Algorithms for Superiorization and their Applications to Image Reconstruction

by

Ran Davidi

A dissertation submitted to the Graduate Faculty in Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy, The City University of New York

2010

This manuscript has been read and accepted for the
Graduate Faculty in Computer Science in satisfaction of the
dissertation requirements for the degree of Doctor of Philosophy.

Date — Dr. Gabor T. Herman, Chair of Examining Committee

Date — Dr. Theodore Brown, Executive Officer

Dr. Robert Haralick

Dr. Ioannis Stamos

Dr. Jose-Maria Carazo
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Algorithms for Superiorization and their Applications to Image Reconstruction

by

Ran Davidi

Adviser: Gabor T. Herman

Computational tractability with limited computing resources is a major barrier for the ever increasing problem sizes of constrained optimization models that seek a minimum of an objective function satisfying a set of constraints. On the other hand, there exist efficient and computationally much less demanding iterative methods for finding a feasible solution that only fulfills the constraints. These methods can handle problem sizes beyond which existing optimization algorithms cannot function. To bridge this gap we present a new concept called *superiorization*, envisioned methodologically as lying between optimization and feasibility seeking. It enables us to use efficient iterative methods to steer the iterates toward a point that is feasible and superior, but not necessarily optimal, with respect to the given objective/merit function.

Using efficient iterative methods to do 'superiorization' instead of 'full constrained optimization' or only 'feasibility' is a new tool for handling mathematical models that include constraints and a merit function. The target improvement of the superiorization methodology is to affect the computational treatment of the mathematical models so that we can reach solutions that are desirable from the point of view of the application at hand at a relatively small computational cost. The key to superiorization is our discovery that two principal prototypical algorithmic schemes, string-averaging projections and block-iterative projections, which

include many projection methods, are bounded perturbation resilient. While stability of algorithms under perturbations is usually made to cope with all kinds of imperfections in the data, here we have taken a proactive approach designed to extract specific benefits from the kind of stability that we term *perturbation-resilience*. Superiorization uses perturbations proactively to reach feasible points that are superior, according to some criterion, to the ones to which we would get without employing perturbations. In this work, we set forth the fundamental principle of the superiorization methodology, give some mathematical formulations, theorems and results, and show potential benefits in the field of image reconstruction from projections.

To my lovely Anat.

Acknowledgments

Writing this dissertation signifies the end of a journey that started in May 2004. I am looking back on it with a smile because it gave me the honor and opportunity to meet and interact with so many great and wonderful people. I wish to thank them all for helping me to reach where I am today and accompanying me during my pursuit of a doctorate degree. It would have not been possible without them.

First and foremost, I wish to express my deepest gratitude to my advisor, Dr. Gabor T. Herman. He has always made himself available and was always very responsive, even while traveling and visiting other parts of the world. His total commitment to guide and mentor me was always above and beyond any expectation. Throughout our joint research projects or in our conversations on topics outside academia, his insight and infinite expertise are truly remarkable. His original ideas and approach to problems always fascinated me, I've learned so much from his advice and encouragement on all aspects of life and am deeply honored he was willing to be my advisor. I will value his guidance and friendship for the rest of my life.

A special thanks also goes to the rest of my committee members for dedicating their time and ideas into this work. I wish to thank Dr. Robert Haralick for attending my talks every semester and for providing his valuable advice and constructive guidance throughout the years. I would also like to thank Dr. Stamos - I first met Dr. Stamos as a freshmen at Hunter College almost ten years ago, I am very honored he took interest in my work over the years and was involved in my academic life. I am grateful to Dr. Carazo for making himself available to participate in my committee despite his busy schedule; I am honored that he has been taking interest in my research.

I am grateful for the interactions I had with the many visitors of the Discrete Imaging and Graphics (DIG) Group throughout the years. I am especially thankful to Dr. Yair Censor who took interest in my work and with whom I had the chance to

collaborate on several occasions. Our fruitful discussions always intrigued me and advanced my thinking and my research work. Another special thanks goes to the late Dr. Dan Butnariu, who taught me the basics of perturbations and also helped shape my research work at its early stages. A thank you goes to Dr. Patrick Combettes, Dr. Ivan Kazantsev and Dr. Touraj Nikazad with whom I had the pleasure to collaborate and publish joint results. I also wish to thank Dr. Andreas Alpers, Laslo Cernetic, Dr. Edgar Garduño, the late Dr. Attila Kuba, Arun Kulshreshth, Dr. Homeira Pajoohesh, Lajos Rodek, László Ruskó and Dr. Sjors Scheres, for their critique and feedback on my presented work.

I would like to express my deepest gratitude to all past and present members of the DIG group. They are: Dr. Wei Chen, Joel Dubowy, James Harlacher, Dr. Miroslaw Kalinowski, Joanna Klukowska, Dr. Hstau Liao, Lucas de Melo Oliveira, Dr. Stuart Rowland, Deniz Sarioz and Dr. Eilat Vardi-Gonen. Their friendship and assistance throughout the doctorate degree was invaluable and made a tremendous contribution to this dissertation.

Allow me to also thank Joseph Driscoll and Lina Garcia for their great enthusiasm whenever they helped with all my needs over the years at the Graduate Center.

I would like to acknowledge the support I received along the way. Specifically, I wish to thank Dr. Gabor T. Herman who supported me through his grant from the National Institute of Health, grant number HL070472, and through his National Science Foundation grant number DMS0306215. I would like to thank Dr. Ted Brown for his generous help throughout the years and the Graduate Center of CUNY, for providing me a science fellowship, a tuition fellowship, a university fellowship and a presidential fellowship. Without all of them - this work would not be possible.

Outside academia, I want to thank my family, who gave me the courage to walk this path. My parents, Leah and David, who've always been there for me and allowed me to get to where I am today. My brother Gal, for his continued

encouragement, help and dedication. My daughter Lia, who made me the luckiest person and shares her joy of life with me everyday. And of course, I am eternally grateful to my wife Anat, for her endless patience, for always being by my side and for believing in me; I dedicate this dissertation to you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Convex feasibility problem

Many significant real-world problems are modeled by constraints that force the sought-after solution vector to fulfill conditions imposed by the physical nature of the problem. Such a modeling approach leads often to a *convex feasibility problem* of the form

$$\text{find } x^* \in C = \cap_{i=1}^{I} C_i, \tag{1.1}$$

where the sets $C_i \subseteq \mathbb{R}^J$ are closed convex subsets of the vector space $\mathbb{R}^J$ and I and J are positive integers. Such points x^* are called *feasible*. This is a fundamental problem in many areas of mathematics and the physical sciences including image reconstruction from projections [35], radiation therapy treatment planning [11], and crystallography [49], to name but a few. In the literature, an iterative method that provides a sequence of points $\{x^k\}_{k\in\mathbb{N}}$ that converges to a point in C is considered to solve the problem in (1.1) [4, 19].

1.2 Projection methods

For any closed convex set $C \subseteq \mathbb{R}^J$ there is an operator $P_C : \mathbb{R}^J \to C$ such that, for any $x \in \mathbb{R}^J$, $P_C x$ is the unique point in C nearest to x. $P_C x$ is referred to as the (orthogonal) *projection* of x onto C.

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., their intersection) that are derived from the given individual sets. Projection algorithms have various algorithmic structures and they possess desirable convergence properties and/or good initial behavior; see, e.g., [4] and [19, Chapter 5]. They also posses in some cases computational advantage over alternative methods that makes them successful in real-world applications [12]. Here we limit our discussion to two classes of projection methods, in particular, string-averaging projection (SAP) methods (introduced in [16], see also [18] and [24]) and block-iterative projection (BIP) methods (proposed in [28, 29]). More specifically, within the SAP methods, we concentrate on a large subset of algorithms that are called amalgamated projection (AP) methods [8].

1.2.1 Amalgamated projection (AP) methods

Among the AP methods are most of the already known metric projection methods for solving the consistent convex feasibility problem. We define what an AP method is, show how some of these well-known algorithms are in this class and later provide some illustrations.

By an *index vector* we mean a vector $t = (t_1, ..., t_p)$ whose coordinates t_j are in

the set $\{1,...,I\}$. If $t=(t_1,...,t_p)$ is an index vector, then we denote

$$P[t]=P_{C_{t_p}}\cdots P_{C_{t_1}}. \tag{1.2}$$

A finite set Ω of index vectors is called *fit* (to the given feasibility problem) if, for each $i\in\{1,...,I\}$, there exists $t=(t_1,...,t_p)\in\Omega$ such that $t_s=i$ for some $s\in\{1,...,p\}$. We denote by Ω_i the set of those $t\in\Omega$ for which i is a coordinate of t. If Ω is a fit set of index vectors, then a function $w:\Omega\to\mathbb{R}_{++}=(0,\infty)$ is called a *fit weight function* if $\sum_{t\in\Omega}w(t)=1$. A pair (Ω,w) consisting of a fit set of index vectors and a fit weight function defined on it is called an *amalgamator*. We define the operator $\mathbf{P}:\mathbb{R}^J\to\mathbb{R}^J$ by

$$\mathbf{P}x=\sum_{t\in\Omega}w(t)P[t]x. \tag{1.3}$$

This operator is continuous because each P_i is a continuous function on $\mathbb{R}^J$. $\mathbf{P}$ is also nonexpansive (we define nonexpansive operator in Subsection 2.1). (The P_i are well known to be nonexpansive and hence so are the $P[t]$, for any index vector t. The result for $\mathbf{P}$ follows from (1.3) and the inequality properties for the norm of a linear combination of vectors.)

With these notions and notations, the *amalgamated projection method for the amalgamator* (Ω,w) is

$$\begin{aligned} x^0 &\in \mathbb{R}^J, \\ x^{k+1} &= \mathbf{P}x^k,\ \forall k\in\mathbb{N}. \end{aligned} \tag{1.4}$$

Clearly, such a method is a SAP scheme, as presented in [16] and further studied in [5, 6, 18, 24, 25].

Many known projection methods can be described as AP methods. For instance,

if $\Omega = \{t\}$ with $t = (1,2,...,I)$, then $w(t) = 1$ and the AP method with the amalgamator (Ω, w) gives the cyclic projection method originating in [64]; if the amalgamator (Ω, w) consists of $\Omega = \{t_1, ..., t_I\}$, where, for each $i \in \{1, ..., I\}$, $t_i = (i)$, and $w(t_i) = w_i > 0$ with $\sum_{i=1}^{I} w_i = 1$, then the AP method gives a generalization of the Cimmino algorithm originating in [20]. More generally, it can be seen easily by direct comparison that the unrelaxed versions of many projection methods for solving convex feasibility problems (such as those discussed in [4, 16, 18, 59]) can be described as AP methods.

In order to see that some relaxed projection methods are also describable as AP methods, we point out that the convex feasibility problem is equivalent to: Find a common point of the sets $C_1, ..., C_I, C_{I+1}$ where

$$C_{I+1} = \mathbb{R}^J. \tag{1.5}$$

In this case, $P_{C_{I+1}} x = x$, for all $x \in \mathbb{R}^J$. If (Ω, w) is an amalgamator of the original problem of finding a common point of the sets $C_1, ..., C_I$, then for any $\alpha \in (0,1)$, the pair (Ω', w'), where $\Omega' = \Omega \cup \{(I+1)\}$ and $w' : \Omega' \to \mathbb{R}_+$ is defined by

$$w'(t) = \begin{cases} \alpha w(t), & \text{if } t \in \Omega, \\ \\ 1 - \alpha, & \text{if } t = (I+1), \end{cases} \tag{1.6}$$

is an amalgamator of the problem of finding a common point of the sets $C_1, ..., C_{I+1}$, and the AP method for the amalgamator (Ω', w') is the relaxed iterative algorithm:

$$\begin{aligned} x^0 &\in \mathbb{R}^J, \\ x^{k+1} &= (1-\alpha)x^k + \alpha \mathbf{P} x^k, \ \forall k \in \mathbb{N}. \end{aligned} \tag{1.7}$$

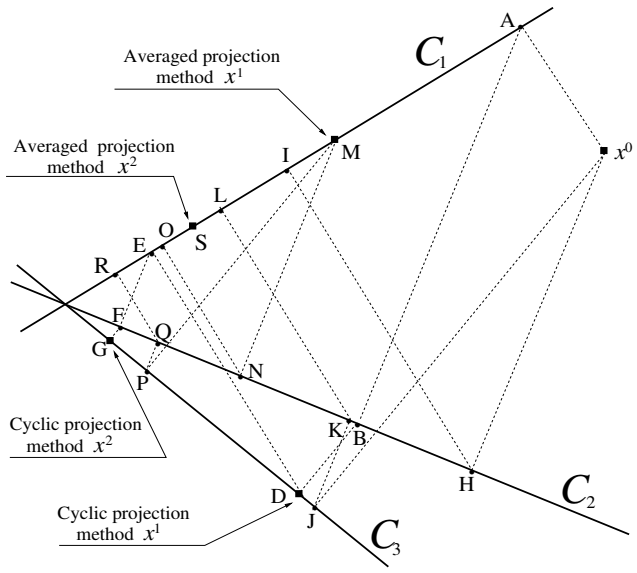


Figure 1.1: Illustration of the cyclic projection and the averaged projection methods for $I = 3$. In the cyclic projection method, we get to x^2 by starting at x^0 and going through A, B, D, E, F to G. In the averaged projection method x^1 is the point M, which is the center of mass of the points A, I and L. The next step of the averaged projection method produces the point S, which is the center of mass of the points M, O and R.

We illustrate the way in which the AP methods function in some particular instances and how they compare with the cyclic projection method whose convergence behavior is well researched. Consider the amalgamator (Ω, w) for which Ω consists of I index vectors of the form

$$(1), (2,1), (3,2,1), \ldots, (I, I-1, ..., 1), \tag{1.8}$$

and $w(t) = 1/I$. The functioning of the corresponding algorithm (which we call here the *averaged projection method*) in the case $I = 3$ is illustrated by Figure 1.1, where two iterates x^1 and x^2 are indicated in parallel with the first two iterates of the cyclic projection method starting from the same initial point x^0. In the illustrated case, the second iterate of the cyclic projection method is closer to the common point of the sets (lines) C_1, C_2 and C_3 than the second iterate of the averaged projection method.

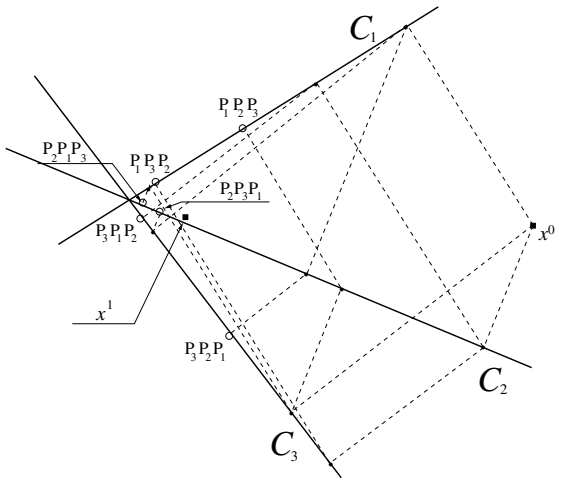


Figure 1.2: A step of the algorithm with amalgamator (Ω, w) for which Ω contains all permutations of $\{1,2,3\}$ and $w(t) = 1/6$ for all t. Here, x^1 is the center of mass of the six points $P_1P_2P_3,\ P_1P_3P_2,\ \ldots,\ P_3P_2P_1$.

Another possible amalgamator (Ω, w) is one for which Ω consists of all $I!$ permutations of the set $\{1,...,I\}$ and $w(t) = 1/(I!)$ for each $t \in \Omega$. The functioning of the resulting AP method is illustrated in Figure 1.2. It is interesting to note that, in Figure 1.2, the point marked $P_3P_2P_1$ is, in fact, the first iterate of the cyclic projection method. This is not as close to the solution of the feasibility problem as the iterate x^1 generated by this variant of the AP method.

1.2.2 Block-iterative projection (BIP) methods

Block-iterative projection methods 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 cyclic projection method discussed earlier (when each set is considered as one block) as well as the fully-simultaneous algorithmic structure (when the entire family of sets is considered as one block), but many additional in-between structures are permitted and convergence to a solution of the convex feasibility problem

(1.1) is guaranteed under reasonable conditions. In the literature many papers have been devoted to their properties, e.g., [1, 2, 14, 17, 28, 29] to name a few.

In Figure 1.3 we illustrate a BIP algorithm using six sets (lines) and two blocks. The algorithm starts from x^0 and get to x^1 by taking a convex combination after projecting onto the first block and get to x^2 by repeating the process onto the second block. We point out that the resulting point of the convex combination does not necessarily satisfy any of the sets as presented in the illustration.

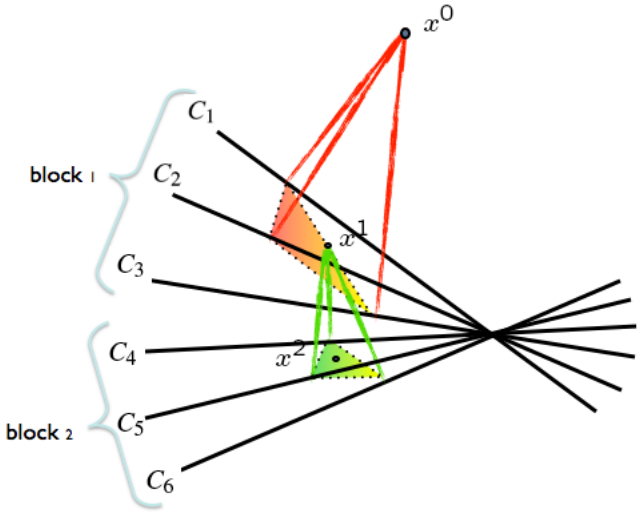


Figure 1.3: Illustration of a BIP algorithm with 6 sets and 2 blocks. The algorithm starts from x^0 and gets to x^1 after taking a convex combination of the projections onto the individual sets belonging to the first block. It gets to x^2 after repeating the same operation starting from x^1.

Chapter 2

Perturbation-resilient (PR) algorithms

We discuss algorithms that are perturbation-resilient and concentrate on the two classes of projection methods described in the previous chapter, namely, AP and BIP methods. We first define terms that are used in the discussion of the chapter, and then provide theorems (followed by proofs to support them).

2.1 Definitions

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

$$d(x,C) = \inf\{\|x-y\| \mid y \in C\}, \tag{2.1}$$

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

$$\|x-y\| = d(x,C). \tag{2.2}$$

This y is called the *projection of x onto C* and is denoted by $P_C x$.

For any operator $\mathbf{O}: \mathbb{R}^J \to \mathbb{R}^J$, we use $Fix(\mathbf{O})$ to denote the set $\left\{x \in \mathbb{R}^J | \mathbf{O}(x) = x\right\}$ of *fixed points* of $\mathbf{O}$. We say that $\mathbf{O}$ is *nonexpansive* if $\|\mathbf{O}x - \mathbf{O}y\| \leq \|x - y\|$, for all $x, y \in \mathbb{R}^J$, and that $\mathbf{O}$ is *attracting with respect to F*, where F is a subset of $\mathbb{R}^J$, if for every $x \in \mathbb{R}^J \backslash F$ and $y \in F$, $\|\mathbf{O}x - y\| < \|x - y\|$.

Let $A \in \mathbb{R}^{J \times J}$ be a matrix. Then $\rho(A)$ denotes the *spectral radius* of A and is defined as

$$\rho(A) = \max_{1 \leq i \leq I} |\mu_i|, \tag{2.3}$$

where μ_i are the eigenvalues of the matrix A. If A is a $J \times J$ symmetric matrix then we say that A is *positive definite* if, for all nonzero $z \in \mathbb{R}^J$, $z^T A z > 0$.

For any $x \in \mathbb{R}^J$, we say that the sequence $\left\{x^k\right\}_{k \in \mathbb{N}}$ is *asymptotically regular* if

$$\lim_{k \to \infty} \left\| x^{k+1} - x^k \right\| \to 0. \tag{2.4}$$

2.2 Perturbation-resilient AP methods

The following result says that the convergence of an AP method to a solution of the convex feasibility problem (1.1) is stable under summable perturbations of the iterates.

Theorem 1 *Let C_i, $1 \leq i \leq I$, be closed convex sets with a nonempty intersection C. If $\{\beta_k\}_{k \in \mathbb{N}}$ is a sequence of nonnegative real numbers such that $\sum_{k=0}^{\infty} \beta_k < \infty$ and $\left\{v^k\right\}_{k \in \mathbb{N}}$ is a bounded sequence of vectors in $\mathbb{R}^J$, then for any amalgamator (Ω, w) the procedure*

$$\begin{aligned} x^0 &\in \mathbb{R}^J, \\ x^{k+1} &= \mathbf{P}(x^k + \beta_k v^k), \ \forall k \in \mathbb{N} \end{aligned} \tag{2.5}$$

converges, and its limit is in C.

Proof The proof is in two stages. In the first, we prove convergence when there are no perturbations of the computational process.

Claim 1 *If $\beta_k = 0$ for all $k \in \mathbb{N}$, then the sequence $\{x^k\}_{k\in\mathbb{N}}$ specified by* (2.5) *(or, equivalently in this case, by* (1.4)*) converges to a fixed point of* **P**.

This claim is a consequence of Theorem 2.2 of [16], here we give an alternative proof.

Let $i \in \{1,...,I\}$ and recall that the projection operator P_{C_i} is 1-attracting with respect to C_i (in the sense given to this term in [4]), that is, it satisfies

$$\|z - P_{C_i}x\|^2 + \|P_{C_i}x - x\|^2 \leq \|z - x\|^2, \tag{2.6}$$

for any $z \in C_i$ and $x \in \mathbb{R}^J$. Therefore, according to [4, Proposition 2.10], the operator $P[t]$ is $2^{-p(t)}$-attracting with respect to the set C, that is, for any $z \in C$ and $x \in \mathbb{R}^J$,

$$\|z - P[t]x\|^2 + 2^{-p(t)}\|P[t]x - x\|^2 \leq \|z - x\|^2, \tag{2.7}$$

where $p(t)$ is the length of the index vector t. In particular, for any $z \in C$ and $k \in \mathbb{N}$, we have that

$$\left\|z - P[t]x^k\right\|^2 + 2^{-p(t)}\left\|P[t]x^k - x^k\right\|^2 \leq \left\|z - x^k\right\|^2. \tag{2.8}$$

Let

$$p_* := \max_{t\in\Omega} p(t). \tag{2.9}$$

By (2.8) we deduce that, whenever $z \in C$ and $k \in \mathbb{N}$,

$$\left\|z - P[t]x^k\right\|^2 + 2^{-p_*}\left\|P[t]x^k - x^k\right\|^2 \leq \left\|z - x^k\right\|^2. \tag{2.10}$$

Multiplying both sides of this inequality by $w(t)$, summing up the resulting inequalities, and taking into account that, for each $u \in \mathbb{R}^J$, the function $x \to \|u-x\|^2$ is convex, we deduce that, for all $z \in C$ and $k \in \mathbb{N}$,

$$\left\|z-x^{k+1}\right\|^2 + 2^{-p_*}\left\|x^{k+1}-x^k\right\|^2 \leq \left\|z-x^k\right\|^2, \tag{2.11}$$

and, consequently,

$$\left\|z-x^{k+1}\right\| \leq \left\|z-x^k\right\|. \tag{2.12}$$

This implies that the sequence $\left\{\left\|z-x^k\right\|\right\}_{k\in\mathbb{N}}$ converges. By (2.11) we have that, for all $z \in C$ and $k \in \mathbb{N}$,

$$2^{-p_*}\left\|x^{k+1}-x^k\right\|^2 \leq \left\|z-x^k\right\|^2 - \left\|z-x^{k+1}\right\|^2, \tag{2.13}$$

and the right hand side converges to zero. Hence,

$$\lim_{k\to\infty}\left\|x^{k+1}-x^k\right\| = 0. \tag{2.14}$$

According to (2.14), the nonexpansive operator $\mathbf{P}$ is asymptotically regular and, clearly, any element of C is a fixed point of $\mathbf{P}$. Therefore, Theorem 2 in [56] applies to $\mathbf{P}$ and shows that $\{x^k\}_{k\in\mathbb{N}}$ converges to a fixed point of $\mathbf{P}$.

To complete the proof of Theorem 1, we prove:

Claim 2 *If $\{\beta_k\}_{k\in\mathbb{N}}$ and $\left\{v^k\right\}_{k\in\mathbb{N}}$ are sequences satisfying the hypotheses of Theorem 1, then any sequence generated according to* (2.5) *converges, and its limit is in C.*

Since the operator $\mathbf{P}$ is nonexpansive, for each $k \in \mathbb{N}$, we have that

$$\begin{aligned} \left\| x^{k+1} - \mathbf{P}x^k \right\| &= \left\| \mathbf{P}(x^k + \beta_k v^k) - \mathbf{P}x^k \right\| \\ &\leq \left\| (x^k + \beta_k v^k) - x^k \right\| = \beta_k \left\| v^k \right\|. \end{aligned} \tag{2.15}$$

Consequently, denoting by $\mathscr{M}$ a finite upper bound of the bounded sequence $\left\{ \left\| v^k \right\| \right\}_{k\in\mathbb{N}}$, we have that

$$\sum_{k=0}^{\infty} \left\| x^{k+1} - \mathbf{P}x^k \right\| \leq \mathscr{M} \sum_{k=0}^{\infty} \beta_k < \infty, \tag{2.16}$$

showing that the sequence $\left\{ \left\| x^{k+1} - \mathbf{P}x^k \right\| \right\}_{k\in\mathbb{N}}$ is summable. According to [9, Theorem 4.1] combined with the already proved Claim 1, the summability of this sequence ensures that the sequence $\{x^k\}_{k\in\mathbb{N}}$ converges and its limit is a fixed point of $\mathbf{P}$. Taking into account [4, Proposition 2.12(i)] and [4, Proposition 2.10(i)] we also have

$$\begin{aligned} C \subseteq \operatorname{Fix}\mathbf{P} &= \bigcap_{t\in\Omega} \operatorname{Fix} P[t] \\ &= \bigcap_{i=1}^{I} \bigcap_{t\in\Omega_i} \operatorname{Fix} P[t] \subseteq \bigcap_{i=1}^{I} C_i = C, \end{aligned} \tag{2.17}$$

and so $C = \operatorname{Fix}\mathbf{P}$ and, hence, the sequences generated by (2.5) converge to points in C. □

We refer to a procedure defined by (2.5) as a *perturbation-resilient amalgamated projection method*. This is only one perturbation model to which Theorem 1 apply, another is the following:

$$y^0 \in \mathbb{R}^n$$
$$y^{k+1} = \mathbf{P}y^k + \beta_k v^k,\ \forall k \in \mathbb{N}. \tag{2.18}$$

By setting $x^0 = \mathbf{P}y^0$, it is easily shown by induction that, for $k \geq 0$,

$$y^{k+1} = x^k + \beta_k v^k, \tag{2.19}$$

where the sequence x^k is generated by (2.5). From the convergence of that sequence to an element in C and from the conditions of the β_k and v^k, the convergence of the sequence of y^k generated by (2.18) to an element of C follows. These two perturbation models (i.e., (2.5) and (2.18)) play an important role when a perturbation-resilient AP method is applied to a real-world problem. For example, the model in (2.18) is better suited to the point of view in which perturbations are considered to be numerical errors due to the fact that inexact projections are computed. The model in (2.5), however, is more appropriate for using the perturbations in such a way as to steer the algorithmic process towards a minimizer of a given convex function. The stability under perturbations of the convergence of projection methods for solving convex feasibility problems was also considered in [4], where the behavior of such procedures is studied under the assumption that the sets C_i are given by approximations. By contrast, Theorem 1 deals with the situation in which the sets C_i are precisely given, but the projections P_{C_i} on them are determined only approximately. A result similar to Theorem 1 was obtained in [21] concerning the following: Fix $w \in \mathbb{R}_{++}$ such that $\sum_{i=1}^{I} w_i = 1$ and use the procedure

$$x^0 \in \mathbb{R}^J$$
$$x^{k+1} = \sum_{i=1}^{I} w_i(P_i x^k + \beta_k v^k),\ \forall k \in \mathbb{N} \tag{2.20}$$

Note that this is just that special case of (2.18) that we get from the choice of amalgamator that gave rise to the generalization of the Cimmino algorithm that we discussed earlier in Subsection 1.2.1.

2.3 Perturbation-resilient BIP methods

We consider two cases of perturbation-resilient BIP methods: one that holds for general closed convex sets and another that holds only for the special case when the sets are hyperplanes. The reason for paying attention to the latter, which is a subset of the first, comes from the need to speed up the computational process, especially when the perturbation-resilient algorithm aims at finding a solution to a real-world problem. Another reason is because we were able to extend our results in this case to systems that are not necessarily consistent. We start with a discussion of perturbation-resilient BIP methods that are good for the general case and later consider the case when the sets are hyperplanes as mentioned above.

2.3.1 PR-BIP for general closed convex sets

In this subsection we present a perturbation-resilient BIP algorithms for solving convex feasibility problems. Since our 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 subsection we use underlined 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

$$D=\bigcap_{z=1}^{Z}D_z \tag{2.21}$$

is nonempty, find a point $\underline{x}$ 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

$$x=\begin{pmatrix} x_1\\ \vdots\\ x_L\\ \vdots\\ x_{(R-1)L+1}\\ \vdots\\ x_{RL}\end{pmatrix}\in\mathbb{R}^J, \tag{2.22}$$

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

$$\begin{pmatrix} x_{(r-1)L+1}\\ \vdots\\ x_{rL}\end{pmatrix}\in S_r. \tag{2.23}$$

Under these circumstances, we use the notations

$$\underline{x}_r=\begin{pmatrix} x_{(r-1)L+1}\\ \vdots\\ x_{rL}\end{pmatrix}\text{ and }x=\begin{pmatrix}\underline{x}_1\\ \vdots\\ \underline{x}_R\end{pmatrix}, \tag{2.24}$$

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

$$\|x-y\|^2=\sum_{r=1}^{R}\left\|\underline{x}_r-\underline{y}_r\right\|^2. \tag{2.25}$$

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 [19, Lemma 5.9.2],

$$P_{\mathbb{P}S} \begin{pmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_R \end{pmatrix} = \begin{pmatrix} P_{S_1}\underline{x}_1 \\ \vdots \\ P_{S_R}\underline{x}_R \end{pmatrix}. \tag{2.26}$$

One more piece of terminology that we need is the following. The *canonical mapping* $\delta : \mathbb{R}^L \to \mathbb{R}^J$ is defined by: For any $\underline{x} \in \mathbb{R}^L$, $\delta(\underline{x}) = x$, with $\underline{x}_1 = \cdots = \underline{x}_R = \underline{x}$ (here we made use of the notation of (2.24)).

We now return to the convex feasibility problem (2.21) that is stated at the beginning of this subsection. For $1 \le u \le U$, let B_u be an ordered set $\left(b_{u,1}, \ldots, b_{u,\ell_u}\right)$ of elements of $\{1, \ldots, Z\}$ (ℓ_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

$$\mathbf{Q} = Q_U \cdots Q_1, \tag{2.27}$$

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

$$Q_u\underline{x} = \frac{1}{R}\sum_{z \in B_u} P_{D_z}\underline{x} + \frac{R - \ell_u}{R}\underline{x}, \tag{2.28}$$

and

$$R = \max\{\ell_u \mid 1 \le u \le U\}. \tag{2.29}$$

An iterative procedure based on $\underline{x}^{k+1} = \mathbf{Q}\underline{x}^k$ is a member of the family of block-iterative projection (BIP) methods discussed in Subsection 1.2.2.

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 $0 < \lambda \le 1$, $\{\gamma_k\}_{k \in \mathbb{N}}$ be a sequence of nonnegative real numbers such that*

$\sum_{k=0}^{\infty} \gamma_k < \infty$, $\left\{\underline{w}^k\right\}_{k\in\mathbb{N}}$ *be a bounded sequence of points in* $\mathbb{R}^L$, *and* $\underline{x}^0 \in \mathbb{R}^L$. *If* $\{1,\ldots,Z\} = \bigcup_{u=1}^{U} B_u$, *then the sequence* $\underline{x}^0, \underline{x}^1, \underline{x}^2, \ldots$ *generated by* the iterative *procedure*

$$\underline{x}^{k+1} = \lambda \mathbf{Q}\left(\underline{x}^k + \gamma_k \underline{w}^k\right) + (1-\lambda)\left(\underline{x}^k + \gamma_k \underline{w}^k\right),\ \forall k \in \mathbb{N}, \tag{2.30}$$

converges, and its limit is in D.

Proof Our proof utilizes Theorem 1 by showing that the procedure in (2.30) is a perturbation-resilient AP method in a product space. (Pierra [58] was the first to use convergence results of sequential algorithms in a product space to prove the convergence of simultaneous algorithms, see also [18] and [19, 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 AP method. For $1 \le u \le U$,

$$C_u = \mathbb{P}\left(D_{b_{u,1}}, \cdots, D_{b_{u,\ell_u}}, \mathbb{R}^L, \cdots, \mathbb{R}^L\right), \tag{2.31}$$

with $R - \ell_u$ copies of $\mathbb{R}^L$ at the end of (2.31). Further,

$$C_{U+1} = \left\{\delta\left(\underline{x}\right) \mid \underline{x} \in \mathbb{R}^L\right\}, \tag{2.32}$$

$$C_{U+2} = \mathbb{R}^J. \tag{2.33}$$

We now specify P_{C_u}, for $1 \leq u \leq U+2$, by making use of (2.26). For $1 \leq u \leq U$,

$$P_{C_u}\begin{pmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_{\ell_u} \\ \underline{x}_{\ell_u+1} \\ \vdots \\ \underline{x}_R \end{pmatrix} = \begin{pmatrix} P_{D_{b_{u,1}}}\underline{x}_1 \\ \vdots \\ P_{D_{b_{u,\ell_u}}}\underline{x}_{\ell_u} \\ \underline{x}_{\ell_u+1} \\ \vdots \\ \underline{x}_R \end{pmatrix}. \tag{2.34}$$

The set C_{U+1} is called the *diagonal convex* [58] or the *diagonal subset* [18] and

$$P_{C_{U+1}}x = \delta\left(\frac{1}{R}\sum_{r=1}^{R}\underline{x}_r\right) \tag{2.35}$$

(for a proof see [58, Lemma 1.1]). Clearly,

$$P_{C_{U+2}}x = x. \tag{2.36}$$

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 $\underline{x} \in D$ then $\delta(\underline{x}) \in C$. 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

$$\Omega = \{(1, U+1, \ldots, U, U+1), (U+2)\}. \tag{2.37}$$

(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 $(t_1,\ldots,t_p) \in \Omega$ such that $t_s = i$ for some $s \in \{1,\ldots,p\}$. 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 $v^k = \delta\left(\underline{w}^k\right)$, for all $k \in \mathbb{N}$, and $x^0 = \delta\left(\underline{x}^0\right)$.

Claim: Given the specifications of the previous paragraph, the sequence $x^0, x^1, x^2, \ldots$ produced by (2.5) of Theorem 1 has the property, that for $k \in \mathbb{N}$, $x^k = \delta\left(\underline{x}^k\right)$, where $\underline{x}^0, \underline{x}^1, \underline{x}^2, \ldots$ is the sequence produced by (2.30) of Theorem 2.

If this claim is true, then Theorem 2 follows as we now show. By Theorem 1, $x^0, x^1, x^2, \ldots$ converges to an $x^* \in C$. Since $x^* \in C_{U+1}$, it is of the form $x^* = \delta(\underline{x}^*)$, for some $\underline{x}^* \in \mathbb{R}^L$. By the Claim, $x^k = \delta\left(\underline{x}^k\right)$, for $k \in \mathbb{N}$. By (2.25),

$$\left\|\underline{x}^k - \underline{x}^*\right\|^2 \leq \left\|x^k - x^*\right\|^2, \tag{2.38}$$

and since the right-hand side converges to zero, so must the left-hand side, and hence $\underline{x}^0, \underline{x}^1, \underline{x}^2, \ldots$ converges to $\underline{x}^*$. Since $x^* = \delta(\underline{x}^*) \in C_{u,}$, for $1 \leq u \leq U$, we have, by (2.31), that $\underline{x}^* \in D_{b_{u,i}}$ for $1 \leq u \leq U$ and $1 \leq i \leq \ell_u$. Since $\{1,\ldots,Z\} = \bigcup_{u=1}^{U} B_u$ and $D = \bigcap_{z=1}^{Z} D_z$, it must be so that $\underline{x}^* \in D$.

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

Using (1.2), (1.3), (2.5), (2.36) and (2.37) with $0 < \lambda < 1$, we obtain

$$\begin{aligned} x^{k+1} &= \lambda P_{C_{U+1}} P_{C_U} \cdots P_{C_{U+1}} P_{C_1}\left(x^k + \beta_k v^k\right) \\ &\quad + (1-\lambda)\left(x^k + \beta_k v^k\right). \end{aligned} \tag{2.39}$$

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

obtain

$$\begin{aligned} x^{k+1} &= \lambda P_{C_{U+1}} P_{C_U} \cdots P_{C_{U+1}} P_{C_1} \left(\delta\left(\underline{x}^k\right) + \gamma_k \delta\left(\underline{w}^k\right)\right) \\ &\quad + (1-\lambda)\left(\delta\left(\underline{x}^k\right) + \gamma_k \delta\left(\underline{w}^k\right)\right). \end{aligned} \tag{2.40}$$

By (2.27) and (2.30) and the obvious linearity of δ we obtain

$$\begin{aligned} \delta\left(\underline{x}^{k+1}\right) &= \lambda \delta\left(Q_U \cdots Q_1\left(\underline{x}^k + \gamma_k \underline{w}^k\right)\right) \\ &\quad + (1-\lambda)\left(\delta\left(\underline{x}^k\right) + \gamma_k \delta\left(\underline{w}^k\right)\right). \end{aligned} \tag{2.41}$$

Since the second terms of (2.40) and (2.41) are identical, the proof that $x^{k+1} = \delta\left(\underline{x}^{k+1}\right)$ is complete if we can show that

$$P_{C_{U+1}} P_{C_U} \cdots P_{C_{U+1}} P_{C_1} \delta\left(\underline{x}^k + \gamma_k \underline{w}^k\right) = \delta\left(Q_U \cdots Q_1\left(\underline{x}^k + \gamma_k \underline{w}^k\right)\right). \tag{2.42}$$

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

$$P_{C_{U+1}} P_{C_u} \cdots P_{C_{U+1}} P_{C_1} \delta\left(\underline{x}^k + \gamma_k \underline{w}^k\right) = \delta\left(Q_u \cdots Q_1\left(\underline{x}^k + \gamma_k \underline{w}^k\right)\right), \tag{2.43}$$

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 $\underline{y}$ abbreviate $Q_u \cdots Q_1\left(\underline{x}^k + \gamma_k \underline{w}^k\right)$. By the induction hypothesis, (2.43), (2.34), (2.35) and (2.28),

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

$$=P_{C_{U+1}}\begin{pmatrix} P_{D_{b_{u+1,1}}}\underline{y} \\ \vdots \\ P_{D_{b_{u+1,\ell_{u+1}}}}\underline{y} \\ \underline{y} \\ \vdots \\ \underline{y} \end{pmatrix}=\delta\left(\frac{1}{R}\sum_{z\in B_{u+1}}P_{D_z}\underline{y}+\frac{R-\ell_{u+1}}{R}\underline{y}\right) \tag{2.44}$$

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

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

$$(1,U+1,\ldots,U,U+1), \tag{2.45}$$

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

2.3.2 PR-BIP for hyperplane convex sets

In this section we propose two accelerated perturbation-resilient BIP algorithms for solving a linear system of equations (that may or may not be consistent) of the form

$$Ax=b, \tag{2.46}$$

where $A \in \mathbb{R}^{I\times J}$, $x \in \mathbb{R}^J$ and $b \in \mathbb{R}^I$. Note that solving (2.46) is a solution to the convex feasibility problem (1.1) for the special case when the C_i are hyperplanes. In this case we partition A and b of the system into blocks of equations that can be written as

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_U \end{pmatrix}, \qquad b = \begin{pmatrix} b^1 \\ b^2 \\ \vdots \\ b^U \end{pmatrix}, \tag{2.47}$$

and then use exactly one block in any atomic step of the algorithm. We state theorems that specify their convergence behaviors early in the section, followed by proofs of these theorems.

Consider blocks of equations as in (2.47). For $u = 1,2,\ldots,U$, we use ℓ_u denote the dimension of the vector b^u. To specify our algorithms, we need to choose positive definite symmetric $\ell_u \times \ell_u$ matrices M_u and real numbers λ_u, such that

$$0 < \varepsilon \leq \lambda_u \leq \frac{2-\varepsilon}{\rho(A_u^T M_u A_u)}, \tag{2.48}$$

for some real number ε, where for any square matrix W, $\rho(W)$ denotes the spectral radius of W. With these choices made, we define the operators $H_u : \mathbb{R}^J \to \mathbb{R}^J$ by

$$H_u x = (\mathbf{I} - \lambda_u A_u^T M_u A_u)x + \lambda_u A_u^T M_u b^u, \tag{2.49}$$

where $\mathbf{I}$ is the identity matrix, and the operator

$$\mathbf{H} = H_U \cdots H_2 H_1. \tag{2.50}$$

Theorem 3 Let $\{\beta_k\}_{k\in\mathbb{N}}$ be a sequence of nonnegative real numbers such that

$\sum_{k=0}^{\infty}\beta_k < \infty$, let $\{v^k\}$ be a bounded sequence of vectors in $\mathbb{R}^J$ and $x^0 \in \mathbb{R}^J$. The sequence defined by

$$x^{k+1} = \mathbf{H}(x^k + \beta_k v^k), \quad \text{for all } k \in \mathbb{N}, \tag{2.51}$$

converges to a fixed point of $\mathbf{H}$. Furthermore, if the system (2.46) of equations is consistent, then the fixed points of $\mathbf{H}$ are the solutions of (2.46).

The first part of this theorem states that the BIP algorithm with perturbations that is specified by (2.51) converges, whether or not the system of equations is consistent. The last sentence implies that if the system is consistent, then the algorithm converges to a solution of it.

To get something similarly satisfactory in the inconsistent case, we need an alternative algorithm: Given the system (2.46) of equations and a symmetric positive definite $I \times I$ matrix M, we call an $x \in \mathbb{R}^J$ an *M-weighted least-squares solution* if, for any $y \in \mathbb{R}^J$,

$$(Ax-b)^T M (Ax-b) \leq (Ay-b)^T M (Ay-b). \tag{2.52}$$

We define (compare with (2.50))

$$\mathbf{S} = H_1 H_2 \cdots H_{U-1} H_U H_U H_{U-1} \cdots H_2 H_1. \tag{2.53}$$

Theorem 4 Let $\{\beta_k\}_{k\in\mathbb{N}}$ be a sequence of nonnegative real numbers such that $\sum_{k=0}^{\infty}\beta_k < \infty$, let $\{v^k\}$ be a bounded sequence of vectors in $\mathbb{R}^J$ and $x^0 \in \mathbb{R}^J$. Then the sequence defined by

$$x^{k+1} = \mathbf{S}(x^k + \beta_k v^k), \quad \text{for all } k \in \mathbb{N}, \tag{2.54}$$

converges to a fixed point of **S**. Furthermore, the fixed points of **S** are the M-weighted least-squares solutions of (2.46), for an appropriately defined symmetric positive definite $I \times I$ matrix M.

The following two Lemmas are used in the proofs of Theorems 3 and 4.

Lemma 1 The operators H_u (for $1 \leq u \leq U$), **H** and **S** are nonexpansive, provided that (2.48) holds.

Proof Clearly, it is sufficient to show that the operators H_u, $1 \leq u \leq U$, are nonexpansive. For any such u, $\mathscr{Q}_u = \mathbf{I} - \lambda_u A_u^T M_u A_u$ is a symmetric matrix. From this and (2.49) it follows that, for $x, y \in \mathbb{R}^J$,

$$\begin{aligned} \|H_u x - H_u y\| &= \|\mathscr{Q}_u (x-y)\| \\ &\leq \|\mathscr{Q}_u\| \|x-y\| = \rho(\mathscr{Q}_u) \|x-y\|. \end{aligned} \tag{2.55}$$

This implies that our proof is complete, provided that we can show that $-1 \leq \mu(\mathscr{Q}_u) \leq 1$, for all eigenvalues $\mu(\mathscr{Q}_u)$ of $\mathscr{Q}_u$.

Let $\mu(\mathscr{Q}_u)$ be such an eigenvalue. Then $1 - \mu(\mathscr{Q}_u)$ is an eigenvalue of the nonnegative-definite matrix $\lambda_u A_u^T M_u A_u$. From this and (2.48) it follows that

$$0 \leq 1 - \mu(\mathscr{Q}_u) \leq \lambda_u \rho(A_u^T M_u A_u) \leq 2 - \varepsilon, \tag{2.56}$$

which yields immediately that $-1 < \mu(\mathscr{Q}_u) \leq 1$. ■

Lemma 2 The operators H_u (for $1 \leq u \leq U$) are attracting with respect to $Fix(H_u)$, provided that (2.48) holds.

Proof For any u, $1 \leq u \leq U$, let $x \in \mathbb{R}^J \backslash Fix(H_u)$ and $y \in Fix(H_u)$. We first note that

$$s = A_u (x - y), \tag{2.57}$$

is not the zero vector, for otherwise we would have $A_u x = A_u y$, which together with

(2.49) would imply that $x \in Fix(H_u)$.

We need to show that

$$\|H_u x - y\| < \|x - y\|. \tag{2.58}$$

Since $y = H_u y$, it is sufficient to show that

$$\|H_u x - H_u y\|^2 < \|x - y\|^2. \tag{2.59}$$

Using (2.49) and (2.57) we have that

$$\begin{aligned} ||H_u x - H_u y||^2 &= \|x - y\|^2 - 2\lambda_u \langle s, M_u s\rangle \\ &\quad + \lambda_u^2 \left\langle A_u A_u^T M_u s, M_u s\right\rangle. \end{aligned} \tag{2.60}$$

Since M_u is positive definite it can be written as $W^T W$ for some symmetric positive definite matrix W, and we get

$$\begin{aligned} \left\langle A_u A_u^T M_u s, M_u s\right\rangle &= \left\langle W A_u A_u^T W^T W s, W s\right\rangle \\ &\leq \rho\left(W A_u A_u^T W^T\right) \langle W s, W s\rangle \\ &= \rho\left(A_u^T M_u A_u\right) \langle M_u s, s\rangle. \end{aligned} \tag{2.61}$$

The inequality is true since $W A_u A_u^T W^T$ is a symmetric matrix. It therefore follows

that

$$\begin{aligned}\|H_u x - H_u y\|^2 &\leq \|x-y\|^2 - 2\lambda_u \langle s, M_u s\rangle \\ &\quad +\lambda_u^2 \rho\left(A_u^T M_u A_u\right)\langle M_u s, s\rangle \\ &= \|x-y\|^2 \\ &\quad -\lambda_u\left(2-\lambda_u\rho\left(A_u^T M_u A_u\right)\right)\langle M_u s_u, s_u\rangle \\ &< \|x-y\|^2. \end{aligned} \tag{2.62}$$

The last inequality holds because, in view of (2.48), $\lambda_u\left(2-\lambda_u\rho\left(A_u^T M_u A_u\right)\right)\langle M_u s_u, s_u\rangle$ is positive. ■

Proof (*Theorem 3*) We first prove that when $\beta_k = 0$, for all $k \in \mathbb{N}$, then the sequence generated by (2.51) converges to a fixed point of the operator $\mathbf{H}$. By Theorem 5 and Proposition 6 of [30], (2.48) guarantees the existence of a limit point of the sequence. Since $\mathbf{H}$ is nonexpansive (Lemma 1), the limit point is a fixed point of $\mathbf{H}$.

To show this result in the general case when $\beta_k \geq 0$, for all $k \in \mathbb{N}$, we consider any sequences of $\{\beta_k\}_{k\in\mathbb{N}}$ and $\left\{v^k\right\}_{k\in\mathbb{N}}$ that satisfy the conditions stated in Theorem 3. Using Lemma 1, we have that, for each $k \in \mathbb{N}$,

$$\begin{aligned}\|x^{k+1} - \mathbf{H}x^k\| &= \|\mathbf{H}(x^k + \beta_k v^k) - \mathbf{H}x^k\| \\ &\leq \|(x^k + \beta_k v^k) - x^k\| = \beta_k\|v^k\|. \end{aligned} \tag{2.63}$$

Consequently, denoting by $\mathscr{M}$ a finite upper bound for the bounded sequence $\{\|v_k\|\}_{k\in\mathbb{N}}$, we have the summability result

$$\sum_{k=0}^{\infty}\|x^{k+1} - \mathbf{H}x^k\| \leq \mathscr{M}\sum_{k=0}^{\infty}\beta_k < \infty. \tag{2.64}$$

According to [9, Theorem 4.1] combined with Lemma 1 and the first part of this

proof, the sequence $\{x^k\}_{k\in\mathbb{N}}$ converges to a fixed point of the operator $\mathbf{H}$.

To complete the proof of Theorem 3, we need to show that in the consistent case the set V of solutions of (2.46) equals $Fix(\mathbf{H})$. If $x \in V$, then, for $u = 1,2,\ldots,U$, $A_u x = b^u$ and hence by (2.49) $H_u x = x$. By (2.50) this implies that $x \in Fix(\mathbf{H})$. Now consider $x \in Fix(\mathbf{H})$. Combining Lemma 2 with [4, Proposition 2.10(i)], we get that $x \in Fix(H_u),$ for $u = 1,2,\ldots,U$. Using (2.49) and the positivity of λ_u in (2.48) it follows that, for $u = 1,2,\ldots,U$,

$$A_u^T M_u (b^u - A_u x) = 0. \tag{2.65}$$

Since (2.46) is assumed to be consistent, there exist a y such that $b^u = A_u y$. For this y (2.65) yields

$$\begin{aligned} 0 &= A_u^T M_u (A_u y - A_u x), \\ &= A_u^T M_u A_u (y - x), \\ &= (M_u^{1/2} A_u)^T (M_u^{1/2} A_u)(y - x). \end{aligned} \tag{2.66}$$

It follows that $\|(M_u^{1/2} A_u)(y-x)\| = 0$ and therefore $A_u x = b^u$. This being true for $u = 1,2,\ldots,U$, implies that $x \in V$. ■

Proof (*Theorem* 4) The proof of the first part of Theorem 4, that the sequence (2.54) converges to a fixed point of $\mathbf{S}$, follows almost verbatim from the equivalent part of the proof of Theorem 3 (here we use Propositions 10 and 11 of [30] instead of Theorem 5 and Proposition 6 from that paper), since $\mathbf{S}$ (like $\mathbf{H}$) is nonexpansive (Lemma 1).

By [30, Proposition 10] the operator $\mathbf{S}$ can be written as

$$\mathbf{S}x = x + A^T \overline{M}_{SB} (b - Ax), \tag{2.67}$$

where $\overline{M}_{SB}$ is a symmetric positive definite $m \times m$ matrix. To see that the fixed points of $\mathbf{S}$ are the $\overline{M}_{SB}$-weighted least-squares solutions of (2.46), we observe that z is an $\overline{M}_{SB}$-weighted least squares solution of (2.46) if, and only if,

$$A^T \overline{M}_{SB}(b - Az) = 0, \tag{2.68}$$

which combined with (2.67), yields the desired result. ■

Chapter 3

The superiorization methodology

The theorems we presented in Chapter 2 guarantee the convergence of the AP methods and BIP methods when the calculation of the iterates is affected by perturbations that are summable and bounded. We can make use of this property to steer the iterates towards the minimizer of a given convex function ϕ; i.e., to the $x \in \mathbb{R}^J$ that provides the solution of the problem

$$\text{minimize } \phi(x), \text{ subject to } x \in C. \tag{3.1}$$

We next discuss the problem (3.1) with respect to all Theorems 1 to 4 presented in the previous chapter. Although Theorem 2 was given for vectors in $\mathbb{R}^L$, it can be assumed and stated for vectors in $\mathbb{R}^J$; this is what we will be doing here. The heuristic provided below is not guaranteed to achieve actual convergence to the minimizer x. However, as demonstrated by examples in Section 3.3, 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 term superiorization was first introduced in [26], see also [8, 13, 36, 55]. Superiorization provides us

with a new algorithm, based on an original one, by steering the iterates (using the perturbations) towards a solution that is superior, according to some criterion, to the one to which we would get without perturbations. We first give the details of a superiorization algorithm and later demonstrate the usefulness of it on problems from tomography, i.e., reconstructing objects using projection data.

3.1 Heuristic approach to superiorization

Consider a convex function $\phi : \mathbb{R}^J \to \mathbb{R}$ which has a minimizer over the set C. For any $k \in \mathbb{N}$, let $s^k \in \partial\phi(x^k)$, be a subgradient of ϕ at x^k, and define

$$v^k = \begin{cases} -\frac{s^k}{\|s^k\|}, & \text{if } s^k \neq 0, \\ \\ 0, & \text{if } s^k = 0. \end{cases} \tag{3.2}$$

Clearly, the sequence $\left\{v^k\right\}_{k\in\mathbb{N}}$ defined by (3.2) is bounded. Therefore, by Theorems 1 to 4 , for any summable sequence of nonnegative real numbers $\{\beta_k\}_{k\in\mathbb{N}}$, the sequence $\left\{x^k\right\}_{k\in\mathbb{N}}$ generated according to (2.5), (2.30), (2.51) and (2.54) converge to an element of C.

In our implementation, we use the following methodology for generating the real numbers $\{\beta_k\}_{k\in\mathbb{N}}$. We define, for an $x \in \mathbb{R}^J$,

$$\text{Res}(x) = \sqrt{\sum_{i=1}^{I} [d(x, C_i)]^2}. \tag{3.3}$$

Clearly, $x \in C$ if, and only if, $\text{Res}(x) = 0$. Furthermore, if $\text{Res}(x) > 0$, then its size indicates how badly x violates the given collection $\{C_1, \ldots, C_I\}$ of constraints. An approximate solution x to the convex optimization problem (for ϕ) under these constraints should have a small value of $\text{Res}(x)$ and should aim at finding, among all

x with similar (or smaller) value of $\mathrm{Res}(x)$, an x for which $\phi(x)$ is small relative to the others. Guided by this principle, we generate $\{\beta_k\}_{k\in\mathbb{N}}$ as follows. We initialize β to be an arbitrary positive number, which we denote by β_{-1}. (In the experiments provided in Section 3.3 we have used $\beta_{-1}=1$, unless otherwise stated.) In the process of the iterative step from x^k to x^{k+1}, we also update the value of β, which is β_{k-1} at the beginning of the iterative step and β_k at its end. This updating is done according to the following pseudocode (in which v^k is defined by (3.2)). We terminate the iterative process when we find an x^k such that $\mathrm{Res}(x^k)<\varepsilon$, where ε is a user-specified small positive number.

```
1: logic = true
2: while (logic)
3:         z = x^k + β v^k
4:         if ( φ(z) ≤ φ(x^k) )
5:             then
6:                 x^{k+1} = Oz
7:                 if ( Res(x^{k+1}) < Res(x^k) )
8:                     then logic = false
9:                     else β = β/2
10:            else β = β/2
```

The operator **O** in Step 6 of the pseudocode stands for **P** of (1.3) for the algorithm of Theorem 1, for **Q** of (2.27) for the algorithm of Theorem 2, for **H** of (2.50) for the algorithm of Theorem 3 and for **S** of (2.53) for the algorithm of Theorem 4. The complete superiorization algorithm consists of the procedure provided by the theorems (i.e., (2.5), (2.30), (2.51) or (2.54), respectively) with the v^k defined by (3.2) and the β_k defined by the pseudocode that makes use of (3.3). What the 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 an element of the

set C, as guaranteed by the theorems for a proper choice of the sequence $\{\beta_k\}_{k\in\mathbb{N}}$ (see Step 7 of the pseudocode). In the next sections we give several examples of the superiorization approach illustrating the use of it for an image processing application.

3.2 Computerized tomography (CT) simulations

We demonstrate the usefulness of the superiorization methodology by applying it to tomographic problems of reconstructing an object from collected projection data. We first provide the 2D object (a phantom image) which we consider for our examples and discuss its nature and different characteristics. We will then give the different modes of data collections for the simulated CT cases and include a description of an evaluation methodology that we will be using in our experiments in the next section.

3.2.1 Images and digitization

In an application of image reconstruction from projections, the *image* is typically represented by a function f of two variables of bounded support. The values of this function are elements of the set of real numbers $\mathbb{R}$ and they represent some physical property (e.g., in CT these are the linear X-ray attenuation coefficients) in a cross-section of the object to be reconstructed. The projections are usually taken with the help of some rays (e.g., X-rays) and can be thought of mathematically as collections of line integrals of the function. The mathematical problem is to reconstruct the function from its (noisy and incomplete) projections [35, 54].

We define the projection in direction $\vartheta \in [0,\pi)$ as follows. Let (s_1,s_2) denote the coordinates of the point $r=(r_1,r_2)\in\mathbb{R}^2$ in the coordinate system rotated by ϑ. Then the *projection of f in the direction ϑ* (the *ϑ-projection* of f) is defined as that

function $[\mathscr{R}f](\bullet,\vartheta)$ of the variable s_1 for which

$$[\mathscr{R}f](s_1,\vartheta) \;=\; \int_{L_{s_1,\vartheta}} f(r)\,ds_2, \tag{3.4}$$

where $L_{s,\vartheta}$ is the line at the distance s from the origin that makes the angle ϑ with the r_2-axis. It can be said that the transform $\mathscr{R}$ defined by (3.4) gives the ϑ-projections of f for any $\vartheta \in [0,\pi)$. The transform $\mathscr{R}$ is called the *Radon transform* of f, after J. Radon who studied this kind of transform in [61].

Let us suppose for now that we have taken projections of f for directions ϑ in the finite set Θ. Let, for $s \in \mathbb{R}$ and $\vartheta \in \Theta$, $g(s,\vartheta)$ denote the approximation to $[\mathscr{R}f](s,\vartheta)$ that we obtain based on our measurements. For any $\vartheta \in \Theta$, we use S_ϑ to denote the set of all s for which we have a projection data item $g(s,\vartheta)$. In practice, the sets S_ϑ have to be finite, but here we also deal with the mathematical idealization in which, for all $\vartheta \in \Theta$, S_ϑ is the set of all real numbers. (The point that we will make is that even such overabundance of data, as compared to what can be obtained in practice, is not in general sufficient for determining f uniquely.) Then we consider the following *reconstruction task*: Suppose f is an unknown image and we are given $g(s,\vartheta)$ for $\vartheta \in \Theta$ and $s \in S_\vartheta$, such that

$$[\mathscr{R}f](s,\vartheta) \;\approx\; g(s,\vartheta), \text{ for all } \vartheta \in \Theta,\, s \in S_\vartheta, \tag{3.5}$$

(where $\approx$ stands for approximately equal), we need to find an image f^* that is a "good" approximation of f.

In discussing our concepts it is essential to have the notion of an $N \times N$ *digital image* p, which is defined as a function from $[0,N-1]^2$ into the real numbers, for a positive integer N. As it is customary in this context, elements of $[0,N-1]^2$ are denoted by row vectors (t_1,t_2) and we consider that (t_1+1,t_2) is "below" and (t_1,t_2+1) is "to the right of" (t_1,t_2). This can be made mathematically precise by

the introduction of a positive real number d, referred to as the *sampling interval*. Given a positive integer N and such a d, we associate with each $(t_1,t_2) \in [0,N-1]^2$ a subset of the plane $\mathbb{R}^2$, which is called the *pixel associated with* (t_1,t_2), defined as

$$\begin{aligned} pix^{N,d}(t_1,t_2) = \quad & \left\{ (r_1,r_2) \in \mathbb{R}^2 \;\middle|\; -\tfrac{(N-2t_2)d}{2} < r_1 \leq -\tfrac{(N-2-2t_2)d}{2} \right. \\ & \left. \text{and} \quad \tfrac{(N-2-2t_1)d}{2} \leq r_2 < \tfrac{(N-2t_1)d}{2} \right\}. \end{aligned} \tag{3.6}$$

Given an image f, a positive integer N and a sampling interval d, we define the $N \times N$ digital image $p_f^{N,d}$ by

$$p_f^{N,d}(t_1,t_2) = \frac{1}{d^2} \int\limits_{pix^{N,d}(t_1,t_2)} f(r)dr, \tag{3.7}$$

for any $(t_1,t_2) \in [0,N-1]^2$. So, the $N \times N$ *digitization* $p_f^{N,d}$ with sampling interval d of f is provided by the averages of f over the pixels. An $N \times N$ digital image p and a $d > 0$ gives rise to an image f_p^d that is defined by

$$f_p^d(r) = \begin{cases} p(t_1,t_2), & \text{if } r \in pix^{N,d}(t_1,t_2) \text{ for some} \\ & (t_1,t_2) \in [0,N-1]^2, \\ 0, & \text{otherwise .} \end{cases} \tag{3.8}$$

For any $N \times N$ digital image p and any sampling interval d, $p_{\left[f_p^d\right]}^{N,d} = p$. However, it is generally not the case that, for an image f, positive integer N, and sampling interval d, $f^d_{\left[p_f^{N,d}\right]} = f$, even if the N and the d are chosen large enough so that $f(r) = 0$ whenever $\max\{|r_1|,|r_2|\} \geq Nd/2$. However, for a "reasonable" image f, there should be an N and a d, such that $f^d_{\left[p_f^{N,d}\right]} \approx f$.

A common approach to solving the reconstruction task is to assume that the image to be reconstructed is f_p^d for some $N \times N$ digital image p and sampling interval d. We will refer to this as the *digital assumption*. The reason why this is helpful is

the following. Let us use an alternative representation of the digital image p as an J-dimensional vector (i.e., an element of $\mathbb{R}^J$) x_p, where $J = N^2$ and, for $1 \leq j \leq J$, the jth component of x_p is $p(t_1,t_2)$, where $(t_1,t_2) \in [0,N-1]^2$ and $j = t_1 N + t_2 + 1$. Note that such a t_1 and t_2 are uniquely determined by j and so we may denote them by t_1^j and t_2^j, respectively. Using this notation, it is easy to see that in such a case we have that, for any $\vartheta \in \Theta$ and any $s \in S_\vartheta$,

$$[\mathscr{R} f_p^d](s,\vartheta) = \langle a, x \rangle , \tag{3.9}$$

(as usual, $\langle a,x \rangle$ denotes $\sum_{j=1}^J a_j x_j$), where a is the J-dimensional vector whose jth component is the length of the segment of the line $L_{s,\vartheta}$ that lies in the pixel $pix^{N,d}(t_1^j,t_2^j)$ (in other words, it is the length of the intersection of the line with the jth pixel). In this fashion, each (approximate) equality in (3.5) gets replaced by an (approximate) linear equation in the unknown vector x. Let x^* denote a "solution" of this system of (approximate) linear equations and let p^* denote the (unique) $N \times N$ digital image such that $x_{p^*} = x^*$. Then one may consider $f^* = f_{p^*}^d$ to be a potential solution of the original problem. The important point here is that we obtain such a solution by solving a system of (approximate) linear equalities, and there is an extremely well-established field of numerical mathematics for solving such systems (see, e.g., [67]).

There is, however, a problem with such an approach. Even in the idealized case when there is no noise in the data (i.e., we have equalities, rather than approximate equalities, in (3.5)), the methodology can lead to a very inaccurate reconstruction due to the digital assumption. That is, the digital assumption can be a source of error: even though this methodology may lead to a unique reconstruction from perfect (noiseless) data, the result is not identical to either the image for which the data have been collected or to its digitization. The error can however be reduced

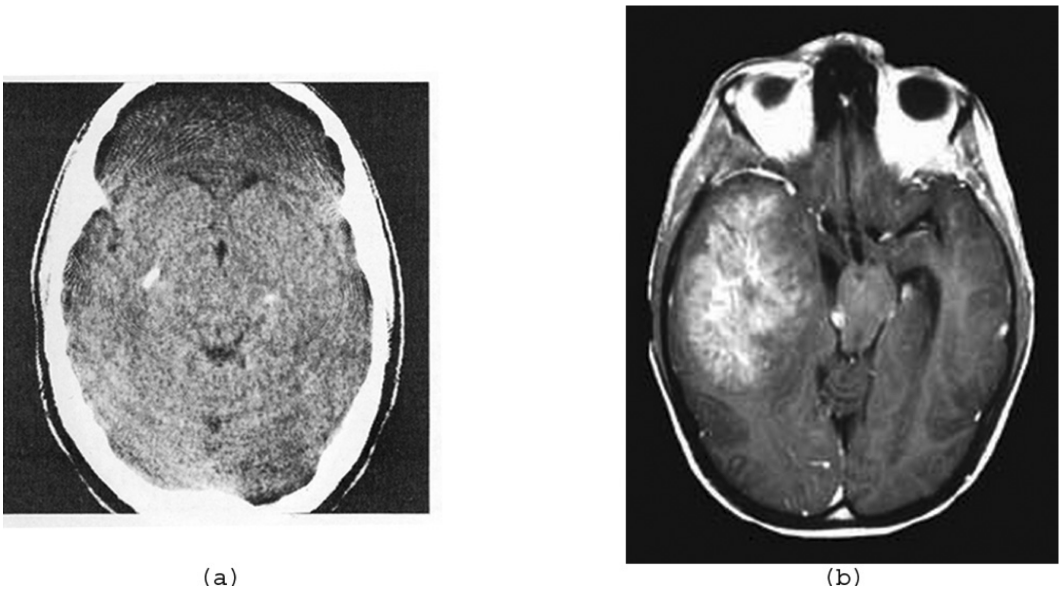


Figure 3.1: Two actual head cross-sections. (a) Figure 4.2 of [35]. (b) Figure 1(b) from [36] is taken from the Roswell Park Cancer Institute website.

by finer sampling and more data, but since an image may not correspond exactly to any digital image, we cannot expect a perfect reconstruction no matter how finely we digitize and how much data we use. To restate this in other words: just because a method can be shown to solve accurately the system of algebraic equations (3.9), one must not conclude that it will also perform well (in the sense of providing a good approximation of f) if the left hand sides of (3.9) were replaced by the $g(s,\vartheta)$ of the integral equations (3.5). To draw such a conclusion would be an *inverse crime*, which is defined in [39] as "the procedure of first simplifying the model, developing an estimator based on this model and then testing it against data produced with the same simplified model."

3.2.2 Phantoms and ghosts

In all the experiments that will be given in this document, we will be using as the basis what we call a *head phantom*. Figure 3.1 shows two actual head cross-sections. Using the software SNARK09 [27] we created a digitized head phantom

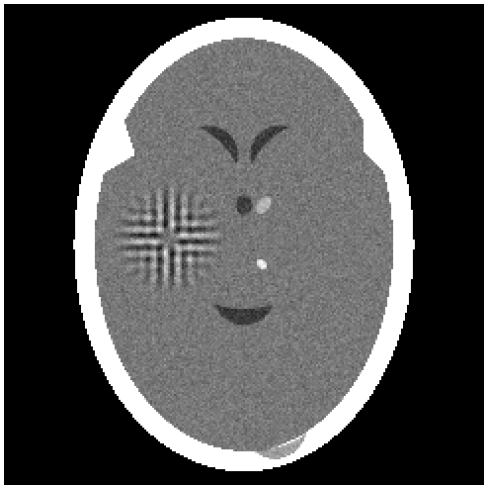


Figure 3.2: A head phantom used for all our experiments.

as given in Figure 3.2. The creation of the head phantom involves specifying a number of elemental objects which are superimposed, each has the freedom to be oriented and placed independently, with shapes and sizes that can be distinct from one another, and with different density value assignment to each. The total density at any given point is the sum of the superimposed densities (for each elemental object) of the different objects at that point. The head phantom in Figure 3.2 was created using 15 basic geometrical shapes (composed from ellipses, segments of circles and triangles), two of the ellipses correspond to the two bright tumors on the right-hand side of the phantom (positioned one above the other). To simulate the various anatomical structures in the brain, (such as gray matter, white matter, blood vessels, capillaries, etc.) we added *inhomogeneity*, that is, to each pixel of the digitized phantom we added an additional density sampled randomly from a zero-mean Gaussian distribution.

To simulate the large tumor on the left-hand side of Figure 3.1(b) we created a *ghost.* Ghosts are images that are invisible from given projection directions. The existence of ghosts have been known and studied since the earliest days of CT; see, e.g., Section 15.4 of [35]. Our particular way of producing the ghost tumor in Figure 3.2 is based on an idea that was first published almost 40 years ago [33].

Using the notation and the definitions of the previous subsection, we say that p is a *ghost for direction* ϑ if, for every line at distance s from the origin with an angle $\vartheta \in [0,\pi)$ and $d \in \mathbb{R}_{++}$

$$[\mathscr{R} f_p^d](s,\vartheta) = 0. \tag{3.10}$$

For the creation of the ghost, it is necessary that the directions ϑ should be of the form $\arctan(u/v)$, where u and v are integers, both not zero. Suppose that we are given L pairs of such integers $(u_1,v_1),\ldots,(u_L,v_L)$. We now construct an image that is a ghost for each of the directions $\arctan(u_\ell/v_\ell)$, for $1 \le \ell \le L$. The construction defines a sequence $h^0,h^1,\ldots,h^L$ of real-valued functions of two integer arguments of finite support (i.e., for $0 \le \ell \le L$, there are only finitely many pairs of integers (t_1,t_2) for which the value of h^l is not zero). We can select h^0 to be any such function and then define, for $1 \le \ell \le L$ and all pairs of integers (t_1,t_2),

$$h^\ell(t_1,t_2) = h^{\ell-1}(t_1,t_2) - h^{\ell-1}(t_1+u_\ell,t_2+v_\ell). \tag{3.11}$$

Clearly, all the functions defined in this way will be of finite support. Now suppose that there exist integers w_1 and w_2 such that $h^L(t_1,t_2) \neq 0$ implies that $w_1 \le t_1 \le w_1+N-1$ and $w_2 \le t_2 \le w_2+N-1$. (Such w_1 and w_2 can always be found, provided only that N is large enough.) If we now define, for all $(t_1,t_2) \in [0,N-1]^2$

$$g(t_1,t_2) = h^L(w_1+t_1,w_2+t_2), \tag{3.12}$$

then the $N \times N$ digital image g is a ghost for the directions $\arctan(u_1/v_1),\ldots,$ $\arctan(u_L/v_L)$.

In Figure 3.2 we added a ghost g for 22 projections. We selected 22 reasonably evenly spaced projection directions by choosing the pairs (4,3), (4,2), (4,1), (4,0), (4,-1), (4,-2), (4,-3), (3,4), (2,4), (1,4), (0,4), (-1,4), (-2,4), (-3,4), (3,2), (3,1), (3,-1), (3,-2), (2,3), (1,3), (-1,3), and (-2,3) as the values for $(u_1,v_1),\ldots,(u_{22},v_{22})$. The

function h^0 was selected to be a digitized *blob*, a generalized Kaiser-Bessel window function [45], with its free parameters assigned to be the default values selected by SNARK09 [27]. (We discuss blobs in Subsection 3.3.5 and give full details about the available parameters).

Finally, we specify the parameters related to the digitization and the display of our phantom. The image in Figure 3.2 has 243×243 pixels (thus $N = 243$ and $J = 59,049$). We picked the sampling interval $d = 0.0752$ and the size of each pixel was set to 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 component values into gray values in between.

3.2.3 Modes of data collections

All the experiments in this document are simulations of various CT reconstructions of a head phantom (such as the one in Figure 3.2). We are examining the usefulness of the reconstructions on two main datasets. For one we calculate the line integrals based on the digitization of the head phantom, but we only consider a subset of views that makes the underlying system of constrains underdetermined (however it is consistent with the phantom). For the second dataset the underlying system of constraints is overdetermined however the phantom is no longer consistent with the collected line integrals. In addition, in some cases we wanted to make our conclusions statistically sound. To do this, we performed a statistical hypothesis testing; we next give the full details of how we performed it along with our two modes of data collections.

3.2.3.1 Underdetermined (UD) dataset

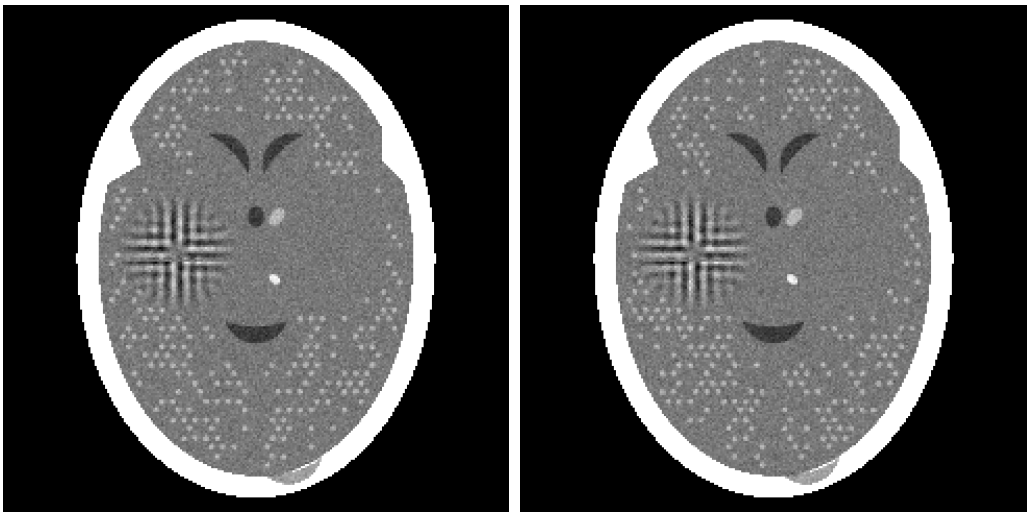
In this set, data are collected by calculating line integrals through the digitized image for either 22 or 82 sets of equally spaced parallel lines. Each such line integral gives rise to a linear equation in the components of x; the set of x that is consistent with such a line integral is a hyperplane in $\mathbb{R}^J$. For the case of the 22 sets, the lines are taken with angle directions that make the large tumor on the left hand side of the phantom in Figure 3.2 a ghost as was described in detail in Subsection 3.2.2 (i.e., its ray-sums are zero for those 22 directions). The data set that has 82 sets have in addition to the 22 directions that creates the ghost mentioned above, an additional 60 equally spaced views with 3° increments between consecutive directions. The vector x that represents the phantom lies in the intersection of all the hyperplanes that are associated with these directions (for the two cases). In the 82 sets case, measurements were taken for $25,452$ lines, making the underlying 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 22 sets case, measurements were taken for only 6,914 lines thus making the problem even more underdetermined than in 82 sets case; the intersection of all the hyperplanes is an at least $52,135$-dimensional subspace of $\mathbb{R}^{59,049}$. We will refer to these two sets of data collection by UD and will indicate which of the two cases we are using.

3.2.3.2 Realistic overdetermined (OD) dataset

Data collected in real-life CT are not at all likely to be consistent; a fundamental reason is that real-life objects are not digitized and hence variations within pixels will exists. In addition, various deviations from the idealized mathematical point of view contribute to the data being inconsistent with the object. In CT the measurements are stochastic in nature (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). Another problem is that the detectors in real instruments will have a width, so even if we ignore the fact that they may not be perfect, they could not be used for measuring line integrals exactly. Furthermore, X-ray beams used in CT are polychromatic and hence their energy spectrum hardens depending on the material they have traveled through [34]. This makes the assignment of a linear attenuation coefficient associated with a point in the body more difficult and yet another estimation takes place to resolve that. There are other (here not listed) additional 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; for a further discussion and in-depth explanation see [35, Chapter 3].

In our simulation we calculated the line integrals based on the geometrical description of the head phantom (the one that was mentioned in Subsection 3.2.2) rather than on its digitization. The width of the detector was simulated by adding, to each line, 10 additional lines (five on each side) with a spacing of $\mathsf{d}/11$ between them, where d ($= 0.0752$ in our case) is the distance between parallel lines along which data are assumed to have been collected in the mathematical formulation. Additionally, statistical noise and scatter were introduced at the levels found in real CT scanners, and furthermore, a beam hardening correction was applied. (The software SNARK09, which we used for all our experiments as mentioned previously, allows us to simulate this kind of data collection, the details of which are explained in [35, Section 4.5].) Data were generated for 360 sets of equally spaced parallel lines (each line gives rise to a hyperplane in $\mathbb{R}^{59,049}$), with a $\frac{1}{2}$ degree increment between consecutive directions. We will refer to this mode of data collection OD.



(a) (b)

Figure 3.3: Random samples from the ensemble of phantoms used for statistical hypothesis testing evaluations.

3.2.4 Statistical hypothesis testing evaluation

General conclusions should not be made based on a few anecdotal experiments. In this subsection we discuss simulation experiments that use statistical hypothesis testing to evaluate the relative efficacy of various reconstruction algorithms for a medically relevant task, in our case, the detection of low-contrast small tumors in the brain from X-ray CT projection data. In accordance with the general description of such statistical hypothesis testing evaluations in Section 5.2 of [35], the following are the steps to carry out such an evaluation:

1. Generation of random samples from a statistically described ensemble of phantoms and simulation of realistic data collection for each sample. Our random samples were similar to the phantom in Figure 3.2, but they included a large number of pairs of potential tumor sites, with the locations of the sites in each pair symmetric around the vertical axis of the head phantom. Only one site of the pair has a tumor in it, these are randomly selected for each sample. Thirty such phantoms will be generated; Figure 3.3 shows two of them.

The small tumors have circular shapes and are of radius 0.1 cm. The data collection for all the experiments we report here used the OD one described in detail in Subsection 3.2.3.2.

2. Reconstruction from each of the generated projection OD data sets by each of the algorithms to be compared.

3. Assignment of *figures of merit* (FOMs) to each reconstruction. FOMs should measure the goodness of the reconstruction for our task, the two that we used for our experiments are *image wise region of interest* (IROI) and *hit ratio* (HITR). It was previously found that IROI has the tendency to agree with the performance human observers in identifying the tumors in a reconstruction [53]. HITR considers the pairs of potential tumors site. Such a pair is a hit if the structure in the pair with the higher average density in the phantom is also the structure in the pair with the higher average density in the reconstruction. The HITR for a reconstruction is the number of hits divided by the total number of pairs. The intuition behind HITR is that a radiologist examining a reconstruction would look at the corresponding symmetric area in order to determine if something is a tumor, as structures in the brain tend to be symmetric.

4. Calculation of statistical significance. For any pair of algorithms and for any FOM, the null hypothesis is that the expected average values (over the 30 reconstructions) of the FOMs for the two algorithms are the same, with the alternative hypothesis that the expected average value is in fact higher for the algorithm for which the experimentally observed average is higher. Based on the 30 pairs of FOMs one can calculate the P-value, which is the probability of observing a difference between the performances (according to the FOM) of the two algorithms that is as high or higher than the observed difference

if the null hypothesis that the two algorithms are equally efficacious were true. The smallness of the P-value measures the significance by which we can reject the null hypothesis in favor of the alternative.

3.3 Experiments

The superiorization algorithm described in Section 3.1 specified how v^k and β_k are selected for all $k \in \mathbb{N}$. The algorithm steers the iterates towards a minimizer of a given convex function ϕ that assigns to each image a number that indicates, in some sense, the "undesirability" of the image. (For example, we may know that most images in our application area should be "piece-wise smooth." In that case, $\phi(x)$ should measure the extent to which piece-wise smoothness is violated in the image represented by x.) We give next two such functionals that were used in our experiments.

Many researchers in image processing have been advocating the use of total variation, e.g., [8, 23, 46, 57, 62]. For a $G \times H$ image q whose pixel values are denoted by $q_{g,h}$ $(1 \leq g \leq G, 1 \leq h \leq H)$, the *total variation* (TV) of q is

$$TV(q) = \sum_{g=1}^{G-1} \sum_{h=1}^{H-1} \sqrt{(q_{g+1,h} - q_{g,h})^2 + (q_{g,h+1} - q_{g,h})^2}. \tag{3.13}$$

By mapping q into a $(G \times H = J)$-dimensional point 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 Section 3.1.

Our second choice for the convex function ϕ to be used in some of the following experiments is based on the maximum entropy formalism, which is a general scientific approach with whole books devoted to it; e.g., [44]. The suggestion that it be used for image reconstruction first appeared in the open literature in [32]. It has been extensively used in the related field of digital image restoration; see, e.g.,

[3]. 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* (NE) of q is

$$negative\text{-}entropy(q) := \sum_{g=1}^{G-1} \sum_{h=1}^{H-1} \left(\frac{q_{g,h}}{\mathsf{Q}} \right) \ln \left(\frac{q_{g,h}}{\mathsf{Q}} \right), \tag{3.14}$$

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 [35, 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 NE, 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 (3.14)), 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 in Section 3.1 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 z 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.)

Theorems 1 and 2 were stated for general closed convex sets, while Theorems 3 and 4 were stated only for the special case when the convex sets are hyperplanes. In the first case we have considered two kinds of constraints for the experiments, one that uses only hyperplane constraints and another that uses both hyperplanes and halfspaces constraints. For the hyperplane case, each of the calculated line integrals mentioned earlier give rise to a linear equation in the components of x; the set of x that is consistent with such a line integral is a hyperplane in $\mathbb{R}^J$. These can

be written mathematically by

$$C_i = \{x : \langle a^i, x\rangle = b_i\}, \tag{3.15}$$

where $a^i \in \mathbb{R}^J$ and $b_i \in \mathbb{R}$. For the second type of constraints, we have considered halfspaces that correspond to nonnegativity constraints on the pixel values; i.e., sets of the form

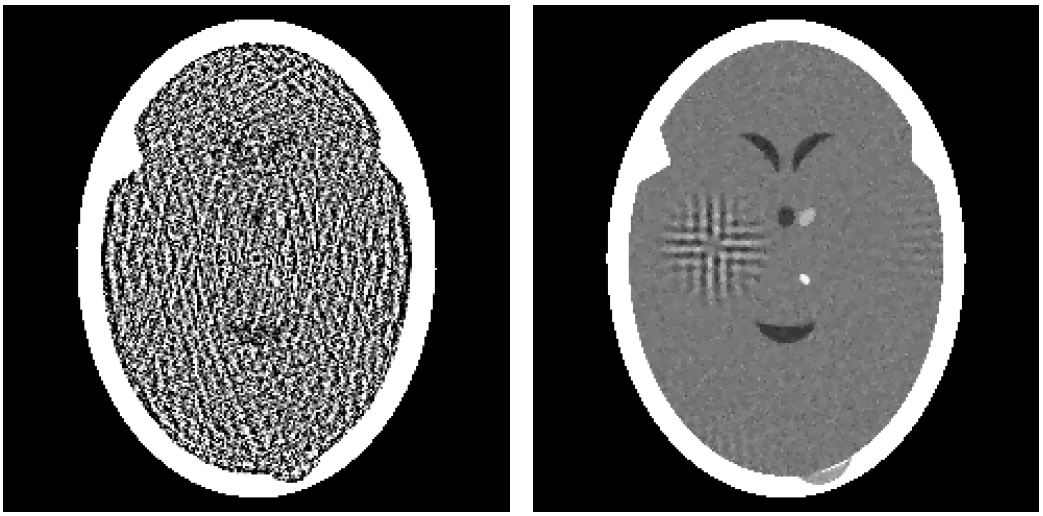
$$\{x : x_j \geq 0\}, \tag{3.16}$$

for $1 \leq j \leq J$.

As a general remark, we point out that all the computational work reported here was done using SNARK09 [27]; the phantom, the data, the reconstructions, displays, and plots were all generated within this same framework. This implies that, for example, differences in the reported reconstruction times are not due to the different algorithms being implemented in different environments.

3.3.1 Experiments with PR-AP methods

We illustrate the efficacy of the AP methods when perturbations are utilized versus when they are not. A classical method for finding a common point in the intersection of a set of hyperplanes is the cyclic projection method, which is commonly known as ART in tomography [35, Chapter 11]. If it is initialized with x^0 being the zero vector, it is known [35, Section 11.2] to converge to that point x in the intersection of the hyperplanes for which $\|x\|$ is minimal. In practice, ART needs to be stopped after a finite number of steps. In Figure 3.4(a) we show the result obtained by ART when we stopped at a k for which $\mathrm{Res}(x^k) < 0.001$. On an Intel Xeon 1.7 MHz processor 1 G RAM workstation, obtaining such a good fit to the data by ART required more than 15 hours. (To demonstrate that 0.001 is indeed a small value in our context, we point out that $\mathrm{Res}(x^0) = 330.204$.)



(a) (b)

Figure 3.4: Illustrative example of superiorization in tomography using AP methods with hyperplanes constraints. (a) Norm-minimizing reconstruction (cyclic projection method, ART). (b) TV-superiorization reconstruction (cyclic projections with perturbations). The corresponding numerical values associated with these images are given in Table 3.1.

There are some obvious differences between the phantom in Figure 3.2 and the ART reconstruction in Figure 3.4(a). This indicates that $\|x\|$ may not be a particularly good measure of the undesirability of x in this situation and therefore we considered for ϕ the TV functional. The only difference between ART and this new algorithm comes from the perturbations aimed at reducing the total variation. Again we started the process with x^0 being the zero vector and stopped it when $\text{Res}(x^k) < 0.001$. Figure 3.4(b) illustrates the output of the TV-superiorizing algorithm. It is visually superior to the reconstruction of Figure 3.4(a). (As a numerical measure, the norm of the difference between the ART reconstruction and the phantom is more than 35 times greater than the norm of the difference between the TV-superiorizing reconstruction and the phantom.)

It is interesting that, for this data set, the TV-superiorizing algorithm is significantly less expensive than ART: The total time required was less than 39 minutes. The reduction in $\text{Res}(x^k)$ as a function of computer time is plotted in Figure 3.5

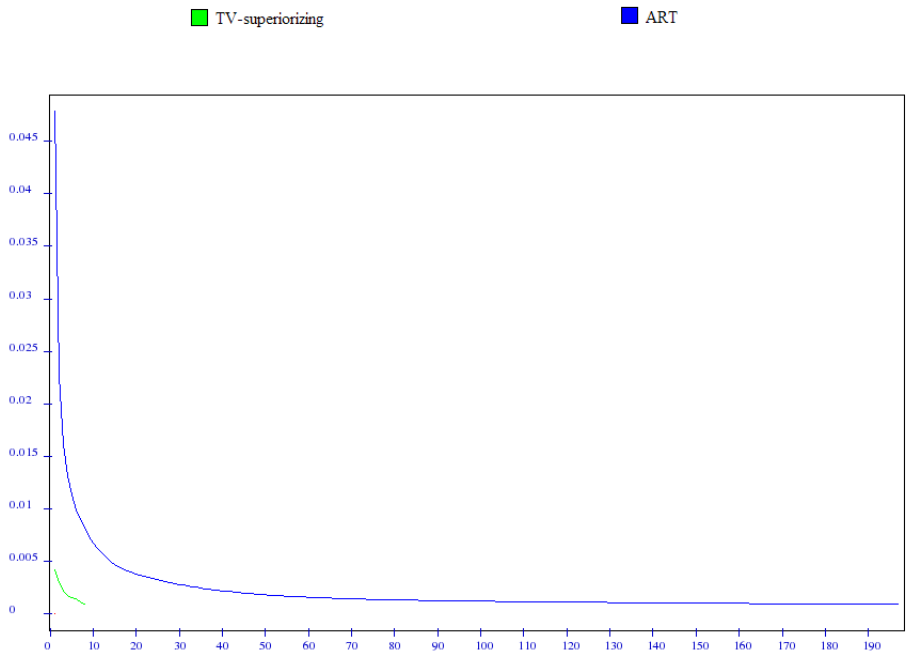


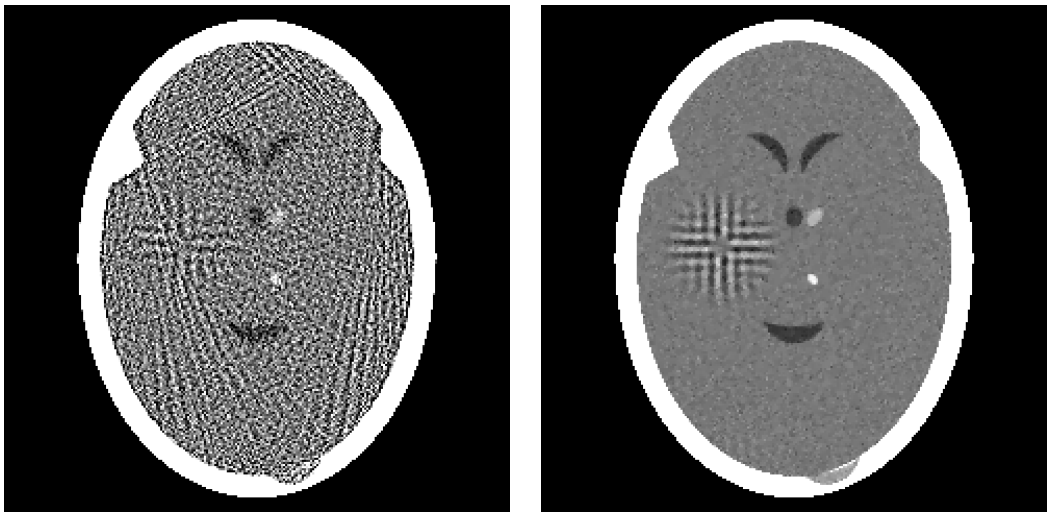
Figure 3.5: Plots of $\mathrm{Res}(x^k)$ for ART (blue, top) and the amalgamated TV-superiorizing algorithm (green, bottom) both plotted against computer time.

Method	norm	TV	distance	time [h]
ART (no perturbations)	46.04	1299.7	3.617	15.77
AP TV-superiorization	46.19	435.9	0.103	0.64

Table 3.1: Numerical values for the outputs of the amalgamated algorithms (without nonnegativity constraints).

for both algorithms. Even though a single iterative step of ART is less expensive than that of the TV-superiorizing algorithm, the perturbations in the latter steer it towards the correct result (i.e., in the general direction of the phantom) and so much fewer steps are needed to get $\mathrm{Res}(x^k)$ below a given ε.

In Table 3.1, we report on the values of the norm and TV for the outputs of the two algorithms (as well as the distance between the reconstruction and the phantom and the time needed to obtain the reconstruction). As can be seen, the algorithms tend to minimize the function that they are supposed to be minimizing; the superiority of the reconstruction in Figure 3.4(b) to that in Figure 3.4(a) is due to TV



(a) (b)

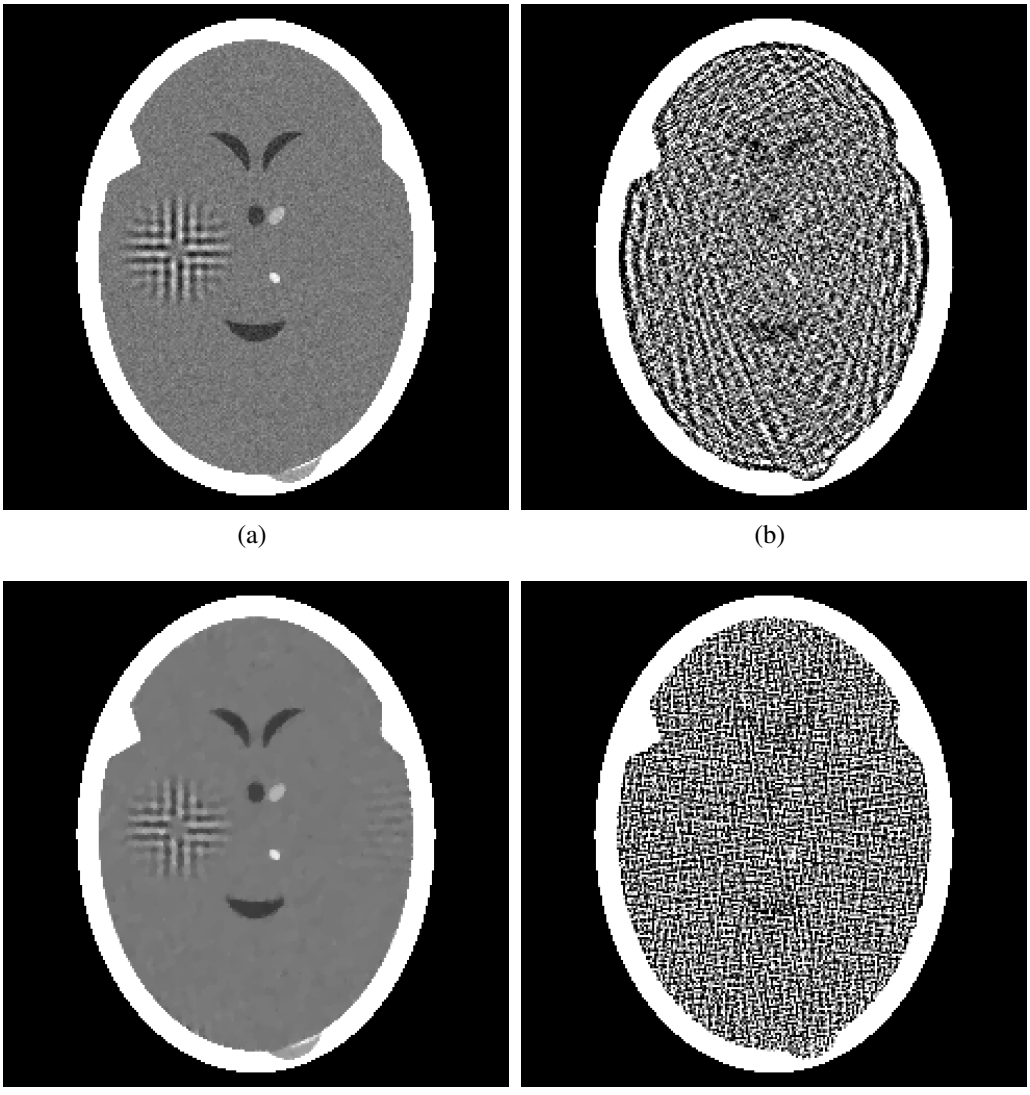
Figure 3.6: Illustrative example of superiorization in tomography using AP methods with hyperplanes and nonnegativity constraints. (a) Norm-minimizing reconstruction (cyclic projection method, ART2). (b) TV-superiorizing reconstruction (cyclic projections with perturbations). The corresponding numerical values associated with these images are given in Table 3.2.

minimization being a more appropriate aim than norm minimization in the current circumstances.

As a further illustration, we also report on results of experiments in which in addition to the hyperplanes of (3.15) we also have half-spaces that correspond to nonnegativity constraints on the pixel values as in (3.16). In setting up the exact formulation of the problem, in the sequence of convex sets each set of the form (3.15) is followed by the J sets of the form (3.16). We compared the performance of our TV-superiorization method with a version of ART (called ART2, introduced in [37], see also [19, page 152]) that minimizes the norm under such mixture of constraints. We again set up the experiments to stop when $\text{Res}(x^k) < 0.001$. The results are shown in Figure 3.6 and reported in Table 3.2.

Method	norm	TV	distance	time [h]
ART2	60.80	3367.1	1.027	0.93
AP TV-superiorization	46.19	437.5	0.087	1.49

Table 3.2: Numerical values for the outputs of the amalgamated algorithms (with nonnegativity constraints).



(a) (b)

(c) (d)

Figure 3.7: Illustration of reconstructions obtained by our BIP algorithm from 82 views, with hyperplane constraints and stopping criterion of $\text{Res}(x^k) < 0.05$ (a) Head phantom for which data were collected (same as in Figure 3.2). (b) Norm-minimizing reconstruction (no perturbations). (c) Reconstruction with TV-superiorizing perturbations. (d) Reconstruction with entropy-superiorizing perturbations.

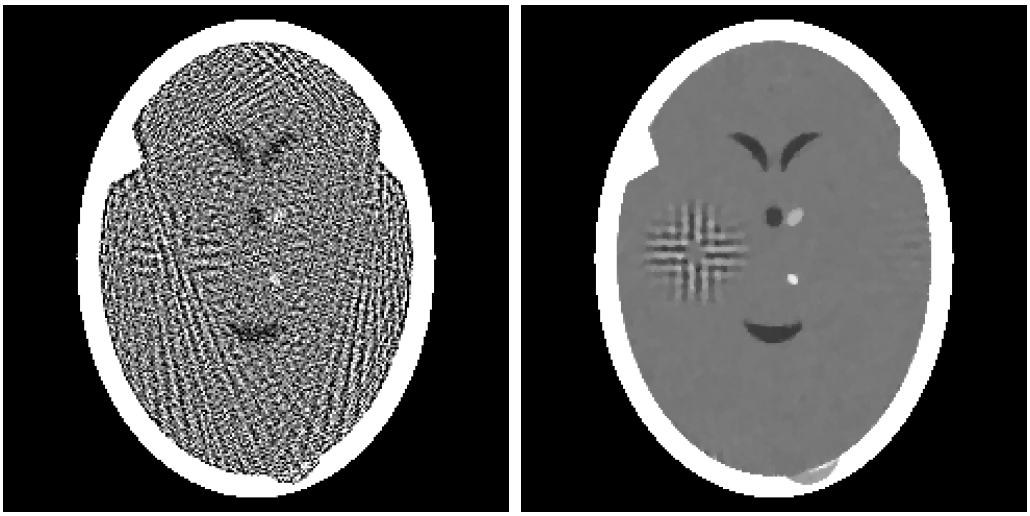
3.3.2 Experiments with PR-BIP for convex sets

In this subsection we give several illustrations to the BIP algorithms discussed in Subsection 2.3.1, which are applicable to general closed convex sets (by Theorem 2). We first considered the UD data collection for 82 views (from Subsection 3.2.3.1) and applied the BIP algorithm (2.30) to the specified data set for the two choices of ϕ given previously, namely TV and NE. Blocks were set so that each block B_u consists of all the indices i associated with the measurements taken in a particular direction (and therefore $\{1,\dots,I\} = \bigcup_{u=1}^{U} B_u$ with $U = 82$). The relaxation parameter for all reconstructions was set to $\lambda = 1$ and all were stopped when $\text{Res}(x^k) < 0.05$. Figure 3.7(b) shows the reconstruction when no perturbations are introduced and Figure 3.7(c) shows the reconstruction when TV-superiorizing perturbations are present. Both runs started with x^0 being the *zero point* (i.e., all its components are 0). Note that $\text{Res}(x^0) = 330.208$, 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., $\beta_k = 0$, for $k \in \mathbb{N}$) converges to the feasible point with minimal norm as was indicated in the previous subsection. Clearly, the reconstruction in Figure 3.7(c) is visually superior to the reconstruction in Figure 3.7(b). As a numerical measure, the norm of the difference between the reconstruction without perturbations in Figure 3.7(b) and the phantom in Figure3.7(a) is 3.764, while the norm of the difference between the reconstruction with the perturbations in Figure 3.7(c) and the phantom is 0.157, making it 24 times smaller than that without perturbations. Moreover, the TV of the image in Figure 3.7(c) is 421.568, which is near to (and is actually less than) the TV of the phantom in Figure 3.7(a) (450.594), whereas the TV of the image in Figure 3.7(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 (3.14). 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 [43]. MART is actively used in various applications; see, e.g., [42, 65]. For this run, x^0 was set to be the point that is also the initial point for MART for which each component is either zero or another constant, selected based on the measurements so that the total density is the Q of (3.14). Figure 3.7(d) presents the resulting image for the chosen stopping criterion. If we compare the two superiorizing reconstruction results (Figures 3.7(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 (3.14) is provided by the entropy-superiorization algorithm. In fact, we ran MART (as implemented in the software package SNARK09 [27]) on the data of this experiment and obtained an entropy of 10.305. The entropy of the image in Figure 3.7(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 3.7(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 I hyperplanes used in the previous experiment, we also have a convex set corresponding to nonnegativity constraints on the pixel values as in (3.16). 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 $I+1$. Our actual algorithm for taking care of nonnegativity (in addition



(a) (b)

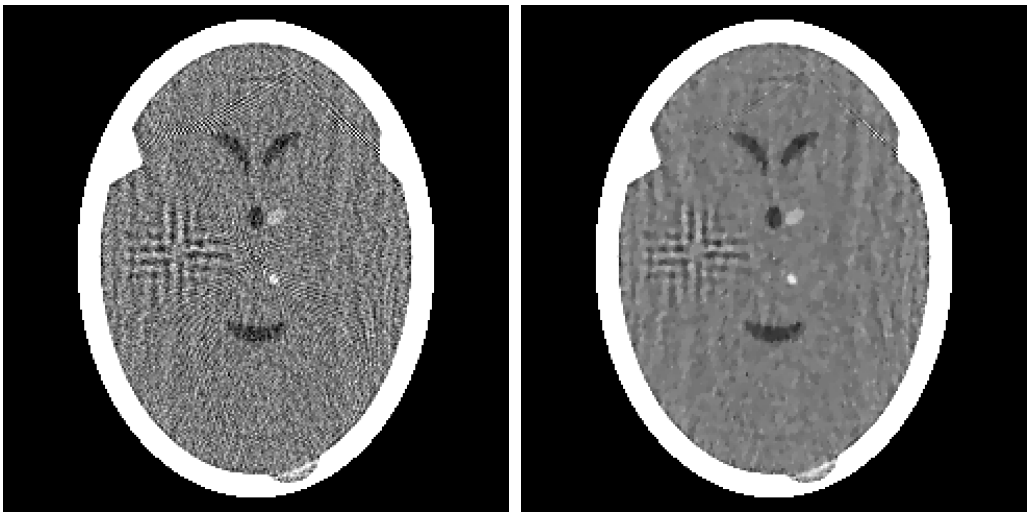
Figure 3.8: Illustration of reconstructions obtained by a BIP algorithm from Theorem 2 using the dataset for 82 views from Subsubsection 3.2.3.1, with hyperplane and nonnegativity constraints and stopping criterion of $\text{Res}\left(x^k < 0.05\right)$ (a) Reconstruction when no perturbations are present. (b) Reconstruction with TV-superiorizing perturbations.

to the hyperplane constraints) uses instead of (2.27) the operator

$$\mathbf{Q} := Q_{U+1}Q_U \cdots Q_{U+1}Q_1, \tag{3.17}$$

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 C_{I+1}.

We compared the performance of the BIP algorithm (2.30) without and with perturbations for the TV functional (3.25). As in the previous experiment, we started both reconstructions with x^0 being the zero point, set $\lambda = 1$, and set the algorithms to stop at $\text{Res}(x^k) < 0.05$. The results of the reconstructions are given in Figure 3.8. Note again the superiority of the reconstruction of the TV-superiorizing algorithm with perturbations present (Figure 3.8(b)) as compared to the one produced without perturbations (Figure 3.8(a)). The norm of the difference between the reconstruc-



(a) (b)

Figure 3.9: Illustration of reconstructions obtained by our BIP algorithm from 360 views of physically realistic projection data, with hyperplane and nonnegativity constraints and stopping criterion of $\text{Res}\left(x^k\right) < 3.75$. (a) Reconstruction when no perturbations are present. (b) Reconstruction with TV-superiorizing perturbations.

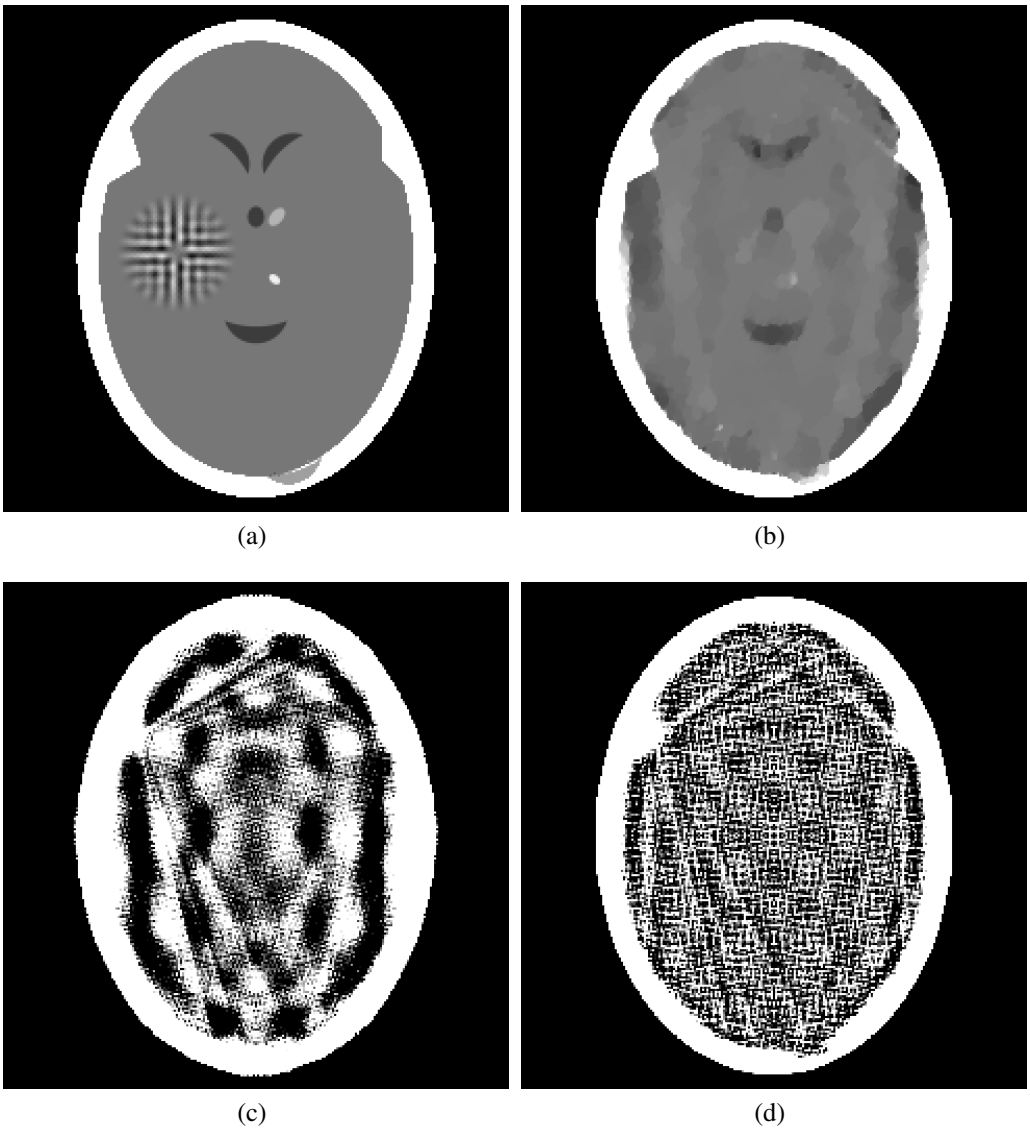
tion without perturbations in Figure 3.8(a) and the phantom is more than 8 times greater than that between the reconstruction with the perturbations in Figure 3.8(b) and the phantom (1.254 and 0.143, respectively). The use of the nonnegativity constraint (3.16) improved the quality of the reconstructions in both cases, although only slightly in the TV-superiorizing case. The TV for the image on the right in Figure 3.8 is 421.185, while for that on the left it is 667.926.

In the third experiment we demonstrate the BIP algorithm from Theorem 2 when the dataset is the OD of Subsubsection 3.2.3.2. Similarly to our previous experiments in this subsection, blocks were formed using all measurements taken from a particular direction (making $U = 360$), and an additional constraint of the form (3.16) was utilized in the same manner as in the former experiment.

The reconstructions from this dataset, both without and with TV-superiorizing perturbations, are shown in Figure 3.9. The stopping criterion in both cases was $\text{Res}\left(x^k\right) < 3.75$, which is reasonable since for this noisy data set the value of Res for the

phantom is actually higher (3.97, to be exact). 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 3.9(a) and Figure 3.9(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 3.2.

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 3.10(a) shows a head phantom for which data were collected. The only difference between the head phantom in Figure 3.2 and this one is that in Figure 3.2 the phantom includes local inhomogeneities; see Subsection 3.2.2 for more details. For the current example we use the decreased number 22 of views of the UD dataset. Here $U = 22$ and $I = 6,914$, and so we have almost an order of magnitude fewer linear equations than unknowns. We again ran our algorithm starting from the zero point ($\text{Res}\left(x^0\right) = 171.557$ in this case) and chose the stopping criterion of $\text{Res}\left(x^k\right) < 0.05$. The TV of the phantom in Figure 3.10(a) is 421.735. Since the system is consistent, the phantom in Figure 3.10(a) lies in the intersection of all the constraints. Figure 3.10(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



(a) (b)

(c) (d)

Figure 3.10: Illustration of reconstructions obtained by the BIP algorithm of Theorem 2 using consistent data with 22 views, hyperplane constraints and stopping criterion of $\mathrm{Res}\left(x^k\right) < 0.05$. (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.

perturbations are used in Figure 3.10(c) and when the perturbations are aimed at maximizing the entropy (Figure 3.10(d)). As a numerical measure, the norm of the difference between the reconstruction with perturbations in Figure 3.10(b) and the phantom in Figure 3.10(a) is 1.684, which is less than a fifth of the norm of the

difference between Figure 3.10(c) and the phantom and less than half of the norm of the difference between Figure 3.10(d) and the phantom.

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 (2.28) is large; it is 345 for our examples. From this it follows that $x^{k+1} = \mathbf{Q}x^k$ is practically the same as 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 and achieve a more than an order of magnitude speedup while retaining the limiting convergence property stated in Theorem 2. To indicate the great potential that exists for speeding up the BIP algorithm used in this section we note that it required 10,400 iterations to reach the very stringent stopping criterion that resulted in Figure 3.7 (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 (2.5), which is the same as the BIP algorithm described here 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 (2.30)), which would slow it down to some extent. Also, it is often the case that special hardware allows us to implement a BIP 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

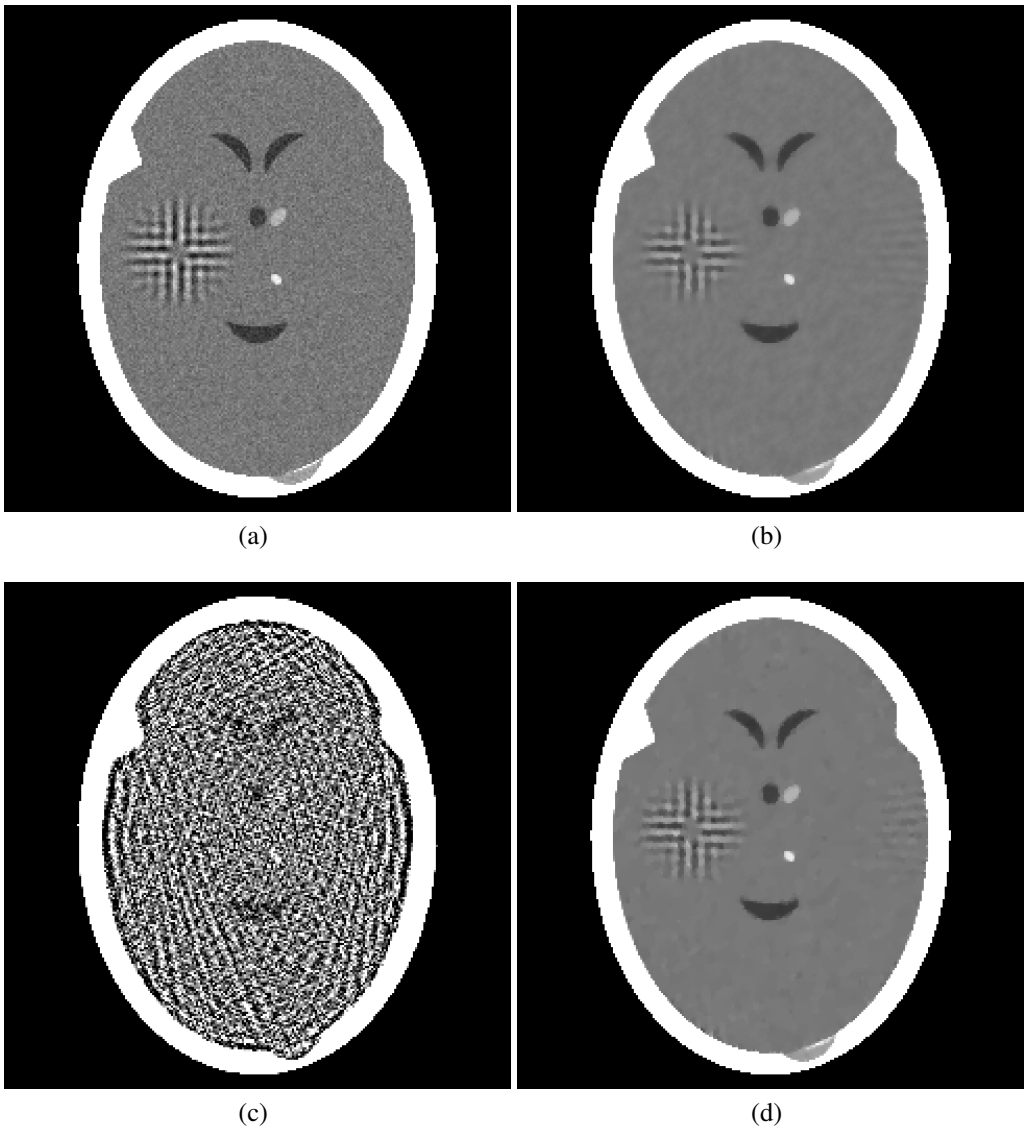
of our perturbed BIP algorithm has the potential of being in practice advantageous; we demonstrate this next.

3.3.3 Experiments with PR-BIP for hyperplane constraints

We next give illustrations when the algorithm is (2.51) from Theorem 3. For the first experiment, we considered the UD dataset for 82 views from Subsubsection 3.2.3.1. In this case, each block A_u consists of all the indices i associated with the measurements taken in a particular direction. Thus $\{1,\dots,I\} = \bigcup_{u=1}^{U} B_u$, with $U = 82$. Note that the number of elements in B_u is ℓ_u as defined above (2.48) in Subsection 2.3.2.

We applied our BIP algorithm based on (2.51) to the specified data set with the selection of TV as the functional ϕ. The choice of the relaxation parameters, λ_u, has an affect on the stopping point for any chosen ε. Choosing the maximum λ_u allowed by (2.48) gets to a solution faster, but it reduces the number of superiorization steps (Step 3 of the pseudocode in Section 3.1) that steer the process towards a solution close to the minimizer of our functional ϕ. We have chosen for all the illustrations of this subsection (except for the ones described in the last paragraph of this subsection) the relaxation parameters $\lambda_u = \lambda = 0.125$, which reflects a reasonable trade-off between speed and quality of the reconstruction. This choice is based on a set of auxiliary experiments, the details of which are omitted. For the stopping criterion we set $\text{Res}\left(x^k\right) < 0.05$.

Figure 3.11(b) shows the reconstruction when TV-superiorizing perturbations are present and Figure 3.11(c) shows the reconstruction when no perturbations are introduced. Both runs started with x^0 being the zero point (i.e., all its components are 0). We again point out that $\text{Res}\left(x^0\right) = 330.208$, demonstrating that the value 0.05 for the stopping criterion is small in our context. Again we indicate that the algorithm without perturbations (i.e., $\beta_k = 0$, for $k \in \mathbb{N}$) converges to the feasible



(a) (b)

(c) (d)

Figure 3.11: Reconstructions from UD consistent data for 82 views with the stopping criterion of $\text{Res}\left(x^k\right) < 0.05$ using accelerated BIP algorithm. (a) Phantom for which data were collected, same as in Figure 3.2. (b) BIP TV-superiorizing accelerated algorithm (Theorem 3). (c) Norm-minimizing reconstruction (Theorem 3, no perturbations). (d) BIP TV-superiorizing without acceleration (Theorem 2).

point with minimal norm [35, Section 11.2]. Clearly, the reconstruction in Figure 3.11(b) is visually superior to the reconstruction in Figure 3.11(c).

In the first two rows of Table 3.4, we report on the values of the norm and TV for the outputs of the two algorithms (as well as the distance, defined as the norm of the difference between the reconstruction and the phantom, and the time in min-

Method	norm	TV	distance	time (m)
BIP TV-superiorization with acceleration (Theorem 3)	46.18	424.22	0.15	14.85
BIP norm-minimizing (no perturbations) with acceleration (Theorem 3)	46.03	1281.18	3.75	42.47
BIP TV-superiorization (Theorem 2) without acceleration	46.18	421.57	0.16	390.29

Table 3.4: Numerical values for the outputs (shown in Figure 3.11) of the three algorithms on consistent data.

utes needed to obtain the reconstruction). As can be seen, the algorithms tend to minimize the functions that they are supposed to be minimizing; the superiority of the reconstruction in Figure 3.11(b) to that in Figure 3.11(c) is due to TV minimization being a more appropriate aim than norm minimization under the current circumstances.

In order to appreciate the speedup of our new method, we ran another TV-superiorizing BIP algorithm (from Theorem 2) on this data set. We point out that using (2.28) of Theorem 2 for the special case when the sets are hyperplanes we obtain that, for $x \in \mathbb{R}^J$ and $1 \leq u \leq U$,

$$Q_u x = \frac{1}{R} \sum_{i \in B_u} \left(x + \frac{b_i - \langle a^i, x \rangle}{\|a^i\|^2} a^i \right) + \frac{R - \ell_u}{R} x, \tag{3.18}$$

where a^i is the transpose of the ith row of A, and R defined in (2.29). Rearranging (3.18) we obtain

$$Q_u x = x + \frac{1}{R} \sum_{i \in B_t} \frac{b_i - \langle a^i, x \rangle}{\|a^i\|^2} a^i. \tag{3.19}$$

Thus, the only distinction between the operators in (3.19) and in (2.49) with the matrices chosen as

$$M_u = diag \left(\frac{1}{\|a^i\|^2} \right)_{i \in B_u} \tag{3.20}$$

comes from the operator in (3.19) being dependent on R, while the operator in (2.49) allows flexibility in the selection of λ_u, with a wider range of choice that depends only on (2.48). To be precise, in our example above, $\frac{1}{R} = \frac{1}{343}$, while the relaxation parameter for the newly proposed operator of Theorem 3 (2.49) was chosen in our implementation to be $\lambda_u = \lambda = \frac{1}{8}$, making the corrections in the subiterative steps 42 times greater than what was used in the unaccelerated method.

Again, we started the process with the zero point and set the stopping criterion to $\text{Res}\left(x^k\right) < 0.05$. Figure 3.11(d) shows the resulting image and the third row of Table 3.4 gives the corresponding numerical values. Both reconstructions with perturbations seem to be superior to the reconstruction without perturbations (Figures 3.11(b) and 3.11(d), respectively, versus 3.11(c)). Note that the TV (reported in the TV column of Table 3.4) of the reconstructions produced by the two superiorization algorithms are nearer to (and are in fact less than) the TV of the phantom in Figure 3.11(a) (which is 450.594), whereas the TV of the image in Figure 3.11(c) that is obtained without perturbations is 1281.18. In addition, distances between the reconstructions by the perturbed algorithms and the phantom (reported in the distance column of Table 3.4) are 24 times less than the distance between the reconstruction without perturbations and the phantom (with a slightly smaller distance for the reconstruction obtained using acceleration).

Figure 3.12 shows a graph of $\text{Res}\left(x^k\right)$ over computer time for the three algorithms reported in Table 3.4. As indicated by the plot (and also by the time column in the table), the accelerated BIP TV-superiorizing method (from Theorem 3, in solid gray at the bottom-left corner) required less than 15 minutes (on an Intel Xeon 1.7-GHz processor 1-G RAM workstation) to reach a Res below 0.05, while the method without acceleration (in black) took more than 28 times longer to obtain a similar Res. Even though the accelerated method without perturbations (i.e., the norm minimizing method from Theorem 3 in broken-line gray) is faster than the

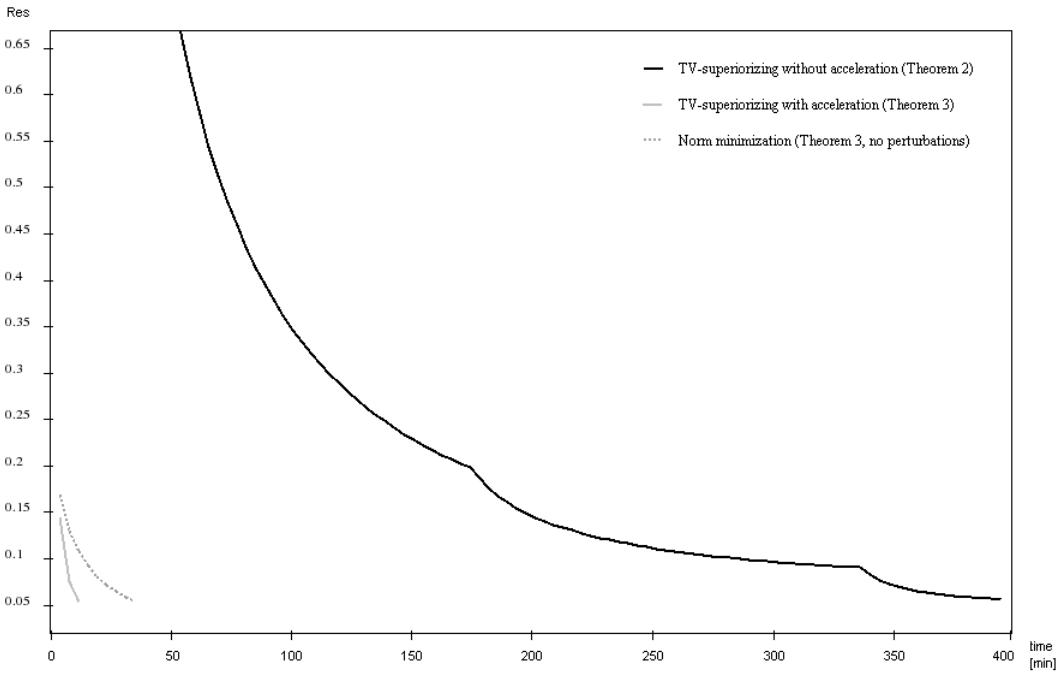
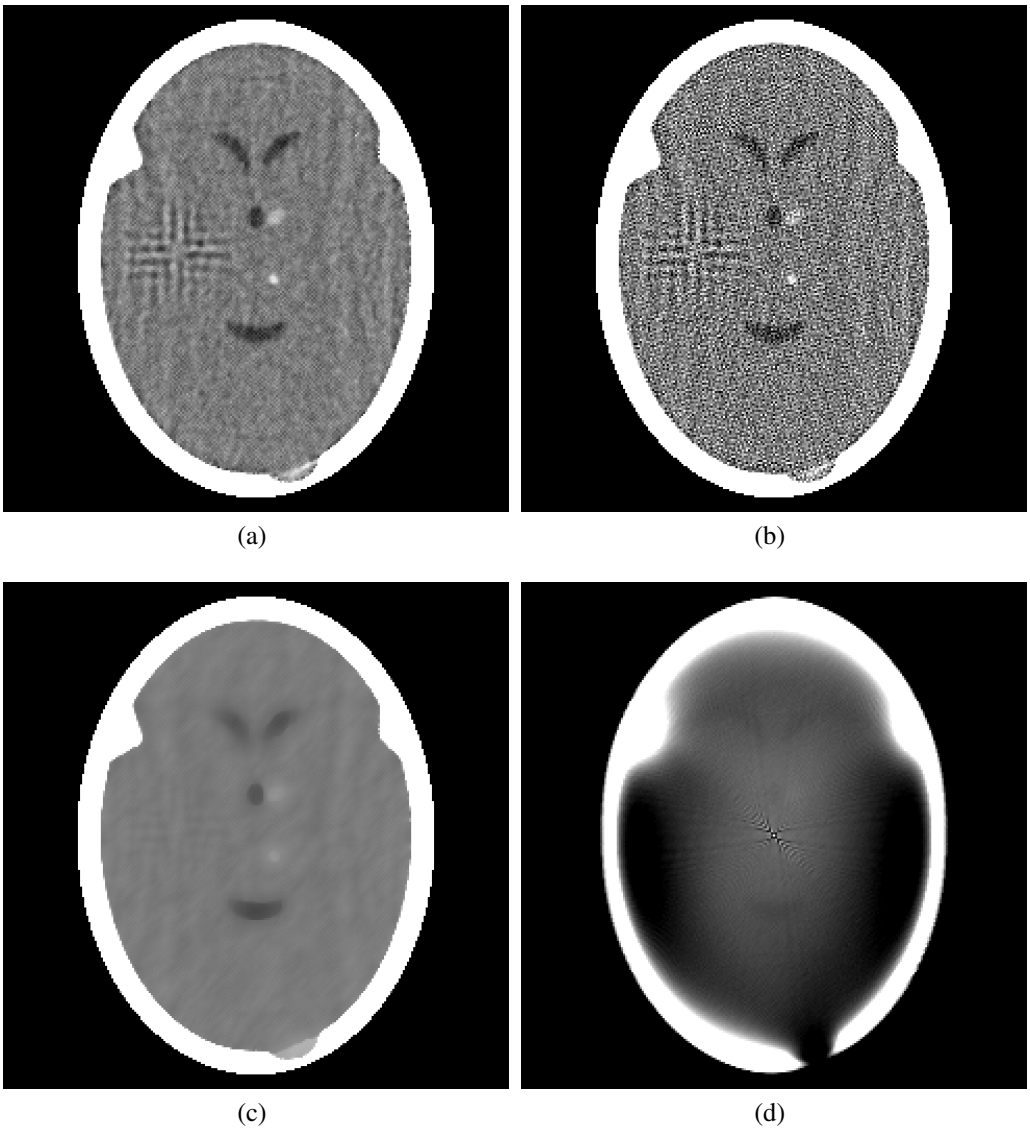


Figure 3.12: Plots of Res$\left(x^k\right)$ against computer time of the three algorithms on 82 views UD consistent data.

TV-superiorizing method without acceleration from Theorem 2, the perturbations in the latter steer it towards the correct result (i.e., in the general direction of the phantom).

In the next experiment we examined our method from Theorem 3 with the OD dataset of Subsubsection 3.2.3.2 by running the accelerated BIP algorithm and comparing the results with and without TV-superiorizing perturbations. Blocks were again formed by using all measurements taken from a particular direction (making $U = 360$). The starting point was the zero point for all runs in this experiment and the stopping criterion was Res $\left(x^k\right) < 5$, which is reasonable since for this noisy data set the value of Res for the phantom is 4.909. Figures 3.13(a) and 3.13(b) show the resulting reconstructions and Table 3.5 provides the corresponding numerical values. The quality of the reconstruction in Figure 3.13(a) is visually better than that of the reconstruction in Figure 3.13(b) by the algorithm without perturbations. The



(a) (b)

(c) (d)

Figure 3.13: Reconstructions of the head phantom from realistic data obtained from 360 views with the stopping criterion of $\text{Res}\left(x^k\right) < 5$. (a) BIP TV-superiorizing accelerated algorithm of Theorem 3. (b) BIP algorithm of Theorem 3 with no perturbations. (c) BIP TV-superiorizing without acceleration from Theorem 2 for $\text{Res}\left(x^k\right) < 5$. (d) BIP TV-superiorizing without acceleration from Theorem 2, stopped after the time it took our accelerated algorithm to obtain (a).

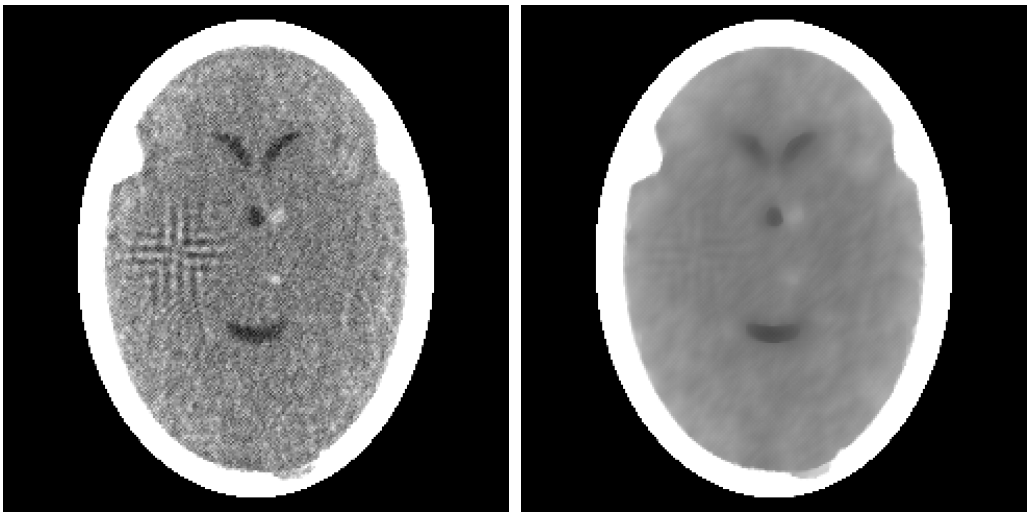
superiority of the reconstruction with perturbations is also expressed numerically by both the distance from the phantom and by the TV value, as indicated in the first two rows of Table 3.5. We also ran the TV-superiorizing algorithm without acceleration from Theorem 2 on these inconsistent data. The resulting image is in Figure

Method	norm	TV	distance	time (s)
BIP TV-superiorization with acceleration (Theorem 3)	45.60	548.36	2.79	37.3
BIP norm-minimizing (no perturbations) with acceleration (Theorem 3)	45.60	790.59	2.96	37.3
BIP TV-superiorization (Theorem 2) without acceleration	45.48	403.11	3.20	1338.6

Table 3.5: Numerical values for the outputs of the algorithms for inconsistent data in Figure 3.13.

3.13(c). As indicated by the third row of Table 3.5, the time that algorithm took to reconstruct that image was more than 35 times the time needed by the accelerated TV-superiorizing algorithm of Figure 3.13(a). As a further illustration, we looked at the image at the iterate reached by the algorithm from Theorem 2 after 37.26 seconds had passed, in order to demonstrate how much the acceleration buys us. The result is seen in Figure 3.13(d), which took approximately 39 seconds to obtain. As can be seen, the algorithm does not provide an acceptable looking reconstruction at that time (that image has a distance of 18.74 from the phantom and the value of Res is 98.938).

Using the evaluation method described in Subsection 3.2.4, we report on the outcome of a comparative evaluation procedure when the algorithms compared are two BIP TV-superiorizing algorithms, the one with acceleration based on Theorem 3 and the one without acceleration from Theorem 2. This is of interest since the results of the experiments we reported so far are anecdotal, they may not be representative of the general situation. Also, in spite of the fact that the TV for Figure 3.13(c) is much less than that for Figure 3.13(a) (see Table 3.5), the visual evaluation of the images indicate that the former is a worse reconstruction than the latter (in particular, Figure 3.13(c) shows hardly any sign of the big tumor in the left half of the phantom shown in Figure 3.2, whereas it is clearly visible in Figure 3.13(a).) The results were conclusive: for both FOMs, the null hypothesis of equal



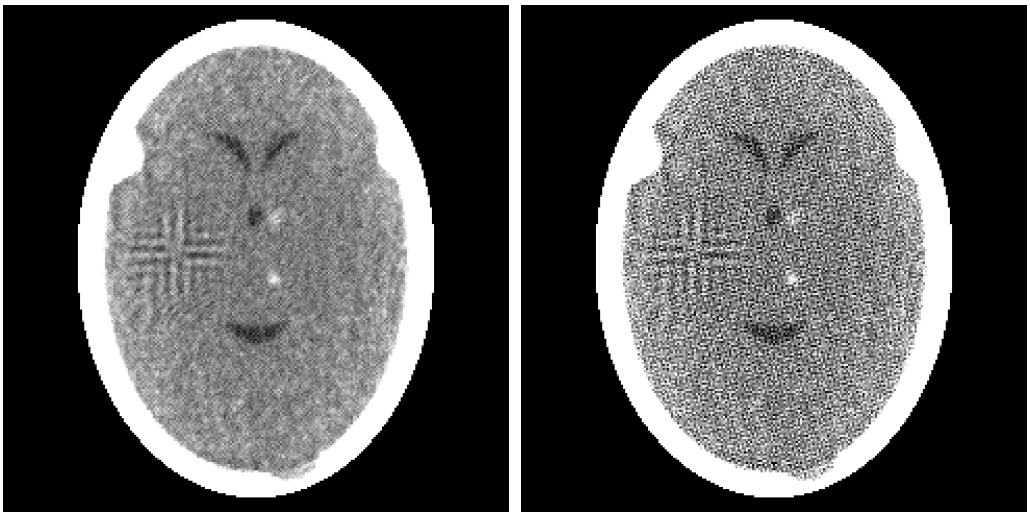
(a) (b)

Figure 3.14: Reconstructions of the phantom shown in Figure 3.3(a). (a) BIP TV-superiorizing accelerated algorithm of Theorem 3. (b) BIP TV-superiorizing without acceleration from Theorem 2.

efficacy can be rejected with extreme significance in favor of the alternative that the method with acceleration performs better for the task at hand (the P-values are 7.72×10^{-15} for IROI and 1.68×10^{-12} for HITR). As an illustration we include in Figure 3.16 the reconstructions of the sample phantom from Figure 3.3(a) by the two algorithms.

Next we report on the statistical comparison of the two accelerated algorithms whose behaviors on a single data set are reported in the first two lines of Table 3.5; they are both based on Theorem 3, but one uses TV-superiorization while the other does not use perturbations. The results were again conclusive: for both FOMs, we can reject the null hypothesis that the two algorithms are equally efficacious with extreme significance in favor of the alternative that the one that superiorizes the TV functional is more helpful for the task at hand (the P-value for HITR is 1.52×10^{-8} and for IROI it is 1.82×10^{-14}). As an illustration, we give in Figure 3.15 examples of reconstructions by the two algorithms.

Finally, we report on statistical experiments that were carried out because of



(a) (b)

Figure 3.15: Reconstructions of a randomly sampled phantom similar to the ones from Figure 3.3. (a) Accelerated algorithm of Theorem 3 with TV-superiorization. (b) Accelerated algorithm of Theorem 3 without perturbations.

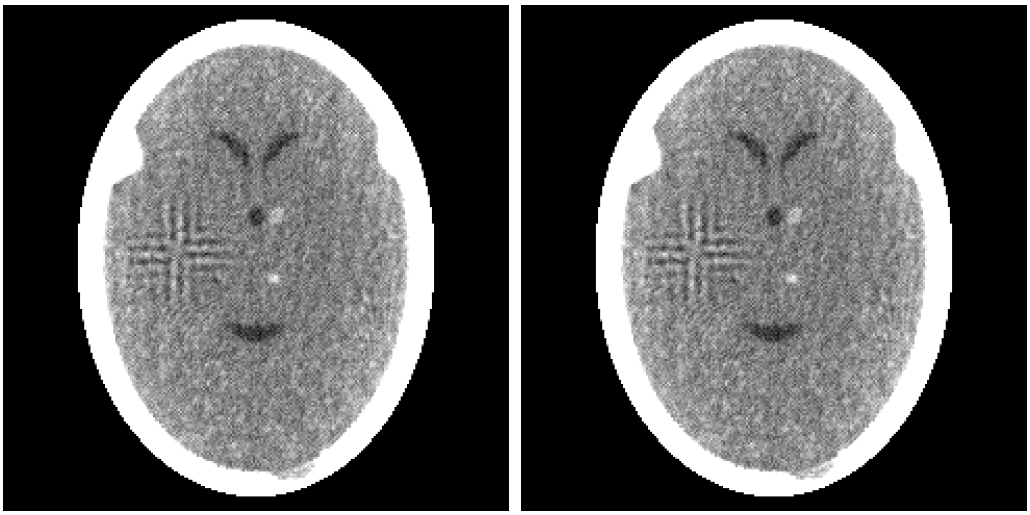
	$\mathbf{S}+p$ and $\mathbf{S}-p$	$\mathbf{H}^2+p$ and $\mathbf{H}^2-p$	$\mathbf{S}+p$ and $\mathbf{H}^2+p$
HITR P-value	1.59×10^{-9}	5.70×10^{-10}	3.76×10^{-3}
IROI P-value	1.52×10^{-14}	9.49×10^{-15}	5.77×10^{-3}

	$\mathbf{S}-p$ and $\mathbf{H}^2-p$	$\mathbf{S}+p$ and $\mathbf{H}^2-p$	$\mathbf{H}^2+p$ and $\mathbf{S}-p$
HITR P-value	3.36×10^{-2}	2.70×10^{-10}	5.38×10^{-9}
IROI P-value	1.75×10^{-3}	1.35×10^{-14}	1.25×10^{-14}

Table 3.6: Results of the statistical hypothesis testing evaluations comparing the algorithms of Theorems 3 and 4. The P-values of the two FOMs indicate the statistical significance at which the null hypothesis that the two algorithms are equally helpful for the task of detecting low contrast tumors in the brain can be rejected in favor of the alternative hypothesis that the one listed first is more helpful. In the first row, $\mathbf{S}$ and $\mathbf{H}^2$ indicate the use of the symmetric and the nonsymmetric operator in the algorithm, respectively, $+p$ stand for TV-superiorization and $-p$ for no perturbations.

our desire to investigate the efficacy of algorithms based on Theorem 4. Since the computational cost of applying $\mathbf{S}$ (see (2.53)) is approximately twice of that of applying $\mathbf{H}$ (see (2.50)), in this set of studies we used $\mathbf{H}^2$ (i.e., two consecutive applications of $\mathbf{H}$) as the operator in the nonsymmetric case and selected $\lambda_u = \lambda = 0.075$ for both the symmetric and the nonsymmetric algorithms. In Table 3.6 we

report on the results of pairwise evaluations of algorithms that are either symmetric or nonsymmetric and that either use or do not use TV-superiorizing perturbations. Summarizing this table, if one algorithm is TV-superiorizing and another one is not, then for both FOMs we can reject the null hypothesis that the two algorithms are equally efficacious with extreme significance in favor of the alternative that the one that superiorizes TV is more helpful for the task at hand. On the other hand, in those cases where either both algorithms are TV-superiorizing or neither is, we can reject the null hypothesis that the two algorithms are equally efficacious in favor of the alternative that the one that uses the symmetric operator is more efficacious, but the significance is now less (the P-values are larger). In other words, we can be extremely certain TV-superiorizing perturbations improve efficacy, but we cannot be quite that certain that the observed improvements due to the use of the symmetric operator is not accidental. As an illustration we give in Figure 3.16 reconstructions produced by TV-superiorizing algorithms with the symmetric and with the nonsymmetric operator, respectively.



(a) (b)

Figure 3.16: Reconstructions from the same realistic projection data set for one of our random phantoms (similar to the ones in Figure 3.3) by two TV-superiorizing algorithms using (a) the symmetric operator $\mathbf{S}$ and (b) the nonsymmetric operator $\mathbf{H}^2$.

3.3.4 Comparison with optimization

An important point that is illustrated by the experiments in the previous subsections is that, from the practical point of view, TV-superiorization is as useful as TV-optimization. This is because a realistic phantom, such as the one in Figure 3.2, is unlikely to be TV-minimizing subject to the constraints provided by the measurements. The TV value of our phantom is 450.594, which is larger than that of many of the TV-superiorizing reconstructions we provided earlier. While an optimization method should be able to find an image with a lower TV value, there is no practical point for doing that. Since the underlying aim of what we are doing is to estimate the phantom from the data, producing an image whose TV value is further from the TV value of the phantom than that of our superiorized reconstructions is unlikely to be helpful towards achieving this aim. Next we provide two comparisons performed for both the AP and BIP methods.

3.3.4.1 AP TV-Superiorizing Vs. TV-optimization

We report on a comparison of the TV-superiorizing AP method (from Subsection 3.3.1) with an adaptation of a TV-minimizing version of Algorithm 6 in [22]. The most essential part of our adaptation was that where the algorithm called for a projection onto C, in our implementation we approximated this by doing a single cycle of consecutive projections onto $C_1, ..., C_I$. We applied both algorithms to the UD dataset for 82 views with hyperplane constraints, starting the processes with the zero vector and stopping them when $\text{Res}(x^k) < \varepsilon = 0.005$. For the TV-minimizing algorithm we obtained an output for which the TV value is 471.3, which is higher than the TV value of the TV-superiorizing amalgamated method (457.2). Also, the time required to get to termination was just over four times longer for the algorithm based on [22] than for the amalgamated TV-superiorizing algorithm. All the relevant numbers for this comparison, including the amalgamated algorithm without

Method	norm	TV	distance	time (m)
ART (no perturbations)	46.04	1300.8	3.644	112.6
AP TV-superiorizing	46.19	457.2	0.223	14.6
TV-optimization (based on [22])	46.17	471.3	0.390	59.9

Table 3.7: Numerical values for the outputs of AP TV-superiorization and TV-optimization algorithm based on [22])

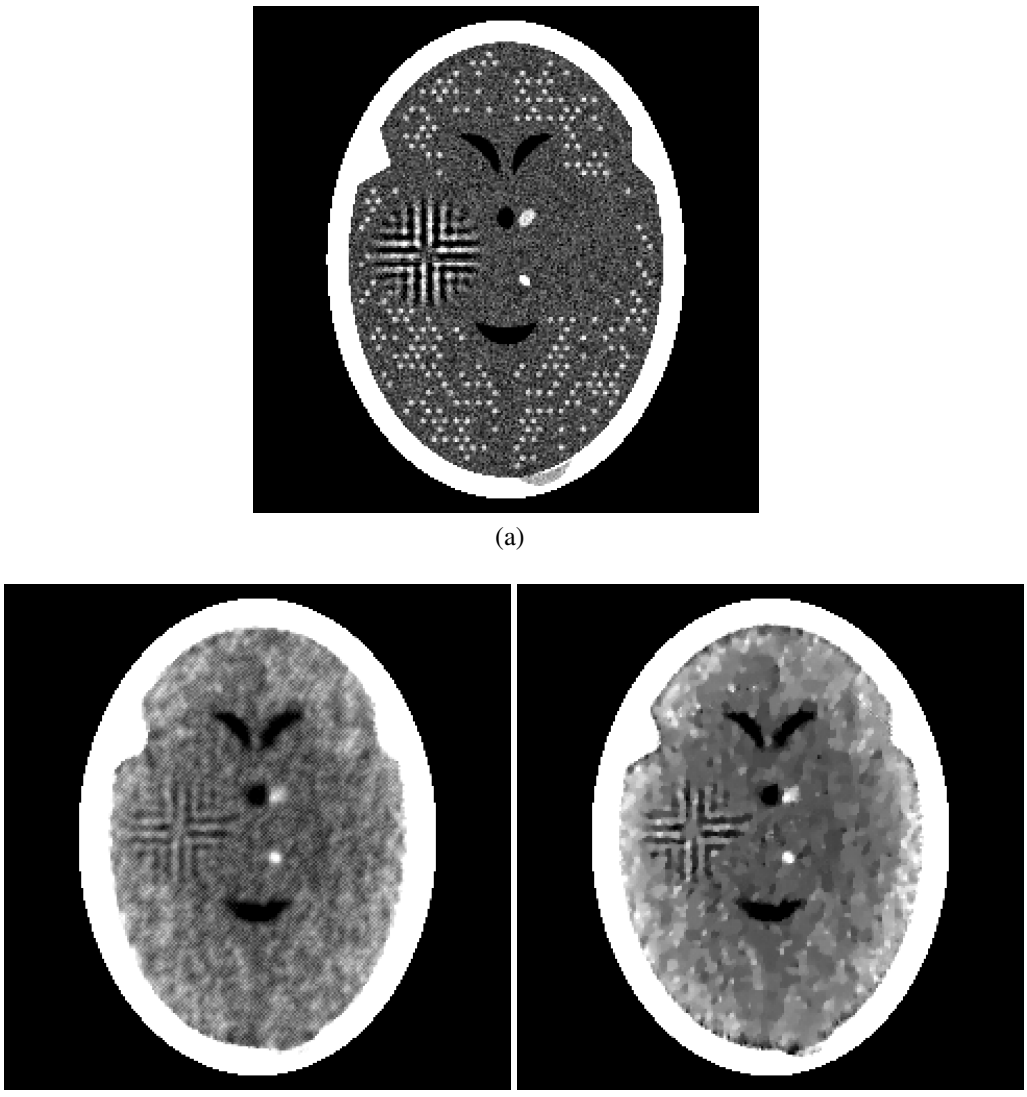
perturbations (i.e., ART), are reported in Table 3.7.

3.3.4.2 BIP TV-Superiorizing Vs. TV-optimization

In our second comparison with an optimization method, we compared the accelerated BIP TV-superiorization algorithm from Subsection 2.3.2 with a TV-optimization algorithm using TwIST [38] and Split Bregman [31] as the substep. In setting up the problem for this TV-optimization algorithm we required a solution to the problem: find an x that specifies an image with a small TV subject to $\text{Res}(x) < 5.5$ (the Res of the phantom we used had Res=5.462 and so requiring such a Res makes the digitization of the head-phantom feasible). By a "small TV" we mean a value that is not larger than 454.036 which is the TV of the digitized head-phantom.

For this experiment we chose a phantom similar to the ones presented in Figure 3.3, presented here in Figure 3.17(a), to be reconstructed with OD dataset for 360 views using hyperplane constraints. (We again set up all the measurements taken from a particular direction to be associated with the uth block, with $U = 360$.) Our particular choice of parameters for this experiment was $\lambda_u = \lambda = 0.031$ (in (2.49)) and in the superiorization algorithm, instead of multiplying the size of beta at steps 9 and 10 of the pseudocode in (Subsection 3.1) by 0.5, we multiplied it by 0.7 and started with $\beta_{-1} = 0.7$.

Figure 3.17(b) shows the result of the reconstruction using the TV-superiorizing BIP algorithm and Figure 3.17(c) has the resulted image of the TwIST and Split Bregman TV-optimization method, both (along with the phantom in (a)) displayed



(a)

(b) (c)

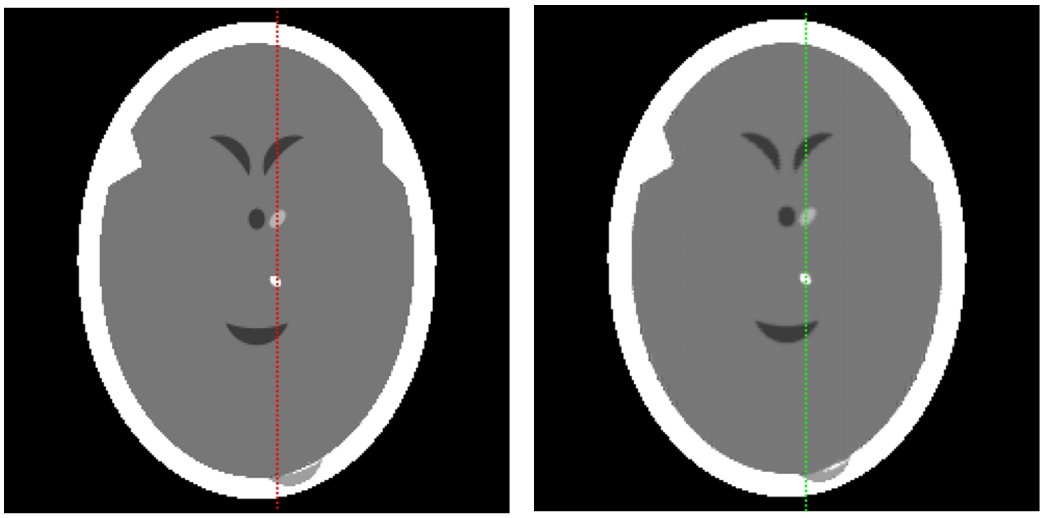
Figure 3.17: Reconstructions of BIP TV-superiorizing and TwIST with Split Bregman TV-optimization. (a) The phantom for which data were collected. (b) Reconstructions using BIP TV-superiorizing algorithm. (c) Reconstructions using TwIST with Split Bregman optimization algorithm. All images are displayed at a window of $[0.2085 - 0.21375]$.

at a narrow window of $[0.2085 - 0.21375]$ (narrower than the display window used in all our previous experiments). The TV value of these images are 451.5 and 443.9 for the image in 3.17(b) and 3.17(c), respectively. While from a mathematical perspective the two algorithms provide a solution to the problem above, the times to produce the two images are substantially different; on an Intel Pentium core two duo

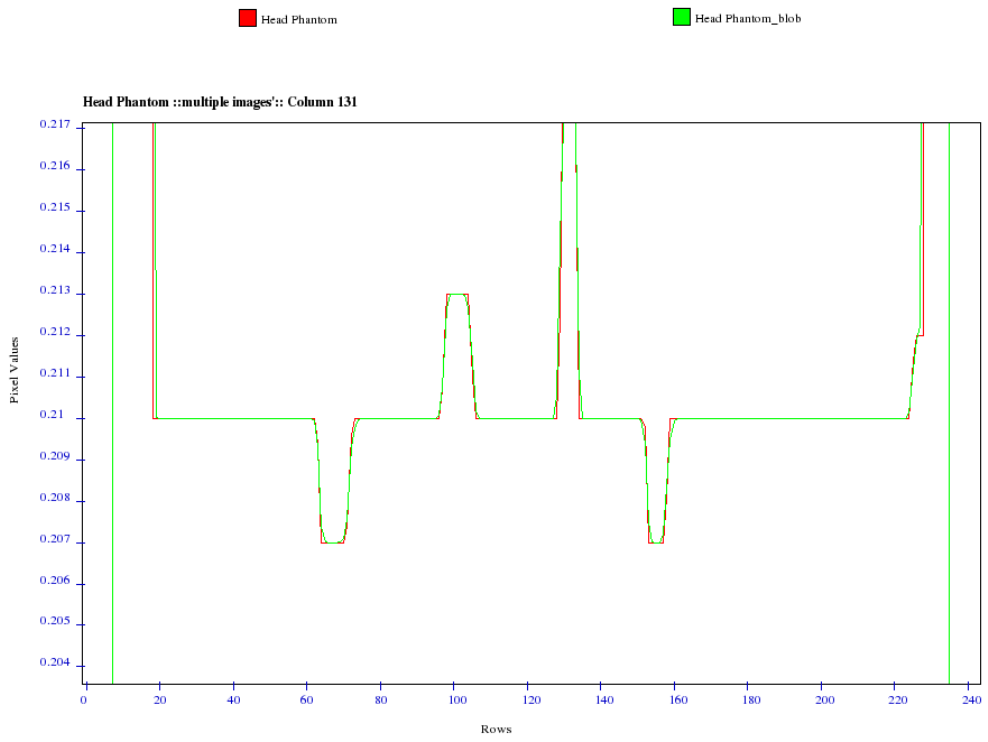
2.8 GHz (using a single CPU) with 4GB memory the image in Figure 3.17(a) took less than 7 seconds to be reconstructed by the TV-superiorizing algorithm versus slightly over 28 seconds for the image in Figure 3.17(b).

3.3.5 Superiorization using blob basis functions

Generalized Kaiser-Bessel window functions, also known by the more simple name *blobs,* were first introduced to the field of image reconstruction and image representation by R. M. Lewitt [45]. He proposed the use of these spherically symmetric basis functions for discrete image representation in a multidimensional space. Blobs are suitable alternative basis functions for both visualization and reconstruction to the standard pixel basis functions (in 2D) and the voxel basis functions (in 3D). It has been repeatedly shown in the literature that the use of blobs as basis functions can produce superior results for different types of applications, see for example [40, 47, 48, 50]. In a reconstruction problem, the quality of the images produced using blobs as the basis is superior to that reconstructed using pixels. Blob-based reconstructions however, suffer from a noticeable drawback: oscillations appear in neighborhood of sharp density changes in the object that we wish to reconstruct. For example, in the head phantom in Figure 3.18(a) (which is similar to the head phantom of Figure 3.2 but without inhomogeneity and without the ghost tumor), this occurs in the reconstruction between the skull and the brain. This undesirable artifact is not due to the nonexistence of a nonoscillatory approximation to the head phantom by a linear combination of a finite number of blobs (which are very smooth functions), this is illustrated in Figure 3.18(b) using a blob approximation (which is not a reconstruction). In Figure 3.19 we include the reconstructions when the basis are pixels and blobs for ART starting from a point with an average density values assigned to all the pixels of x^0. Note the superiority of the reconstruction when blobs are used as the basis in Figure 3.19(b). We are interested in the answer

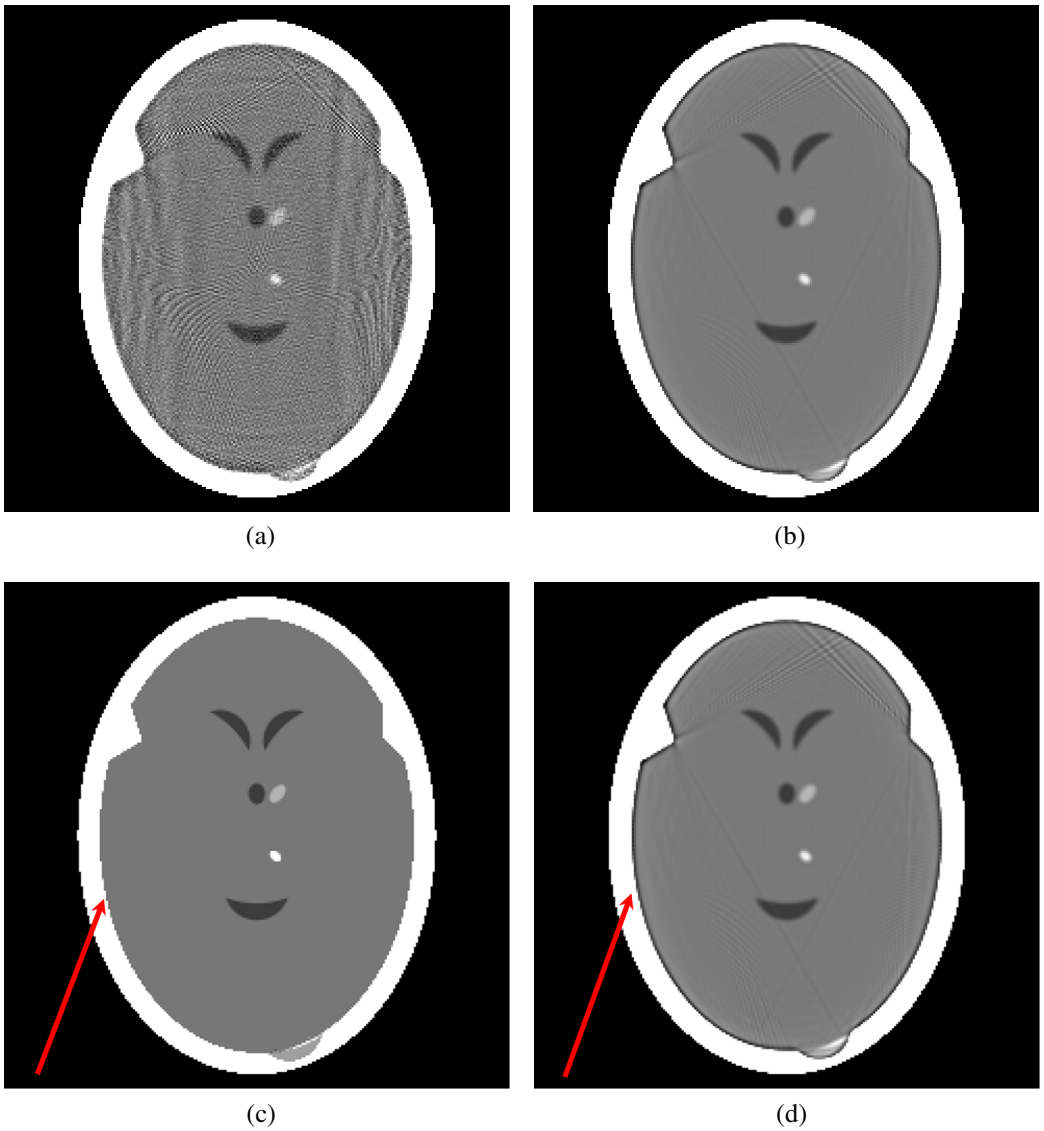


(a) (b)



(c)

Figure 3.18: (a) A digitized head phantom similar to the one in Figure 3.2 without the ghost tumor and without inhomogeneity. (b) A blob based approximation to the digitized head phantom, illustrating that there exist a linear combination of blobs that produces a desirable image. (c) Graphs of the values in the 131^{st} column of the digitized head phantom in (a) (colored red) and of the blob approximation (colored green).



(a) (b)

(c) (d)

Figure 3.19: Reconstructions of the head phantom from Figure 3.18(a) using pixels and blobs as basis functions. (a) Reconstruction using pixels as the basis functions. (b) Reconstruction using blobs as the basis functions. (c) The head phantom from Figure 3.18(a) with a red arrow marking a clean transition between the skull and the brain. (d) The reconstructed image from (b) using blobs as basis functions with a red arrow indicating the appearance of the artifact seen as a black ring.

to the following question: can a superiorization algorithm help to reduce the black-ring artifact (indicated with a red arrow in Figure 3.19(d)) when the reconstruction uses blobs as the basis? Unfortunately, our findings does not answer this question positively; we provide the details next.

3.3.5.1 Blob properties

Blobs have the following form:

$$b_{(m,a,\alpha)}(r) = \begin{cases} \frac{1}{I_m(\alpha)} \left[\sqrt{1-\left(\frac{r}{a}\right)^2} \right]^m I_m \left[\alpha \sqrt{1-\left(\frac{r}{a}\right)^2} \right], & \text{if } 0 \le r \le a, \\ 0, & \text{otherwise,} \end{cases} \tag{3.21}$$

where $I_m(\cdot)$ denotes the modified Bessel function of the first kind of order m, a stands for the nonnegative radius of the blob and α is a nonnegative real number that controls the blob's taper (the shape of the blob). The parameter m is a nonnegative integer that controls the continuity of the blob (the number of times it is continuously differentiable at $r = a$). More specifically, if $m = 0$ then the function will have a discontinuity at $r = a$, if $m > 0$ then it will be continuous at that point, and if $m > 1$ then the first derivative of the blob will be continuous. In general, the kth derivative of a blob is continuous if $m > k$. The blob has the value zero for all $r > a$, where r is the radial distance from the center of the blob. It can be seen from (3.21) that a blob is controlled by the choice of its parameters.

3.3.5.2 Image representation using blobs

The series expansion approach to image reconstruction from projections assumes that the image can be represented as

$$\sum_{n=1}^{N} c_n b_n(x,y), \tag{3.22}$$

where, for $1 \le n \le N$, $b_n(x,y)$ is the value of the nth blob basis function at the grid point (x,y) and c_n is its coefficient for the image that we are trying to reconstruct (the recommended grid for blobs is the hexagonal grid as oppose to a rectangular grid for pixels [51, 52]).

3.3.5.3 Implementation of superiorization using blobs

The implementation of the superiorization algorithm described previously in Subsection 3.1 require several changes to fit to a blob-based reconstruction. If c^k is the blob-based reconstruction image at the kth iteration, then the following pseudocode is the equivalent to the pixel-based superiorization algorithm described in Subsection 3.1 for the iterative step going from c^k to c^{k+1}:

1: logic = true;
2: while (logic)
3: $y = c^k + \beta v^k$
4: if ($\phi(z) \leq \phi(x^k)$)
5: then
6: $c^{k+1} = \mathbf{O}y$
7: if ($\text{Res}(c^{k+1}) < \text{Res}(c^k)$)
8: then logic = false
9: else $\beta = \beta/2$
10: else $\beta = \beta/2$

In step 3 the algorithm steers the process in the negative subgradient direction of ϕ at c^k and produces the point y. In step 4 it checks if the value of ϕ of the corresponding pixel image of y (denoted by z) is less than or equal to the pixel image of c^k (denoted by x^k). In step 6 the algorithmic operator is applied to y with c^{k+1} as the resulting point. In step 7 the algorithm checks the Res of the newly obtained point and compares it to Res of c^k. The algorithm exits if $\text{Res}(c^k) < \varepsilon$, for some small positive user-specified ε. Based on results reported at conferences on the usefulness of TV-minimization in avoiding overshoots when using harmonic functions in image processing, it was our hope that TV-superiorization might be helpful in reducing the undesirable black ring inside the skull. In our implementation, similar to what we did in all the experiments so far, we are basing the TV on the pixel im-

age at the kth step (step 4 of the pseudocode), whereas the subgradient is calculated based on the blob image (in step 3). Consider a $G \times H$ rectangular grid for the pixel image q, whose pixels are denoted by $q_{g,h}$ $(1 \leq g \leq G, 1 \leq h \leq H)$. Using (3.22) above, let $f(c,g,h)$ and $t(c,g,h)$ be defined as

$$f(c,g,h) = \sum_{n=1}^{N} c_n \left(b_n \left(x_{g,h+1}, y_{g,h+1} \right) - b_n \left(x_{g,h}, y_{g,h} \right) \right), \tag{3.23}$$

and

$$t(c,g,h) = \sum_{n=1}^{N} c_n \left(b_n \left(x_{g+1,h}, y_{g+1,h} \right) - b_n \left(x_{g,h}, y_{g,h} \right) \right), \tag{3.24}$$

where $b_n \left(x_{g,h}, y_{g,h} \right)$ denotes the value of the nth blob at the coordinate of the center of the (g,h)th pixel. Then the TV of a blob-based image c (with N hexagonal grid points) is given by

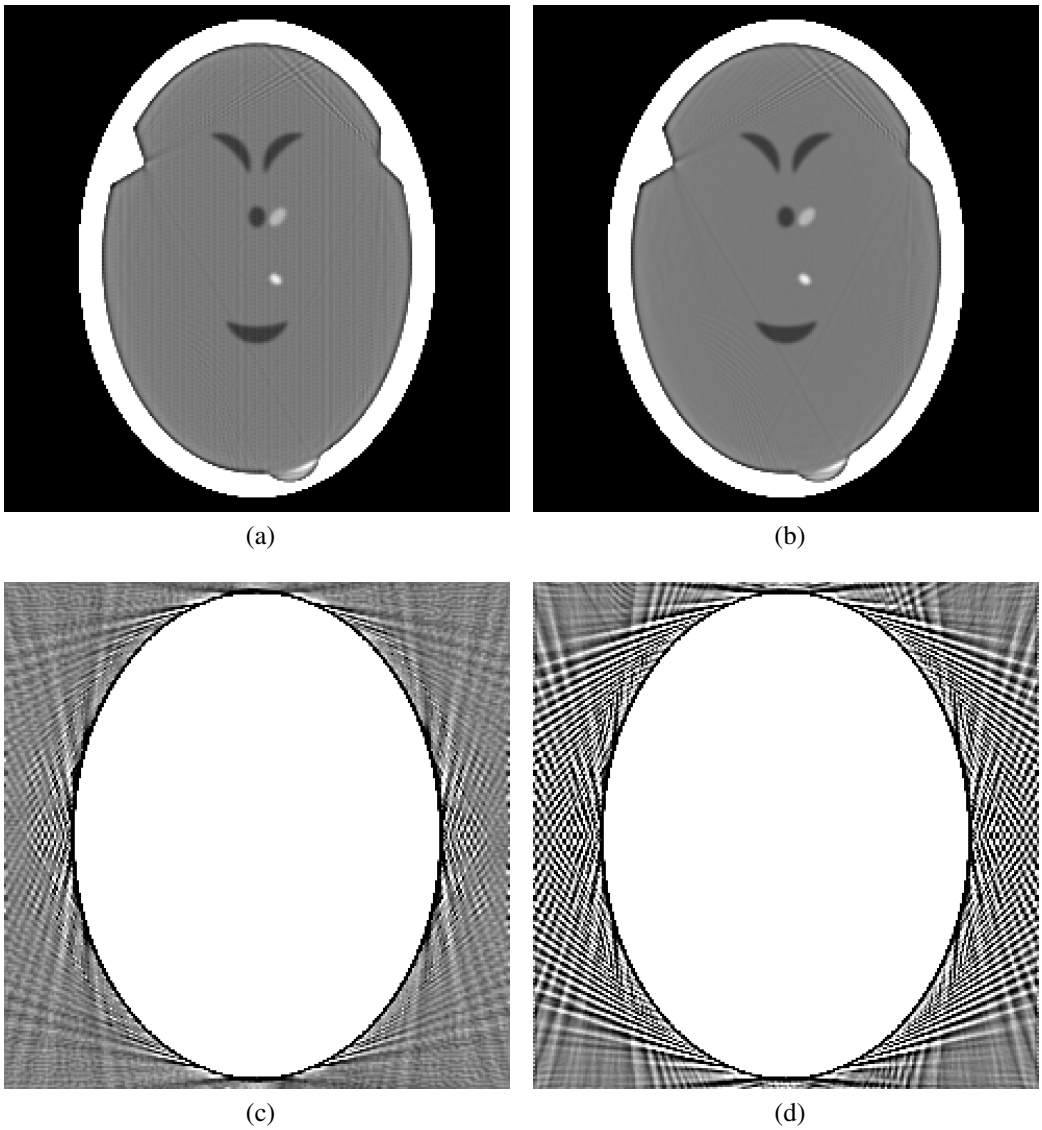
$$TV_b(c) = \sum_{g=1}^{G-1} \sum_{h=1}^{H-1} \sqrt{(f(c,g,h))^2 + (t(c,g,h))^2}. \tag{3.25}$$

For $1 \leq m \leq N$, the mth coordinate of the subgradient of TV_b at c is

$$\begin{aligned} \frac{\partial TV_b}{\partial c_m}(c) = & \sum_{g=1}^{G-1} \sum_{h=1}^{H-1} \frac{1}{\sqrt{(f(c,g,h))^2 + (t(c,g,h))^2}} \\ & \left[f(c,g,h) \left(b_m \left(x_{g+1,h}, y_{g+1,h} \right) - b_m \left(x_{g,h}, y_{g,h} \right) \right) + \right. \\ & \left. t(c,g,h) \left(b_m \left(x_{g,h+1}, y_{g,h+1} \right) - b_m \left(x_{g,h}, y_{g,h} \right) \right) \right]. \end{aligned} \tag{3.26}$$

3.3.5.4 Experiments of superiorization with blobs

Consider the phantom from Figure 3.18(a) and the reconstructions of ART using pixels and blobs as the basis in Figure 3.19(a) and (b) respectively. The dataset on which the two algorithms reconstructed was the OD with 360 views from Subsubsection 3.2.3.2 using hyperplanes constraints as in (3.15). Note that in this case the phantom does not lie in the intersection of all the constraints and its Res=3.38.



(a) (b)

(c) (d)

Figure 3.20: Reconstructions of the head phantom from Figure 3.18(a) using (a) TV-superiorizing with blobs and (b) ART without perturbations. The corresponding images displayed with the narrow window of $[-0.001$ to $0.001]$ show that the TV-superiorizing algorithm in (c) reduces the oscillations outside the skull as compared to the ART reconstruction in (d).

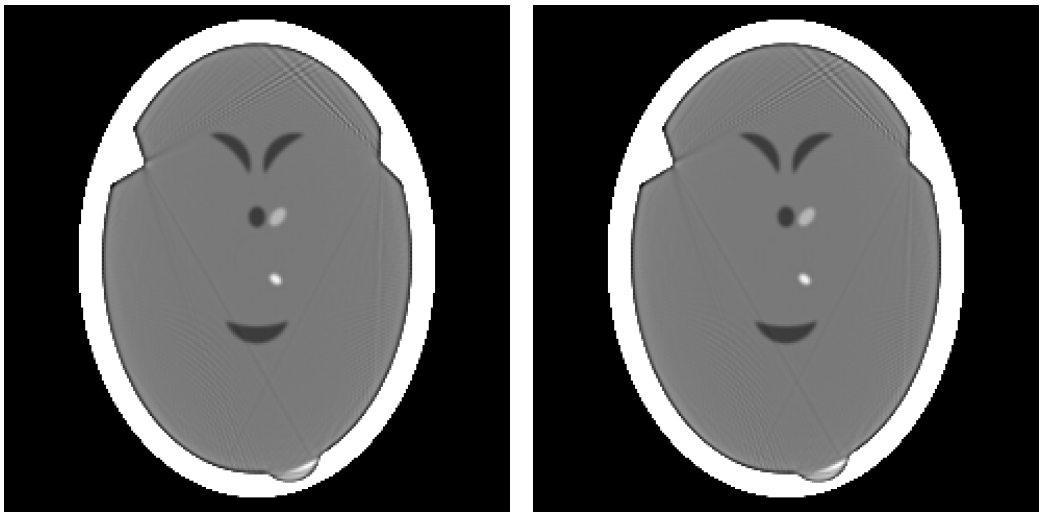
Since the data are noisy, we used $\alpha = 0.05$ in (1.7). We implemented the TV-superiorizing algorithm using blobs as was described in the previous subsection and ran the cyclic projection method with perturbations, starting from the average point for x^0, and stopping at $\text{Res}\left(c^k\right)<\varepsilon = 3.38$. In Figure 3.20(a) we give the resulting image of the reconstruction by TV-superiorizing with blobs and in Figure

3.20(b) we provide the image of the reconstruction of ART with blobs without perturbations (same as Figure 3.19(b)). Mathematically speaking, the value of the TV functional in the superiorization case is indeed superior with 444.4 for the image in 3.20(a) versus 473.2 for ART without perturbations for the image in 3.20(b) (as a base to compare, the TV of the phantom is 408.9), however the quality of the image is not better and in particular the black ring between the skull and the brain did not disappear. Examining the two images at the very narrow window of -0.001 to 0.001 reveals that much of the reduction of the TV is due to the reduced oscillations in the air outside the skull, which is medically not useful.

As a second experiment we examined the TV-superiorization of the cyclic projection method when both hyperplanes and halfspace constraints are present. Since we know that the digitized phantom does not have negative pixel values, we obtain constraints of the form

$$pixel_j = \sum_{n=1}^{N} c_n b_n\left(x_j, y_j\right) \geq 0, \tag{3.27}$$

for $1 \leq j \leq J$. In other words, the linear combination of the blobs that contribute to the jth pixel has to satisfy the inequality in (3.27). Since one pixel influences many blobs, we incorporated a second relaxation parameter, $\alpha' = 0.5$, to slow down the process of fitting these sets of inequalities. With α' fixed we performed another set of auxiliary experiments to pick an appropriate relaxation parameter α. In Figure 3.21(a) we give the resulting image of the TV-superiorizing algorithm and in Figure 3.21(b) we provide the reconstructed image when no perturbations are present. Although the images look very similar they are different and behave as they advertised: the TV of the image with TV-superiorization is 437.5 and the one without has TV of 444.2. Unfortunately, even after this incorporation of the nonnegativity of pixel values, the oscillations inside the skull have remained and manifest them-



(a) (b)

Figure 3.21: Reconstructions of the head phantom from Figure 3.18(a) using hyperplanes and nonnegativity constraints. (a) TV-superiorizing with blobs reconstruction. (b) ART with blobs without perturbations reconstruction. Although the two look very similar, the TV value of the one obtained by TV-superiorization is lower than that obtained by ART without perturbations.

selves in the two images as black rings between the skull and the brain.

Chapter 4

Automatic superiorization of iterative methods

The work presented so far on superiorization has been very specific: superiorization has been applied to certain algorithms for certain tasks. In this chapter we introduce new terminology and the rudiments of a theory that allow us to superiorize automatically any iterative algorithm for a problem set from a very general class of problem sets. It is shown that under reasonable conditions such superiorized algorithms are guaranteed to halt. In fact, the method of superiorization introduced in the current chapter is slightly different from the specific superiorized algorithms presented in Chapter 3; in particular, in order to guarantee that those algorithms halt we needed more complex termination conditions than what is needed for the algorithms produced by the method of the current chapter. This new method of superiorization is illustrated at the end of the chapter by applying it to an AP and to a BIP algorithm. These illustrations are similar to the ones presented in the previous chapter for image reconstruction from projections and demonstrate an important practical aspect of our work, which is that the output of superiorization can be as useful as anything that may be obtained by full optimization because the value of the objective func-

tion for the image provided by superiorization is already smaller than that for the true image that we are trying to reconstruct. All this is explained in greater detail below.

4.1 Specifications

The superiorization principle relies on the bounded perturbation resilience of algorithms. Therefore we define this notion next in a general setting within $\mathbb{R}^J$. We introduce the notion of a *problem structure* $\langle \mathbb{T}, \mathscr{P}r \rangle$, where $\mathbb{T}$ is a nonempty *problem set* and $\mathscr{P}r$ is a function on $\mathbb{T}$ such that, for all $T \in \mathbb{T}$, $\mathscr{P}r_T : \mathbb{R}^J \to \mathbb{R}_+$, where $\mathbb{R}_+$ is the set of nonnegative real numbers. Intuitively we think of $\mathscr{P}r_T(x)$ as a measure of how "far" x is from being a solution of T. In fact, we call x a *solution* of T if $\mathscr{P}r_T(x) = 0$.

For example, for the convex feasibility problem (1.1)

$$\mathbb{T} = \{\{C_1, \ldots, C_I\} | \quad I \text{ is a positive integer and, for } 1 \leq i \leq I, \\ C_i \text{ is a closed convex subset of } \mathbb{R}^J\} \tag{4.1}$$

and

$$\mathscr{P}r_{\{C_1,\ldots,C_I\}}(x) = \sqrt{\sum_{i=1}^{I} \left(d(x, C_i)\right)^2}, \tag{4.2}$$

where $d(x, C_i)$ is the Euclidean distance of x from the set C_i. Clearly, in this case x is a solution of $\{C_1, \ldots, C_I\}$ as defined in the previous paragraph if, and only if, $x \in C$ as defined in (1.1).

Definition 1 An *algorithm* $\mathbf{P}$ for $\langle \mathbb{T}, \mathscr{P}r \rangle$ assigns to each $T \in \mathbb{T}$ an algorithmic operator $\mathbf{P}_T : \mathbb{R}^J \to \mathbb{R}^J$. $\mathbf{P}$ is said to be *bounded perturbations resilient* if, for all $T \in \mathbb{T}$, the following is the case: if the sequence $\left((\mathbf{P}_T)^k x\right)_{k=0}^{\infty}$ converges to a solution of T for all $x \in \mathbb{R}^J$, then any sequence $\left(x^k\right)_{k=0}^{\infty}$ of points in $\mathbb{R}^J$ also converges

to a solution of T provided that, for all $k \geq 0$,

$$x^{k+1} = \mathbf{P}_T \left(x^k + \beta_k v^k \right), \tag{4.3}$$

where $\beta_k v^k$ are bounded perturbations, meaning that β_k are real nonnegative numbers such that $\sum_{k=0}^{\infty} \beta_k < \infty$ and the sequence $\left(v^k\right)_{k=0}^{\infty}$ is bounded.

We give next specific instances of bounded perturbations resilient algorithms for solving the convex feasibility problem as in (4.1) and (4.2), from the classes of AP and BIP methods. For our discussion of AP methods in this more general context, recall the discussion and notation from Subsection 1.2.1. Associate with each $\{C_1, \ldots, C_I\} \in \mathbb{T}$ an $\left(\Omega_{\{C_1,\ldots,C_I\}}, \omega_{\{C_1,\ldots,C_I\}}\right)$, which is an amalgamator for the problem $\{C_1, \ldots, C_I\}$. For each such association of amalgamators with problems, there is an algorithm $\mathbf{P}$ for $\langle \mathbb{T}, \mathscr{P}r \rangle$, defined by: for any $\{C_1, \ldots, C_I\} \in \mathbb{T}$,

$$\mathbf{P}_{\{\mathbf{C_1},\ldots,\mathbf{C_I}\}} x = \sum_{t \in \Omega_{\{C_1,\ldots,C_I\}}} w_{\{C_1,\ldots,C_I\}}(t) P[t] x; \tag{4.4}$$

this should be compared with (1.3).

Corollary 1 *For any association of amalgamators with problems, the algorithm* $\mathbf{P}$ *defined by (4.4) is bounded perturbations resilient.*

Proof Assume that for $T = \{C_1, \ldots, C_I\}$ the sequence $\left(\left(\mathbf{P}_T\right)^k x \right)_{k=0}^{\infty}$ converges to a solution of T for all $x \in \mathbb{R}^J$. This implies, in particular, that C of (1.1) is nonempty. By Definition 1, we need to show that any sequence $\left(x^k\right)_{k=0}^{\infty}$ of points in $\mathbb{R}^J$ also converges to a solution of T provided that, for all $k \geq 0$, (4.3) is satisfied when the $\beta_k v^k$ are bounded perturbations. Under our assumptions, this follows from Theorem 1. ■

For our discussion of BIP methods in this more general context, recall the discussion and notation from Subsection 2.3.1. Associate with each $\{C_1, \ldots, C_I\} \in \mathbb{T}$ a

positive integer $U_{\{C_1,\ldots,C_I\}}$ of blocks (ordered subsets of $\{1,\ldots,I\}$) whose union is $\{1,\ldots,I\}$. For each such association of blocks with problems, define the algorithm $\mathbf{Q}$ for $\langle \mathbb{T}, \mathscr{P}r \rangle$ by: for any $\{C_1,\ldots,C_I\} \in \mathbb{T}$, $\mathbf{Q}_{\{C_1,\ldots,C_I\}}$ is equal to the right hand side of (2.27).

Corollary 2 *For each association of blocks with problems, the resulting algorithm* $\mathbf{Q}$ *is bounded perturbations resilient.*

Proof Replace in the proof of Corollary 1 $\mathbf{P}$ by $\mathbf{Q}$ and Theorem 1 by the special case of Theorem 2 when $\lambda = 1$. ■

For a problem structure $\langle \mathbb{T}, \mathscr{P}r \rangle$, $T \in \mathbb{T}$, $\varepsilon \in \mathbb{R}_{++}$ and a sequence $S = \left(x^k\right)_{k=0}^{\infty}$ of points in $\mathbb{R}^J$, we use $O(T,\varepsilon,S)$ to denote the $x \in \mathbb{R}^J$ that has the the following properties: $\mathscr{P}r_T(x) \leq \varepsilon$ and there is a nonnegative integer K such that $x^K = x$ and, for all nonnegative integers $\ell < K$, $\mathscr{P}r_T\left(x^\ell\right) > \varepsilon$. Clearly, if there is such an x, then it is unique. If there is no such x, then we say that $O(T,\varepsilon,S)$ is undefined. The intuition behind this definition is the following: if we think of S as the (infinite) sequence of points that is produced by an algorithm (intended for the problem T) without a termination criterion, then $O(T,\varepsilon,S)$ is the output produced by that algorithm when we add to it instructions that make it terminate as soon as it reaches a point at which the value of $\mathscr{P}r_T$ is not greater than ε. The following result is obvious.

Lemma 3 *If* $\mathscr{P}r_T$ *is continuous and the sequence* S *converges to a solution of* T*, then* $O(T,\varepsilon,S)$ *is defined and* $\mathscr{P}r_T\left(O(T,\varepsilon,S)\right) \leq \varepsilon$.

Given an algorithm $\mathbf{P}$ for a problem structure $\langle \mathbb{T}, \mathscr{P}r \rangle$, a $T \in \mathbb{T}$ and an $\bar{x} \in \mathbb{R}^J$, let $R(T,\bar{x}) = \left(\left(\mathbf{P}_T\right)^k \bar{x}\right)_{k=0}^{\infty}$. For a function $\phi : \mathbb{R}^J \to \mathbb{R}$, the *superiorization methodology* should provide us with an algorithm that produces a sequence $S(T,\bar{x},\phi) = \left(x^k\right)_{k=0}^{\infty}$, such that, for any $\varepsilon \in \mathbb{R}_{++}$ and $\bar{x} \in \mathbb{R}^J$ for which $\mathscr{P}r_T(\bar{x}) > \varepsilon$ and

$O(T,\varepsilon,R(T,\bar{x}))$ is defined, $O(T,\varepsilon,S(T,\bar{x},\phi))$ is also defined and

$$\phi\left(O(T,\varepsilon,S(T,\bar{x},\phi))\right) < \phi\left(O(T,\varepsilon,R(T,\bar{x}))\right). \tag{4.5}$$

This is of course too ambitious in its full generality and so here we analyze only a special case, but one that is still quite general. We now list our assumptions for the special case for which we discuss details of the superiorization methodology.

Assumptions

1. $\langle \mathbb{T}, \mathscr{P}r \rangle$ is a problem structure such that $\mathscr{P}r_T$ is continuous for all $T \in \mathbb{T}$.

2. $\mathbf{P}$ is a bounded perturbation resilient algorithm for $\langle \mathbb{T}, \mathscr{P}r \rangle$ such that, for all $T \in \mathbb{T}$, $\mathbf{P}_T$ is continuous and, if x is not a solution of T, then $\mathscr{P}r_T\left(\mathbf{P}_T x\right) < \mathscr{P}r_T\left(x\right)$.

3. $\phi : \mathbb{R}^J \to \mathbb{R}$ is an everywhere real-valued convex function, defined on the whole space.

Under these Assumptions, we now describe the algorithm to produce the sequence $S(T,\bar{x},\phi) = \left(x^k\right)_{k=0}^{\infty}$ and present and prove Theorem 5 below.

The algorithm assumes that we have available a summable sequence $(\gamma_\ell)_{\ell=0}^{\infty}$ of positive real numbers. It is easy to generate such sequences; e.g., we can use $\gamma_\ell = a^\ell$, where $0 < a < 1$. The algorithm generates, simultaneously with the sequence $\left(x^k\right)_{k=0}^{\infty}$, sequences $\left(v^k\right)_{k=0}^{\infty}$ and $(\beta_k)_{k=0}^{\infty}$. The latter will be generated as a subsequence of $(\gamma_\ell)_{\ell=0}^{\infty}$. Clearly, the resulting sequence $(\beta_k)_{k=0}^{\infty}$ of positive real numbers will be summable. We first specify the algorithm and then discuss it. The algorithm depends on the specified $\bar{x}$, ϕ, $(\gamma_\ell)_{\ell=0}^{\infty}$, $\mathscr{P}r_T$ and $\mathbf{P}_T$. It makes use of a logical variable called *continue* and also of the concept of a subgradient of the convex function ϕ. $\|\cdot\|$ is the Euclidean norm.

Superiorized Version of Algorithm P

1. **set** $k = 0$
2. **set** $x^k = \bar{x}$
3. **set** $\ell = 0$
4. **repeat**
5. **set** g to a subgradient of ϕ at x^k
6. **if** $\|g\| > 0$
7. **then set** $v^k = -g/\|g\|$
8. **else set** $v^k = g$
9. **set** *continue* = *true*
10. **while** *continue*
11. **set** $\beta_k = \gamma_\ell$
12. **set** $y = x^k + \beta_k v^k$
13. **if** $\phi(y) \leq \phi\left(x^k\right)$ **and** $\mathscr{P}r_T(\mathbf{P}_T y) < \mathscr{P}r_T\left(x^k\right)$ **then**
14. **set** $x^{k+1} = \mathbf{P}_T y$
15. **set** *continue* = *false*
16. **set** $\ell = \ell + 1$
17. **set** $k = k + 1$

Sometimes it is useful to emphasize the function ϕ for which we are superiorizing, in which case we refer to the algorithm above as the *ϕ-superiorized version of algorithm* **P**. It is important to bear in mind that the sequence S produced by the algorithm depends also on the initial point $\bar{x}$, the selection of the subgradient in Line (5) of the algorithm, the summable sequence $(\gamma_\ell)_{\ell=0}^{\infty}$, and the problem T. In addition, the output $O(T,\varepsilon,S)$ of the algorithm depends on the stopping criterion ε.

Theorem 5 *Under the Assumptions listed above, the Superiorized Version of Algorithm* **P** *will produce a sequence $S(T,\bar{x},\phi)$ of points in $\mathbb{R}^J$ that either contains a solution of T or is infinite. In the latter case, if the sequence $\left((\mathbf{P}_T)^k x\right)_{k=0}^{\infty}$ converges to a solution of T for all $x \in \mathbb{R}^J$, then, for any $\varepsilon \in \mathbb{R}_{++}$, $O(T,\varepsilon,S(T,\bar{x},\phi))$ is defined and $\mathscr{P}r_T\left(O(T,\varepsilon,S(T,\bar{x},\phi))\right) \leq \varepsilon$.*

Proof Assume that the sequence $S(T,\bar{x},\phi)$ produced by the Superiorized Version of Algorithm **P** does not contain a solution of T. We first show that in this case the algorithm generates an infinite sequence $\left(x^k\right)_{k=0}^{\infty}$. This is equivalent to saying that, for any x^k that has been generated already, the condition in Line (13) of the algorithm will be satisfied sooner or later (and hence x^{k+1} will be generated). This needs to happen, because as long as the condition is not satisfied we keep resetting (in Line (11)) the value of β_k to γ_ℓ, with ever increasing values of ℓ. However, $(\gamma_\ell)_{\ell=0}^{\infty}$ is a summable sequence of positive real numbers, and so γ_ℓ is guaranteed to be arbitrarily small if ℓ is sufficiently large. Since v^k is either a unit vector in the direction of the negative subgradient of the convex function ϕ at x^k or is the zero vector (see Lines (5)–(8)), $\phi\left(x^k+\beta_k v^k\right) \leq \phi\left(x^k\right)$ must be satisfied if the positive number β_k is small enough. Also, since $\mathscr{P}r_T\left(\mathbf{P}_T x^k\right) < \mathscr{P}r_T\left(x^k\right)$ and $\mathbf{P}_T$ and $\mathscr{P}r_T$ are continuous (Assumptions (2) and (1), respectively), we also have that $\mathscr{P}r_T\left(\mathbf{P}_T\left(x^k+\beta_k v^k\right)\right) < \mathscr{P}r_T\left(x^k\right)$ if β_k is small enough. This completes the proof that the condition in Line (13) of the algorithm will be satisfied and so the algo-

rithm will generate an infinite sequence $S(T,\bar{x},\phi)$. Observing that we have already demonstrated that the $\beta_k v^k$ are bounded perturbations, and comparing (4.3) with Lines (12) and (14), we see that (by the bounded perturbation resilience of **P**) the assumption that the sequence $\left((\mathbf{P}_T)^k x\right)_{k=0}^{\infty}$ converges to a solution of T for all $x \in \mathbb{R}^J$ implies that $S(T,\bar{x},\phi)$ also converges to a solution of T. Thus, applying Lemma 3 we obtain the final claim of the theorem. ■

Unfortunately, this theorem does not go far enough. To demonstrate that a methodology leads to superiorization we should be proving (under some assumptions) a result like $\phi\left(O\left(T,\varepsilon,S(T,\bar{x},\phi)\right)\right) < \phi\left(O\left(T,\varepsilon,R(T,\bar{x})\right)\right)$ in place of the weaker result at the end of the statement of the theorem. Currently we do not have any such proofs and so we are restricted to providing practical demonstrations that our methodology leads to superiorization in the desired sense. In the next section we provide such demonstrations for the Superiorized Version of Algorithm **P**, for two different **P**s.

4.2 Illustrations

We illustrate the superiorization methodology using the algorithm of the previous section on a problem that we have considered in the rest of this document, namely the problem of reconstructing a head cross-section (Figure 3.2) from its projections using both an AP and a BIP algorithm. Consider the 82-views UD data collection described in Subsection 3.2.3.1.

For our illustration, we chose the AP algorithm **P** to be the cyclic projection method as determined by (4.4) with $\Omega = \{(1,\ldots,I)\}$ and $\omega(1,\ldots,I) = 1$. This is, as was mentioned earlier, a classical method that in tomography would be considered a variant of the algebraic reconstruction techniques (ART) [35, Chapter 11].

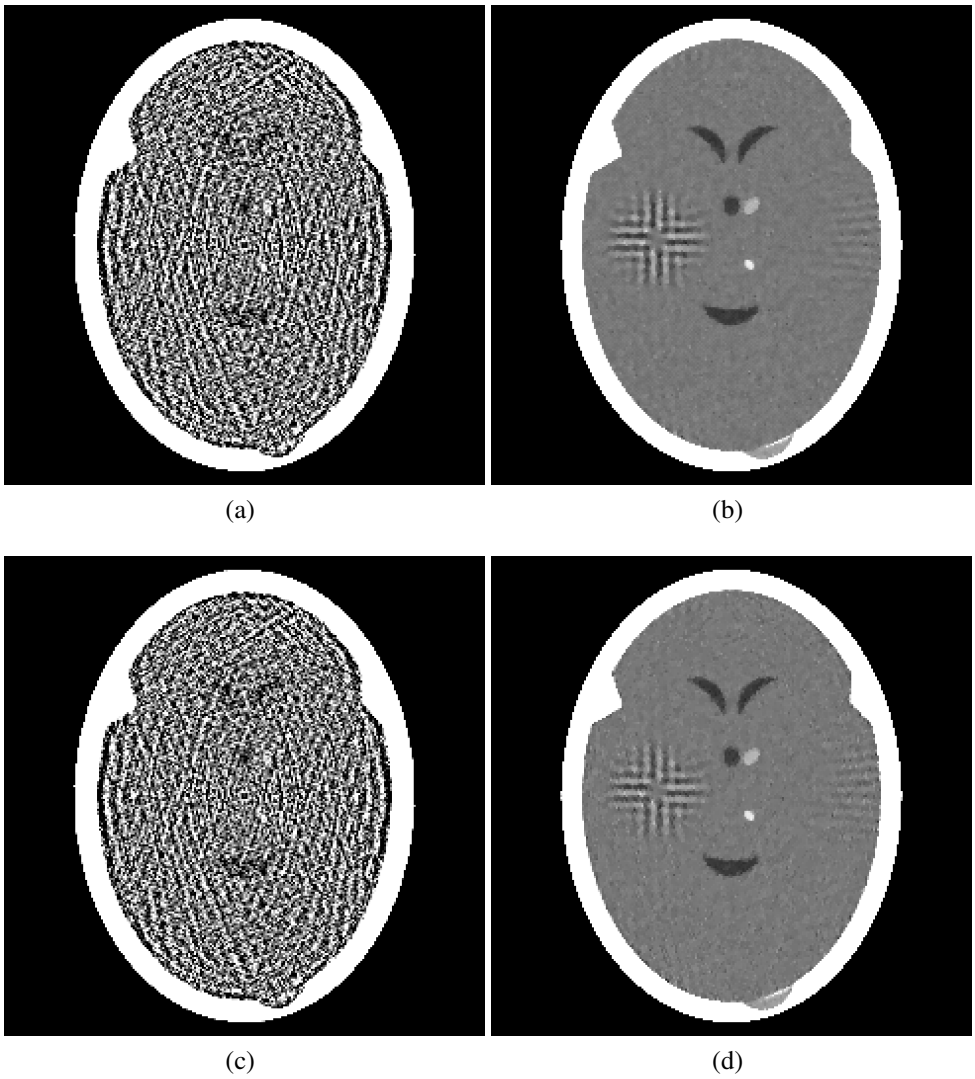
Algorithm	$\phi\left(O\left(T,\varepsilon,S\left(T,\bar{x},\phi\right)\right)\right)$	$\phi\left(O\left(T,\varepsilon,R\left(T,\bar{x}\right)\right)\right)$
Variant of ART	441.50	1,296.44
Variant of BIP	444.15	1,286.44

Table 4.1: Values of TV for the outputs of the various algorithms. The second column is for the superiorized versions and the third column is for the original versions.

For the BIP algorithm we chose $\mathbf{Q}$ as determined by (2.27) with $U = 82$ and each block corresponding to one of the 82 sets of parallel lines along which the data are collected.

The function ϕ for which we superiorized was TV. For the TV-superiorized versions of the algorithms $\mathbf{P}$ and $\mathbf{Q}$ of the previous paragraph we selected $\bar{x}$ to be the origin (the vector of all zeros) and $\gamma_\ell = 0.999^\ell$. Also , we set $\varepsilon = 0.01$ for the stopping criterion, which is small compared to the $\mathscr{P}r_T$ of the initial point ($\mathscr{P}r_T\left(\bar{x}\right) = 330.208$).

For each of the four algorithms ($\mathbf{P}$, $\mathbf{Q}$ and their TV-superiorized versions), the sequence S that is produced by it is such that the output $O\left(T,\varepsilon,S\right)$ is defined; see Figures 4.1(a)-(d) for the images that correspond to these outputs. Clearly, the superiorized reconstructions in Figures 4.1(b) and 4.1(d) are visually superior to their not superiorized versions in Figures 4.1(a) and 4.1(c), respectively. More importantly from the point of view of our theory, consider Table 4.1. As stated in the last paragraph of the previous section, we would like to have that $\phi\left(O\left(T,\varepsilon,S\left(T,\bar{x},\phi\right)\right)\right) < \phi\left(O\left(T,\varepsilon,R\left(T,\bar{x}\right)\right)\right)$. While we are not able to prove that this is the case in general, Table 4.1 clearly shows it to be the case for the two algorithms discussed in this section.



(a) (b)

(c) (d)

Figure 4.1: Reconstructions from underdetermined consistent data obtained for 82 views using: (a) a variant of ART, (b) TV-superiorized version of the same variant of ART, (c) a BIP method, and (d) TV-superiorized version of the same BIP method. The same initial point and stopping criterion were used in all cases; see the text for details.

Chapter 5

Conclusions

5.1 Contributions

The newly proposed superiorization methodology is envisioned as lying between the methods of optimization and of feasibility seeking. With a feasible solution one settles for a point that just fulfills a set of constraints, whereas solving a constrained optimization problem calls for finding a feasible point that optimizes a given objective function. Generally speaking, optimization is a computationally more demanding task than that of finding just any feasible point. We demonstrated that, without employing an optimization algorithm, it is possible to use certain iterative methods, designed for (the less demanding) feasibility problems, in a way that will steer the iterates towards a point that is superior, but not necessarily optimal, in a well-defined sense. The advantage of superiorization is that it allows us to solve significant problems by using powerful feasibility seeking methods and reach a superior feasible point without resorting to optimization techniques. We proved that two large classes of projection methods, namely, AP and BIP algorithms are perturbations resilient. Moreover, for the BIP class of algorithms we provided convergence results even when the underlying system is inconsistent (for the special

case when the C_i in (1.1) are hyperplanes) and further, we were able to accelerate their performances in such linear cases, and provided convergence results that translated into orders of magnitude faster algorithms as compared to the ones that used perturbations without the acceleration. Finally, we took an initial step towards making superiorization a nonheuristic technique, and showed how any algorithm in these classes can be automatically turned into a "superiorized version" of it .

All the mathematical results mentioned above were applied to problems in medical imaging of reconstructing images from projection data, of the type that occurs in computerized tomography. We illustrated the power of this methodology by finding superior solutions in reconstructing objects (phantoms) from data collected according to various geometries and levels of noise, similar to the ones found in real instruments. We found that when ϕ corresponds to the TV of the discrete representation of the phantom, and when we chose v^k to be a unit vector in the direction of a negative subgradient of ϕ at x^k, we were able to get superior reconstructions in terms of quality and speed. As part of our evaluation, we performed statistical hypothesis testing and illustrated that methods that use superiorization are superior (in a strong statistical sense) to the ones that do not use superiorization for the task of detecting low contrast tumors in a CT reconstruction problem.

Many of the results reported in this dissertation can also be found in [8, 13, 26, 36, 55].

5.2 Future work

The research work presented in this dissertation can be extended and generalized in a number of ways. The superiorization methodology needs to be studied further from the mathematical, algorithmic and computational points of view in order to unveil its general capability as a tool for solving inverse problems. In particu-

lar, we need to investigate the computational efficiency of the superiorized versions of algorithms compared to their original versions and, more importantly, further compared to actual optimization algorithms. The initial study of comparison with optimization methods that have been reported in Subsection 3.3.4 for specific algorithms illustrated the importance of the applicability of superiorization for being such a general tool (e.g., in our comparisons of TV-superiorization methods with TV-optimization algorithms we found the superiorization instances to be four times faster than the optimization ones). However, further testing of the computational efficiency of the algorithms produced by this new tool will have to be carried out under a variety of circumstances.

General superiorization algorithmic structure can have additional specific algorithms, for example, if a convex feasibility problem (1.1) has closed convex sets that are not linear (not hyperplanes nor halfspaces) then using *subgradient projections* (see, e.g., [66]) instead of orthogonal projections (2.2) will be important because they are much easier to calculate in each iterative step. Additionally, the use of different ϕs, for example, smoothness of the vectors using *selective smoothing* as discussed in [60] or using central finite differences or other schemes that have vertical and horizontal blurs may be more suitable to different types of applications. Superiorization can be extended to handle several ϕs for which we desire to superiorize, in which case they need to be ordered by priority and/or weighted for their relative importance. Investigating the superiorization algorithms' computational efficiency should also be examined when they are implemented using special hardware that makes use of their parallel nature (i.e., being either index vectors or blocks). This we expect to be most efficacious when superiorization is used to reconstruct 3D objects from 2D projections. As algorithms are developed and tested a dialog on algorithmic developments must be accompanied by mathematical validation and applications to simulated and real data from various relevant fields of

applications.

Validating the concept means proving precise statements about the behavior of iterates $\left(x^k\right)_{k=0}^{\infty}$ generated by the superiorized versions of algorithms. Under what conditions do they converge? Can their limit points be characterized? How would different choices of the perturbation coefficients β_k and the perturbation vectors v^k affect the superiorization process? Can different schemes for generating the β_ks be developed, implemented, investigated? How can the blocks/index vectors be formed so that the iterative process will converge most efficiently? Enlarging the arsenal of bounded perturbations resilient algorithms means generalizing existing proofs for such algorithms and developing new theories that will bring additional ones into the family of bounded perturbations resilient algorithms. Such extensions may include the entire SAP class of algorithms, the extension of accelerated perturbation-resilient BIP algorithms to other kinds of convex sets (not just hyperplanes), proving convergence to other operators that are not necessarily orthogonal (such as Bregman projections/distances, see, e.g., [7, 10]) and proving convergence for sequences that are perturbed at the end of each block or index vector rather than at the end of each iteration. Further developments should include extension to the problem of finding a common fixed point of a family of operators (a direct generalization of the convex feasibility problem, see, e.g., [41]), the possibility to generalize the concept of superiorization so that it will be applicable to the split feasibility problem, see [15, 63, 68], and further studying the behavior of superiorization algorithms in inconsistent situations when the underlying solution set is empty. Such generalization and extensions to the superiorization methodology will make it a promising new field of endeavor for solving inverse problems.

Bibliography

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

[5] 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.

[6] 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.

[7] L.M. Bregman. The relaxation method of finding the common point of con-

vex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

[8] 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.

[9] D. Butnariu, S. Reich, and A. Zaslavski. Convergence to fixed points of inexact orbits of Bregman-monotone and nonexpansive operators in Banach spaces. In H. Fetter Nathansky, B. Gamboa de Buen, K. Goebel, W.A. Kirk, and B. Sims, editors, *Fixed Point Theory and Applications*, pages 11–32. Yokohama Publishers, 2006.

[10] C. Byrne. Iterative projection onto convex sets using multiple Bregman distances. *Inverse Problems*, 15:1295–1313, 1999.

[11] 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.

[12] Y. Censor, W. Chen, P.L. Combettes, R. Davidi, and G.T. Herman. On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints. *Computational Optimization and Applications*, 2009. (Submitted).

[13] Y. Censor, R. Davidi, and G.T. Herman. Perturbation resilience and superiorization of iterative algorithms. *Inverse Problems*, 26:065008, 2010.

[14] Y. Censor, T. Elfving, G.T. Herman, and T. Nikazad. On diagonally relaxed or-

thogonal projection methods. *SIAM Journal on Scientific Computing*, 30:473–504, 2007.

[15] Y. Censor, T. Elfving, N. Kopf, and T. Bortfeld. The multiple-sets split feasibility problem and its applications for inverse problems. *Inverse Problems*, 21:2071–2084, 2005.

[16] Y. Censor, T. Elvfing, 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*, pages 101–113. Elsevier Science Publishers, 2001.

[17] 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.

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

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

[20] G. Cimmino. Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari. *La Ricerca Scientifica (Roma)*, 1:326–333, 1938.

[21] P.L. Combettes. On the numerical robustness of a parallel projection method in signal synthesis. *IEEE Signal Processing Letters*, 8:45–47, 2001.

[22] P.L. Combettes and J. Luo. An adaptive level set method for nondifferentiable constrained image recovery. *IEEE Transactions on Image Processing*, 11:1295–1304, 2002.

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

[24] 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.

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

[26] R. Davidi, G.T. Herman, and Y. Censor. Perturbation-resilient block-iterative projection methods with application to image reconstruction from projection. *International Transactions in Operational Research*, 16:505–524, 2009.

[27] R. Davidi, G.T. Herman, and J. Klukowska. *SNARK09: A Programming System for the Reconstruction of 2D Images from 1D Projections*. New York: CUNY Institute for Software Design and Development, 2009.

[28] 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.

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

[30] T. Elfving and T. Nikazad. Properties of a class of block-iterative methods. *Inverse Problems*, 25:115011, 2009.

[31] T. Goldstein and S. Osher. The split Bregman method for L1 regularized problems. *UCLA CAM report*, 08-29, 2008.

[32] 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.

[33] R. Gordon and G.T. Herman. Reconstruction of pictures from their projections. *Communications of the ACM*, 14:759–768, 1971.

[34] G.T. Herman. Demonstration of beam hardening correction in computerized reconstruction of the head,. *Journal of Computer Assisted Tomography*, 3:373–378, 1979.

[35] G.T. Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections. 2nd edition*. Springer, 2009.

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

[37] G.T. Herman, A. Lent, and S.W. Rowland. ART: Mathematics and applications. *Journal of Theoretical Biology*, 42:1–32, 1973.

[38] M. Figueiredo J. Bioucas-Dias. A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16:2992–3004, 2007.

[39] L. Kaipio and E. Somersalo. Statistical inverse probems: discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198:493–504, 2007.

[40] P.E. Kinahan, S. Matej, J.P. Karp, G.T. Herman, and R.M. Lewitt. A comparison of transform and iterative reconstruction techniques for a volume-imaging PET scanner with a large axial acceptance angle. *IEEE Transactions on Nuclear Science*, 42:2181–2287, 1995.

[41] K.C. Kiwiel and B. Łopuch. Surrogate projection methods for finding fixed points of firmly nonexpansive mappings. *SIAM Journal on Optimization*, 7:1084–1102, 1997.

[42] 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.

[43] 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.

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

[45] R.M. Lewitt. Multidimensional digital image representations using generalized Kaiser-Bessel window functions. *Journal of the Optical Society of America A*, 7:1834–1846, 1990.

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

[47] R. Marabini, G.T. Herman, and J.M. Carazo. 3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs). *Ultramicroscopy*, 72:53–65, 1998.

[48] R. Marabini, E. Reitzel, R. Schroeder, G.T. Herman, and J.M. Carazo. Three-dimensional reconstruction from reduced sets of very noisy images acquired following a single-axis tilt schema: Application of a new three-dimensional

reconstruction algorithm and objective comparison with weighted backprojection. *Journal of Structural Biology*, 120:363–371, 1997.

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

[50] S. Matej, G.T. Herman, T.K. Narayan, S.S. Furuie, R.M. Lewitt, and P.E. Kinahan. Evaluation of task-oriented performance of several fully 3D PET reconstruction algorithm. *Physics in Medicine and Biology*, 39:355–367, 1994.

[51] S. Matej and R.M. Lewitt. Efficient 3-D grids for image reconstruction using spherically symmetrical volume elements. *IEEE Transactions on Nuclear Science*, 42:1361–1370, 1995.

[52] S. Matej and R.M. Lewitt. Practical consideration for 3D image-reconstruction using spherically-symmetrical volume elements. *IEEE Transactions on Medical Imaging*, 15:68–78, 1996.

[53] T.K. Narayan and G.T. Herman. Prediction of human observer performance by numerical observers: an experimental study. *Journal of the Optical Society of America A*, 16:679–693, 1999.

[54] F. Natterer. *The Mathematics of Computerized Tomography*. Teubner, 1986.

[55] T. Nikazad, R. Davidi, and G.T. Herman. Accelerated perturbation-resilient block-iterative projection methods with application to image reconstruction. *IEEE Transactions on Image Processing*, 2010. (Submitted).

[56] Z. Opial. Weak convergence of the sequences of successive approximations for nonexpensive mappings. *Bulletin of the American Mathematical Society*, 53:591–597, 1967.

[57] 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.

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

[59] A.R. De Pierro and A.N. Iusem. A parallel projection method for finding a common point of a family of convex sets. *Pesquisa Operational*, 5:1–20, 1985.

[60] A.R. De Pierro and A.N. Iusem. On the asymptotic behavior of some alternate smoothing series expansion iterative methods. *Linear Algebra and its Applications*, 130:3–15, 1990.

[61] J. Radon. Über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Ber. Sächs. Akad. Wiss., Leipzig, Math. Phys. Kl.*, 69:262–277, 1917.

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

[63] F. Schöpfer, T. Schuster, and A.K. Louis. An iterative regularization method for the solution of the split feasibility problem in Banach spaces. *Inverse Problems*, 24:055008, 2008.

[64] J. von Neumann. *Functional Operators - Vol. II, The Geometry of Orthogonal Spaces*. Princeton University Press, 1950. Reprint of mimeographed lecture notes distributed in 1933.

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

[66] I. Yamada and N. Ogura. Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions. *Numerical Functional Analysis and Optimization*, 25:593–617, 2004.

[67] D.M. Young. *Iterative Solution of Large Linear Systems*. Dover Publication, 2003.

[68] J. Zhao and Q. Yang. Several solution methods for the split feasibility problem. *Inverse Problems*, 21:1791–1799, 2005.

Index