MASTER THESIS

Evaluation of Shearlet-based Superiorization with ART

Master thesis submitted to the Salzburg University of Applied Sciences, in partial fulfilment of the requirements for the degree Master of Science in Engineering, MSc.

Nikolai Janakiev, BSc

Programme Director: FH-Prof. Univ.-Doz. Dr. Stefan Wegenkittl
Supervisor SUAS: FH-Prof. Univ.-Doz. Dr. Stefan Wegenkittl
Supervisor CUNY: Gabor T. Herman Ph.D.

Salzburg, September 2017
Declaration of Academic Honesty

I, Nikolai JANAKIEV, born on December 22, 1991 in Salzburg, hereby declare, that with the exception of chapter 3 and 6, which were co-authored by Clemens Havas, I have written this master thesis entirely on my own and that I have not used any other sources apart from those given.

Salzburg, September 2017

Nikolai JANAKIEV

1410721002
Matriculation Number
General Information

Last- and Given Name: Nikolai JANAKIEV
Institution: Salzburg University of Applied Sciences and Paris Lodron University of Salzburg
Degree Programme: Applied Image and Signal Processing
Full Title: Evaluation of Shearlet-based Superiorization with ART
Keywords: Superiorization, Constrained Optimization, Inverse Problems, Medical Imaging, SNARK, Algebraic Reconstruction Technique, Computerized Tomography, Shearlet Transform, Compressed Sensing, Proximal Operator
Supervisor SUAS: FH-Prof. Univ.-Doz. Dr. Stefan Wegenkittl
Supervisor CUNY: Gabor T. Herman Ph.D.

Abstract

In computerized tomography (CT) it is desired to reduce the x-ray dose, but when using reconstruction algorithms the amount of constraint-compatible solutions increases with the decrease of measurements. One approach to solve the reconstruction problem is the recently-developed superiorization methodology. It is an automated procedure that takes an iterative and perturbation resilient algorithm and creates an algorithm that produces constraint-compatible solutions that are superior according to some secondary criterion. The proposed secondary criterion used in this thesis is based on the shearlet transform, which provides optimally sparse approximations for images with anisotropic features and piecewise $C^2$ singularity curves with bounded curvature. As a result the minimization of the $l_1$-norm of the shearlet transform aims to make tissues like brain tumors better visible. The goal in this thesis is to evaluate the efficacy of the superiorization of the Algebraic Reconstruction Technique (ART) with secondary criteria based on the Total Variation (TV) and the $l_1$-norm of the shearlet transform. The proposed secondary criterion is based on the proximal operator of the $l_1$-norm of the shearlet transform. The experiments conclude that for ART the proposed proximal operator variation of the shearlet-based superiorization performs better than the previous shearlet-based superiorization but it does not outperform the TV based superiorization according to the Image-wise Region of Interest (IROI) metric.
Acknowledgement

I would like to express my sincere gratitude towards my supervisor Distinguished Professor Gabor T. Herman, Ph.D at the Graduate Center of the City of New York, for providing his patience and profound expertise and giving me a chance to do my master project at his department.

Furthermore I would like to thank FH-Prof. Univ.-Doz. Mag. Dr. Stefan Wegenkittl, who was my supervisor at the Salzburg University of Applied Sciences, for his helpful advice and support throughout the project.

Additionally I would like to thank Marcelo Zibetti, Ph.D and Chuang Lin, Ph.D at the Graduate Center of the City University of New York for the conversations and insights which helped my understand the research topics in a lot more depth.

I would also like to thank the Austrian Marshall Plan Foundation for their generous funding, which made this master project financially possible.
# Contents

1 Introduction  
2 Computerized Tomography  
   2.1 Radon Transform  
   2.2 Reconstruction Problem  
3 Superiorization  
   3.1 Problem Sets, Proximity Functions, and $\varepsilon$-Compatibility  
   3.2 Algorithms and Outputs  
   3.3 Bounded Perturbation Resilience  
   3.4 Optimization Criterion and Nonascending Vector  
   3.5 Superiorized Version of an iterative Algorithm  
4 Optimization  
   4.1 Unconstrained and Constrained Optimization  
   4.2 Least squares approximation  
   4.3 Pseudoinverses and least squares  
   4.4 Spectral filtering and regularization  
5 Algebraic Reconstruction Algorithms  
   5.1 Kaczmarz’s method  
   5.2 Affine Spaces and Projections  
   5.3 ART Algorithm  
   5.4 Variations of the Algebraic Reconstruction Technique  
6 Shearlet Transform  
   6.1 $l_1$-Norm of the Shearlet Transform Criterion  
   6.2 Nonascending Vector for $l_1$-Norm of the Shearlet Transform Criterion  
   Version 1  
7 Proximal Operator for Shearlets  
   7.1 Gradients, Subgradients and Subdifferentials  
   7.2 Proximal Algorithms  
   7.3 Proximal Operator for the $l_1$-norm  
   7.4 Algorithm for the Nonascending Proximal Shearlet
1 Introduction

Superiorization is a methodology for a large class of constrained optimization problems. It is a heuristic solver producing constraint-compatible solutions which are optimized according to a secondary optimization criterion. This means superiorization is producing a solution, which is compatible to a primary criterion that is provided by physically or otherwise obtained constraints while the solution is superior according to a secondary criterion [1]. This is done by taking an iterative algorithm and turning it into a superiorized one, where the iterative algorithm is steered by perturbations to a superior solution according to the secondary criterion, while ensuring to reach a constraint-compatible solution with regard to the primary criterion. Therefore the iterative algorithm needs to be perturbation resilient in order to be superiorized.

The implementations in this project are done with SNARK14 which is a framework for image reconstruction of 2D images from 1D projections [2]. SNARK14 incorporates the superiorization methodology and allows the use of existing algorithms as well as the incorporation of new algorithms. These algorithms can be also compared and evaluated with the framework. The image reconstruction experiments are simulating a computerized tomography scanner using SNARK14, which will be discussed in Chapter 8.

The iterative algorithm (covered in Chapter 5) in this project for superiorization is the Algebraic Reconstruction Technique (ART), which is a common algorithm in the field of computerized tomography [3]. ART is based on linear algebra and matrix theory in order to solve a variety of reconstruction problems. This algorithm will be superiorized with two variations of the $l_1$-norm minimization of the shearlet transform.

The shearlet transform provides optimally sparse approximations for cartoon-like images, namely for images with anisotropic features and piecewise $C^2$ singularity curves with bounded curvature [4]. The $l_1$-norm minimization of the shearlet transform aims to amplify these cartoon-like features in the reconstructed image. This will be discussed in Chapter 6.

The $l_1$-norm minimization of the shearlet transform will be done by taking the algorithm which was proposed in [5] and applying the proximal operator for the $l_1$-norm minimization. The proximal algorithms discussed in Chapter 7, is a class of algorithms for solving nonsmooth large-scale convex optimization problems which are well suited for high-dimensional datasets [6].
2 Computerized Tomography

This chapter explains Computerized Tomography (CT), a branch of radiology, diagnostic medicine and medical imaging. CT refers to the imaging of the cross-sections (or slices) of a person or an object in a non-destructive way. This is done for a chosen cross-section by measuring the attenuation of multiple X-ray beams for multiple angles and lines through the medium. From this measured data the image of the cross-section can be reconstructed with various mathematical methods. In the reconstructed image the different tissues or materials are represented by their attenuation coefficients.

When a single X-ray beam passes through a medium, such as body tissue or other materials, the energy of the beam gets attenuated while passing through. This transmitted energy is measured by a detector at the other end of the medium which derives the attenuation of the transmitted energy of the beam. This can be seen in Figure 2.1 which illustrates projections of two cylinders for two angles with parallel beams. This results in two graphs of the attenuation of the energy over the object. In CT, multiple projections are collected over a range of angles, which results in a so-called sinogram. An example for a sinogram from a cross-section can be seen in Figure 2.2, where each column of the 2D image represents the projection of a single angle and each value on that column represents a single beam transmitted through the material. The brighter the value is the more the beam is attenuated when passing through the medium.

With the help of the sinogram, the cross-section of the medium can be reconstructed. The first mathematical solution for reconstructing images from projections was first published in 1917 by Johann Radon in [8], whereas the first X-ray CT scanner was much later invented by Godfrey Hounsfield for which he received the Nobel Prize in 1979 [9]. More to CT can be found in [3], [7].

Figure 2.1: Two projections from two cylinders adapted from [7, p.2]
The way measurements are taken for a CT scan is depicted in Figure 2.3. Each X-ray source sends its beam to some corresponding detector, where both the source and detector lie in the plane of the selected cross-section to be imaged. It is assumed that the material in the reconstruction region is homogeneous (such as water or air) except for the object of interest.

2.1 Radon Transform

Following from Figure 2.3, each line that a beam follows is described in the \((r, \theta)\) space for a given function \(f\) which is defined in the plane of the cross-section. Such a function \(f\) can represent the attenuation coefficients of a cross-section. The Radon transform \(R_f(r, \theta)\) describes the line integral for each beam for the pairs \(r\) and \(\theta\).
In order to calculate the Radon transform, the integral over these lines needs to be calculated as shown in the following definition.

**Definition 2.1. Radon transform**
The Radon transform of a function \( f \) is defined for the pairs of \((r, \theta)\) as

\[
\mathcal{R}f(r, \theta) = \begin{cases} 
\int_{-\infty}^{\infty} f(\sqrt{r^2 + z^2}, \theta + \tan^{-1}(\frac{z}{r}))dz & r \neq 0 \\
\int_{-\infty}^{\infty} f(z, \theta + \frac{\pi}{2})dz & r = 0
\end{cases}
\] (2.1)

where \( \mathcal{R}f(r, \theta) \) is the line integral over the distance \( z \) along the line \( L \) as shown in Figure 2.3 [3].

The Radon transform of \( f \) describes the sinogram for the function \( f \). This becomes useful in the following section where the reconstruction problem is described. Further details on the Radon Transform can be found in [7], [8] and [3].

### 2.2 Reconstruction Problem

In this section the general definitions of the reconstruction problem are discussed. Many of the definitions are necessary for the following chapters in order to describe the solutions for the image reconstruction of projections.

**Definition 2.2. Image vector \( x \)**
The image vector \( x \in \mathbb{R}^J \) is the digitized representation of an image, which has \( n \times n \) pixels and is \( J = n^2 \) dimensional. The numbering of the image vector iterates over the image pixels, as shown in Figure 2.4. [3]

**Definition 2.3. Pixel basis functions \( b_1, ..., b_J \)** can be both

\[
b_j(r, \theta) = \begin{cases} 
1, & \text{if } (r, \theta) \text{ is inside the } j\text{-th pixel} \\
0, & \text{otherwise}
\end{cases}
\] (2.2)

for \( j = 1, 2, ..., J \) and points with polar coordinates \((r, \theta)\) in the plane. [3]

Using this definition of a pixel basis function an image can be represented in a digitized form by a set of pixel basis functions as shown in the following equation.
If then the radon transform $\mathcal{R}$ is applied to both sides of such a digitized version of an image, the following equation can be derived due to the linearity of $\mathcal{R}$.

$$\mathcal{R}f(r, \theta) = \sum_{j=1}^{J} x_j b_j(r, \theta)$$

(2.4)

The measurements collected by the CT scanner are a vector of the form of $\mathcal{R}f(r, \theta)$ for a finite set of lines $l_{r, \theta}$. These lines correspond to $(r_1, \theta_1), (r_2, \theta_2), \ldots, (r_I, \theta_I)$ for some positive integer $I$. Following the equation (2.4) the measurement vector can be derived as follows.

$$y_i = \mathcal{R}f(r_i, \theta_i), \text{ for } i = 1, \ldots, I$$

(2.5)

**Definition 2.4. Measurement vector $y$**

For the measurement vector $y \in \mathbb{R}^I$, the element $y_i$ is the $i$-th measurement of the CT scanner. The measurement vector $y$ is also called the ray-sum, where the equation for $y_i$ is shown in (2.5). [3]

The following system of equations can be derived from (2.5)

$$y_i = \sum_{j=1}^{J} x_j \mathcal{R}b_j(r_i, \theta_i) \text{ for } i = 1, \ldots, I$$

(2.6)
Since the pixel basis function $b_j$ has the value 1 on the corresponding pixel and 0 everywhere else, the integral of $Rb_j(r_i, \theta_i)$ corresponds to the length of the intersection of the line $l_{r_i,\theta_i}$. Or in other words, it is the length of the intersection of the $j$-th pixel with the ray of the $i$-th source-detector pair. This value is denoted in the following equation for $r_{i,j}$.

$$r_{i,j} = Rb_j(r_i, \theta_i), \text{ for } i = 1, ..., I \text{ and } j = 1, ..., J$$ (2.7)

Equation (2.5) can then be rewritten as follows. From this equation the definition for the projection matrix can be denoted.

$$y_i = \sum_{j=1}^{J} r_{i,j} x_j$$ (2.8)

**Definition 2.5. Projection Matrix $R$**

The $(i,j)$-th element of the projection matrix $R$ is defined in equation (2.7) as $r_{i,j}$ where $R \in \mathbb{R}^{I \times J}$. The equation for $r_{i,j}$ is stated in (2.7). [3]

Since any particular line $l_{r_i,\theta_i}$ goes through relatively few of the pixels of the grid, most values of $r_{i,j}$ are equal to 0. This means that the projection matrix $R$ is a sparse matrix.

**Definition 2.6. Error vector $e \in \mathbb{R}^I$**

The error vector $e$ is the $I$-dimensional column vector where the $i$-th component, $e_i$ is the difference between the component of the measurement vector $y_i$ and $\sum_{j=1}^{J} r_{i,j} x_j$ [3]

The next definition describes the reconstruction problem formulated as an inverse problem. This can be used for many different algorithms, such as ART, which is described in Chapter 5.

**Definition 2.7. Reconstruction Problem**

The discrete reconstruction problem is defined by finding the solution for $x$ in the following equation

$$y = Rx + e$$ (2.9)

where $y, e \in \mathbb{R}^I$, $x \in \mathbb{R}^J$ and $R \in \mathbb{R}^{I \times J}$. [3]

For further details on the basic concepts and definitions refer to [7], [3] and [10].
3 Superiorization

The superiorization methodology is an automatic and heuristic procedure for taking an iterative algorithm and turning it into an superiorized one. The iterative algorithm needs to fulfil the properties of producing constraint-compatible solutions and being resilient to bounded perturbations in order for its superiorized version to produce constraint-compatible solutions [1].

Superiorization is needed when a problem has a set of different solutions which are constraint-compatible, but are not suitable for the user. This can mean noisy solutions for some given problems as the case can be in medical physics. In order to handle this issue and still reach a constraint-compatible solution, some prior knowledge can be added into the procedure for improvement. This can be any criterion which evaluates some quality measure of the solution (e.g. measurement of noise). Superiorization incorporates this prior knowledge by steering the solution with bounded perturbations between each iteration of the given algorithm towards a solution, which is superior in the secondary criterion [1].

As an heuristic approach, the superiorization methodology does not guarantee an optimal solution in regard to its secondary criterion. Superiorization will produce a solution which is both constraint-compatible and also superior to the given secondary criterion. Thus, the superiorized algorithm will produce better solutions, which are as good as the solutions of the original algorithm, but are better in regard to one secondary criterion [1].

This chapter, which is based on [1], focuses on all the necessary declarations, definitions and concepts which are required for the superiorization methodology and for all further experiments with superiorization in this thesis.

3.1 Problem Sets, Proximity Functions, and $\varepsilon$-Compatibility

In optimization in medical physics the solution space can be restricted to $\mathbb{R}^J$ and in practice the solution space is further restricted to a non-empty subset $\Omega$ of $\mathbb{R}^J$ such as the space $\mathbb{R}^J_+$. In this section, basic definitions are described for the explanation of the functionality of superiorization in the next chapter.

Definition 3.1. Problem set $\mathcal{T}$

Each problem $T \in \mathcal{T}$ is a description of the constraints of a particular application [1].
In the case of a tomographic scanner, the reconstruction problem is determined by the measurements taken by the scanner and $T$ is then the set of all possible measurements. For further details see [1].

**Definition 3.2. Proximity function $\mathcal{P}r$ on the problem set $T$**

$\mathcal{P}r$ is defined such that, for every $T \in T$, $\mathcal{P}r_T(x) : \Omega \mapsto \mathbb{R}_+$. $\mathcal{P}r_T$ can be thought as an indicator of how incompatible $x$ is with the constraints of $T \in T$. The case $\mathcal{P}r_T(x) = 0$ means that $x$ is perfectly compatible with the constraints given by $T$ [1].

In tomography, $\mathcal{P}r_T(x)$ should indicate the violation of the proposed reconstruction $x \in \Omega$ with regard to the constraints of $T$ that are provided by the measurements of the scanner.

**Definition 3.3. Problem structure $\langle T, \mathcal{P}r \rangle$**

The pair $\langle T, \mathcal{P}r \rangle$ is a combination of a non-empty set $T$ and a proximity function $\mathcal{P}r$ [1].

**Definition 3.4. $\varepsilon$-compatible solution**

For a problem structure $\langle T, \mathcal{P}r \rangle$, a problem $T \in T$ and a non-negative $\varepsilon$, $x \in \Omega$ is a $\varepsilon$-compatible solution if it satisfies the condition that $\mathcal{P}r_T(x) \leq \varepsilon$ [1].

In the case of tomography, the definition of an $\varepsilon$-compatible solution is needed since measurements can be noisy and a perfectly compatible solution is not always desired or possible for the problem $T \in T$. The $\varepsilon$ can be considered as a stopping criterion for the iterative algorithm.

### 3.2 Algorithms and Outputs

In this section the general concept of algorithms is defined for the problem structures $\langle T, \mathcal{P}r \rangle$. Furthermore an iterative process including such an algorithm and the definition of the output for such a sequence of an iterative process are explained. An additional set $\Delta$ needs to be defined, such that $\Omega \subseteq \Delta \subseteq \mathbb{R}^J$, where both $\Omega$ and $\Delta$ are assumed to be fixed for any particular problem structure $\langle T, \mathcal{P}r \rangle$.

**Definition 3.5. Algorithm $P$**

For each problem $T \in T$ and a problem structure $\langle T, \mathcal{P}r \rangle$ an algorithm $P$ assigns an operator $P_T : \Delta \mapsto \Omega$ [1].

**Definition 3.6. Iterative process**
An iterative process of an algorithm \( P_T \) is defined for any initial point \( x \in \Omega \) as an (potentially) infinite sequence
\[
((P_T)^k x)_{k=0}^\infty
\]
that is the sequence \((x, P_T(x), P_T(P_T(x)), P_T(P_T(P_T(x))),...)\) of points in \( \Omega \) [1].

**Definition 3.7. Output \( O(T, \varepsilon, R) \)**

For a problem structure \( \langle T, Pr \rangle \), a \( T \in T \), an \( \varepsilon \in \mathbb{R}_+ \), and a sequence \( R = (x^{(k)})_{k=0}^\infty \) of points in \( \Omega \), the output \( O(T, \varepsilon, R) \) is used to denote \( x \in \Omega \) with the following properties:
\( P_T(x) \leq \varepsilon \) and there is a non-negative integer \( K \) such that \( x^{(K)} = x \) and, for all non-negative integers \( k < K \), \( Pr(x^{(k)}) > \varepsilon \). If there is no such \( x \), then \( O(T, \varepsilon, R) \) is undefined, otherwise it is defined [1].

As described in [1], if \( R \) is an (infinite) sequence of points in \( \Omega \) that are produced by an algorithm \( P_T \) for the problem \( T \) without a termination criterion, then \( O(T, \varepsilon, R) \) is the output produced by the algorithm \( P_T \) when \( Pr(x^{(k)}) \leq \varepsilon \) is reached, or in other words when the first \( \varepsilon \)-compatible solution is reached.

### 3.3 Bounded Perturbation Resilience

The paper on perturbation resilience [11] discusses the notion of bounded perturbation, but the definitions in the paper are not sufficient for a specific application i.e. in medical physics or more generally for problems involving noisy data. This is because bounded perturbation is only useful for problems \( T \) for which there is a perfectly compatible solution as in \( Pr_T(x) = 0 \). Therefore, an extension of the definition of bounded perturbation is introduced in [1] and defined in the following definition.

**Definition 3.8. Strongly perturbation resilient algorithm \( P \)**

For a problem structure \( \langle T, Pr \rangle \) and for a problem \( T \in T \), the definition of \( P \), a strongly perturbation resilient algorithm, is as follows

1. there exists an \( \varepsilon \in \mathbb{R}_+ \) such that \( O(T, \varepsilon, ((P_T)^k x)_{k=0}^\infty) \) is defined for every \( x \in \Omega \);
2. for all \( \varepsilon \in \mathbb{R}_+ \) such that \( O(T, \varepsilon, ((P_T)^k x)_{k=0}^\infty) \) is defined for every \( x \in \Omega \), there is also the case that \( O(T, \varepsilon', R) \) is defined for every \( \varepsilon' > \varepsilon \) and for every sequence \( R = (x^{(k)})_{k=0}^\infty \) of points in \( \Omega \) generated by
\[
x^{(k+1)} = P_T(x^{(k)} + \beta_k v^{(k)}), \text{ for all } k \geq 0,
\]
where $\beta_k v^{(k)}$ are bounded perturbations, meaning that the sequence $(\beta_k)_{k=0}^{\infty}$ of non-negative real numbers is summable (that is $\sum_{k=0}^{\infty} \beta_k < \infty$), the sequence $(v^{(k)})_{k=0}^{\infty}$ of vectors in $\mathbb{R}^J$ is bounded and, for all $k \geq 0$, $x^{(k)} + \beta_k v^{(k)} \in \Delta$ [1].

The second property means that for strongly perturbation resilient algorithms which are guaranteed to produce solutions that are $\varepsilon$-compatible (for non-negative real number $\varepsilon$), there exists a $\varepsilon'$-compatible solution (where $\varepsilon' > \varepsilon$) for every initial point $x \in \Omega$ after the perturbation in (3.2). This is discussed in more detail in [11] and [1].

### 3.4 Optimization Criterion and Nonascending Vector

For the constrained optimization problem with the problem structure $\langle T, P \rangle$ the secondary optimization criterion and the nonascending vector is defined as follows.

**Definition 3.9.** *Secondary optimization criterion* $\phi$

$\phi$ is a function $\phi : \Delta \mapsto \mathbb{R}$ which measures the undesirability of a point $x \in \Delta$ with regard to the secondary optimization criterion. Further a point $x_1 \in \Delta$ is superior to a point $x_2 \in \Delta$ if $\phi(x_1) < \phi(x_2)$ [1].

**Definition 3.10.** *Nonascending vector* $d \in \mathbb{R}^J$

Given a function $\phi : \Delta \mapsto \mathbb{R}$ and a point $x \in \Delta$, a nonascending vector $d \in \mathbb{R}^J$ for $\phi$ at $x$ satisfies $||d|| \leq 1$ and there is a $\delta > 0$ such that for all $\lambda \in [0, \delta]$, $(x + \lambda d) \in \Delta$ and $\phi(x + \lambda d) \leq \phi(x)$ [1].

In order to steer a strong perturbation resilient algorithm, perturbation as defined in (3.2) is needed. The produced superiorized algorithm seeks to be equally good in regard to its constraint-compatibility and also superior according to the secondary optimization criterion. This is done by making sure that $\beta_k v^{(k)}$ have to be bounded and that $\phi(x^{(k)} + \beta_k v^{(k)}) \leq \phi(x^{(k)})$ for all $k \geq 0$.

For the superiorized version of the algorithm $P$ to be in the solution space $\Omega$, the algorithm $P$ needs to be defined as $P : \Delta \mapsto \Omega$ since $(x + \beta d) \in \Delta$ can be outside the set of $\Omega$.

One important nonascending vector is the zero-vector, which is necessary for the proof for convergence. In order for the algorithm to steer the solution towards a smaller or equal of value $\phi$, a vector $d$ needs to be found that fulfils $\phi(x + \lambda d) \leq \phi(x)$. 
3.5 Superiorized Version of an iterative Algorithm

In this section the superiorized version of an iterative algorithm $P$ is discussed.

**Algorithm 1** Superiorized Version of Algorithm P [1]

1: set $k = 0$
2: set $x^{(k)} = \bar{x}$
3: set $l = -1$
4: while $P_{rT}(x^{(k)}) > \varepsilon$ do
5: set $n = 0$
6: set $x^{(k,n)} = x^{(k)}$
7: while $n < N$ do
8: set $v^{(k,n)}$ to be a nonascending vector for $\phi$ at $x^{(k,n)}$
9: set loop = true
10: while loop do
11: set $\beta_{k,n} = \gamma_l$
12: set $z = x^{(k,n)} + \beta_{k,n} v^{(k,n)}$
13: if $z \in \Delta$ and $\phi(z) \leq \phi(x^{(k)})$ then
14: set $n = n + 1$
15: set $x^{(k,n)} = z$
16: set loop = false
17: end if
18: end while
19: end while
20: set $x^{(k+1)} = P_{rT}x^{(k,N)}$
21: set $k = k + 1$
22: end while

In the initialization the iteration number $k$ is set to 0 in line 1 and then increased with every iteration in line 22 of the main loop (line 4-23). This main loop is iterated as long as the condition of the proximity function $P_{rT}(x^{(k)}) > \varepsilon$ is fulfilled, where $\varepsilon$ is a defined threshold. $x^{(k)} = x^{(0)}$ is initialized with its initial value $\bar{x}$ in line 2. Then in line 3 the integer index $l$ is set to $-1$ which is responsible for the next element in the summable sequence $(\gamma_l)_{l=0}^{\infty}$. Such a summable sequence can be defined as $\gamma_l = a^l$ where $0 < a < 1$.

The nonascending vector $v^{(k,n)}$ is computed in line 8 and the innermost loop (line 10-19) is executed until the condition $z \in \Delta$ and $\phi(z) \leq \phi(x^{(k)})$ is met. This means that the index $l$ for the sequence $\gamma_l$ in $\beta_{k,n} = \gamma_l$ is increased until the vector $z$, which is assembled by $x^{(k,n)} + \beta_{k,n} v^{(k,n)}$, is superior to $x^{(k)}$ with regard to the secondary criterion $\phi$. 
4 Optimization

Optimization is a subject which aims to construct computational methods to find an optimal solution to a given problem. It has wide applications in many different fields such as science, engineering, business management, military and space technology. Optimization plays also a large role in image analysis and medical imaging where it is applied to tomographic reconstruction, radiation therapy treatment planning, image registration and segmentation.

The goal of this chapter is to provide a broad summary to the theoretical foundation to optimization and also some methods and algorithms which are used for solving unconstrained and constrained optimization problems.

4.1 Unconstrained and Constrained Optimization

Optimization can be separated into unconstrained optimization problems and constrained optimization problems, which will be discussed in this section. Unconstrained optimization is defined as follows

Definition 4.1. Unconstrained optimization problem

A unconstrained optimization problem is described as

\[
\text{minimize}_{x \in \mathbb{R}^n} \quad f(x)
\]

where \(x \in \mathbb{R}^n\) is the optimization variable and \(f\) represents the objective function which is also called the cost function or the loss function. The goal is to find some solution \(x^*\) where \(f(x^*) \leq f(x)\) for all \(x \in \mathbb{R}^n\). [12]

This means the aim is to find a minimizer for the problem in Definition 4.1. When some restrictions are required for the optimization problem and some constraints are introduced then it is a constrained optimization problem

Definition 4.2. Constrained optimization problem

A constrained optimization problem has the form

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad \phi_i(x) \leq \varepsilon_i
\end{align*}
\]

(4.2)
where \( f(x) : \mathbb{R}^n \to \mathbb{R} \) is the objective function. The functions \( \phi_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, m \) are the constraint functions with the bounds \( \varepsilon_1, \ldots, \varepsilon_m \). The vector \( x \in \mathbb{R}^n \) is the \( n \)-dimensional optimization variable [13].

The vector \( x^* \) is called optimal solution of the optimization problem stated in the previous Definition 4.2, if it reaches the smallest objective value \( f(x) \) while satisfying all the constraints \( \phi_i(x) \leq \varepsilon_i, i = 1, \ldots, m \). In optimization there is a class of optimization problems called convex optimization problems which aims to minimize convex functions over convex sets. This means that all the objective functions and constraint functions are convex.

**Definition 4.3. Convex function**

The function \( f : X \to \mathbb{R} \) is called convex if

\[
f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)
\]

for all \( x_i, x_2 \in X \) and for all \( t \in [0,1] \) [13].

**Definition 4.4. Convex set**

A convex set \( C \) contains the line segment between any two points \( x_1, x_2 \in C \), that is

\[
\theta x_1 + (1-\theta)x_2 \in C
\]

for \( 0 \leq \theta \leq 1 \) [13].

### 4.2 Least squares approximation

The least squares approximation of the following system of equations

\[
Ax = b
\]

can be expressed as an unconstrained problem involving the following Euclidean or \( l_2 \)-norm minimization

\[
\text{minimize } ||Ax - b||_2^2 = r_1^2 + r_2^2 + \cdots + r_M^2
\]

where \( A \in \mathbb{R}^{M \times N} \) and \( b \in \mathbb{R}^M \) describe the linear system of equation and \( x \in \mathbb{R}^N \) is
the variable that needs to be solved or approximated. A solution to the least squares
approximation is sometimes called an approximate solution of \( Ax \approx b \), in the Euclidean
norm \( || \cdot ||_2 \). The residual \( r \in \mathbb{R}^M \) of the problem is defined as follows

\[
r = Ax - b
\]

where its components are sometimes called the individual residuals associated with
\( x \). The problem in equation (4.6) is a convex problem, which means that there exists
a minimum which is also the global minimum. This can be proved by showing that
the second order derivative is positive semi-definite if it exists. This is shown in the
following equation

\[
f : x \mapsto ||Ax - b||_2^2 = x^T A^T A x - 2b^T A x + b^T b
\]

\[
\frac{\partial^2 f}{\partial x} = A^T A
\]

where the second derivative of \( f \) is a positive semi-definite matrix, which means that
\( f \) is a convex function. In the case when \( b \) is in the column-space of the matrix \( A \),
(i.e. \( b \in \text{col}(A) \)), then the equation \( Ax = b \) has at least one solution vector \( x \). If
\( b \notin \text{col}(A) \) then \( x \) can only be approximated up to some residual \( r \) of length \( ||r||_2 > 0 \).
Which leads to the following definition of the normal equation:

**Definition 4.5. Normal equation**

The normal equation is defined by

\[
A^T Ax = A^T b
\]

where \( A^T A \) is a symmetric matrix, since \( (A^T A)^T = A^T (A^T)^T = A^T A \). The normal
equation has always at least one solution [13].

When \( A^T A \) is invertible, there exists a unique solution to the normal equation and
therefore the solution \( x^* \) to the least squares problem is

\[
x^* = (A^T A)^{-1} A^T b.
\]

When the matrix \( A^T A \) is not invertible, the solution \( x^* \) is not unique.
One way to solve (4.6) is by expressing the objective function as the convex quadratic function \( f : \mathbb{R}^N \rightarrow \mathbb{R} \) defined by

\[
f(x) := ||Ax - b||^2_2 = (Ax - b) \cdot (Ax - b)
\]

which is further \( f(x) = x^T A^T Ax - 2b^T Ax + b^T b \). This function \( f \) is minimized by \( x \) if and only if \( x \) satisfies

\[
\nabla f(x) = 2A^T Ax - 2A^T b = 0
\]

which also satisfies the normal equation in (4.9).

The least squares approximation has one computational concern when dealing with CT reconstruction of the form of \( y = Rx + e \) (2.9), which is that the matrix \( R^T R \) is typically very large and \( R^T R \) is not guaranteed to be sparse. Another problem is that \( R^T R \) might not be invertible or that the inverse of this matrix could potentially be difficult to compute. Further literature on least squares approximation can be found in [13].

### 4.3 Pseudoinverses and least squares

One way to solve the previous equation in (4.9) is to use the pseudoinverse which is also called the Moore-Penrose pseudoinverse. This can be achieved by Singular Value Decomposition (SVD)

**Definition 4.6.** *Singular Value Decomposition*

For the matrix \( A \in \mathbb{R}^{M \times N} \) the SVD is defined as

\[
A = U \Sigma V^T
\]

where \( U \in \mathbb{R}^{M \times M} \), \( \Sigma \in \mathbb{R}^{M \times N} \) is a rectangular diagonal matrix with non-negative real numbers \( \sigma_j \) and \( V \in \mathbb{R}^{N \times N} \). \( U \) and \( V \) are unitary matrices, meaning that \( U^T = U^{-1} \) and \( V^T = V^{-1} \). Every matrix \( A \) has an SVD and the singular values \( \sigma_i \) are uniquely determined.

The equation (4.5) can be expressed with SVD as
\[ U \Sigma V^T x = b. \]  

**Definition 4.7. Pseudoinverse of a matrix $A$**

The pseudoinverse of a matrix $A$ is given by

\[ A^+ = V \Sigma^+ U^T \]  

where $\Sigma^+ \in \mathbb{R}^{M \times N}$ is the pseudoinverse of $\Sigma$ with diagonal entry $1/\sigma_j$ where $\sigma_j \neq 0$ and 0 where $\sigma_j = 0$.

From this follows the Moore-Penrose solution to the normal equation (4.9) where $x^+$ also satisfies the normal equation

\[ x^+ = A^+ b = (V \Sigma^+ U^T) b \]  

### 4.4 Spectral filtering and regularization

The previously used SVD approach can be further extended by using truncated SVD. This comes from the notion that the largest singular values of the matrix $A$ capture the dominant behavior of $A$. Because of the reciprocal of a small singular value, these values can have strong influence on the computation of $x^+$ which might not be desired. From this idea the matrix $A$ can be approximated by truncating the SVD as follows

**Definition 4.8. Truncated SVD**

The truncated SVD approximation $A_k$ of a matrix $A$ is given by

\[ A_k = U_k \Sigma_k V_k^T = \sum_{j=1}^{k} \sigma_j u_j v_j^T \]  

where $A, A_k \in \mathbb{R}^{M \times N}$, $U_k$ and $V_k$ are created from the first $k$ columns of $U$ and $V$, $\Sigma_k \in \mathbb{R}^{k \times k}$ is the diagonal matrix with entries $\sigma_1 \geq \cdots \geq \sigma_k > 0$.

Besides truncating the SVD, regularization is often applied to select a solution among the possible solutions with respect to a second criterion. Tikhonov Regularization is a common form of regularization of (4.5) which results in the following convex quadratic optimization problem

**Definition 4.9. Tikhonov regularization**
The Tikhonov regularization is defined for the problem $Ax = b$ as

$$\text{minimize } ||Ax - b||_2^2 + \alpha||x||_2^2 = x^T (A^T A + \alpha I)x - 2b^T Ax + b^T b$$

(4.18)

where $\alpha > 0$ and $I$ represents the identity matrix.

The analytical solution of the Tikhonov regularization problem is defined as

$$x_\alpha = (A^T A + \alpha^2 I)^{-1} A^T b$$

(4.19)

This solution can be expressed with singular value decomposition, where the Tikhonov regularization applies specific damping to $\Sigma$ where the diagonal entries of $\Sigma_\alpha$ are given by

$$\begin{cases} 
\left( \frac{\sigma_j^2}{\sigma_j^2 + \alpha^2} \right) & \text{if } \sigma_j > 0 \\
0 & \text{if } \sigma_j = 0
\end{cases}$$

(4.20)

The solution $x_\alpha$ of the minimization problem is then given by

$$x_\alpha = V \Sigma_\alpha U^T b = \sum_{j:\sigma_j > 0} \frac{1}{\sigma_j} \left( \frac{\sigma_j^2}{\sigma_j^2 + \alpha^2} \right) (u_j^T b) v_j$$

(4.21)

The $\alpha$ parameter controls the damping of the components of the solution, namely controlling the smoothness of the solution. Setting $\alpha = 0$ is equivalent to the previous Moore-Penrose solution. There are also other ways to choose the weights $\phi_j = \left( \frac{\sigma_j^2}{\sigma_j^2 + \alpha^2} \right)$ for the Tikhonov regularization, which could for example incorporate the minimization of the Total Variation (TV). Spectral filtering is further discussed in [14] and [15].

As discussed in Chapter 3, the superiorization methodology is one approach for solving constrained optimization problems. A problem is discussed in [16] that arises with common constrained optimization approaches. The problem when designing a constrained optimization algorithm, is that a new algorithm needs to be investigated and developed for each new combinations of constraints and for each new problem.

The superiorization methodology skips the step of creating a new algorithm for each problem and uses existing algorithms and secondary criteria instead. If there is an iterative, perturbation resilient and constraint-compatible algorithm, it can be steered after each iteration towards a superior solution with regard to some secondary criterion. Such existing iterative algorithms are the algebraic reconstruction algorithms which are
discussed in Chapter 5.

The essential difference between the conventional approach to constrained optimization and superiorization is that the procedure of the conventional approach is to design an algorithm that depends simultaneously on both the constraints and the optimization criterion. In contrast, superiorization takes an iterative algorithm for the constraints and produces from this algorithm a superiorized version for the given optimization criterion, which can be seen in [16].
5 Algebraic Reconstruction Algorithms

The focus of this chapter is to introduce and discuss ART, which forms a large family of image reconstruction algorithms. ART is a procedure for the discrete reconstruction problem as described before in (2.9). This reconstruction problem is the estimation of the image vector \( x \) in the equation \( y = Rx + e \), where \( y \) is the given measurement vector, \( e \) is the error vector and \( R \) is the reconstruction matrix.

The algebraic reconstruction techniques are based on linear algebra and on matrix theory in contrast to other methods using transforms. They rely on a system of constraints which are derived by the measurements from the tomographic scanner. ART treats the problem of image reconstruction as a discrete problem from the start where each produced image will be digitized as a rectangular grid of pixels.

One concern with an iterative approach with such large systems of equations is that the computational cost tends to be very high. If the system of equations is overdetermined, namely with more equations than unknowns, then the system likely does not have an exact solution and needs to produce an approximation. On the other hand if that system of equations is underdetermined, namely more unknowns than equations, then there may be infinitely many solutions, of whose a solution needs to be found. ART is mostly suggested in literature for the use of such an underdetermined case [17].

5.1 Kaczmarz’s method

Kaczmarz’s method describes an iterative procedure or algorithm for solving the approximation of a linear equation system \( Ax = b \). In the case of the reconstruction problem (2.9) the Kaczmarz’s method is used to solve the equation of the form \( y = Rx \).

If \( r_i \) is denoted as the \( i \)-th row of the projection matrix \( R \) and if \( y_i \) is denoted as the \( i \)-th coordinate of the measurement vector \( y \), then the system can be written as \( y_i = r_i x \) for every value of \( i \).

ART as well as Kaczmarz’s method works with a system of equations. In this case this system consists of \( I \) linear equations from the measurement vector \( y \in \mathbb{R}^I \) and with \( J \) unknowns, namely the image vector \( x \in \mathbb{R}^J \). Typically both \( I \) and \( J \) are in order of \( 10^5 \), so this creates a large system. As previously discussed in Section 2.2, the projection matrix \( R \in \mathbb{R}^{I \times J} \) is a sparse matrix. Kaczmarz’s method depends on the system being sparse [3].
5.2 Affine Spaces and Projections

The discrete reconstruction problem of the form \( y = Rx + e \) can be written as a system of equations with \( J \) unknowns from the image vector \( x \) and \( I \) equations for each measurement component \( y_i \) from \( y \). This system of equations can be written in the following way.

\[
\begin{align*}
y_1 &= r_{1,1}x_1 + r_{1,2}x_2 + \cdots + r_{1,J}x_J \\
y_2 &= r_{2,1}x_1 + r_{2,2}x_2 + \cdots + r_{2,J}x_J \\
\vdots \\
y_I &= r_{I,1}x_1 + r_{I,2}x_2 + \cdots + r_{I,J}x_J
\end{align*}
\]  

Each equation of the system of equations represents a hyperplane. When there is a unique solution for the whole reconstruction problem, then all hyperplanes intersect in a unique point. This can be simply illustrated in the case for the two dimensional case where both hyperplanes are straight lines in \( \mathbb{R}^2 \).

\[
\begin{align*}
y_1 &= r_{1,1}x_1 + r_{1,2}x_2 \\
y_2 &= r_{2,1}x_1 + r_{2,2}x_2
\end{align*}
\]  

A system of equations for \( J = 2 \) is illustrated in Figure 5.2. The application of Kaczmarz’s method can be already seen in this figure where the initial point is iteratively projected onto each hyperplane converging to the intersection of the hyperplanes.
For the overdetermined case where the hyperplanes do not intersect in one point Kaczmarz’s method does not converge to a single point. As seen in Figure 5.2, the algorithm does not converge but is projecting each iteration from hyperplane to hyperplane. The projection onto hyperplanes can be formalized in the following definitions for the affine space and affine projection.

**Definition 5.1. Affine space $S_{r,y}$**

For a fixed vector $r \in \mathbb{R}^n$, a real number $y$ the affine space $S_{r,y}$ is defined by

$$S_{r,y} = \{x \in \mathbb{R}^n : \langle r, x \rangle = y \}$$

(5.3)

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors i.e.

$$\langle r, x \rangle = \sum_{i=1}^{n} r_i x_i \text{ for } r, x \in \mathbb{R}^n.$$ 

(5.4)

The affine space $S_{r,y}$ is also called a hyperplane and is a subspace of $\mathbb{R}^n$ if and only if $y = 0$. [3][10]

**Definition 5.2. Affine projection**

Given a vector $x$ and an affine space $S_{r,y}$ for some vector $r$ and some real number $y$, the affine projection of $x$ to $S_{r,y}$ is the vector $x^*$ in $S_{r,y}$ that is closest to $u$ amongst all vectors in $S_{r,y}$ [10].

The closest point $x^*$ has a connecting vector to $x$ that is orthogonal to the affine space.
and this connecting vector is parallel to $r$. According to this the following equation can be derived for the affine projection to the point $x^*$ with

$$x^* = x + r \frac{y - \langle x, r \rangle}{\langle r, r \rangle}.$$  \hfill (5.5)

The following equation is the generalization the previous equation for the undefined case when $\langle r, r \rangle = 0$ and applied to the reconstruction problem $y = Rx + e$

$$x^{(k)} = \begin{cases} 
  x^{(k)} + r \frac{y - \langle x^{(k)}, r_i \rangle}{\langle r_i, r_i \rangle}, & \langle r_i, r_i \rangle \neq 0 \\
  x^{(k)}, & \langle r_i, r_i \rangle = 0
\end{cases}$$ \hfill (5.6)

where $r_i$ is the $i$-th row of the projection matrix $R$. \footnote{1}

### 5.3 ART Algorithm

The ART algorithm is taking the previous reconstruction problem $y = Rx + e$ from (2.9) where $y \in \mathbb{R}^I$ is the measurement vector or ray-sum, $x \in \mathbb{R}^J$ is the image vector or estimate, $e \in \mathbb{R}^I$ is the error vector and $R \in \mathbb{R}^{I \times J}$ is the projection matrix with $r_i \in \mathbb{R}^J$ being the $i$-th row of the projection matrix. Since ART is an iterative algorithm, $x^{(k)}$ denotes the current estimate of the image vector and $x^{(k+1)}$ denotes the estimate of the next iteration.

One common variation of Kaczmarz’s method introduces a relaxation parameter $\lambda \in \mathbb{R}$ to control the amount of error correction and improve the performance. This relaxation parameter is in the range $0.0 < \lambda < 2.0$. This introduces the following equation for ART with relaxation parameter $\lambda$

$$x^{(k+1)} = \begin{cases} 
  x^{(k)} + \lambda r_i \frac{y - \langle x^{(k)}, r_i \rangle}{\langle r_i, r_i \rangle}, & \langle r_i, r_i \rangle \neq 0 \\
  x^{(k)}, & \langle r_i, r_i \rangle = 0
\end{cases}$$ \hfill (5.7)

Different relaxation parameters produce different behaviors with regard to the projection to the hyperplanes $S_{r_i,y_i}$. This means the projection of $x^{(k)}$ to the new estimate $x^{(k+1)}$ has the following geometric properties with regard to the hyperplane $S_{r_i,y}$ \footnote{1}
\[
\lambda < 0 \quad x^{(k+1)} \text{ moves away from } S_{r_i,y_i}
\]
\[
\lambda = 0 \quad \text{there is no movement of } x^{(k+1)}
\]
\[
0 < \lambda < 1 \quad \text{movement of } x^{(k+1)} \text{ is towards } S_{r_i,y_i} \text{ but does not reach it}
\]
\[
\lambda = 1 \quad x^{(k+1)} \text{ gets projected onto } S_{r_i,y_i}
\]
\[
1 < \lambda < 2 \quad x^{(k+1)} \text{ moves past } S_{r_i,y_i} \text{ but is closer than } x^{(k)} \text{ to hyperplane } S_{r_i,y_i}
\]
\[
\lambda = 2 \quad x^{(k+1)} \text{ gets reflected by } S_{r_i,y_i}
\]
\[
2 < \lambda x^{(k+1)} \text{ moves to other side of } S_{r_i,y_i} \text{ and further away from } S_{r_i,y_i} \text{ than } x^{(k)}
\]

The implementation of the ART algorithm with relaxation parameter can be seen in Algorithm 2. The initialization in the algorithm starts with setting the iteration number \( k = 0 \) and setting \( x^{(k)} = x^{(0)} \) to some initial point \( \bar{x} \). The main loop (lines 3-16) does each ART iteration. In the innermost loop (6-13) the vector \( x^{(k,i-1)} \) gets projected to each of the \( I \) hyperplanes where \( i \) iterates over each hyperplane. Before \( x^{(k,i-1)} \) gets projected the condition \( \langle r_i, r_i \rangle \neq 0 \) needs to be fulfilled in order to handle the undefined case \( \langle r_i, r_i \rangle = 0 \). The main loop (lines 3-16) gets executed as long as the condition \( C(x^{(k)}, y) \) is fulfilled, which represents some stopping criteria for the algorithm. This stopping criteria could the least squares distance \( ||R_x^{(k)} - y||_2^2 < \varepsilon \) for some arbitrary threshold \( \varepsilon \).

**Algorithm 2 ART Algorithm**

1: set \( k = 0 \)
2: set \( x^{(k)} = \bar{x} \)
3: while \( C(x^{(k)}, y) \) do
4: set \( i = 1 \)
5: set \( x^{(k,0)} = x^{(k)} \)
6: while \( i \leq I \) do
7: if \( \langle r_i, r_i \rangle \neq 0 \) then
8: \( x^{(k,i)} = x^{(k,i-1)} + \lambda r_i \frac{y_i - (x^{(k,i-1)}, r_i)}{\langle r_i, r_i \rangle} \)
9: else
10: \( x^{(k,i)} = x^{(k,i-1)} \)
11: end if
12: \( i = i + 1 \)
13: end while
14: \( x^{(k+1)} = x^{(k,I)} \)
15: \( k = k + 1 \)
16: end while

The iteration step of ART can be rewritten in order to be better suited for implementations by the following equation:
\[ x_j^{(k+1)} = x_j^{(k)} + \lambda_j r_{i,j} \frac{y_i - y_i^{(k)}}{\sum_{j'=1}^{J} r_{i,j'}} \]  

(5.8)

where \( r_{i,j} \) corresponds to the \( i \)-th row and \( j \)-th column of the projection matrix \( R \) and the calculation of the ray-sum \( y_i^{(k)} \) of the \( i \)-th ray after \( k \)-th iteration is written as follows:

\[ y_i^{(k)} = \langle x^{(k)}, r_i \rangle = \sum_{j=1}^{J} x_j^{(k)} r_{i,j}. \]  

(5.9)

For further details on ART refer [3], [10] and [17].

### 5.4 Variations of the Algebraic Reconstruction Technique

There are different variations of ART which are closely related to ART, but which are not applied in this thesis. One such variation is the simultaneous iterative reconstruction technique (SIRT), where the algorithm produces less noisy images than ART for the expense of slower convergence. The main idea of the algorithm is to average the projections to all hyperplanes in each iteration. Further resources can be found in [7][18].

Another version of ART, namely Simultaneous Algebraic Reconstruction Technique (SART) presented in the paper by Andersen and Kak in [19] is a compromise between ART and SIRT. The basic principle behind SART is that instead of projecting the estimate \( x^{(k)} \) onto one hyperplane, the estimate gets projected onto all hyperplanes simultaneously with specific weighting. The iteration procedure for SART is denoted as follows:

\[ x_j^{k+1} = x_j^{(k)} + \frac{1}{\sum_{i=1}^{I} r_{i,j}} \sum_{j'=1}^{J} r_{i,j'} \frac{y_i - \sum_{j'=1}^{J} r_{i,j'} x_j^{(k)}}{\sum_{j'=1}^{J} r_{i,j'}} \]  

(5.10)

A previous implementation with experiments of SART can be seen below in [20]. Further literature that covers SART can be found in [19] and [7].
6 Shearlet Transform

Shearlets are a multiscale framework used for the optimal encoding of several classes of multivariate data. In comparison to the wavelet transform, the shearlet transform extracts anisotropic singularities. For this, shearlets have parameters different from those of other waveforms. These parameters change the waveform in the scale, as well in orientation and in location with the ability to become very elongated. Thus, there are three different operators, namely a scaling operator to scale the analysing elements to the desired size, an orthogonal operator to change the orientation of the elements and a translation operator to put the elements on a specific place on the 2D plane [4].

The dilation operator $D_{A_a}$, where $a > 0$, is based on the following parabolic scaling matrices $A_a$

$$A_a = \begin{pmatrix} a & 0 \\ 0 & a^{1/2} \end{pmatrix}$$

and is defined in the same way as the dilation operator family in wavelets

$$D_{A_a}\psi(x) = |\det A_a|^{1/2}\psi(A_a^{-1}x)$$

As one of the major applications of shearlets is the analysis of cartoon-like images, introduced in [21], with the purpose to use a simplified model of natural images, in which the focus is on anisotropic features, and most notably, edges. The parabolic scaling is best suited for edges with $C^2$-regularities and this in a model class with discontinuities along those curves. This means that the shearing operator enables shearlets to approximate optimally sparse images and to extract the anisotropic features [4].

The next operator is the orthogonal operator. An intuitive example for the orthogonal would be a rotation operator, which changes the orientation of the waveform. This encounters the problem when the the operator is used in the digital setting, because it destroys the structure of the integer lattice $\mathbb{Z}^2$. Thus, the shearing operator $D_s$, where $s \in \mathbb{R}$ is used for this operation, depends on the following shearing matrix $S_s$

$$S_s = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$$

This shearing matrix works for the digital setting too, because it leaves the integer
lattice invariant, if \( s \) is set as an integer. The letter \( s \) is used for the slope rather than the angles [4].

The following last operator is the translation operator \( T_t \), which is defined like the translation operator in wavelet theory.

\[
T_t \psi(x) = \psi(x - t)
\] (6.4)

These three operators are forming the continuous shearlet system [4].

**Definition 6.1.** For \( \psi \in L^2(\mathbb{R}^2) \), the continuous shearlet system \( SH(\psi) \) is defined by the following equation

\[
SH(\psi) = \{ \psi_{a,s,t} = T_tD_A D_S \psi : a > 0, s \in \mathbb{R}, t \in \mathbb{R}^2 \} \] (6.5)

The corresponding continuous shearlet transform is given by the following mapping [4]:

\[
L^2(\mathbb{R}^2) \ni f \mapsto SH_f(a, s, t) = \langle f, \psi_{a,s,t} \rangle, f \in L^2(\mathbb{R}^2), (a, s, t) \in \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}^2 \] (6.6)

The continuous shearlet transform is an isometry and a reconstruction formula, if it fulfils certain conditions. Therefore, an admissible shearlet, which is also called continous shearlet is defined in the following way [4]:

**Definition 6.2.** If \( \psi \in L^2(\mathbb{R}^2) \) is satisfied then the integral converges,

\[
\int_{\mathbb{R}^2} \left| \hat{\psi}(\xi_1, \xi_2) \right|^2 \frac{\xi_2^2}{\xi_1^2} d\xi_2 d\xi_1 < \infty \] (6.7)

An advantage of admissible shearlets is their easy way to construct and that they include examples of admissible shearlets which are well localized. Furthermore, any function \( \psi \) whose Fourier transform \( \hat{\psi} \) is compactly supported away from the origin, is an admissible shearlet. An example for an admissible shearlet is the classical shearlet [4].

**Definition 6.3.** Let \( \psi \in L^2(\mathbb{R}^2) \) be defined by [4]:

\[
\hat{\psi}(\xi) = \hat{\psi}(\xi_1, \xi_2) = \hat{\psi}_1(\xi_1) \hat{\psi}_2(\frac{\xi_2}{\xi_1})
\] (6.8)
(a) Support of the Fourier transform of a classical shearlet. [4]

(b) Fourier domain support of several elements of the shearlet system, for different values. [4]

Figure 6.1: Classical shearlet in the fourier domain

where $\psi_1 \in L^2(\mathbb{R})$ is a wavelet with $\hat{\psi}_1 \in C^\infty(\mathbb{R})$ and supp $\hat{\psi}_1 \subseteq [-\frac{1}{2}, -\frac{1}{16}] \cup [\frac{1}{16}, \frac{1}{2}]$, and $\psi_2 \in L^2(\mathbb{R})$ is a bump function that satisfies $\hat{\psi}_2 \in C^\infty(\mathbb{R})$ and supp $\hat{\psi}_2 \subseteq [-1, 1]$. Then $\psi$ is called a classical shearlet.

The behaviour of the frequency domain and the support of several elements of the shearlet systems of classical shearlets are illustrated in Figure 6.1a and 6.1b. The x-axis represents the values of the Fourier transformed wavelet function and the y-axis represents the values of the Fourier transformed bump function [4].

A problem arises if the values of a function $\psi$ are close to the y-axis but the values of the wavelet function are close to zero. This means that $s \rightarrow \infty$ and then it is not possible that an edge along the x-axis in the spatial domain is detected. This could be a drawback for certain applications, but it is avoidable with cone-adapted continuous shearlet systems, where the coordinate system is split up into five different parts. The first part is a square around the origin which is there to analyse the low frequencies. The rest is cut into four different areas. Thus, the shearlets can be sheared by a maximum of $45^\circ$ and cannot elongate like in the typical shearlet example before. This requires the cone-adapted shearlet system to have three different components: the centre around the origin, a vertical and a horizontal shearlet, which is illustrated in Figure 6.2 [4].

The dependence of $\psi$ from $x \in \mathbb{R}, \psi(x)$ will be abbreviated by using the dot notation $\psi(\cdot) = \ldots \cdot \ldots$. The continuous shearlet system can be discretized. It is defined by the
following equation [4]:

$$SH(\psi) = \{ \psi_{j,k,m} = 2^{\frac{4}{3}j} \psi(S_k A_{2j} \cdot -m) : j, k \in \mathbb{Z}, m \in \mathbb{Z}^2 \}.$$  

(6.9)

The corresponding discrete shearlet transform of \( f \in L^2(\mathbb{R}^2) \) is then defined by

$$f \mapsto SH_\psi f(j, k, m) = \langle f, \psi_{j,k,m} \rangle, (j, k, m) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}^2.$$  

(6.10)

where \( \langle \cdot, \cdot \rangle \) denotes the inner product in \( L^2(\mathbb{R}^2) \). The used parameters \( j, k, m \) represent the scale index, the orientation index \( k \) and the position index \( m \). For more details on the shearlet transform and its attributes, see [4].

6.1 \( l_1 \)-Norm of the Shearlet Transform Criterion

In medical physics many researchers try to reduce the radiation dose of the scanning procedure as in the case of CT. There are different ways to decrease the radiation dose, namely to decrease the current of the emitting hardware or to reduce the x-ray pulse. However, there is also the possibility to decrease the number of projections needed for the reconstruction. There are various regularization approaches to aid this. One possible choice is the \( l_1 \)-norm of any sparse transformation, which was suggested by two research groups, see [22] and [23]. For the sparse transformation, the shearlet transform is a reasonable choice for images, because it is a multiscale geometric
analysis method that can extract anisotropic features. The goal is to use the $l_1$-norm of the shearlet transform as a secondary criterion with any superiorized reconstruction algorithm which is defined in the following definition.

**Definition 6.4. $l_1$ Norm of the Shearlet Transform (SH)**

The SH-norm is defined as the $l_1$-norm of the shearlet transform as the following function $\phi : \mathbb{R}^J \mapsto \mathbb{R}$:

$$
\phi(x) = ||Sx||_1 = \sum_{i=1}^{I} \left| \sum_{j=1}^{J} s_{ij} x_j \right|,
$$

(6.11)

where the $S = I \times J$ matrix, such as $Sx$, is the discrete shearlet transform [4], [5].

### 6.2 Nonascending Vector for $l_1$-Norm of the Shearlet Transform

Criterion Version 1

The nonascending vector has to be defined for the new secondary criterion. Therefore, the theory of the nonascending vectors defined in the superiorization methodology has to be applied on SH.

Edgar Garduño\(^1\) and Gabor T. Herman\(^2\) observed that $\phi(x)$ is a convex function and applied the Definition 3.10 of the nonascending vector on $\phi$. This requires the variable $\zeta$, to avoid numerical difficulties if a shearlet coefficient is close to zero. $\zeta$ should be chosen as a small positive real number, but the actual value is chosen by the user of the secondary criterion [5].

Next, for any $x \in \mathbb{R}^J$, three sets have to be defined:

$$
P_{\zeta} = \{ i | 1 \leq i \leq I \text{ and } \sum_{j=1}^{J} s_{ij} x_j > \zeta \},$$

(6.12)

$$
N_{\zeta} = \{ i | 1 \leq i \leq I \text{ and } \sum_{j=1}^{J} s_{ij} x_j < \zeta \},$$

(6.13)

$$
Z_{\zeta} = \{ i | 1 \leq i \leq I \text{ and } \left| \sum_{j=1}^{J} s_{ij} x_j \right| \leq \zeta \}$$

(6.14)

The derivative of (6.11) with respect to $x$ is computed by $x$ to get the gradient. The

---

\(^1\)Former member of Digital Imaging Group (DIG)

\(^2\)Head of DIG at the Graduate Center of the City University of New York
$|x|$ results in the sign function in the derivative, which is defined as followed

$$
\text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0. \\
0, & \text{if } x = 0. \\
1, & \text{if } x > 0. 
\end{cases}
$$

(6.15)

The sum $\sum_{j=1}^{J}$ can be split into two sums and in the combination with the sets defined before in (6.12), (6.13) and (6.14), the nonascending vector is defined by using $g = (g_1, g_2, \ldots, g_J)^T \in \mathbb{R}^J$, where

$$
g_j = \begin{cases} 
0, & \text{if } Z_\zeta(x) \neq 0 \\
(\sum_{i \in P_\zeta} s_{ij}) - (\sum_{i \in N_\zeta} s_{ij}), & \text{otherwise}
\end{cases}
$$

(6.16)

This result leads to Algorithm 3, which is used to create the nonascending vector in superiorization.

**Algorithm 3** procedure NONASCENDINGSHEARLET1($x, d, \zeta$) [5]

1: set $\text{cont} = \text{true}$
2: set $d = 0$
3: set $a = Sx$
4: set $i = 0$
5: while $i < I$ and $\text{cont} == \text{true}$ do
6: set $i = i+1$
7: if $|a_i| \leq \zeta$ then
8: set $g = 0$
9: set $\text{cont} = \text{false}$
10: else
11: if $a_i > \zeta$ then
12: set $z_i = 1$
13: else
14: set $z_i = -1$
15: end if
16: end if
17: if $\text{cont} == \text{true}$ then
18: set $g = S^\dagger z$
19: end if
20: if $\|g\|_2 > \zeta$ then
21: set $d = -\frac{g}{\|g\|_2}$
22: end if
23: end while
In line 3 the shearlet transform is applied to the current values of the reconstruction. In line 5 a while-loop iterates over all shearlet coefficients as long as none of them is below $\zeta$. If a shearlet coefficient is below $\zeta$, the nonascending vector $d$ is set to 0 and returns it. If the value of the shearlet coefficient is above $\zeta$ and positive, the value 1 is assigned to the shearlet coefficient in the $z$-vector. If the value of the shearlet coefficient is above $\zeta$ and negative, the value -1 is assigned to the shearlet coefficient in the $z$-vector. The values 1 and -1 result from the differentiation of the $l_1$-norm and represent the directionality of the gradient, which increases the sparsity. In the next step, the inverse shearlet transform is applied on the $z$-vector, which is then the gradient. At the end of the procedure, the negative of the gradient is chosen as the nonascending direction and is normalized [5].

This procedure was not explicitly created for the shearlet matrix $S$, but rather for any sparse transform [5].
CHAPTER 7. PROXIMAL OPERATOR FOR SHEARLETS

7 Proximal Operator for Shearlets

The focus in this chapter is on the optimization of non-differentiable functions like the $l_1$-norm of the shearlet transform as discussed in Section 6.2. Further this chapter is introducing an alternative to the previously shown nonascending vector for the $l_1$-norm of the shearlet transform of Section 6.2. This alternative solution proposed by Marcelo V. W. Zibetti\(^1\) is based on the proximal operator which is discussed in this chapter.

### 7.1 Gradients, Subgradients and Subdifferentials

The gradient of a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ which is denoted by $\nabla f$ is given by

$$
\text{grad}(f) = \nabla f = \begin{pmatrix}
\frac{\partial f}{\partial x_1} \\
\vdots \\
\frac{\partial f}{\partial x_n}
\end{pmatrix}
$$

(7.1)

where $x \in \mathbb{R}^n$. If the function $f$ is differentiable, then the gradient represents the direction for the steepest ascending slope. If $f$ represents the cost function of a minimization problem, then the negative direction of the gradient is taken. This is the basic concept for the gradient methods [24] like steepest descent [25] or the Armijo gradient method [26].

For example the least-squares problem $Ax = b$ of the form

$$
\text{minimize} \ ||Ax - b||^2_2
$$

(7.2)

where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ has the gradient

$$
\nabla ||Ax - b||^2_2 = 2A^T(Ax - b)
$$

(7.3)

For non-differentiable functions like the function $f$ in Figure 7.1, there are points where a unique gradient cannot be obtained. This can be seen at the position of the point $x_2$ where multiple gradients like $g_2$ and $g_3$ are possible. One such gradient is called a subgradient.

---

\(^1\)Visiting Scholar at the Graduate Center (Date: June 2016)
Definition 7.1. Subgradient
A vector $g \in \mathbb{R}^n$ is a subgradient of the function $f$, when for some fixed $x_0 \in \mathbb{R}^n$ and all $x \in \mathbb{R}^n$

$$f(z) \geq f(x) + g^T(z - x)$$

(7.4)

where $f : \mathbb{R}^n \to \mathbb{R}$ [27].

Such a subgradient is part of a subdifferential $\partial f$ of a function $f$.

Definition 7.2. Subdifferential
The subdifferential $\partial_x f$ of a function $f$ of $x$ is the set of all subgradients $g \in \mathbb{R}^n$

$$\partial_x f = \{ g \mid f(z) \geq f(x) + g^T(z - x) \}$$

(7.5)

for all $x \in \mathbb{R}^n$, where $f : \mathbb{R}^n \to \mathbb{R}$ [27].

When taking the one-dimensional function $f(x) = |x|$, the subdifferential $\partial_x f$ of $f$ is the following sign function in (7.6) which is shown in Figure 7.2. Figure 7.2a shows the function $f$ for which the corresponding subgradient is shown in Figure 7.2b.

Definition 7.3. Sign function
The sign function is defined as the set of subgradients $\partial_x$ of the absolute function
CHAPTER 7. PROXIMAL OPERATOR FOR SHEARLETS

(a) Absolute function $|x|$ in one dimension

(b) Subgradient of $|x|$ in one dimension (adapted from [27, p.2])

Figure 7.2: Subgradient of the absolute function $|x|$ in one dimension

$f(x) = |x|$ as follows

\[ \partial_x|x| = \text{sign}(x) = \begin{cases} 
-1 & x < 0 \\
1 & x > 0 \\
[-1, 1] & x = 0 
\end{cases} \tag{7.6} \]

One algorithm for minimizing a non-differentiable convex function $f$ is the subgradient method [28]. For a convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$, the subgradient method has the following iteration step

\[ x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)} \tag{7.7} \]

where $x^k$ is the $k$-th iterate, $g^{(k)}$ is any subgradient of $f$ at the point $x^{(k)}$ fulfilling the condition in (7.4) and where $\alpha_k > 0$ is the $k$-th step size. In the case that $f$ is differentiable, the subgradient method becomes the gradient method with $g^{(k)} = \nabla f(x^{(k)})$. The difference then is in the step size $\alpha_k$. The subgradient method can use different step size rules, like constant step size ($\alpha_k = h$ where $h > 0$), constant step length and other rules as stated in [28].

One problem with the subgradient method is, that it is not a descent method, that means that $x^{(k+1)}$ is not necessarily smaller than $x^{(k)}$. This can be a problem for the non-ascending vector in the superiorization methodology (Chapter 3), because the non-ascending vector might not steer the algorithm in each iteration to a superior solution.
with regard to the secondary criterion. When applying the subgradient method, the best point after each iteration \( k \) is tracked by

\[
f_{\text{best}}^{(k)} = \min\{f_{\text{best}}^{(k-1)}, f(x^{(k)})\}.
\]

(7.8)

This means that \( f_{\text{best}}^{(k)} \) is the best objective value found after \( k \) iterations. In descending methods like the gradient descent method there is no need for tracking the best objective value after each iteration [28].

## 7.2 Proximal Algorithms

The proximal algorithms as stated in [6] are a class of algorithms for solving convex optimization problems. In contrast to the Newton’s method which is well suited for unconstrained smooth minimization problems of modest size, the proximal algorithms are suited for non-smooth, constrained, large-scale minimization problems. They are generally applicable to a large set of problems and particularly well suited for large or high-dimensional datasets. Proximal algorithms involve the evaluation of the proximal operator of a function, which involves solving a convex optimization problem. Proximal operators can also be viewed as generalized projections which are minimizing convex functions over convex sets.

**Definition 7.4. Closed proper convex function**

The epigraph of a closed proper convex function \( f: \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\} \) or the set of points lying on or above its graph

\[
\text{epi} \; f = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq t\}
\]

(7.9)

is a nonempty closed set. Further the effective domain of \( f \) is given by

\[
\text{dom} \; f = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}
\]

(7.10)

which corresponds to the set of points for which \( f \) takes on finite values [6].

For such a closed proper convex function the proximal operator is defined as follows

**Definition 7.5. Proximal operator**

The proximal mapping or the proximal operator \( \text{prox}_f: \mathbb{R}^n \mapsto \mathbb{R}^n \) of a convex function
Figure 7.3: Evaluation of the proximal operator for different points, adapted from [6, p.125]

The function in (7.11) on the right side is convex and not for every point infinite which means that it has a unique minimizer for every $y \in \mathbb{R}^n$. It is common to encounter the proximal operator of the scaled function $\lambda f$ with parameter $\lambda > 0$, which is shown in the following equation

$$\text{prox}_{\lambda f}(x) = \arg \min_y \left\{ f(y) + \frac{1}{2\lambda} ||y - x||_2^2 \right\}$$

(7.12)

One important property of the proximal operator which will be necessary later on in this chapter, is that if $f$ is fully separable, meaning that $f(y) = \sum_{i=1}^{n} f_i(y_i)$, then the proximal operator can be reduced by evaluating proximal operators of scalar functions [6] as in the following equation

$$(\text{prox}_f(x))_i = \text{prox}_{f_i}(x_i)$$

(7.13)

The application of the proximal operator is shown in Figure 7.3, where the thin black
lines correspond to the level curves of a convex function \( f \) and the thick black line corresponds to the boundary of its domain. After the application of the proximal operator on each point in the figure, the point is moved towards the minimum of the function. The points outside of the domain move to the boundary of the domain. The resulting points at the tip of the arrows are each called the proximal point of \( y \) with respect to \( f \). The extent to which the points are mapped by the proximal operator is controlled by the parameter \( \lambda \). This parameter can be also interpreted as a relative weight or trade-off parameter between the proximal mapping and the function \( f \) [6].

7.3 Proximal Operator for the \( l_1 \)-norm

In order to compute the proximal mapping of the \( l_1 \)-norm of the shearlet transform the first step is to derive the proximal mapping of the \( l_1 \) norm function \(|x|_1\) which is the solution \( \hat{y} \) of the equation

\[
\hat{y} = \text{prox}_{||y||_1, \beta} = \arg\min_y \left\{ \frac{1}{2} ||y-x||^2 + \beta ||y||_1 \right\},
\]

which can be separated into each dimension due to the property (7.13) by the proximal operator into

\[
\hat{y}_i = \text{prox}_{|x_i|, \beta} = \arg\min_{y_i} \left\{ \frac{1}{2} |y_i-x_i|^2 + \beta |y_i| \right\}.
\]

From this separation, the solution \( \hat{y} \) can be calculated by finding the subdifferentials of \( 0 \in \partial_{y_i} \left( \frac{1}{2} |y_i-x_i|^2 + \beta |y_i| \right) \) as follows

\[
\begin{align*}
0 &\in \partial_{y_i} \left( \frac{1}{2} |y_i-x_i|^2 + \beta |y_i| \right) \\
0 &\in (y_i-x_i) + \beta \text{sign}(y_i) \\
x_i &\in y_i + \beta \text{sign}(y_i) \\
x_i &\in (1 + \beta \text{sign})(y_i) \\
\hat{y} &\in (1 + \beta \text{sign})^{-1}(x_i) \\
\hat{y} &\in \text{ST}_\beta(x_i)
\end{align*}
\]

where the shrinkage or soft-thresholding operator of the derivation in (7.16) is defined
CHAPTER 7. PROXIMAL OPERATOR FOR SHEARLETS

Figure 7.4: Soft-thresholding function $\text{ST}_\beta(x_i)$

\[
\text{ST}_\beta(x_i) = \begin{cases} 
  x_i - \beta & x_i > \beta \\
  x_i + \beta & x_i < -\beta \\
  0 & |x_i| < \beta 
\end{cases} \tag{7.17}
\]

The graph of the shrinkage operator can be seen in Figure 7.4. This derivation will be applied in the next step to minimize the $l_1$-norm of the shearlet transform. The proximal operator of $l_1$-minimization problems finds its use in image de-noising as in [29].

The Figures 7.5 depict a single iteration step of both the proximal operator and the subgradient method applied to the $l_1$-minimization. It can be seen that the proximal operator in Figure 7.5b does not move further than the corners and follows the corners down to the minimum with increasing step sizes. The subgradient method in Figure 7.5a on the other hand does not stop at the edges and can be even ascending for larger step sizes.

7.4 Algorithm for the Nonascending Proximal Shearlet

This algorithm is an alternative to the previously discussed algorithm in Section 6.2 which is based on the subgradient. This alternative shown in Algorithm 4 is based on the previously discussed proximal operator, but needs to be adapted in order to function in the superiorization methodology from Chapter 3.

In Algorithm 4 the $\beta$ value, which corresponds to the step size is calculated by the
(a) Single superiorization step with the negative subgradient for the $l_1$-norm with different stepsizes

(b) Single superiorization step with the proximal operator for the $l_1$-norm with different stepsizes

Figure 7.5: Subgradient of the absolute function $|x|$ in one dimension
following equation which describes the step size of the proximal operator

\[ \beta = \beta_0 \alpha^l \]  \hspace{1cm} (7.18)

where \( \beta_0 \) and \( \alpha \) are the input parameters for the superiorization. The value \( l = k \) since the superiorization steps \( N = 1 \). This is because the proximal operator is performed for only one superiorization step. The soft-thresholding operator from (7.17) can be seen in lines 6-14, which is performed for each value \( a_i \) of the vector \( a \).

In line 17 the nonascending vector \((g-x)\) is divided by \( \beta \), which comes from the previous Algorithm 1 where the nonascending vector is multiplied by \( \beta \). The reason for this is because the step size \( \beta \) is already incorporated into the vector \((g-x)\) through the soft-thresholding operator. This makes sure that the proximal operator can be incorporated into the superiorization.

The proximal algorithm of the nonascending shearlet transform is referred in this thesis as the Proximal Operator of the \( l_1 \) Norm of the Shearlet Transform (PSH) superiorization.

**Algorithm 4** procedure PROXIMALNONASCENDINGSHEARLET(x,d,\( \beta \))

1: set \( d = 0 \)
2: set \( a = Sx \)
3: set \( i = 0 \)
4: while \( i < I \) do
5:   set \( i = i + 1 \)
6:   if \( |a_i| \leq \beta \) then
7:     set \( z_i = 0 \)
8:   else
9:     if \( a_i > \beta \) then
10:        set \( z_i = a_i - \beta \)
11:     else
12:        set \( z_i = a_i + \beta \)
13:     end if
14:   end if
15: end while
16: set \( g = S^\dagger z \)
17: set \( d = (g - x)/\beta \)

The proximal algorithm of the nonascending shearlet transform is referred in this thesis as the Proximal Operator of the \( l_1 \) Norm of the Shearlet Transform (PSH) superiorization.
8 Experiments

The focus of this chapter are the experiments for the concepts and theory described in the previous chapters. The first part of the chapter introduces the SNARK14 framework which is the software for the experiments. Afterwards the scope of the performed experiments is explained, which describes the data generation and the stopping criteria for the experiments. The first two experiments focus on the comparison of different parameters with their corresponding results and the final experiment is in form of an anecdotal study for all selected algorithms.

8.1 SNARK14

SNARK14 provides for researchers a framework for image reconstruction of 2D images from 1D projections. Multiple frequently used algorithms are already implemented in SNARK14 and it provides the opportunity to evaluate the performance of the algorithms according to different metrics and figures of merit. Researchers can also extend SNARK14 by implementing their own algorithms by using the provided interfaces and classes in SNARK14. The process in SNARK14 consists of

- **Phantom Generation and Simulation**
  SNARK14 provides the tools to produce the phantom and the corresponding projection data. SNARK14 can either use previously created data or can be configured manually. The configuration allows the creation of geometrical shapes and densities of the phantom and the geometric objects inside the phantom. Besides the configured shapes, it allows to introduce different forms of noise to the data to simulate a real scanner.

- **Reconstruction**
  In this phase the reconstruction algorithm can be chosen which will be performed on the data. This can be either an existing algorithm like ART or a user defined algorithm. SNARK14 generates files during the reconstruction, where all iterations of the reconstruction process are saved.

- **Analysis**
  In this phase the results from the previously generated files are evaluated, which can either be done by comparing the used metrics or by visually comparing the resulting image reconstructions of the projections.
SNARK14 is described in further detail in [2]. The framework is distributed in a VirtualBox version in order to be usable in many application areas and to be transportable to a large variety of machines at the expense of efficiency. SNARK14 offers additionally two further tools, namely SNARK14UserDefined and SNARK14 Experimenter.

8.2 SNARK14UserDefined

SNARK14 allows besides the already implemented algorithms and classes, the additional use and testing of its own implemented algorithms. This is covered by SNARK14UserDefined, which is a set of classes and interfaces which enable the user to extend SNARK14 by own procedures and algorithms. This enables the user to implement reconstruction algorithms, termination tests and superiorization-related functions.

The user can implement the iterative reconstruction algorithm and its primary stopping criterion which can be used with the SNARK14 framework. When using superiorization, the user can implement a custom secondary criterion, as well as a custom non-ascending vector. This was used for the implementation of the non-ascending vector and secondary criterion of the PSH algorithm, which was discussed in Chapter 7. The whole process for using SNARK14UserDefined is discussed in further detail in [2].

8.3 SNARK14 Experimenter

SNARK14 Experimenter is a tool in SNARK14 which enables the user to perform statistical comparisons such as a task-oriented comparison of algorithm performance which is discussed in Chapter 9. The details for using the SNARK14 Experimenter can be found in [2].

In order to perform the evaluation on a single algorithm or a set of different algorithms, it is required to run the selected algorithms on a large set of randomly generated phantoms and using a Figure of Merit (FOM) for the results. This evaluates the efficacy of the algorithms. The procedure of the comparative evaluation by SNARK14 Experimenter is as follows.

- Generation of randomly created phantoms from statistically described collections of phantoms with their projection data.
- Reconstruction from the projection data with each selected algorithm.
- Measure the image quality of each reconstructed image by measuring the image
quality with a FOM.

- Calculate the statistical significance based on the previously calculated FOM which is then used to reject the hypothesis that the methods are equally good for the reconstruction of the image.

8.4 Implementation

In SNARK14, the algorithms such as ART or SART and the superiorization procedures such as TV superiorization can be switched without directly implementing a new algorithm. This is done by expressing an algorithm with a desired superiorization procedure in an input file that specifies the combined algorithm for the reconstruction. The algorithms for the following experiments are either existing algorithms in SNARK14 or implemented with SNARK14UserDefined.

The algorithms for the experiments, namely Filtered Back Projection (FBP), ART and TV superiorization were already part of SNARK14, whereas the algorithm for the SH superiorization was realized by Edgar Garduno with SNARK14UserDefined. The SH superiorization is using the Discrete Shearlet Transform implemented by Bert Vandeghinste and Bart Goosens. The proposed algorithm PSH superiorization is also using the Discrete Shearlet Transform and was realized with SNARK14UserDefined.

The secondary criteria for the used algorithms are minimizing the SH norm which was defined in Section 6.1 and the TV norm, where the TV norm is defined as follows.

**Definition 8.1.** *Total Variation norm*

The total variation norm is defined by the following function

$$TV(x) = \sum_{i \in T} \sqrt{(x_{r(i)} - x_i)^2 + (x_{b(i)} - x_i)^2}$$ (8.1)

where $T$ is the set of all indices of pixels except the ones in the rightmost column and the bottom row of a square $n \times n$ pixel image and where $x_{r(i)}$ is the pixel right to the current pixel and $x_{b(i)}$ the one below it [5].

8.5 Data Generation

The following experiments use ART as the iterative algorithm for the image reconstruction experiments which was introduced in Chapter 5. The FBP algorithm is additionally used for comparison. The generated data used for the experiments is
based on the settings for the phantom and simulated CT scanner from the paper [5] for comparison. The used projection data for the generated phantom from the simulated CT scanner uses the following settings:

- Divergent projection data
- 110.735 cm source-to-detector distance
- 78 cm source-to-center-of-rotation distance
- View angles with 693 rays per view
- 0.0533 cm detector spacing
- Simulated projection data using integrals over the original structures instead of the digitized versions of them
- Simulated using 1000000 photons for estimating each line integral
- 180, 360 and 720 numbers of views (number of projections).

The generated phantom has a digitized resolution of 485 by 485 and consists of different geometric shapes and local inhomogeneities which represent the neuroanatomical structure of the brain. The local inhomogeneities are generated using a Gaussian random variable. The small circles scattered inside the brain represent tumours which play also an important role for evaluating the efficacy of the selected reconstruction algorithms which is later addressed in Chapter 9. The digitization of the resulting phantom can be seen in Figure 8.1. Each value of the phantom maps the density of the matter inside the phantom. These densities are mapped in the range of 0.204 to 0.21675, where the densities below 0.204 are represented with black and the densities above or equal to 0.21675 are represented with white. The reason for this range is the high density of the skull enclosing the brain.

### 8.6 Stopping Criteria

The stopping criterion is necessary to indicate when the iterative reconstruction algorithm reaches an acceptable solution and stops the procedure. This stopping criterion is denoted by the proximity function $\mathcal{P}_{\tau}(x) : \Omega \mapsto \mathbb{R}_+$ which is discussed previously in Chapter 3. The proximity function for the experiments is based on the $l_2$-norm which is given by the proximity function.
Figure 8.1: Digitization of the generated Phantom

<table>
<thead>
<tr>
<th>Number of Angles</th>
<th>180</th>
<th>360</th>
<th>720</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td></td>
<td>y - Rx</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.1: Calculated residuals from filtered back-projection reconstructions for 180, 360 and 720 views

\[
P_{rt}(x) = ||y - Rx^{(k)}||_2, \tag{8.2}\]

where $y \in \mathbb{R}^L$ is the measurement vector, $R \in \mathbb{R}^{L \times J}$ is the system matrix and $x^{(k)} \in \mathbb{R}^J$ is the current estimate for the reconstructed image.

In order to produce comparable results besides choosing the same phantom, the same stopping criteria amongst all algorithms in the experiments are chosen for each view as in [5]. These stopping criteria are created by running FBP with the same phantom for different numbers of views. The resulting residual after calculating the FBP can be seen in Table 8.1 and the resulting reconstructions can be seen in Figure 8.2. These residuals are used as stopping criteria for the following simulations and evaluations.

### 8.7 Comparison of Shearlet and TV Superiorized ART

The first experiments were done in order to compare the behaviour of the SH norm and the TV norm for the SH superiorized ART and TV superiorized ART. Through experiments by Gabor T. Herman and Edgar Garduño they showed, that the TV superiorized ART performed better than the SH superiorized ART with regard to the SH norm. This behaviour is investigated in this section because it is unexpected. Different
experiments for a set of parameters have been realized to look into the performance of both the TV norm and the SH norm for the two algorithms. The setup of the experiment is as follows. The two algorithms (TV superiorized ART, SH superiorized ART) will be computed for the following set of $N$ superiorization steps and fixed $\alpha$ and $\beta$ values

- $N = [5, 10, \ldots, 60]$
- $\alpha = 0.9999$
- $\beta = 0.03$

where $\alpha$ and $\beta$ have been taken from the paper [5].

The results for these experiments can be seen in Figure 8.3 and Figure 8.4. Figure 8.3 displays the relationship of the superiorization steps $N$ with regard to the TV norm for both algorithms. This behaviour shows that the TV superiorized ART performs better than the SH superiorized ART with regard to the TV norm, which was expected due to the fact that the TV superiorization aims to decrease the TV-norm.

Figure 8.4 shows the direct comparison to the previous figure with regard to the SH-norm. In this figure it can be seen that SH superiorized ART needs higher superiorization steps than TV superiorized ART in order to perform better with regard to the SH-norm. This means that TV superiorized ART performs better as the SH superiorized ART with lower superiorization steps $N$. Also it can be seen that with the increase of the views the SH superiorized ART needs higher number of superiorization steps $N$.

The reconstructed phantoms of both algorithms for superiorization steps $N = 10$, \ldots
Figure 8.3: Comparison of SH superiorized ART and TV superiorized ART with regard to their TV norm for 180, 360 and 720 views

Figure 8.4: Comparison of SH superiorized ART and TV superiorized ART with regard to their SH norm for 180, 360 and 720 views

$N = 30$ and $N = 50$ are displayed in Figure 8.5 and Figure 8.6. Figure 8.5 shows the influence of the increase in the superiorization steps $N$ using the TV superiorized ART and Figure 8.6 shows on the other hand the influence of $N$ by using the SH superiorized ART.

Figure 8.5: Reconstructions with TV superiorized ART for $N = 10$ (left), $N = 30$ (middle) and $N = 50$ (right)
8.8 Parameter Comparison of Proximal Shearlet Superiorized ART

The goal of this experiment is to investigate the behaviour of the PSH superiorized ART which was previously discussed in Chapter 7. This algorithm is tested with various parameters $\alpha$ and $\beta_0$ as stated in (7.18) in order to choose appropriate parameters for the following anecdotal experiment. For simplicity $\beta$ is corresponding to $\beta_0$ from (7.18) in this section.

The experiment consists of different values for $\alpha$ in the range between 0.3 and 0.9 and different values for $\beta$ in the range between 0.01 and 5. From these ranges of $\alpha$ and $\beta$ all combinations are simulated and evaluated. The same projection data is used as in the previous experiments and as discussed in Section 8.5 and Section 8.6. The experiments are evaluated for the TV norm, the SH norm and the number of iterations to reach the defined stopping criteria.

Figure 8.7 illustrates the influence of the simulated parameter combinations $\alpha$ and $\beta$ for the SH norm for 180, 360 and 720 views. Since the goal is to minimize the norm, it can be seen that higher $\alpha$ values are in favour of a minimization of the SH norm, whereas a higher $\beta$ have only noticeable influence with combination of $\alpha$ values in the lower range, namely between 0.3 and 0.5. This trend is also visible with the increase of the views. This seems to suggest to use lower $\beta$ values and reasonable high $\alpha$ values. Figure 8.8 shows a visually similar result on the visualizations for the TV norm, showing a similar relationship of the parameter choice of $\alpha$ and $\beta$ with regard to a minimization of the TV norm.

In order to compare the results based on the duration which the computations needed,
Figure 8.7: The SH norm for a set of given $\alpha$ and $\beta$ values for 180, 360 and 720 views

Figure 8.8: The TV norm for a set of given $\alpha$ and $\beta$ values for 180, 360 and 720 views

The number of iteration is evaluated for each parameter combination of $\alpha$ and $\beta$. This can be seen in Figure 8.9. Here it can be clearly seen that the number of iterations increases as $\alpha$ increases. For higher $\beta$ the increase in number of iterations is even higher. This can be clearly seen for $\beta$ values in combination with large $\alpha$ values, even though the larger $\beta$ values have visibly no large influence to the minimization of the corresponding norm as seen in Figure 8.7 and Figure 8.8.

The reconstructed images for this experiment can be seen in Figure 8.10, which shows the reconstructed phantoms for the parameter combinations $\alpha$ and $\beta$ in the case of 180 views. In this figure it can be clearly seen that the $\beta$ parameter does not have a visually significant influence on the resulted reconstructions, whereas an increasing $\alpha$ value has a more pronounced effect on the reconstructed phantom. Based on the visual quality, the midrange of $\alpha = 0.6$ seems to have a visually better reconstruction in terms of visibility of the tumours, but this will be evaluated in Chapter 9.
Figure 8.9: The number of iteration needed for a set of given $\alpha$ and $\beta$ values for 180, 360 and 720 views.

8.9 Anecdotal Experiment

The focus of this section is to perform and evaluate an anecdotal experiment for various algorithms for 180, 360 and 720 views. The used algorithms involve

- Filtered Backprojection (FBP) with sinc window with linear interpolation as described in [3]
- Pixel ART, which is a pixel based ART algorithm
- TV superiorized ART, SH superiorized ART with both having the parameters $\alpha = 0.9999$, $\beta = 0.03$, $\zeta = 10^{-20}$ and $N = 40$
- PSH superiorized ART, which will be separated into the following three different parameter combinations for comparison
  - PSH1 ART with parameters $\alpha = 0.9$ and $\beta = 0.03$
  - PSH2 ART with parameters $\alpha = 0.6$ and $\beta = 0.03$
  - PSH3 ART with parameters $\alpha = 0.3$ and $\beta = 0.03$

The parameters for the PSH superiorized ART are chosen with regard to the results of the previous experiments in Section 8.8. These three parameter sets are the main focus of this anecdotal experiment.

The results can be seen in Table 8.2 which compare all the algorithms to their TV-norm, SH-norm and number of iterations needed to complete the image reconstruction below the previously defined stopping criteria from Section 8.6. The corresponding reconstructed images for 180 views can be seen in Figure 8.11 and Figure 8.12. Figure
8.11 shows the reconstructed phantoms reproduced for the parameters and algorithms as described in [5] which focuses mainly on the comparison of TV superiorized ART and SH superiorized ART in their efficacy.

The proposed three versions of PSH are shown in Figure 8.12. As previously shown in Section 8.8 the results seem to perform visually better than the SH superiorized ART, but the results do not perform better according to either the TV-norm nor SH-norm as seen in Table 8.2.

<table>
<thead>
<tr>
<th>Views</th>
<th>Measure</th>
<th>FBP</th>
<th>Pixel ART</th>
<th>TV ART</th>
<th>SH ART</th>
<th>PSH1 ART</th>
<th>PSH2 ART</th>
<th>PSH3 ART</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>$</td>
<td></td>
<td>y - Rx</td>
<td></td>
<td>_2$</td>
<td>3.6380</td>
<td>3.0088</td>
<td>3.5388</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td></td>
<td>x</td>
<td></td>
<td>_{TV}$</td>
<td>3007.6751</td>
<td>3565.0785</td>
<td>926.5725</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td></td>
<td>Sx</td>
<td></td>
<td>_1$</td>
<td>7289.0860</td>
<td>6779.7905</td>
<td>4935.7246</td>
</tr>
<tr>
<td>Iterations</td>
<td>1</td>
<td>15</td>
<td>20</td>
<td>35</td>
<td>45</td>
<td>19</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>360</td>
<td>$</td>
<td></td>
<td>y - Rx</td>
<td></td>
<td>_2$</td>
<td>4.0314</td>
<td>3.9267</td>
<td>3.9089</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td></td>
<td>x</td>
<td></td>
<td>_{TV}$</td>
<td>1797.8089</td>
<td>3259.0070</td>
<td>955.4885</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td></td>
<td>Sx</td>
<td></td>
<td>_1$</td>
<td>5840.9464</td>
<td>6672.6732</td>
<td>5278.3822</td>
</tr>
<tr>
<td>Iterations</td>
<td>1</td>
<td>12</td>
<td>15</td>
<td>23</td>
<td>42</td>
<td>16</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>720</td>
<td>$</td>
<td></td>
<td>y - Rx</td>
<td></td>
<td>_2$</td>
<td>5.9793</td>
<td>5.3769</td>
<td>5.7150</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td></td>
<td>x</td>
<td></td>
<td>_{TV}$</td>
<td>1331.3471</td>
<td>2900.1962</td>
<td>1016.4669</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td></td>
<td>Sx</td>
<td></td>
<td>_1$</td>
<td>5389.4274</td>
<td>6444.8295</td>
<td>5227.7271</td>
</tr>
<tr>
<td>Iterations</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>28</td>
<td>10</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.2: Overview of the measurements for the anecdotal experiment
(a) $\alpha = 0.3$ and $\beta = 0.05$
(b) $\alpha = 0.3$ and $\beta = 0.5$
(c) $\alpha = 0.3$ and $\beta = 5$
(d) $\alpha = 0.6$ and $\beta = 0.05$
(e) $\alpha = 0.6$ and $\beta = 0.5$
(f) $\alpha = 0.6$ and $\beta = 5$
(g) $\alpha = 0.9$ and $\beta = 0.05$
(h) $\alpha = 0.9$ and $\beta = 0.5$
(i) $\alpha = 0.9$ and $\beta = 5$

Figure 8.10: Reconstructed phantoms for parameter combinations of $\alpha$ and $\beta$

(a) FBP    (b) Pixel ART    (c) TV superiorized ART    (d) SH superiorized ART

Figure 8.11: Comparison of reconstructed phantoms for 180 views for the reproduced anecdotal experiment from [5]
Figure 8.12: Comparison of reconstructed phantoms for 180 views for SH and PSH superiorized ART
9 Statistical Comparison

In order to evaluate the efficacy of a reconstruction algorithm, it is necessary to measure its performance in a clinical diagnostic situation. The methodology used for evaluation in this thesis is Statistical Hypothesis Testing (SHT) which allows the evaluation of the relative efficacy of the different reconstruction methods for a given task.

9.1 Task-Oriented Comparison of Algorithm Performance

The SHT is performed with the help of the SNARK14 Experimenter which is previously discussed in Section 8.3. The steps as in [3] that need to be taken for a SHT of an experiment are as follows:

1. Generation of random samples of different statistically generated phantoms and their projection data. In this project the projection data is created from generated phantoms as described in Section 8.5.

2. Reconstruction from the generated projection data by each selected algorithm.

3. Assignment of a FOM to each reconstruction. This FOM is a measure of how well the reconstruction algorithm performed for the given task.

4. Calculation of the statistical significance between two reconstructions (based on the previously assigned FOM) by which the null hypothesis that the reconstructions have equal performance can be rejected in favour of the alternative.

It is important to choose a FOM that is appropriate for the given task in order to measure the efficacy of the reconstruction algorithm. In this thesis the task is to detect low-contrast tumors in the brain based on reconstructions from X-ray CT projection data. The FOM for this comparative evaluation is discussed in the following Section 9.2.

In order to obtain statistically significant results, multiple experiments need to be performed, where the results are then used to determine the statistical significance by which the claim of equal performance of a pair of algorithms with regard to a FOM can be rejected in favour of the alternative. If two algorithms have different performance with large enough statistical significance, the null-hypothesis can be rejected. This means that the null-hypothesis is that the expected average values for a number of reconstructions of two algorithms are the same. Based on the calculated FOM
for each experiment, the p-value can be also calculated. The p-value represents the probability of wrongly rejecting the Null hypothesis and defines the highest possible significance for a rejection given the data. The more obvious the difference between the measurements, the smaller the p-value will be, expressing the fact that the probability of wrongly rejecting the Null hypothesis can be made small in case of clearly different measurements. For more details on the SHT methodology, see [3].

9.2 Image-wise Region of Interest

The Image-wise Region of Interest (IROI) is a figure of merit which compares the performance of CT reconstruction algorithms by measuring the detection of small, low-contrast regions. These regions are simulating tumors in the brain. The evaluation needs specially designed test data, which is based on a large number of pairs of potential tumors placed symmetrically on either left or right side of the brain. The probability for a tumor to be placed on either side has equal probability. These two sites for a tumor to be placed are considered a pair.

Figure 8.1 shows such a generated phantom with the small filled discs representing the generated tumors. The generated phantom has additionally local inhomogeneities which are based on a Gaussian random variable.

Definition 9.1. The image-wise region of interest (IROI)

IROI is defined as

\[
IROI = \frac{\sum_{b=1}^{B} (\alpha^{r}_{t}(b) - \alpha^{n}_{t}(b))}{\sqrt{\sum_{b=1}^{B} \left(\frac{\alpha^{r}_{t}(b) - \frac{1}{B} \sum_{b'=1}^{B} \alpha^{r}_{n}(b')}^{2}} \right) \sqrt{\sum_{b=1}^{B} \left(\frac{\alpha^{n}_{t}(b) - \frac{1}{B} \sum_{b'=1}^{B} \alpha^{n}_{n}(b')}^{2}} \right)}}
\]

(9.1)

where for \(1 \leq b \leq B\), \(B\) is the number of tumor-site pairs. \(\alpha^{r}_{t}(b)\) is the average density for the tumor site of the \(b\)-th pair, where \(r\) and \(p\), denote the reconstructed image and the phantom respectively and \(t\) and \(n\) denote that there is a tumor or not respectively [3].

For IROI the pixels inside the specified tumor sites are the ones of interest which is the reason why the calculations are restricted to those. IROI uses the densities \(\alpha\) for the calculation which represent the densities of either a site with or without a tumor,
9.3 Comparison of TV and SH Superiorized ART

The goal in this section is to evaluate the anecdotal experiments in Section 8.9 with statistical hypothesis testing. In the statistical hypothesis testing, a set of statistically generated experiments are computed and evaluated for a given FOM. In this project the statistical variation in the phantoms is given by the distribution of the tumors inside the brain and by the local inhomogeneities for each generated phantom. The FOM used for this statistical hypothesis testing is the previously discussed IROI. The resulting IROI values for the whole statistical hypothesis testing experiment can be seen in Table 9.1.

From these results it can be seen that the results for FBP, Pixel ART, TV superiorized ART and SH superiorized ART from [5] are reproduced within a certain accuracy. Further it can be seen that all the PSH superiorized ART algorithms perform better in most of the cases than SH superiorized ART. The algorithm PSH2 superiorized ART yield the best results with regard to the IROI. This could be an indicator that the superiorization of the SH needs more calibration, since the $\alpha$ value controls the strength of the SH superiorization. These results confirm the visual results in the
anecdotal experiment, where PSH2 superiorized ART seemed to perform visually better than the other PSH superiorized ART algorithms and SH superiorized ART. The TV superiorized ART performed best for 180 Views and the FBP algorithm outperformed all the selected algorithms for 360 Views and 720 Views.
### Table 9.2: Statistical Hypothesis Testing comparing all SH and PSH superiorized ART versions

<table>
<thead>
<tr>
<th>Views</th>
<th>FBP</th>
<th>Sh ART</th>
<th>PSh1 ART</th>
<th>PSh2 ART</th>
<th>PSh3 ART</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>0.067567</td>
<td>0.057384</td>
<td>0.070243</td>
<td>0.080904</td>
<td>0.075087</td>
<td>$3.447025 \times 10^{-7}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.057384</td>
<td>0.070243</td>
<td></td>
<td></td>
<td>$6.733804 \times 10^{-7}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.057384</td>
<td></td>
<td>0.080904</td>
<td>0.075087</td>
<td>$9.437760 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.057384</td>
<td></td>
<td></td>
<td>0.075087</td>
<td>$4.538794 \times 10^{-7}$</td>
</tr>
<tr>
<td>360</td>
<td>0.163981</td>
<td>0.103882</td>
<td>0.123305</td>
<td>0.152605</td>
<td>0.143863</td>
<td>$3.449420 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.103882</td>
<td>0.123305</td>
<td></td>
<td>0.143863</td>
<td>$5.232484 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.103882</td>
<td></td>
<td>0.152605</td>
<td>0.143863</td>
<td>$1.842080 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.103882</td>
<td></td>
<td></td>
<td>0.143863</td>
<td>$7.067300 \times 10^{-8}$</td>
</tr>
<tr>
<td>720</td>
<td>0.230594</td>
<td>0.177130</td>
<td>0.132185</td>
<td>0.198070</td>
<td>0.197488</td>
<td>$4.007811 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.177130</td>
<td>0.132185</td>
<td></td>
<td>0.197488</td>
<td>$2.783923 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.177130</td>
<td></td>
<td>0.198070</td>
<td>0.197488</td>
<td>$7.038603 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.177130</td>
<td></td>
<td></td>
<td>0.197488</td>
<td>$5.998789 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.132185</td>
<td>0.198070</td>
<td>0.197488</td>
<td>$2.798320 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.132185</td>
<td></td>
<td>0.197488</td>
<td>$4.211129 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.198070</td>
<td>0.197488</td>
<td>$2.685698 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.197488</td>
<td>$2.962616 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$2.962615 \times 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$2.926271 \times 10^{-8}$</td>
</tr>
<tr>
<td>Views</td>
<td>FBP</td>
<td>Pixel ART</td>
<td>TV ART</td>
<td>SH ART</td>
<td>PSH2 ART</td>
<td>P-value</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
<td>-----------</td>
<td>--------</td>
<td>--------</td>
<td>----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>180</td>
<td>0.067567</td>
<td>0.062939</td>
<td>0.087908</td>
<td>0.057384</td>
<td>0.080904</td>
<td>$1.325424 \times 10^{-05}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.062939</td>
<td>0.087908</td>
<td>0.057384</td>
<td>0.080904</td>
<td>$1.085350 \times 10^{-07}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.062939</td>
<td>0.087908</td>
<td>0.057384</td>
<td>0.080904</td>
<td>$3.447025 \times 10^{-07}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.062939</td>
<td>0.087908</td>
<td>0.057384</td>
<td>0.080904</td>
<td>$9.437760 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.062939</td>
<td>0.087908</td>
<td>0.057384</td>
<td>0.080904</td>
<td>$5.666351 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.062939</td>
<td>0.087908</td>
<td>0.057384</td>
<td>0.080904</td>
<td>$9.985165 \times 10^{-06}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.062939</td>
<td>0.087908</td>
<td>0.057384</td>
<td>0.080904</td>
<td>$3.359058 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.067567</td>
<td>0.062939</td>
<td>0.087908</td>
<td>0.057384</td>
<td>0.080904</td>
<td>$2.554454 \times 10^{-06}$</td>
</tr>
<tr>
<td>360</td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$3.011842 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$9.814118 \times 10^{-07}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$3.449420 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$1.842080 \times 10^{-06}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$7.339467 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$9.954257 \times 10^{-07}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$4.030707 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$3.624714 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$8.343526 \times 10^{-03}$</td>
</tr>
<tr>
<td></td>
<td>0.163981</td>
<td>0.119628</td>
<td>0.148439</td>
<td>0.103882</td>
<td>0.152605</td>
<td>$4.360051 \times 10^{-08}$</td>
</tr>
<tr>
<td>720</td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$3.613128 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$3.140910 \times 10^{-06}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$4.007811 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$7.038603 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$4.170885 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$3.932270 \times 10^{-06}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$8.563044 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$7.362680 \times 10^{-08}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$3.324880 \times 10^{-06}$</td>
</tr>
<tr>
<td></td>
<td>0.230594</td>
<td>0.162550</td>
<td>0.212577</td>
<td>0.177130</td>
<td>0.198070</td>
<td>$4.211129 \times 10^{-08}$</td>
</tr>
</tbody>
</table>

Table 9.3: Statistical Hypothesis Testing comparing the best PSH superiorized ART version according to IROI to the other algorithms
10 Conclusion

This thesis covered two variations of the superiorization with the $l_1$-norm minimization of the shearlet transform. The data used for the experiments are projection data from SNARK14 which are simulating a CT scanner. The projection data are then reconstructed using superiorization with ART. The SH superiorized ART which was proposed in [5] uses a subgradient method for the minimization, whereas the proposed method PSH superiorized ART applies the proximal operator to the $l_1$-norm minimization of the shearlet transform.

The first results compare the TV-norm minimization and the SH-norm minimization with regard to the number of superiorization steps $N$, which showed the impact of the superiorization steps for the SH-norm minimization. This means that SH superiorization needs more superiorization steps than the TV superiorization for ART in order to perform better with regard to the SH-norm.

The next experiments focus on the different parameters for the proposed PSH superiorization using ART. The results show the comparison of their performance with regard to the TV-norm, SH-norm and number of iterations.

In the anecdotal experiment it can be seen that the PSH superiorized ART does not outperform either the SH superiorized ART with regard to the SH-norm or the TV superiorized ART with regard to the TV-norm. A reason could be the parameter selection for the PSH superiorized ART, which could be varied in further investigations. Additionally the results do not visually affirm that the SH-norm improves the image quality significantly. This behaviour has been shown in [5], where the TV superiorized ART turned out to be more efficacious than SH superiorized ART with regard to the IROI.

The major benefit of the proximal operator to the subgradient method is that the superiorization can be done in one step while maintaining the property of a non-increasing SH-norm. For the subgradient more superiorization steps are necessary to achieve the same effect, which can be seen in the first experiment. The behaviour of the algorithms is further researched in Chapter 9, where the algorithms are evaluated according to the their IROI value. The proposed PSH superiorized ART performed better than SH superiorized ART, but did not outperform the existing TV superiorized ART with regard to the IROI.
An investigation, which was not covered in this thesis, is the combination of TV-norm minimization with the SH-norm minimization. Similar research has been done in [30] which covered the combination of the minimization of the TV-norm and SH-norm into one general CT reconstruction algorithm. Such a combined minimization as a non-ascending vector for the superiorization methodology could be incorporated and evaluated. This can be applied to CT reconstruction problems using algorithms such as ART or SART.

Another point left for investigation, is to evaluate different sets of parameters for SH superiorized ART and PSH superiorized ART in order to optimize their performance with regard to a FOM like IROI.
Acronyms

DIG  Digital Imaging Group
CT   Computerized Tomography
FBP  Filtered Back Projection
SVD  Singular Value Decomposition
ART  Algebraic Reconstruction Technique
SART Simultaneous Algebraic Reconstruction Technique
TV   Total Variation
SH   $l_1$ Norm of the Shearlet Transform
PSH  Proximal Operator of the $l_1$ Norm of the Shearlet Transform
FOM  Figure of Merit
SHT  Statistical Hypothesis Testing
IROI Image-wise Region of Interest
List of Figures

2.1 Two projections from two cylinders adapted from [7, p.2] .......................... 7
2.2 Sinogram for a simulated cross-section, done with SNARK14 ...................... 8
2.3 Projection of a single beam, adapted from [3, p.30] ............................... 8
2.4 Numbering of the image vector over each pixel ..................................... 10

5.1 Example for a linear system of equations with two lines. Adapted from [10, p.143] ............................................................................................................. 25
5.2 Example for a overdetermined linear system of equations with three lines. Adapted from [10, p.143] .............................................................. 26

6.1 Classical shearlet in the fourier domain ................................................. 32
6.2 Cone-adapted shearlet system ................................................................. 33

7.1 Subgradients of a function \( f \) (adapted from [27, p.2]) .......................... 38
7.2 Subgradient of the absolute function \( |x| \) in one dimension (adapted from [27, p.2]) .......................................................... 39
7.3 Evaluation of the proximal operator for different points, adapted from [6, p.125] .............................................................. 41
7.4 Soft-thresholding function \( \text{ST}_\beta(x_i) \) .................................................. 43
7.5 Subgradient of the absolute function \( |x| \) in one dimension ..................... 44

8.1 Digitization of the generated Phantom ................................................... 50
8.2 Resulting phantoms from filtered back-projection for 180 (left), 360 (middle) and 720 (right) views .............................................................. 51
8.3 Comparison of SH superiorized ART and TV superiorized ART with regard to their TV norm for 180, 360 and 720 views ............................. 52
8.4 Comparison of SH superiorized ART and TV superiorized ART with regard to their SH norm for 180, 360 and 720 views ............................. 52
8.5 Reconstructions with TV superiorized ART for \( N = 10 \) (left), \( N = 30 \) (middle) and \( N = 50 \) (right) .............................................................. 52
8.6 Reconstructions with SH superiorized ART for \( N = 10 \) (left), \( N = 30 \) (middle) and \( N = 50 \) (right) .............................................................. 53
8.7 The SH norm for a set of given \( \alpha \) and \( \beta \) values for 180, 360 and 720 views 54
8.8 The TV norm for a set of given \( \alpha \) and \( \beta \) values for 180, 360 and 720 views 54
8.9 The number of iteration needed for a set of given α and β values for 180, 360 and 720 views .......................................................... 55
8.10 Reconstructed phantoms for parameter combinations of α and β ...... 57
8.11 Comparison of reconstructed phantoms for 180 views for the reproduced anecdotal experiment from [5] ................................. 57
8.12 Comparison of reconstructed phantoms for 180 views for SH and PSH superiorized ART ..................................................... 58
List of Tables

8.1 Calculated residuals from filtered back-projection reconstructions for 180, 360 and 720 views ........................................ 50
8.2 Overview of the measurements for the anecdotal experiment .......... 56
9.1 IROI measurements for all algorithms .................................. 61
9.2 Statistical Hypothesis Testing comparing all SH and PSH superiorized ART versions ............................................. 63
9.3 Statistical Hypothesis Testing comparing the best PSH superiorized ART version according to IROI to the other algorithms ............... 64
Bibliography


A Anecdotal Experiments

A.1 Comparison of Algorithms for CT from 180 Views Resolution Data

* COMPARISON OF ALGORITHMS FOR CT FROM 180-VIEWS RESOLUTION DATA

PICTURE TEST

PROJECTION OLD

EXECUTE DCON

FBP FROM 180 ANGLES

STOP TERMINATION RESI 2.0

STOP TERMINATION RESI 3.638

STOP ITERATION 2

EXECUTE ART

Pixel ART from 180 angles

ART3 relaxation constant 0.05

constraint bound

SUPERIORIZE 40 0.9999 0.03 TVAR

EXECUTE ART

TV Superiorized ART from 180 angles

ART3 relaxation constant 0.05

constraint bound

SUPERIORIZE 40 0.9999 0.03 SCR5

EXECUTE ART

Shearlet Superiorized ART from 180 angles

ART3 relaxation constant 0.05

constraint bound

SUPERIORIZE 1 0.9 0.03 SCR3 RPRT 1
EXECUTE ART
Proximal1 Shearlet Superiorized ART from 180 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 1 0.6 0.03 SCR3
*
EXECUTE ART
Proximal2 Shearlet Superiorized ART from 180 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 1 0.3 0.03 SCR3
*
EXECUTE ART
Proximal2 Shearlet Superiorized ART from 180 angles
ART3 relaxation constant 0.05
constraint bound
*
EVALUATE RESOLUTION
EVALUATE ON 180 VIEWS
WHOLEPIC
2222222222222222222222222222222222222222222222222222222222222222

SKUNK PHANTOM MINIMUM 0.204 MAXIMUM 0.21675
1
END

Listing A.1: comparison.angles.0180.in

A.2 Comparison of Algorithms for CT from 360 Views Resolution Data
FBP FROM 360 ANGLES
2 1 -1 1 0
SINC
1
*
*STOP TERMINATION RESI 3.0
STOP TERMINATION RESI 4.0314
*STOP ITERATION 2
*
EXECUTE ART
Pixel ART from 360 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 40 0.9999 0.03 TVAR
*
EXECUTE ART
TV Superiorized ART from 360 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 40 0.9999 0.03 SCR5
*
EXECUTE ART
Shearlet Superiorized ART from 360 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 1 0.9 0.03 SCR3 RPRT 1
*
EXECUTE ART
Proximal1 Shearlet Superiorized ART from 360 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 1 0.6 0.03 SCR3
*
EXECUTE ART
Proximal2 Shearlet Superiorized ART from 360 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 1 0.3 0.03 SCR3
*
EXECUTE ART
Proximal \textsuperscript{3} Shearlet Superiorized ART from 360 angles

\begin{verbatim}
ART3 relaxation constant 0.05
constraint bound

EVALUATE RESOLUTION
EVALUATE ON 360 VIEWS
WHOLEPIC
2222222222222222222222222222222222222222222222222222222222222222

SKUNK PHANTOM MINIMUM 0.204 MAXIMUM 0.21675

END
\end{verbatim}

Listing A.2: comparison.angles.0360.in

A.3 Comparison of Algorithms for CT from 720 Views Resolution Data

\begin{verbatim}
*  
* COMPARISON OF ALGORITHMS FOR CT FROM 720-VIEWS RESOLUTION DATA
*  
PICTURE TEST
*  
PROJECTION OLD
*  
EXECUTE DCON
FBP FROM 720 ANGLES
2 1 -1 1 0
SINC
1
*
*STOP TERMINATION RESI 4.0
STOP TERMINATION RESI 5.9793
*STOP ITERATION 2
*
EXECUTE ART
Pixel ART from 720 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 40 0.9999 0.03 TVAR
*
\end{verbatim}
EXECUTE ART
TV Superiorized ART from 720 angles
ART3 relaxation constant 0.05
constraint bound
SUPERIORIZE 40 0.9999 0.03 SCR5

EXECUTE ART
Shearlet Superiorized ART from 720 angles
ART3 relaxation constant 0.05
constraint bound
SUPERIORIZE 1 0.9 0.03 SCR3 RPRT 1

EXECUTE ART
Proximal1 Shearlet Superiorized ART from 720 angles
ART3 relaxation constant 0.05
constraint bound
SUPERIORIZE 1 0.6 0.03 SCR3

EXECUTE ART
Proximal2 Shearlet Superiorized ART from 720 angles
ART3 relaxation constant 0.05
constraint bound
SUPERIORIZE 1 0.3 0.03 SCR3

EXECUTE ART
Proximal3 Shearlet Superiorized ART from 720 angles
ART3 relaxation constant 0.05
constraint bound
EVALUATE RESOLUTION
EVALUATE ON 720 VIEWS
WHOLEPIC
SKUNK PHANTOM MINIMUM 0.204 MAXIMUM 0.21675
END
B Statistical Comparison

These files in this chapter are necessary for the SNARK14 Experimenter to execute the statistical comparison. The files here are for statistical hypothesis evaluation for 180 Views.

B.1 Input File

The Listing B.1 is the main configuration of the files that are used in the statistical comparison.

```
1 SEED -1
2 EnSEMBLE brain.ens
3 EXPERIMENT 1 0 485 0.0376 11 1 30
4 DATA brain.projection.ss
5 RECONSTRUCTION brain.recon.ss
6 ANALYSIS brain.compare.ss
7 END
```

Listing B.1: sht.in

B.2 Ensemble

The brain.atl file specified in Listing B.2 describes the geometry of the phantom.

```
1 brain.atl
```

Listing B.2: brain.ens

B.3 Data

The Listing B.3 describes the settings for generating the projection data.

```
1 RAYSUM AVERAGE 1
2 1
3 GEOMETRY
4 divergent ARC 78 110.735
5 RAYS USER 693 DETECTOR SPACING 0.0533
6 ANGLES 180 EQUAL SPACING
7 0 358.00
8 MEASUREMENT NOISY
9 QUANTUM 1000000 180 CALIBRATION 2
10 SEED 0
11 BACKGROUND 0
```
RUN
PICTURE TEST
PROJECTION REAL

Listing B.3: brain.projection.ss

B.4 Reconstruction

The Listing B.4 describes the superiorization settings for the reconstruction algorithms.

EXECUTE DCON
dcon: FBP FROM 180 ANGLES
2 1 -1 1 0
SINC
1
*
STOP TERMINATION RESI 3.638 RPRT
*
EXECUTE ART
artp: Pixel ART from 180 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 40 0.9999 0.03 TVAR RPRT
*
EXECUTE ART
artt: TV Superiorized ART from 180 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 40 0.9999 0.03 SCR5 RPRT
*
EXECUTE ART
arts: Shearlet Superiorized ART from 180 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 1 0.9 0.03 SCR3 RPRT
*
EXECUTE ART
apr1: Proximal Shearlet Superiorized ART from 180 angles
ART3 relaxation constant 0.05
constraint bound
*
SUPERIORIZE 1 0.6 0.03 SCR3 RPRT
EXECUTE ART
apr2: Proximal2 Shearlet Superiorized ART from 180 angles
ART3 relaxation constant 0.05
constraint bound

SUPERIORIZE 1 0.3 0.03 SCR3 RPRT
EXECUTE ART
apr3: Proximal3 Shearlet Superiorized ART from 180 angles
ART3 relaxation constant 0.05
constraint bound

Listing B.4: brain.recon.ss

B.5 Analysis

The Listing B.5 describes the setting for the analysis of resulting reconstructions from the algorithms specified in Listing B.4.

Results_brain
COMPARE dcon artp IROI 1 0
COMPARE dcon artt IROI 1 0
COMPARE dcon arts IROI 1 0
COMPARE dcon apr1 IROI 1 0
COMPARE dcon apr2 IROI 1 0
COMPARE dcon apr3 IROI 1 0
COMPARE artp arts IROI 0 0
COMPARE artp artt IROI 0 0
COMPARE artp apr1 IROI 0 0
COMPARE artp apr2 IROI 0 0
COMPARE artp apr3 IROI 0 0
COMPARE arts artt IROI 0 0
COMPARE arts apr1 IROI
  0 0
COMPARE arts apr2 IROI
  0 0
COMPARE arts apr3 IROI
  0 0
COMPARE artt apr1 IROI
  0 0
COMPARE artt apr2 IROI
  0 0
COMPARE artt apr3 IROI
  0 0
COMPARE apr1 apr2 IROI
  0 0
COMPARE apr1 apr3 IROI
  0 0
COMPARE apr2 apr3 IROI
  0 0
END

Listing B.5: brain.compare.ss