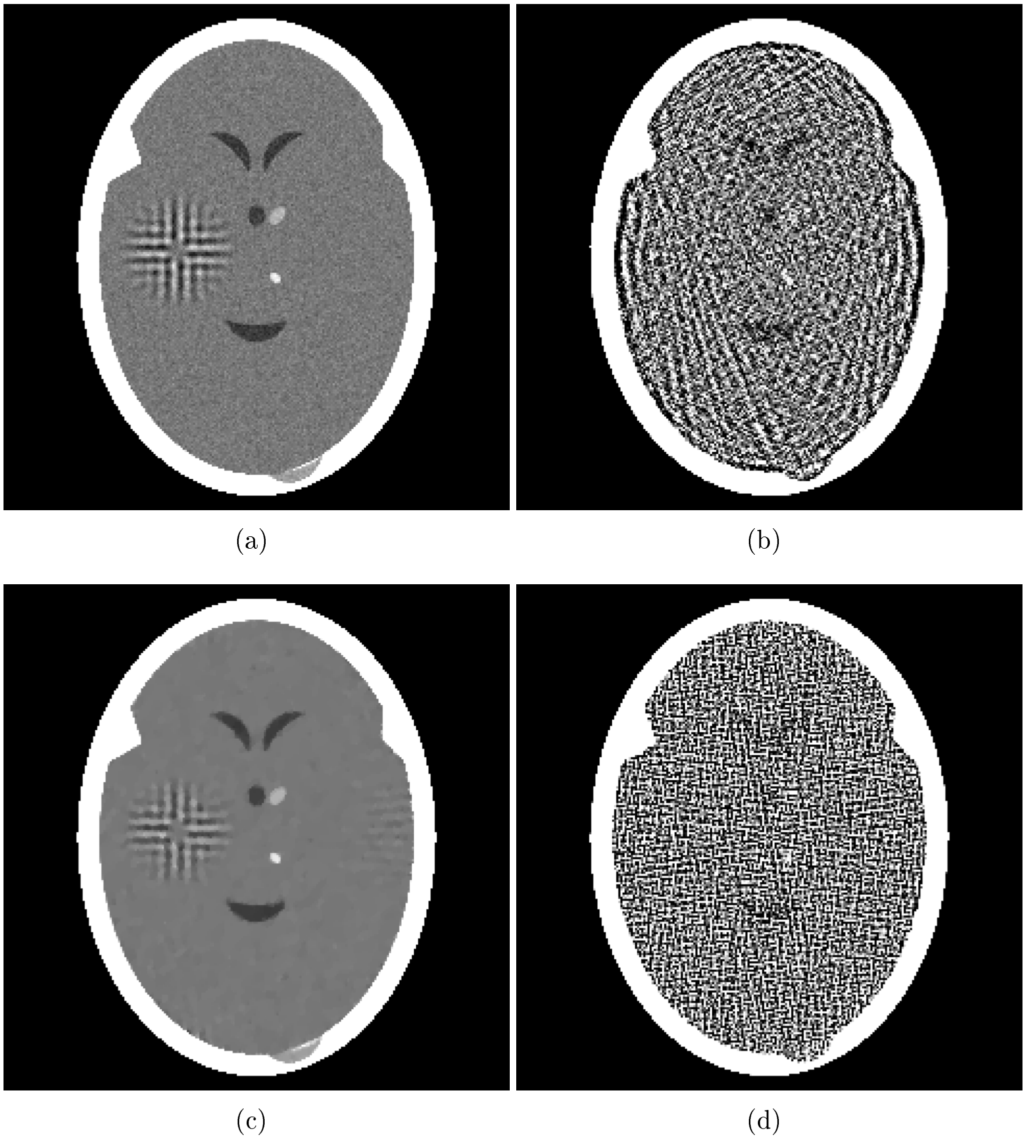


Figure 1: Illustration of reconstructions obtained by our block-iterative algorithm from 82 views, with hyperplane constraints and stopping criterion of $\mathrm{Res}(\underline{\boldsymbol{x}}^k) < 0.05$. (a) Head phantom for which data were collected. (b) Norm-minimizing reconstruction (no perturbations). (c) Reconstruction with TV-superiorizing perturbations. (d) Reconstruction with entropy-superiorizing perturbations.

Our second choice for the convex function ϕ to be used in this experiment is based on the maximum entropy formalism, which is a general scientific approach with whole books devoted to it; e.g., [33]. The suggestion that it be used for image reconstruction first appeared in the open literature in [26]. It has been extensively used in the related field of digital image restoration; see, e.g., [3]. An algorithm that finds the maximum entropy solution of a consistent system of equations with nonnegativity constraints is the Multiplicative Algebraic Reconstruction Technique (MART), as was proved in [32]. MART is actively used in various applications; see, e.g., [31, 39]. For images q that satisfy $q_{g,h} \geq 0$, for $1 \leq g \leq G, 1 \leq h \leq H$, with at least one value strictly greater than 0, the *negative-entropy* of q is

$$\textit{negative-entropy}(q) := \sum_{g=1}^{G-1}\sum_{h=1}^{H-1}\left(\frac{q_{g,h}}{Q}\right)\ln\left(\frac{q_{g,h}}{Q}\right), \tag{37}$$

where Q is a constant provided to us by estimating the the sum of the $q_{g,h}$ from the measured data (see the end of [28, Section 6.4] for a discussion as to why we may assume in image reconstruction from projections that this estimate is extremely accurate). We use negative entropy, rather than entropy, to define ϕ , since our algorithm is written for minimizing a function. Nevertheless, in reporting on our experiments we will be giving the value of the entropy (which is minus the value provided by (37)), thus a higher value will indicate a more desirable solution according to the maximum entropy principle. We need to point out that the previously-specified algorithm needs to be altered for entropy maximization, due to the fact that in that case ϕ is defined only on the nonnegative orthant. For example, there is nothing to prevent Step 3 of the pseudocode to produce a $\underline{\boldsymbol{y}}$ that is not in the domain of ϕ and this would make it impossible to execute Step 4. To avoid such difficulties, we put in additional conditions controlling the flow of the algorithm so that the ϕ based on negative entropy need never be evaluated for an argument outside the nonnegative orthant. We omit the technical details.

We applied our block-iterative algorithm (17) to the specified data set for the two choices of ϕ given above. The relaxation parameter for all reconstructions was set to $\lambda = 1$ and all were stopped when $\mathrm{Res}(\underline{\boldsymbol{x}}^k) < 0.05$. Figure 1(b) shows the reconstruction when no perturbations are introduced and Figure 1(c) shows the reconstruction when TV-superiorizing perturbations are present, set as specified in the previous section. Both runs started with $\underline{\boldsymbol{x}}^0$ being the *zero point* (i.e., all its components are 0). Note that

$\mathrm{Res}(\underline{\boldsymbol{x}}^0) = 330.208$, demonstrating that the value 0.05 for the stopping criterion is small in our context. Since the convex sets are hyperplanes, the algorithm without perturbations (i.e., $\gamma_k = 0$, for $k \in \mathbb{N}$) converges to the feasible point with minimal norm (see, [28, Section 11.2]). Clearly, the reconstruction in Figure 1(c) is visually superior to the reconstruction in Figure 1(b). As a numerical measure, the norm of the difference between the reconstruction without perturbations in Figure 1(b) and the phantom in Figure 1(a) is 3.764, while the norm of the difference between the reconstruction with the perturbations in Figure 1(c) and the phantom is 0.157, making it 24 times smaller than that without perturbations. Moreover, the TV of the image in Figure 1(c) is 421.568, which is near to (and is actually less than) the TV of the phantom in Figure 1(a) (450.594), whereas the TV of the image in Figure 1(b) is 1270.240.

We further report on the the results of the reconstruction when the perturbations are aimed at maximizing the entropy as defined by (37). For this run, $\underline{\boldsymbol{x}}^0$ was set to be the point that is also the initial point for MART [32] for which each component is either zero or another constant, selected based on the measurements so that the total density is the Q of (37). Figure 1(d) presents the resulting image for the chosen stopping criterion. If we compare the two superiorizing reconstruction results (Figures 1(c) and (d)), the TV-superiorizing reconstruction appears clearly preferable. However it is worth noting that the entropy-superiorizing algorithm behaves as advertised: a superior result relative to the function of (37) is provided by the entropy-superiorization algorithm. In fact, we ran MART (as implemented in the software package SNARK05 [11]) on the data of this experiment and obtained an entropy of 10.305. The entropy of the image in Figure 1(d) is 10.307, which is higher than that obtained by MART for the same data and the same stopping criterion, indicating that our algorithm performs better in the superiorization sense. However, looking at the image produced by TV-superiorization (Figure 1(c)) we see that, for this data set, the TV-minimization criterion characterizes the phantom better than the maximum entropy criterion.

As a further illustration, we report on results of an experiment in which in addition to the Z hyperplanes used in the previous experiment, we also have a convex set corresponding to nonnegativity constraints on the pixel values; i.e., the set

$$D_{Z+1} = \{\underline{\boldsymbol{x}} \mid x_l \geq 0, \text{ for } 1 \leq l \leq L\}. \tag{38}$$

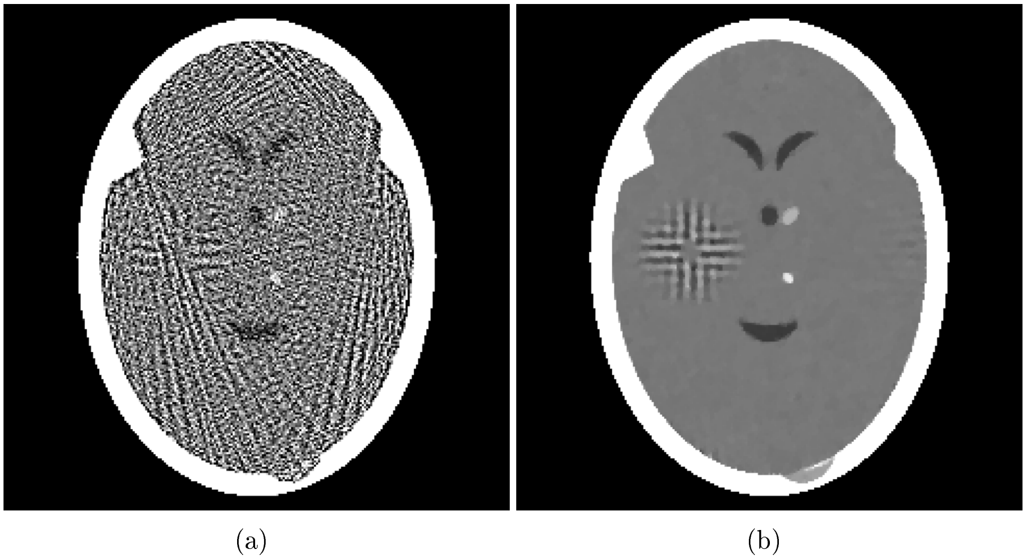


Figure 2: Illustration of reconstructions obtained by our block-iterative algorithm from 82 views, with hyperplane and nonnegativity constraints and stopping criterion of Res($\underline{\boldsymbol{x}}^k$)< 0.05. (a) Reconstruction when no perturbations are present. (b) Reconstruction with TV-superiorizing perturbations.

It is easy to see that if $\underline{\boldsymbol{y}} = P_{D_{Z+1}}\underline{\boldsymbol{x}}$, then $y_l = \max(x_l, 0)$, for $1 \leq l \leq L$. In the exact formulation of our algorithm for this case, we introduce in addition to the U blocks used in the previous case one more block B_{U+1} that contains R copies of the index $Z+1$. From (15) it follows that, for all $\underline{\boldsymbol{x}} \in \mathbb{R}^L$, $Q_{U+1}\underline{\boldsymbol{x}} = P_{D_{Z+1}}\underline{\boldsymbol{x}}$, which implies that the implementation of the operator Q_{U+1} is trivial. Our actual algorithm for taking care of nonnegativity (in addition to the hyperplane constraints) uses instead of (14) the operator

$$\mathbf{Q} := Q_{U+1}Q_U \cdots Q_{U+1}Q_1, \tag{39}$$

which means that in the sequence of the blocks of the convex sets, each B_u (i.e., each block of hyperplanes, as defined for the previous experiment) is followed by a projection onto D_{Z+1}.

We compared the performance of our block-iterative algorithm without and with perturbations for the TV functional (36). As in the previous experiment, we started both reconstructions with $\underline{\boldsymbol{x}}^0$ being the zero point, set $\lambda = 1$, and set the algorithms to stop at Res($\underline{\boldsymbol{x}}^k$)< 0.05. The results of the reconstructions are given in Figure 2. Note again the superiority of the reconstruction of the TV-superiorizing algorithm with perturbations present (Figure 2(b)) as compared to the one produced without perturbations (Figure 2(a)). The norm of the difference between the reconstruction without perturbations in Figure 2(a) and the phantom is more than 8 times greater than that between the reconstruction with the perturbations in Figure 2(b) and the phantom (1.254 and 0.143, respectively). The use of the nonnegativity constraint (38) improved the quality of the reconstructions in both cases, although only slightly in the TV-superiorizing case. The TV for the image on the right in Figure 2 is 421.185, while for that on the left it is 667.926.

Until now, all the reconstructions were from data sets in which the line integrals were calculated exactly based on a digitized phantom. The reconstruction algorithms that we used were in fact developed using the assumption that this is indeed the nature of the data. However, real data in applications of image reconstruction from projections will not be such. A basic reason is that the object from which the data are collected will not be a digitized one (i.e., there will be variations even within pixels). Also, the detectors used in real instruments for collecting data will have a width and so, even if they were otherwise perfect, they could not be used for measuring line integrals exactly. In addition, measurements are stochastic in nature; in computerized tomography (CT), for example, the line integrals are estimated by the use of a, by necessity, finite number of x-ray photons, resulting in statistical noise in these estimates. There are further sources of discrepancy between the assumed mathematical model and physically collected data, such as the presence of scattered x-ray photons corrupting the readings by the individual detectors; see [28, Chapter 3].

In the third experiment we demonstrate our block-iterative algorithm when the data are realistic from these points of view. Calculations of the line integrals were based on a geometrical description of the head phantom rather than on its digitization. In order to simulate the width of the detector, for each line for which the algorithm assumes that the data had been collected, we introduce 10 additional lines (five on both sides) with spacing $a/11$ between them, where a ($= 0.0752$ in our case) is the distance between parallel lines along which data are assumed to have been collected in the mathematical formulation. In addition, statistical noise and scatter were introduced at the levels found in real CT scanners. (The software SNARK05 [11], which we used for all our experiments, allows us to simulate this kind of data collection, the details of which are explained in [28, Section 4.4].) Data were generated for 360 sets of equally spaced parallel lines (each line gives

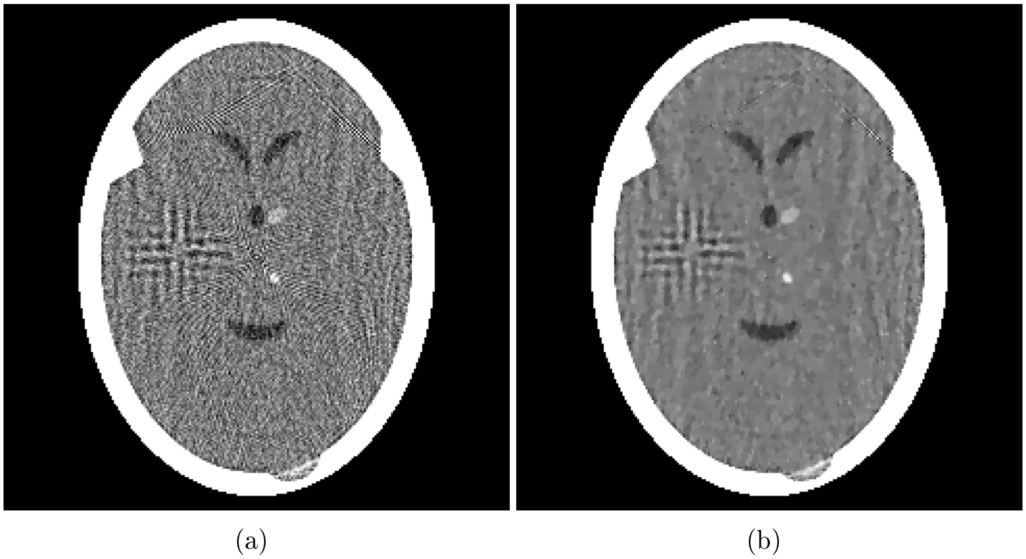
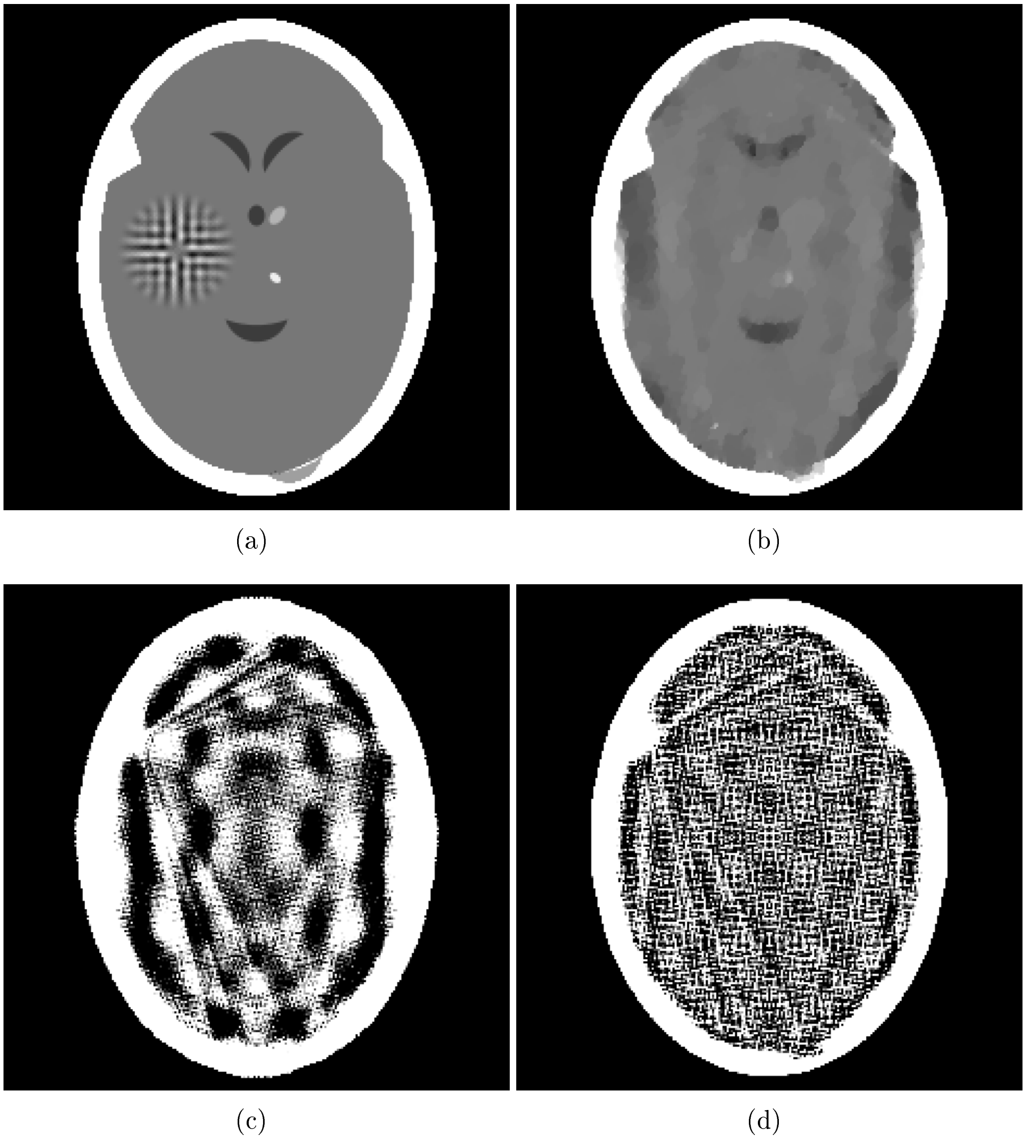


Figure 3: Illustration of reconstructions obtained by our block-iterative algorithm from 360 views of physically realistic projection data, with hyperplane and nonnegativity constraints and stopping criterion of Res$(\underline{\boldsymbol{x}}^k)< 3.75$. (a) Reconstruction when no perturbations are present. (b) Reconstruction with TV-superiorizing perturbations.

rise to a hyperplane in $\mathbb{R}^{59,049}$), with 0.5° increments between consecutive directions. Similarly to our previous experiments, blocks were formed using all measurements taken from a particular direction (making $U = 360$), and an additional constraint of the form (38) was utilized in the same manner as in the second experiment.

The reconstructions from such realistic data, both without and with TV-superiorizing perturbations, are shown in Figure 3. The stopping criterion in both cases was Res $\left(\underline{\boldsymbol{x}}^k\right) < 3.75$, which is reasonable since for this noisy data set the value of Res for the phantom is actually higher (3.97, to be exact). This means in particular that, due to the discrepancy between the actual mode of data collection and the mathematical model that is assumed for it, the phantom no longer lies in the intersection of all the constraints that our algorithm is designed to satisfy simultaneously. Nevertheless, both reconstructions seem to capture the essential features of the phantom, with the reconstruction on the right (obtained using TV-superiorizing perturbations) somewhat more pleasing to the eye than the reconstruction on the left; in particular, it seems to be smoother. This is reflected by the norms of the differences between the reconstructions and the phantom; they are 1.914 and 1.877, respectively, for Figure 3(a) and Figure 3(b). Comparing the TVs of the two reconstructed images, we note that the image on the right has a much lower TV (440.411) than the image on the left (518.779). (We point out again that the TV of the phantom is 450.594.) This (matching the results of the previous two experiments) indicates that the use of TV-superiorizing perturbations steers the iterates towards an image with a lower TV, which appears to be a desirable feature when attempting to reconstruct an object of the type shown in Figure 1(a).

In all the examples above, when the algorithm stops it provides an image with a value of the functional ϕ less than that of the phantom. The following example shows that this is not always so. Figure 4(a) shows a head phantom for which data were collected. The only difference between the head phantom in Figure 1(a) and this one is that in Figure 1(a) the phantom includes local inhomogeneities; i.e., each of the 243×243 pixels was multiplied by a sample from a normal distribution with mean 1 and a standard deviation that is appropriate to produce the variability found in actual CT scans of the human head. For the current example we decreased the number of views to 22 and calculated ideal line integrals through the digitized head phantom (similarly to the way we collected the data for the first experiment at the beginning of the section). This gave rise to 6,914 hyperplanes, thus making the problem even more underdetermined than in our previous examples. Here $U = 22$ and $Z = 6,914$, and so we have almost an order of magnitude fewer linear equations than unknowns. We again ran our algorithm starting from the zero point ($\mathrm{Res}(\underline{\boldsymbol{x}}^0) = 171.557$ in this case) and chose the stopping criterion of $\mathrm{Res}\left(\underline{\boldsymbol{x}}^k\right) < 0.05$. The TV of the phantom in Figure 4(a) is 421.735. Since the system is consistent, the phantom in Figure 4(a) lies in the intersection of all the constraints. Figure 4(b) shows the result of the TV-superiorizing when perturbations are present. The TV of that image (427.372) is higher than the TV of the phantom. Even though the reconstruction is visually poor, the superiority of it is seen by comparing it to the reconstruction when no perturbations are used in Figure 4(c) and when the perturbations are aimed at maximizing the entropy (Figure 4(d)). As a numerical measure, the norm of the difference between the reconstruction with perturbations in Figure 4(b) and the phantom in Figure 4(a) is 1.684, which is less than a fifth of the norm of the difference between Figure 4(c) and the phantom and less than half of the norm of the difference between Figure 4(d) and the phantom.



(a)

(b)

(c)

(d)

Figure 4: Illustration of reconstructions obtained by our block-iterative algorithm using consistent data with 22 views, hyperplane constraints and stopping criterion of Res($\underline{\boldsymbol{x}}^k$)$< 0.05$. (a) Head phantom for which data were collected. (b) Reconstruction with TV-superiorizing perturbations. (c) Reconstruction when no perturbations are present. (d) Reconstruction with entropy-superiorizing perturbations.

Up to this point we have been concentrating on the quality of our algorithm from the point of view of it achieving its aim of providing a point for which the values of both Res and the specified ϕ are small, without paying any attention to the computational costs involved. It happens to be the case that the timing of our algorithm can be greatly improved under certain circumstances. The reason for this is that the number R in (15) is large; it is 345 for our examples. From this it follows that $\underline{\boldsymbol{x}}^{k+1} = \mathbf{Q}\underline{\boldsymbol{x}}^{k}$ is practically the same as $\underline{\boldsymbol{x}}^{k}$ and so it takes quite a number of iterations to reach the stopping criterion. Fortunately, under many reasonable circumstances (for example, if all the convex sets are hyperplanes), it is possible to replace R by a much smaller number (such as 2 in our case) and achieve a more than an order of magnitude speed up while retaining the limiting convergence property stated in Theorem 2. We will publish the exact formulation, proof and illustration of this claim in a forthcoming paper; here we note only that similar approaches have been published already for algorithms without perturbations; e.g., in [14]. To indicate the great potential that exists for speeding up the block-iterative algorithm used in this section we note that it required 10,400 iterations to reach the very stringent stopping criterion that resulted in Figure 1 (we note that essentially the same image would have been obtained if we stopped much earlier), but when we used instead the perturbed fully sequential algorithm of [9], which is the same as the algorithm described in this paper with all blocks being of size 1, then the stopping criterion was reached in only 20 iterations. This is not necessarily a good thing from the point of view of superiorization, since perturbations are made only at the end of each iteration and so the value of the ϕ is likely to be much higher if the algorithm stops after relatively few iterations. In practice one would use an underrelaxed version of the fully sequential algorithm (i.e., a smallish λ in (17)), which would slow it down to some extent. Also, it is often the case that special hardware allows us to implement a block-iterative algorithm so that one iteration of it requires less time than one iteration of its fully sequential version. For these reasons, the speeded up versions of our perturbed block-iterative algorithm has the potential of being in practice advantageous.

6 Discussion

In this section we discuss the nature of our approach to solving our problem bearing in mind the existence of alternative approaches. We give reasons why feasibility problems of various kinds are looked at from the viewpoint of projection methods. Projections onto sets are used in a wide variety of methods in optimization theory, but not every method that uses projections really belongs to the class of projection methods. As mentioned earlier, *projection methods* are iterative algorithms that use projections onto individual sets, relying on the general principle that when a family of (usually closed and convex) sets is present then projections onto the given individual sets are easier to perform than projections onto other sets (intersections, image sets under some transformation, etc.) that are derived from the given individual sets.

The main advantage of projection methods, which makes them successful in real-world applications, is computational. They commonly have the ability to handle huge-size problems of dimensions beyond which other, more sophisticated currently available, methods cease to be efficient. This is so because the building bricks of a projection algorithm are the projections onto the given individual sets (assumed and actually easy to perform) and the algorithmic structure is either sequential or simultaneous (or in-between, i.e., block-iterative, as discussed in the current paper). Sequential algorithmic structures cater for the row-action approach (see Section 1.3 on classification of parallel algorithms in [17]) while simultaneous algorithmic structures favor parallel computing platforms, see, e.g., [15].

In the problems discussed in the previous section, the number of unknowns was 59,049. In the examples given in [29] (a paper devoted to radiation therapy planning), the number of unknowns was 128,668 and the feasible constraint set consisted of approximately three times as many linear inequality constraints. In four of the six cases reported there, the projection method called "ART3+" found a feasible point in less than three seconds, and in the remaining two cases a feasible point was found in less than 34 seconds. These times are for a standard PC, using an Intel Xeon 1.7 GHz processor and 1 G RAM. The problem in [29] is actually small compared to some of the other applications for which projection methods have been successfully used. In [10] (a paper devoted to reconstruction from electron microscopic projections), there are examples in which 16,777,216 unknowns are to be recovered from 4,587,520 measurements (each giving an approximate linear equality) and others in which 884,436 unknowns are to be recovered from 92,160,000 measurements. Projection methods (similar to the algorithm presented in this paper, but without perturbations) were used in [10] to handle such large problems in a reasonable time.

As true for all methodologies, projection methods are not necessarily the approach of choice in all applications. However, in the important applications mentioned in this paper (bio-medicine and image processing), projection methods work well and have been used successfully for a long time. A mathematical reason for this is that for these problems the angles between hyperplanes or half-spaces, represented by the linear equalities or linear inequalities, are in general large due to the high sparsity of the system matrix.

In a recent paper [27] Gould wrote, referring to projection methods for convex feasibility problems: "Unfortunately, particularly given the large literature which might make one think otherwise, numerical tests indicate that in general none of the variants considered are especially effective or competitive with more sophisticated alternatives." Gould also stated there: "We do not want to suggest here that successive and simultaneous projection methods are not useful, since in particular they appear to have been applied successfully for many years in medical imaging, radiation therapy planning and signal processing [references]. But our experience suggests that despite the large literature devoted to theoretical analysis, they should not be considered as the method of choice for a given application without further strong empirical evidence to support such a claim." The largest problem tried in [27] has $22,275$ unknowns and $6,630$ linear constraints and, for that problem, all methods reported in [27] needed 42 seconds or more to reach the stopping point used there on an 3.06 GHz Dell Precision 650 Workstation, which should be nearly twice as fast as the computer used in [29]. Comparing this with the results from [29] quoted above indicates that Gould's criticism of projection methods may very well be invalid for the applications mentioned in the previous paragraph.

Projection methods are not restricted to convex feasibility problems. The same general principle of reaching a goal associated with a family of sets by performing projections onto individual members of the family works also when solving certain optimization problems, see, e.g., [17, Chapter 6]. Another problem that is related to the convex feasibility problem is the *best approximation problem* of finding the projection of a given point onto the nonempty intersection of a family of closed convex subsets, see, e.g., [23]. The convex sets commonly represent mathematical constraints obtained from the modeling of a real-world problem. In the convex feasibility approach any point in the intersection is an acceptable solution to the real-world problem, whereas the best approximation formulation is usually appropriate if some initial point has been obtained from modeling and computational efforts that did not take into account the constraints, and now we wish to incorporate them by seeking a point in the intersection of the convex sets that is closest to the initial point. Iterative projection algorithms for finding a projection of a point onto the intersection of sets are more complicated then algorithms for finding just any feasible point in the intersection because they must have, in their iterative steps, some built-in "memory" mechanism to remember the original point whose projection is sought after. The sequential or parallel algorithms of Dykstra (see [23, p. 207]), of Haugazeau (see [6]), and of Halpern-Lions-Wittmann-Bauschke (see [4]) employ variants of such memory mechanisms. It is well-known that the convex feasibility problem is a special case of the *common fixed points problem.* In that problem a common fixed point of a family of operators, not necessarily projectors, is sought. This is a fundamental problem in fixed point theory and there is a large, and ever growing, body of literature available on theoretical issues of generalizations and convergence associated with projection methods in that mathematical field as well as in optimization oriented papers. In our paper we have not considered the generalizations and extensions discussed in this paragraph, but it appears to us that perturbation-resilient block-iterative projection methods are likely to play a role in their future developments.

Acknowledgments. This work was supported by grant No. 2003275 of the United States-Israel Binational Science Foundation (BSF) and by Award Number R01HL070472 from the National Heart, Lung, And Blood Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Heart, Lung, And Blood Institute or the National Institutes of Health. The authors are grateful to Eilat Vardi-Gonen for comments on earlier versions of the paper.

References

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

[2] A. Aleyner and S. Reich. Block-iterative algorithms for solving convex feasibility problems in Hilbert and in Banach spaces. *Journal of Mathematical Analysis and Applications*, 343:427–435, 2008.

[3] H.C. Andrews and B.R Hunt. *Digital Image Restoration.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1977.

[4] H.H. Bauschke. The approximation of fixed points of compositions of nonexpansive mappings in Hilbert space. *Journal of Mathematical Analysis and Applications*, 202:150–159, 1996.

[5] H.H. Bauschke and J.M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38:367–426, 1996.

[6] H.H. Bauschke and P.L. Combettes. A weak-to-strong convergence principle for Fejér-monotone methods in Hilbert spaces. *Mathematics of Operations Research*, 26:248–264, 2001.

[7] H.H. Bauschke, E. Matoušková, and S. Reich. Projection and proximal point methods: convergence results and counterexamples. *Nonlinear Analysis: Theory, Methods and Applications*, 56:715–738, 2004.

[8] J.R. Bilbao-Castro, J.M. Carazo, I. García, and J.J. Fernández. Parallel iterative reconstruction methods for structure determination of biological specimens by electron microscopy. *Proceedings of the International Conference on Image Processing (ICIP)*, 1:1565–1568, 2003.

[9] D. Butnariu, R. Davidi, G.T. Herman, and I.G. Kazantsev. Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems. *IEEE Journal of Selected Topics in Signal Processing*, 1:540–547, 2007.

[10] J.M. Carazo, G.T. Herman, C.O.S. Sorzano, and R. Marabini. Algorithms for three-dimensional reconstruction from imperfect projection data provided by electron microscopy. In : J. Frank, editor, *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell.* 2nd edition, Springer Science+Business Media, LLC, New York, NY, USA, 2006, pages 217–243.

[11] B. Carvalho, W. Chen, J. Dubowy, G.T. Herman, M. Kalinowski, H.Y. Liao, L. Rodek, L. Ruskó, S.W. Rowland, and E. Vardi-Gonen. *SNARK05: A Programming System for the Reconstruction of 2D Images from 1D Projections.* www.snark05.com/SNARK05.pdf.

[12] Y. Censor, M.D. Altschuler, and W.D. Powlis. On the use of Cimmino's simultaneous projections method for computing a solution of the inverse problem in radiation therapy treatment planning. *Inverse Problems*, 4:607–623, 1988.

[13] Y. Censor, T. Elfving, and G.T. Herman. Averaging strings of sequential iterations for convex feasibility problems. In : D. Butnariu, Y. Censor, and S. Reich, editors, *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications.* Elsevier Science Publishers, Amsterdam, The Netherlands, 2001, pages 101–113.

[14] Y. Censor, T. Elfving, G.T. Herman, and T. Nikazad. On diagonally relaxed orthogonal projection methods. *SIAM Journal on Scientific Computing*, 30:473–504, 2007.

[15] Y. Censor, D. Gordon, and R. Gordon. BICAV: A block-iterative, parallel algorithm for sparse systems with pixel-related weighting. *IEEE Transactions on Medical Imaging*, 20:1050–1060, 2001.

[16] Y. Censor and E. Tom. Convergence of string-averaging projection schemes for inconsistent convex feasibility problems. *Optimization Methods and Software*, 18:543–554, 2003.

[17] Y. Censor and S. Zenios. *Parallel Optimization: Theory, Algorithms and Applications.* Oxford University Press, New York, NY, USA, 1997.

[18] P.L. Combettes. The foundations of set-theoretic estimation. *Proceedings of the IEEE*, 81:182–208, 1993.

[19] P.L. Combettes. The convex feasibility problem in image recovery. *Advances in Imaging and Electron Physics*, 95:155–270, 1996.

[20] P.L. Combettes and J.C. Pesquet. Image restoration subject to a total variation constraint. *IEEE Transactions on Image Processing*, 13:1213–1222, 2004.

[21] G. Crombez. Finding common fixed points of strict paracontractions by averaging strings of sequential iterations. *Journal of Nonlinear and Convex Analysis*, 3:345–351, 2002.

[22] G. Crombez. Finding common fixed points of a class of paracontractions. *Acta Mathematica Hungarica*, 103:233–241, 2004.

[23] F. Deutsch. *Best Approximation in Inner Product Spaces.* Springer-Verlag, New York, NY, USA, 2001.

[24] P.P.B. Eggermont, G.T. Herman, and A. Lent. Iterative algorithms for large partitioned linear systems, with applications to image reconstruction. *Linear Algebra and Its Applications*, 40:37–67, 1981.

[25] T. Elfving. Block-iterative methods for consistent and inconsistent linear equations. *Numerische Mathematik*, 35:1–12, 1980.

[26] R. Gordon, R. Bender, and G.T. Herman. Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and x-ray photography. *Journal of Theoretical Biology*, 29:471–481, 1970.

[27] N.I.M. Gould. How good are projection methods for convex feasibility problems? *Computational Optimization and Applications*, 40:1–12, 2008.

[28] G.T. Herman. *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography.* Academic Press, New York, NY, USA, 1980.

[29] G.T. Herman and W. Chen. A fast algorithm for solving a linear feasibility problem with application to intensity-modulated radiation therapy. *Linear Algebra and Its Applications*, 428:1207–1217, 2008.

[30] G.T. Herman and R. Davidi. On image reconstruction from a small number of projections. *Inverse Problems*, 24:045011, 2008.

[31] A.B. Konovalov, V.V. Vlasov, D.V. Mogilenskikh, O.V. Kravtsenyuk, and V.V. Liubimov. Algebraic reconstruction and postprocessing in one-step diffuse optical tomography. *Quantum Electronics*, 38:588–596, 2008.

[32] A. Lent. A convergent algorithm for maximum entropy image restoration, with a medical x-ray application. In R. Shaw, editor, *Image Analysis and Evaluation*, pages 249–257. Society of Photographic Scientists and Engineers (SPSE), Washington, DC, USA, 1977.

[33] R.D. Levine and M. Tribus, editors. *The Maximum Entropy Formalism.* MIT Press, Cambridge, MA, USA, 1979.

[34] F. Malgouyres. Minimizing the total variation under general convex constraints for image restoration. *IEEE Transactions on Image Processing*, 11:1450–1456, 2002.

[35] L.D. Marks, W. Sinkler, and E. Landree. A feasible set approach to the crystallographic phase problem. *Acta Crystallographica*, A55:601–612, 1999.

[36] M. Persson, D. Bone, and H. Elmqvist. Total variation norm for three-dimensional iterative reconstruction in limited view angle tomography. *Physics in Medicine and Biology*, 46:853–866, 2001.

[37] G. Pierra. Decomposition through formalization in a product space. *Mathematical Programming*, 28:96–115, 1984.

[38] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[39] N.A. Worth and T.B. Nickels. Acceleration of Tomo-PIV by estimating the initial volume distribution. *Experiments in Fluids*, 45:847–856, 2008.