MARSHALL PLAN
SCHOLARSHIP REPORT

Verification and evaluation of superiorized algorithms used in biomedical imaging

Comparison of iterative algorithms with and without superiorization for image reconstruction from projections

submitted by:

Ing. Bernhard Prommegger, 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, October 2014

Declaration of Academic Honesty

I, Bernhard PROMMEGGER, born on May 2, 1978 in Schwarzach im Pongau, hereby declare, that with the exception of chapter 5, which was co-authored by Oliver Langthaler [11], I have written this master thesis entirely on my own and that I have not used any other sources apart from those given.

Salzburg, October 10, 2014

Bernhard PROMMEGGER

Matriculation number

General Information

Last- and Given Name:	Bernhard PROMMEGGER
Institution:	Salzburg University of Applied Sciences
Degree Programme:	Applied Image and Signal Processing
Full Title:	Verification and evaluation of superiorized algorithms used in biomedical imaging
Keywords:	Superiorization, Algebraic reconstruction algorithms, ART, SART, SNARK
Supervisor SUAS:	FH-Prof Univ.-Doz. Dr. Stefan Wegenkittl
Supervisor CUNY:	Gabor T. Herman, Ph. D.

Abstract

The thesis aims at efficient methodologies for obtaining accurate biomedical images by mathematical/algorithmic development, computer implementation and practical evaluation of the novel methodology of superiorization. The underlying idea is the following: in many biomedical imaging applications, there exist efficient iterative algorithms that produce constraints-compatible images for given constraints. Often, the algorithm is perturbation resilient in the sense that, even if a number of changes are made at the end of each iterative step, the algorithm still produces a constraints-compatible image. Superiorization exploits this property by using the perturbations to steer the algorithm to an image that is not only constraints-compatible, but also preferable according to an optimization criterion reflecting some biomedically desirable property.

The specific aim of this master's thesis is the comparison of iterative algorithms with and without superiorization for image reconstruction from projections. The comparison is performed for two algorithms that are used very frequently in CT: algebraic reconstruction technique (ART) and simultaneous algebraic reconstruction technique (SART). Although both algorithms have been used in a lot of scientific papers, the performance of the two algorithms compared to each other has never been evaluated - not even in their non-superiorized versions.

The project is jointly organized by The City University of New York (CUNY) and the Salzburg University of Applied Sciences (SUAS) and should also deepen the academic cooperation between the two institutions. The project was supported by the Austrian Marshall Plan Foundation.

Acknowledgement

I would like to express my sincere gratitude towards Gabor T. Herman for providing me the chance to do my master's thesis at the department of Computer Science at The Graduate Center from The City University of New York. I want to thank my supervisor, Stefan Wegenkittl, for his support and encouraging attitude.

My main thanks go to my girlfriend Elisabeth, however. Thank you for keeping me on track during my very time demanding studies and motivating me for meeting the challenge of doing two master degrees.

Additionally I would like to thank the Austrian Marshall Plan Foundation for their financial support of my academic work in New York.

Contents

1 Introduction 1

2 Computerized Tomography 2

2.1 Data Collection 4

2.1.1 CT Data 5

2.1.2 Polychromaticity vs Monochromaticity 6

2.2 Reconstruction Algorithms 8

3 Basic Concepts of Reconstruction Algorithms 10

3.1 Image and Image Function 10

3.2 Radon Transform 13

3.3 Projection Data 14

3.4 Transform Methods 15

3.5 Series Expansion Methods 17

3.6 Optimization Criterion 20

4 Algebraic Reconstruction Algorithms 21

4.1 Algebraic Reconstruction Technique 23

4.2 Simultaneous Algebraic Reconstruction Technique 26

5 Superiorization 29

5.1 Problem sets, proximity function and ε-compatibility 29

5.2 Algorithms and Output 30

5.3 Bounded Perturbation Resilience 31

5.4 Optimization criterion and non-ascending vector 32

5.5 Superior Version of an Algorithm 33

5.6 Adapted Version of the Superiorized Algorithm 35

6 Implementation 39

6.1 SNARK 39
6.2 Simultaneous Algebraic Reconstruction Technique 39
6.3 Weighted Squared Distance 39

7 Comparison of ART and SART 40

7.1 Test Image Set 40
7.2 Methods of Image Comparison 40
    7.2.1 Task-Oriented Comparison of Algorithm Performance 42
7.3 Image-wise region of interest 43
7.4 Total Variation Minimization 44
7.5 Weighted Squared Distance 45
7.6 SNARK Experimenter 46
7.7 The Experiment 46
    7.7.1 Data generation 47
    7.7.2 Image reconstruction 47
    7.7.3 Figures of Merit 49
    7.7.4 Statistical Hypothesis Testing 50
7.8 Evaluation 50
7.9 Results 51
    7.9.1 ART with BLOB against all other algorithms 56
    7.9.2 ART vs SART 57
    7.9.3 Superiorized vs. Unsuperiorized Algorithms 58
    7.9.4 Superiorized vs. Superiorized Algorithms 61

8 Conclusion 62

List of Figures i

List of Tables iii

Acronyms iv

Bibliography v

A SNARK Input Files **viii**

A.1 experiment_ART_vs_SART.in viii

A.2 ART_vs_SART_projection.ss viii

A.3 ART_vs_SART_recon.ss ix

A.4 ART_vs_SART_compare.ss xi

B Mathematical Details **xiii**

B.1 Proof of Theorem 2 xiii

1 Introduction

This thesis compares the performance of two iterative algorithms with and without superiorization for image reconstruction from projections. The examined algorithms are algebraic reconstruction technique (ART) and simultaneous algebraic reconstruction technique (SART). Both algorithms are used very frequently in the field of computerized tomography. The comparison of their performance has never been subject of scientific work. The aim of this thesis is to close this gap[1].

The algorithms will be compared - using statistical hypothesis testing - against each other in their superiorized and non-superiorized versions. As a gold standard, the implementation of ART using generalized Kaiser-Bessel window functions (blobs) from [6] was defined.

The superiorization methodology is a heuristic solver for a large class of constrained optimization problems. The constraints come from the desire to produce a solution that is compatible to requirements provided by physically or otherwise obtained constraints. The underlying idea is that many iterative algorithms for finding such a solution are perturbation resilient in the sense that, even if certain kinds of changes are made at the end of each iterative step, the algorithm still produces a constraints-compatible solution. This property is exploited by using permitted changes to steer the algorithm to a solution that is not only constraints-compatible, but is also desirable according to specified optimization criteria [7].

All implementations, data generation and evaluation were done in SNARK14[2]. As SART has not been available in SNARK, it has been implemented and incorporated as new standard algorithm. The same is valid for the weighted squared distance (WSQD) as stopping criterion and figure of merit.

---

[1] Publication in peer-reviewed journal together with Prof. Gabor Herman is currently in preparation.

[2] For more details on SNARK see www.dig.cs.gc.cuny.edu.

2 Computerized Tomography

Computerized tomography (CT) is a common methodology for image creation. It aims at obtaining information regarding the exact positions of different matter inside an object. It was discovered in the field of diagnostic medicine but is also used in numerous non-medical imaging applications e.g. nondestructive testing [17]. CT produces images by taking cross-sections (slices) from the obtained data by measuring the attenuation of x-rays along a large number of lines through the cross-section. The resulting image is then reconstructed from the measured data. Figure 2.1 shows the reconstruction of a cross-section through the upper body of a human from its projection data. The different organs (materials) are visible through several gray levels which result from the different densities of the organs.

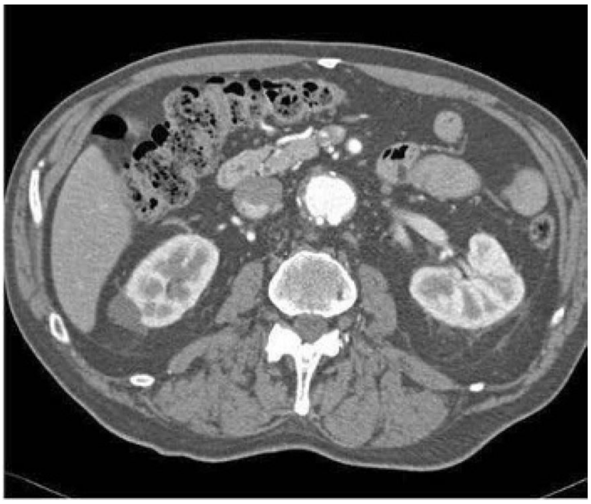


Figure 2.1.: Typical CT image [6, p. 6]

The set-up for acquiring computerized tomography images is shown in Figure 2.2. A narrow beam (ray) is passed from the source to the detector through the object. The measured value at the detector depends on the total amount and type of material placed anywhere along the path of the ray. Different materials block different amounts of the ray. One projection consists of the measurement of a number of parallel rays. In order to be able to reconstruct a complete CT image more projections (300-1000) are required. The divergent projections are generated by rotating source and detector around the fixed object. As a result of the rotation, acquired projections are taken from various angles. Figure 2.3a visualizes how the different projections are taken.

It shows three different projections taken form three distinct angles resulting in three measurements with different density distributions. Each measurement is related to the specific position (translation and rotation) of the source and the detector.

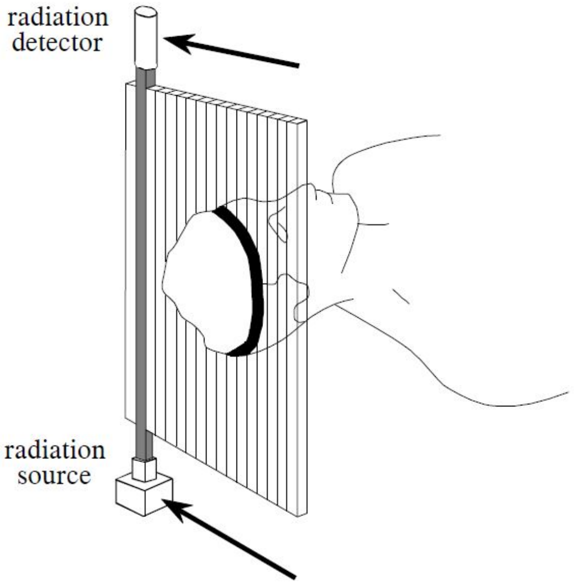
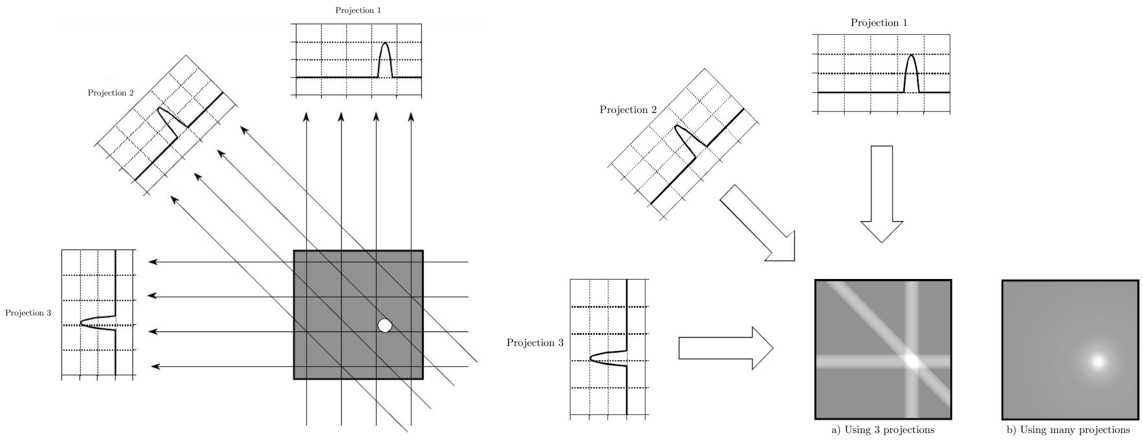


Figure 2.2.: CT data acquisition [18, p. 443]

The actual image is reconstructed by applying all projections back on the image region. Figure 2.3b illustrates how the concept of backprojection works. Every projection is smeared along the path at which it was actually acquired. The result is a blurry version of the original image. Sub-image 2.3a shows the backprojection using three projections, sub-image 2.3b shows the result after applying all projections.



(a) Different projection angles for CT data acquisition [18, p. 445]

(b) Using backprojection to generate a CT Image [18, p. 446]

Figure 2.3.: Projection and backprojection of CT images

The mathematical solution to the reconstruction problem of images from its projections has been published in a paper by Johann Radon in 1917 [14], but all main advances in the field of tomographic imaging originated with Hounsfield's invention of the x-ray computed tomographic scanner for which he received a Nobel prize in 1972, see [2], [6], [9] and [18].

2.1 Data Collection

The aim of CT is the reconstruction of a clean image of the exact position of different materials within an object. The necessary data is collected by the measurement of rays which are fired through the object of interest. As mentioned above, the reconstruction of a CT needs a lot of ray information. Figure 2.4 schematically shows the set-up for one source detector pair. Source and detector are positioned in the same plane as the cross-section to be imaged. The ray passes the reconstruction region in a distance of l from the origin and crosses the y-axes in an angle of θ. The reconstruction region defines the area, in which the object of interest can be located. It is not allowed, that any part of the object is located outside of the reconstruction region. In order to get reliable measurements, it is necessary, that the material within the reconstruction region (except of the object of interest) is homogeneous.

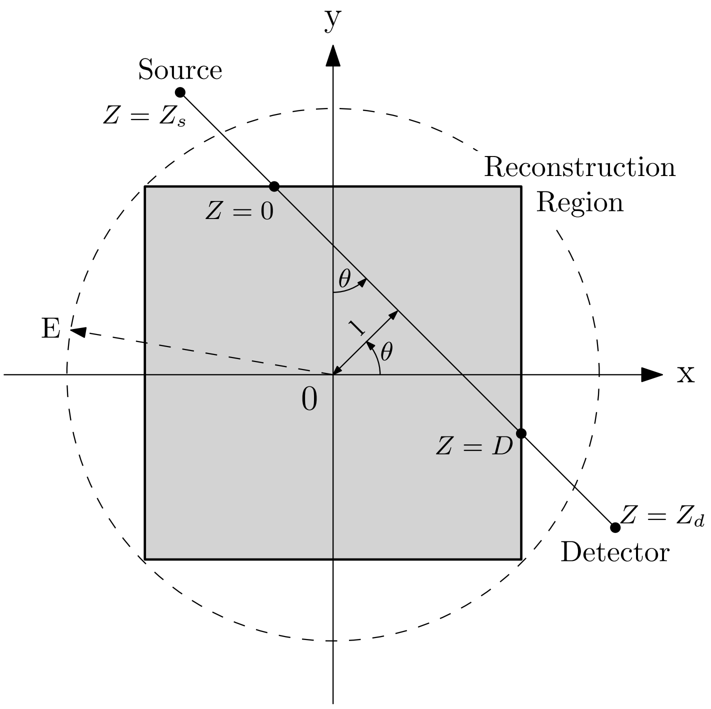


Figure 2.4.: CT data collection (adopted from [6, p. 30])

The actual measurement is done by counting the photons that reach the detector. This number is influenced by the object of interest within the reconstruction region as the different materials inside it block different amounts of the ray. The actual stored value is the ray measurement relative to the result of a calibration measurement without the object of interest, confer [2] and [6].

2.1.1 CT Data

In vacuum, all x-ray photons leaving the source in direction of the detector, will reach it. If a material is placed between the source and the detector, some of the photons might be removed from the ray by absorption or scattering. The amount of removed photons depends on the material which is in the path of the ray and the energy of the photon. The linear attenuation coefficient is used as a measurement of absorption.

Definition 2.1. *The* ***linear attenuation coefficient*** $\mu_{\bar{e}}^{t}$ *of a tissue* t *at energy* $\bar{e}$ *is defined as follows. Let* ρ *be the probability that a photon of energy* $\bar{e}$*, which enters a uniform slab of tissue* t *of unit thickness, on a line* L *perpendicular to the face of the slab, will not be absorbed or scattered in the slab. It is defined as*

$$\mu_{\bar{e}}^{t} = -\ln \rho \,, \tag{2.1}$$

where ln *denotes the natural logarithm [6].*

In order to be able to extract the influence of the object of interest from the measurements, the linear attenuation is taken relative to a calibration measurement with an empty region of interest. The relative value is called the relative linear attenuation.

Definition 2.2. *The* ***relative linear attenuation*** $\alpha_{\bar{e}}^{t}$ *at any point of space is defined as*

$$\alpha_{\bar{e}}^{t} = \mu_{\bar{e}}^{t} - \mu_{\bar{e}}^{a} \,, \tag{2.2}$$

where t *is the tissue occupying the point of space during the actual measurement and* a *is the material occupying the point during the calibration measurement [6].*

As the material inside the reconstruction region is homogeneous, $\mu_{\bar{e}}^{a}$ is constant in that region. Furthermore, since the complete objects of interest must be inside of the reconstruction region, the relative linear attenuation outside of the reconstruction region is 0.

The actual measurements stored by CT scanners are integer values of the type Hounsfield unit.

Definition 2.3. *The **Hounsfield unit (HU)** of a tissue* t *at energy* $\overline{e}$ *is defined as*

$$H = \frac{\mu_{\overline{e}}^{t} - \mu_{\overline{e}}^{water}}{\mu_{\overline{e}}^{water}} \cdot 1000\,, \tag{2.3}$$

where $\mu_{\overline{e}}^{t}$ *is the linear attenuation coefficient of the tissue* t *and* $\mu_{\overline{e}}^{water}$ *is the linear attenuation coefficient of water [9].*

The value of Hounsfield units is in the range between -1000 and 3000. The HU of water is $H = 0$ and the value $H = -1000$ corresponds to the linear attenuation coefficient of air $\mu_{\overline{e}}^{air}$ which is approximately 0.

When we take cross-sections of the body, we divide the cross-section into equal, square-shaped blocks and reference them as volume elements (voxels). The CT number is proportional to the average relative linear attenuation in a voxel. A picture element (pixel) is the representation of a voxel in a 2D image. Its gray level is proportional to the CT number of the corresponding voxel. See [6] and [9] for further details.

2.1.2 Polychromaticity vs Monochromaticity

As defined in Definition 2.1, the attenuation at any point depends on the material at that point and the energy distribution of the ray. In CT, the energy spectrum is not constant (polychromatic) and it changes as the ray passes through the object. As a result of this, the attenuation at a point may vary depending on the direction of the ray that passes it. If there would be only one energy level (monochromatic), this would not be the case and every point would have a unique attenuation, and the construction of the attenuation distribution would be well defined. In order to simplify the development of mathematical procedures, it is assumed that there is only one energy level. Therefore the CT number assigned to a voxel is only a property of the tissue occupying the voxel and it is independent of the position within the region of interest.

Definition 2.4. *The **monochromatic ray sum*** m *for a fixed position of the source* *and detector pair at a specific, monochromatic energy* $\overline{e}$ *is defined as:*

$$m = -\ln \frac{A_m}{C_m}\,, \tag{2.4}$$

where C_m *denotes the calibration measurement and* A_m *the actual measurement. The set of all source and detector pair positions is called* ***monochromatic projection data*** *[6].*

As stated above, the actual energy level changes when a ray passes through an object. This change can be caused by *beam hardening* or *scattering*. Due to these changes, the polychromatic measurements differ from the ideal monochromatic measurements.

Definition 2.5. *The* ***polychromatic ray sum*** m *for a fixed position of the source and detector is defined as:*

$$p = -\ln \frac{A_p}{C_p}\,, \tag{2.5}$$

where C_p *denotes the calibration measurement and* A_p *the actual measurement. The set of all source and detector pair positions is called* ***polychromatic projection data*** *[6].*

Beam hardening arises from the fact that - when a ray passes through an object - preferably low-energy photons are absorbed. As a result the average energy of a photon of the ray at the detector is higher than the average photon energy of the ray leaving the source. Attenuation by scattering occurs because some energy of the ray is deflected onto a new path and therefore the energy of the ray is reduced. The scatter angle is random, but most of the x-rays are scattered in the forward direction. Both - beam hardening and scattering - causes artifacts in the reconstruction. A more detailed description of the effects can be found in [9].

As Herman points out in [6] it is not possible to uniquely determine the monochromatic projection data from the polychromatic projection data, but the approximation of it leads to diagnostically usable CT numbers. All details of this section can be found in [6] and [9].

2.2 Reconstruction Algorithms

> "CT is the reconstruction of a clean image of the density from digital computational operations on measurements of emanations that have passed through the body." [2, p. 356]

In order to be able to create a mathematical model for image reconstruction from projections, the following assumptions have been made:

- There is only one energy level (monochromaticity).
- The cross-sections (slices) are infinitely thin.
- For all source-detector-pairs, all x-ray photons travel in the same straight line in the infinitely thin slice (perfectly calibrated system).

As a result of the second assumption, the difference between voxel and pixel disappears and therefore the gray level of any point (x, y) is proportional to the linear attenuation $\mu_{\overline{e}}(x, y)$.

Theorem 1. *Let* L *be the straight line that is the path of all the x-ray photons for a particular source-detector pair, and let* m *be the corresponding monochromatic ray sum. Based on the definition of the linear attenuation coefficient (see Definition 2.1), it is proven in [6, p. 260] that the monochromatic ray sum can be calculated as:*

$$m \simeq \int_0^D \mu_{\overline{e}}(x, y) dz \,, \tag{2.6}$$

where z *is the distance of the point* (x, y) *on the line* L *and the limits of the integration are defined as shown in Figure 2.4.* 0 *is the point where the ray enters the reconstruction region,* D *is the point where the ray leaves the reconstruction region [6].*

Since $\mu_{\overline{e}}(x, y) = 0$ outside of the reconstruction region (see Section 2.1.1) these points do not need to be considered.

In general this problem has been solved by Radon in 1917 [14]:

Theorem 2. *Let* l *denote the distance of the line* L *from the origin and let* θ *denote the angle made with the x-axis by the perpendicular drawn from the origin to* L *(see*

Figure 2.4) and let $m(l,\theta)$ *denote the integral of* $\mu_{\overline{e}}(x,y)$ *along the line L, then*

$$\mu_{\overline{e}}(x,y) = -\frac{1}{2\pi^2} \lim_{\varepsilon \to 0} \int_{\varepsilon}^{\infty} \frac{1}{q} \int_{0}^{2\pi} m_1(x\cos\theta + y\sin\theta + q, \theta)\, d\theta\, dq\,, \qquad (2.7)$$

where $m_1(l,\theta)$ *denotes the partial derivative of* $m(l,\theta)$ *with respect to* l *[6].*

Details of the proof can be found in Appendix B.1.

This formula implies, that the distribution of the linear attenuation in an infinitely thin slice is uniquely determined by the set of *all* of its line integrals and generally solves the problem of the reconstruction of cross-sections from its line integrals. But computerized tomography (CT) does not fulfill these requirements and therefore some difficulties arise:

- When measuring data with a CT scanner only a finite set of line integrals are measured and therefore the measured data is not sufficient for a reconstruction with Radons formula.
- The measurements of the CT scanner are only estimates. Inaccuracies arise due to beam hardening, scattering, detector inaccuracies, etc. For further details on the different error sources see [6] and [9].

3 Basic Concepts of Reconstruction Algorithms

This chapter discusses the prerequisites for image reconstruction from projections. Additionally, the two main types of reconstruction methods, namely transfer methods and series expansion methods, are explained.

3.1 Image and Image Function

Definition 3.1. *An* ***image*** *is defined by*

- *the* ***image region*** *which is a square with its origin in the center of the coordinate system and*
- *an* ***image function*** f *of two variables whose value is zero outside the image region [6].*

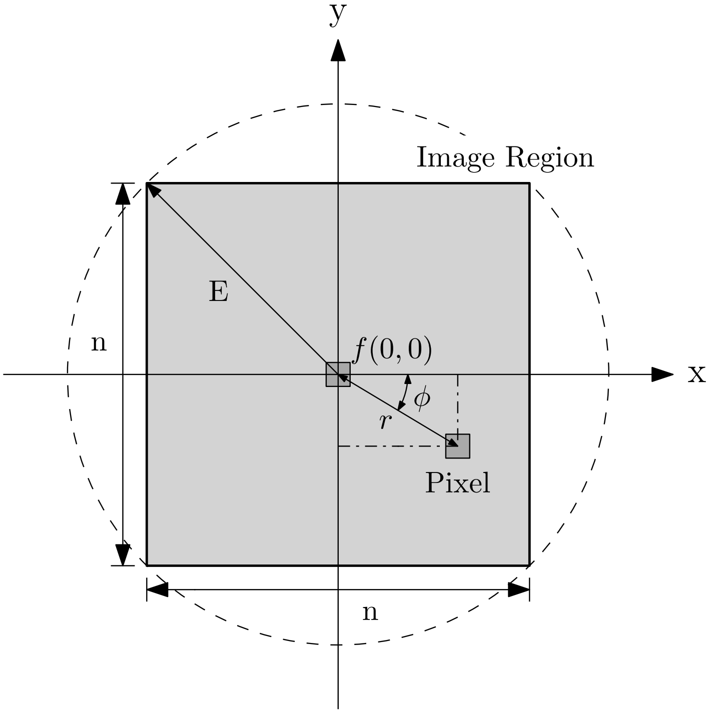


Figure 3.1.: Definition of an Image

Figure 3.1 shows an image. The image region is the square light gray area. The center of the image region is at the origin of the coordinate center. The image function at that point is called $f(0,0)$. The size of its digitization is n x n and therefore it holds n^2 picture elements (pixels). E is the radius of the circumscribed circle of the image region.

In computerized tomography the image region is also called reconstruction region and the gray level of the pixel at (x, y) is proportional to the relative linear attenuation (see Definition 2.1).

It is often more convenient to use polar coordinates and to describe the image I as a function of (r, ϕ). r and ϕ are defined as shown in Figure 3.1: $r = \sqrt{x^2 + y^2}$ and $\phi = \arctan \frac{y}{x}$.

The relationship between the (r, ϕ) (see Figure 3.1) and the (l, θ) space (see Figure 2.4) is shown in Figures 3.2 and 3.3. The point (r, ϕ) lies on the straight L. l is the distance of L to the origin, and therefore the points (r, ϕ) and P form a right triangle with the origin. l can be expressed as:

$$l = r \cos(\theta - \phi) \tag{3.1}$$

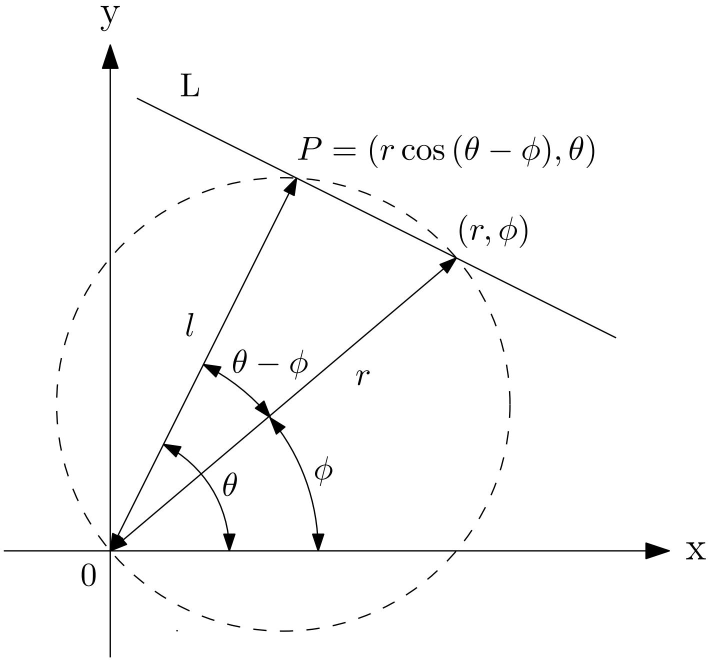


Figure 3.2.: Relationship (part I) between (r, ϕ) and (l, θ) space (adapted from [6, p. 104])

All points P that corresponds to the lines through (r, ϕ) are located on the circle defined by the origin and the point (r, ϕ). In Figure 3.2 this circle is visualized with a dashed line. In the (l, θ) space this circle is the sinusoidal $l = r \cos(\theta - \phi)$. The point P is the point on the sinusoidal at θ. The sinusoid is visualized in Figure 3.3 [6].

Definition 3.2. *It is assumed that the* ***image function*** f ***is square integrable****.*

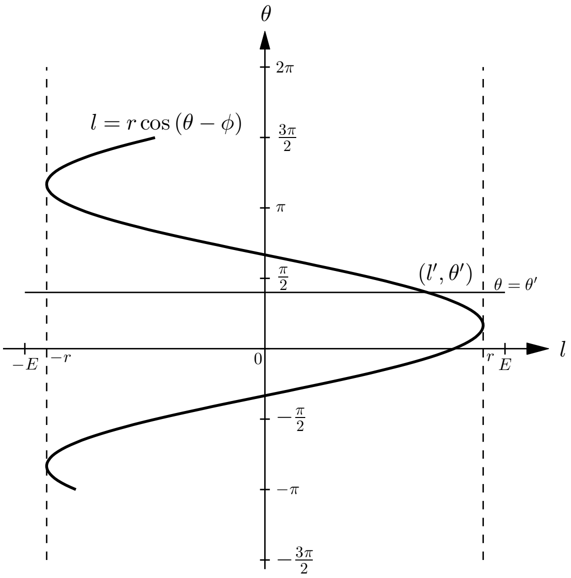


Figure 3.3.: Relationship (part II) between (r, ϕ) and (l, θ) space (adapted from [6, p. 104])

This means that

$$\int_0^{2\pi} \int_0^E f(r,\phi)^2 r \, dr \, d\phi \tag{3.2}$$

exists and its value is a real number [6].

From Definition 3.2 it follows that the difference between two arbitrary images can be calculated.

Definition 3.3. *The difference d of two images is defined as*

$$d(f_1, f_2) = \sqrt{\int_0^{2\pi} \int_0^E (f_1(r,\phi) - f_2(r,\phi))^2 r \, dr \, d\phi}\,, \tag{3.3}$$

where f_1 and f_2 are image functions [6].

In case of computerized tomography the image function is interpreted as the relative linear attenuation at the point (r, ϕ) which is 0 outside of the reconstruction region (see Section 2.1).

$$f(r,\phi) = 0 \text{ if } |r\cos\phi| > E \text{ or } |r\sin\phi| > E \tag{3.4}$$

3.2 Radon Transform

The Radon transform is a two-dimensional transformation which was introduced by Radon [14] in 1917.

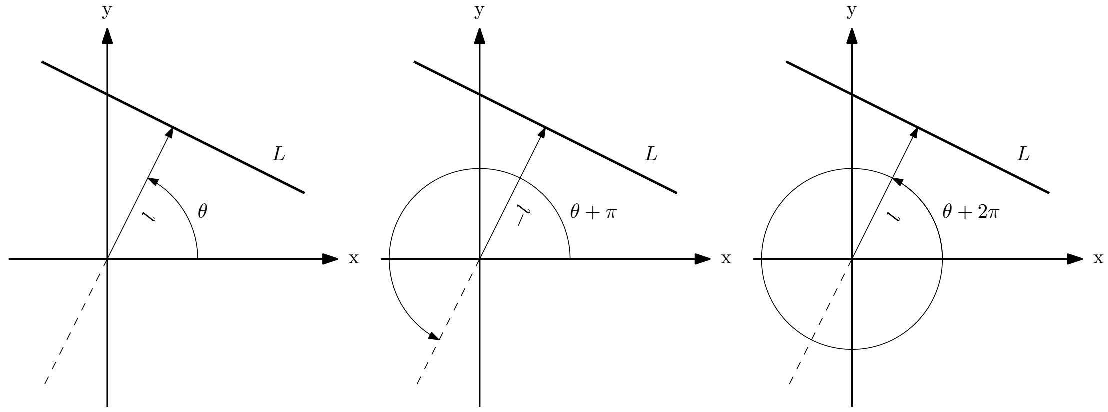
Definition 3.4. *The* ***Radon transform*** *of f is defined for real number pairs (l, θ) as follows:*

$$\mathscr{R}f(l,\theta) = \begin{cases} \int_{-\infty}^{\infty} f\left(\sqrt{l^2+z^2}, \theta + \tan^{-1}\left(\frac{z}{l}\right)\right) dz & ,\text{if } l \neq 0 \\ \int_{-\infty}^{\infty} f\left(z, \theta + \frac{\pi}{2}\right) dz & ,\text{if } l = 0 \end{cases} \tag{3.5}$$

$\mathscr{R}f(l,\theta)$ is the line integral of f along the line L, where z is the distance along the line L (see Figure 2.4) [6].

As visually shown in Figure 3.4, the following variants of the Radon transform are equal:

$$\mathscr{R}f(l,\theta) = \mathscr{R}f(-l,\theta+\pi) = \mathscr{R}f(l,\theta+2\pi) \tag{3.6}$$



(a) $\mathscr{R}f(l,\theta)$ (b) $\mathscr{R}f(-l,\theta+\pi)$ (c) $\mathscr{R}f(l,\theta+2\pi$

Figure 3.4.: Equivalents of the Radon transform

Equation (3.4) implies that the Radon transform outside of the image region is 0.

$$\mathscr{R}f(l,\theta) = 0, \text{ if } |l| \geq E. \tag{3.7}$$

Due to (3.6) and (3.7), $\mathscr{R}f$ is completely defined by its values at the points (l,θ) with $-E < l < E$ and $0 \leq \theta < \pi$ [6].

For more details on the Radon transform see [6] and [15].

3.3 Projection Data

In CT the locations in the (l, θ) space correspond to the lines, for which the CT scanner collects measurements. It is assumed, that there are $2N + 1$ parallel measurements with a step size of d. To cover the complete reconstruction region Nd must be greater than the radius of the circumscribed circle of the reconstruction region E. After the projection data has been collected for all $2N + 1$ lines, the source and detector are rotated by an angle of Δ, and again data for $2N+1$ lines is collected. This is repeated for a total of M views where $M\Delta = \pi$. Figure 3.5 shows the location of the collected data (values of the line integrals) in the (l, θ) space. $(id, m\Delta)$ refers to one measurement value. It lies on the intersection of the straight lines $l = id$ and $\theta = m\Delta$ where $-N \leq i < N$ and $0 \leq m \leq M - 1$.

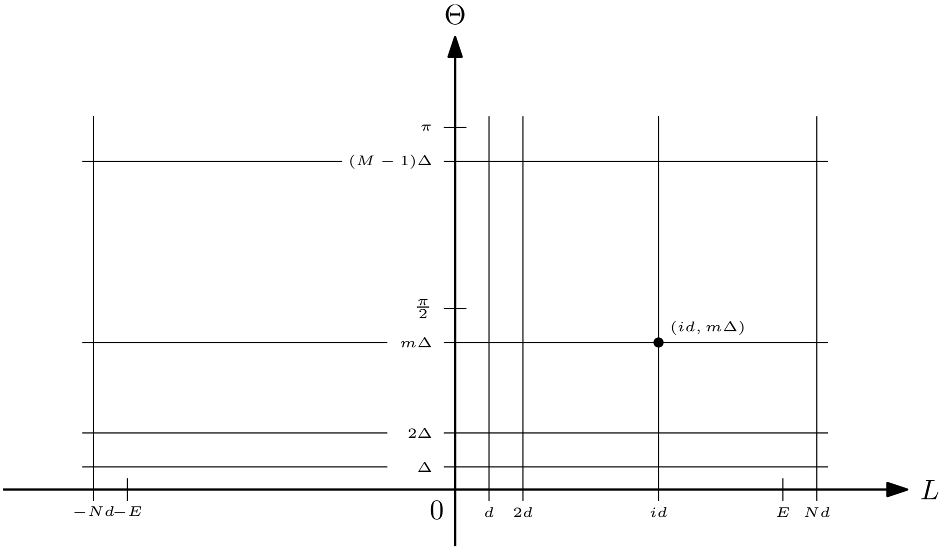


Figure 3.5.: Location of points in the (l, θ) space (adapted from [6, p. 106])

The input data (projection data) of a reconstruction algorithm are estimates of the values $\mathscr{R}f(l, \theta)$ for a finite number of I pairs of (l, θ): $(l_1, \theta_1), \ldots, (l_I, \theta_I)$.

Definition 3.5. *The I-dimensional column vector y is called* ***measurement vector****. Its i-th component y_i is the estimation of the i-th measurement of the CT scanner. It is defined as*

$$y_i = \mathscr{R}_i f = \mathscr{R}f(l_i, \theta_i)\,. \tag{3.8}$$

For reconstruction algorithms it is assumed that the method of data collection is fixed and therefore the set $(l_1, \theta_1), \ldots, (l_I, \theta_I)$ is fixed as well. The reconstruction problem is

given the data y, **estimate** the image f.

The estimation of the image f is denoted as f^*. More details can be found in [6].

3.4 Transform Methods

One way to define the estimate f^* is to give a formula, that expresses the value of $f^*(r, \phi)$ in terms of $r, \phi, y_1, \ldots, y_I$. Such a formula may be a "discretized" inversion formula of its Radon transform $\mathscr{R}f$. Reconstruction methods based on this approach are referred to as transform methods.

Definition 3.6. *The* ***inverse Radon transformation*** $\mathscr{R}^{-1}p(l, \theta)$ *is the inverse of the Radon transformation* $\mathscr{R}f(r, \phi)$*. It reproduces the original function* $f(r, \phi)$ *from its Radon transformed values. By this,*

$$f(r, \phi) = \mathscr{R}^{-1}\mathscr{R}f(r, \phi) \tag{3.9}$$

When $p(l, \theta) = \mathscr{R}f(r, \phi)$ *then the inverse Radon transformation is defined as*

$$\mathscr{R}^{-1}p(l, \theta) = \frac{1}{2\pi^2} \int_0^{\pi} \int_{-\infty}^{\infty} \frac{p_1(l, \theta)}{r \cos(\theta - \phi) - l} \, dl \, d\theta \,, \tag{3.10}$$

where $p_1(l, \theta)$ *is the first partial derivative of* $p(l, \theta)$ *with respect to* l *[6].*

The inverse Radon transformation can also be expressed as a sequence of simpler operators:

The first operator D_Y denotes the *partial differentiation* with respect to the first variable of a function of two real variables. The function value of a pair of any two real numbers (l, θ) is

$$\mathscr{D}_Y p(l, \theta) = p_1(l, \theta) = \lim_{\Delta l \to 0} \frac{p(l + \Delta l, \theta) - p(l, \theta)}{\Delta l} \tag{3.11}$$

assuming that the limit on the right-hand side exist.

The next operator is the Hilbert transform $\mathscr{H}_Y q$ with respect to the first variable of the function q. It is defined for any number pair (l, θ) as

$$\mathscr{H}_Y q(l', \theta) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{q(l, \theta)}{l' - l} \, dl \,. \tag{3.12}$$

Since its integrand becomes infinite at $l = l'$, this is an improper integral and it has to be evaluated in the *Cauchy principal value* sense:

$$\mathscr{H}_Y q(l', \theta) = -\frac{1}{\pi} \lim_{\varepsilon \to 0} \left(\int_{-\infty}^{l'-\varepsilon} \frac{q(l,\theta)}{l'-l}\, dl + \int_{l'+\varepsilon}^{\infty} \frac{q(l,\theta)}{l'-l}\, dl \right). \tag{3.13}$$

The last operator $\mathscr{B}t$ is the so called *backprojection* whose value is defined at any point (r, ϕ) by

$$\mathscr{B}t(r, \phi) = \int_0^{\pi} t(r\cos(\theta - \phi), \theta)\, d\theta\,. \tag{3.14}$$

Figure 3.2 visualizes backprojection. The term $r\cos(\theta - \phi)$ (see (3.1)) describes the function of the dashed circle through the origin and the point (r, ϕ). All possible lines through (r, ϕ) can be described using the point itself and any point on the dashed circle. Setting the integral limits to 0 and π results in a half circle and therefore (3.14) considers all line integrals L through (r, ϕ).

When combining all three operators the result is - apart from a factor $-\frac{1}{2\pi}$ - the inverse Radon transform as defined in Definition 3.6.

$$\mathscr{B}\mathscr{H}_Y\mathscr{D}_Y(r, \phi) = -\frac{1}{\pi} \int_0^{\pi} \int_{-\infty}^{\infty} \frac{p_1(l,\theta)}{r\cos(\theta-\phi) - l}\, dl\, d\theta \tag{3.15}$$

Therefore the inverse Radon transformation of a function p can be interpreted as a sequence of the following operations:

1. partial differentiation of p with respect to the first variable,
2. a Hilbert transform with respect to the first variable,
3. a backprojection and
4. a normalization with the factor $-\frac{1}{2\pi}$.

$$\mathscr{R}^{-1} = -\frac{1}{2\pi}\mathscr{B}\mathscr{H}_Y\mathscr{D}_Y \tag{3.16}$$

Definition 3.7. *In computerized tomography, the inverse Radon transformation is interpreted as*

$$\mathscr{R}^{-1}p(r, \phi) = \frac{1}{2\pi^2} \int_0^{\pi} \int_{-E}^{E} \frac{p_1(l,\theta)}{r\cos(\theta-\phi) - l}\, dl\, d\theta\,, \tag{3.17}$$

where E is the radius of the circle circumscribing the reconstruction region (see Figure

2.4).

In order to get an exact reconstruction of the reconstruction region, the precise values for all $p(l, \theta)$ need to be available. As this is not the case, the reconstruction is just an estimate for the obtained projection data. The algorithms used are based on (3.16) or are alternative representations of the inverse Radon transform $\mathscr{R}^{-1}$, see [6] [15].

3.5 Series Expansion Methods

The series expansion approach differs from the transform methods described in the previous section as the problem itself is discretized at the very beginning: the image reconstruction is translated into finding a set of numbers - the so called image vector - for a defined set of basis functions.

Definition 3.8. *For any specific image region the fixed set B of functions $\{b_1, \ldots, b_J\}$, $b_i = b_i(r, \phi)$, is called* ***basis functions****. For any image f with the specified image region, there exists a linear combination of the basis functions that produces an adequate approximation f^* for f.*

One possible basis function is the so called pixel basis function as defined in (3.18). The image region is divided into n x n square pixels (see Figure 3.1). The numbering of the basis function is the same as for the image vector x (see Figure 3.6). The pixel basis function has the value 1 if (r, ϕ) is inside of the j-th pixel and 0 otherwise. The pixel basis function has $J = n^2$ components.

$$b_j(r, \phi) = \begin{cases} 1 & \text{, if } (r, \phi) \text{ is inside the j-th pixel,} \\ 0 & \text{, otherwise} \end{cases} \tag{3.18}$$

Another commonly used basis function are generalized Kaiser-Bessel window functions (blobs). Blobs are more complex and require more computational effort for the image reconstruction but deliver more accurate results. More details on blobs can be found in [12].

Definition 3.9. ***The** n **x** n **digitization** $\widehat{f}$ **of the image** f using a set of basis functions b is defined as*

$$\widehat{f}(r, \phi) = \sum_{j=1}^{J} x_j b_j(r, \phi)\,, \tag{3.19}$$

where x_j *is the average value of* f *inside the* j*-th pixel and* b_j *is the* j*-th basis function (component) of* B*.*

Once the basis functions are defined, any digitized image $\widehat{f}$ of the specified image region can be represented as a linear combination of the basis functions $(b_j)_{j=1}^{J}$. The image $\widehat{f}$ is uniquely determined by the choice of the coefficients x_j for $1 \leq j \leq J$. The vector x with its j-th component x_j is denoted as the image vector.

Definition 3.10. *The* ***image vector*** x *is the digitized representation of an image region. It divides the image region into* n *x* n *pixels where* n *has to be an odd number. The image vector is* $J = n^2$ *dimensional. Each component of the image vector represents one pixel. The numbering of the pixels is defined in Figure 3.6 [6].*

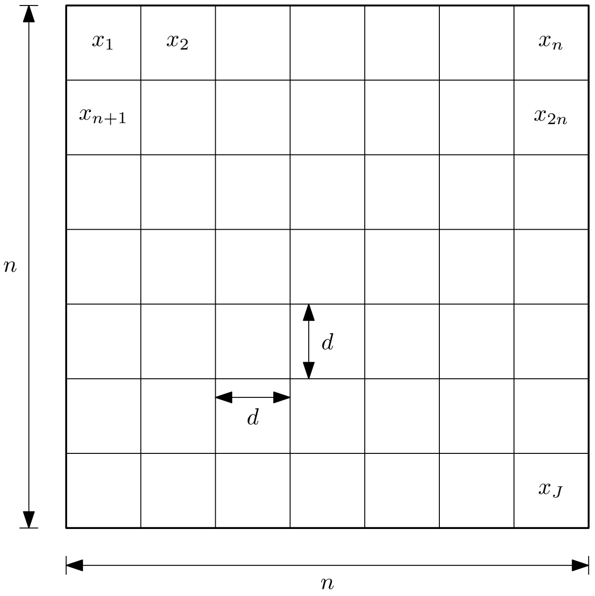


Figure 3.6.: Numbering of the components of the image vector (adapted from [9, p. 276])

Independent of the choice of the basis functions, for any image f there is only one image $\widehat{f}$ with

$$d\left(f, \widehat{f}\right) \leq d\left(f, \widehat{f}_i\right) , \tag{3.20}$$

where $\widehat{f}$ is a linear combination of the basis functions and $\widehat{f}_i$ is any other linear combination of the same basis functions.

Furthermore, if the basis functions are linearly independent, then there is a unique image vector x that fulfills (3.19) for this $\widehat{f}$.

In the best case, the series expansion approach would aim to find the image vector for which the approximation $\widehat{f}$ is closest to f. But as the digitized projection data normally does not uniquely determine the image f, the aim is to find a sufficient close image vector regarding to some optimization criterion.

In order to show how image reconstruction is translated into a discrete problem two properties of $\mathscr{R}_i$ defined in (3.8) are necessary:

1. The Radon transform is $\mathscr{R}_i$ linear

$$\mathscr{R}_i(c_1 f_1 + c_2 f_2) = c_1 \mathscr{R}_i f_1 + c_2 \mathscr{R}_i f_2 \tag{3.21}$$

2. If two images f_1 and f_2 are close to each other, then their two Radon transforms are $\mathscr{R}_i f_1$ and $\mathscr{R}_i f_2$ close to each other as well.

For further details on those two properties see [6].

The above properties imply that the Radon transform of the discretized version $\mathscr{R}_i \widehat{f}$ of the image f will be approximately the same as the Radon transform of the original image $\mathscr{R}_i f$.

$$\mathscr{R}_i f \simeq \mathscr{R}_i \widehat{f} = \sum_{j=1}^{J} x_j \mathscr{R}_i b_j \tag{3.22}$$

As b_j are user-defined functions, usually they are chosen in a way that the calculation of $\mathscr{R}_i b_j$ is computational easy. For example when pixel basis functions (see (3.18)) are used, $\mathscr{R}_i b_j$ is just the length of the intersection of the j-th pixel with the ray of the i-th source-detector pair. $r_{i,j}$ is denoted to the calculated value of $\mathscr{R}_i b_j$.

$$r_{i,j} \simeq \mathscr{R}_i b_j \tag{3.23}$$

By combining the definition of the measurement vector y and its components y_i as the measured estimate of $\mathscr{R}_i$ (see Definition 3.5) and (3.22) and (3.23), we get for $1 \leq i \leq I$

$$y_i \simeq \sum_{j=1}^{J} r_{i,j} x_j \,. \tag{3.24}$$

Definition 3.11. *The **projection matrix** R is the matrix whose (i, j)-th element is*

$r_{i,j}$ where $r_{i,j}$ is defined by (3.23) [6].

Definition 3.12. *The **error vector** e is the I-dimensional column vector whose i-th component e_i is the difference the i-th component of the measurement vector and the calculated sum $\sum_{j=1}^{J} r_{i,j}x_j$ (see (3.24)) [6].*

By using the above Definitions 3.11 and 3.12 combined with (3.24), the reconstruction problem can be stated as

$$y = Rx + e\,. \tag{3.25}$$

Based on (3.25), the series expansion approach solves the following discrete reconstruction problem:

given the data y, **estimate** the image vector x.

If the estimation for the solution of the discrete reconstruction problem is the vector x^*, then the estimated reconstruction f^* of the image f is

$$f^* = \sum_{j=1}^{J} x_j^* b_j\,, \tag{3.26}$$

see [6].

3.6 Optimization Criterion

The optimization criterion is used to estimate the image vector of the series expansion methods. In order to get a good estimation for the image vector in (3.25), the error vector e needs to be known. There are different approaches to find e. The aim is always to choose an image vector x as the solution of (3.25) for which the value of some function $\phi_1(x)$ (optimization criterion) is minimal. If there is more than one image vector x, that minimizes $\phi_1(x)$, then a secondary criterion $\phi_2(x)$ is used to find the best vector. Some proposed optimization functions are:

- prior probability density functions,
- least squares or
- total variation (TV).

For more detailed information on optimization criteria see [6].

4 Algebraic Reconstruction Algorithms

The following sections discuss algebraic reconstruction algorithms which are series expansion methods (see Section 3.5). Furthermore two algorithms - algebraic reconstruction technique (ART) and simultaneous algebraic reconstruction technique (SART) - are discussed in more details.

All series expansion methods are iterative algorithms which try to solve discrete reconstruction problems in the form of $y = Rx + e$ (see (3.25)). They produce a sequence of image vectors $x^{(0)}, x^{(1)}, \ldots$ which should convert to the required estimate x^*. y is the given measurement vector and e is the so called error vector and denotes the difference of the reconstruction Rx and the actual measurements [6].

For algebraic algorithms the reconstruction problem $y = Rx + e$ from (3.25) can also be written as system of equations. As previously defined, there are J pixel in the picture region (see Figure 3.6) and the projection data consists of M projections with $2N + 1$ rays per projection. This makes a total sum of $Z = M * (2N + 1)$ rays. The corresponding system of equations is

$$\begin{aligned} y_1 &= r_{1,1}x_1 + r_{1,1}x_2 + \cdots + r_{1,J}x_J \\ y_2 &= r_{2,1}x_1 + r_{2,1}x_2 + \cdots + r_{2,J}x_J \\ &\vdots \\ y_Z &= r_{Z,1}x_1 + r_{Z,1}x_2 + \cdots + r_{Z,J}x_J \end{aligned} \tag{4.1}$$

A picture region with J pixel leads to a system with J degrees of freedom. And therefore, the image represented by the image vector x may be considered as a single point in a J-dimensional space. Every equation of the system represents a hyperplane. If a unique solution exists, the intersection of all hyperplanes is a single point and represents the solution.

For the sake of simplicity, the problem is explained for the two-dimensional case where the hyperplanes are represented as straight lines:

$$\begin{aligned} y_1 &= r_{1,1}x_1 + r_{1,1}x_2 \\ y_2 &= r_{2,1}x_1 + r_{2,1}x_2 \end{aligned} \tag{4.2}$$

There are a lot of different algorithms to solve such a problem. Figure 4.1 shows the Kaczmarz method. The starting point of the algorithm is some arbitrary initial guess. This arbitrary point is then projected onto the first hyperplane resulting in a point G, which is now projected onto the next hyperplane resulting in point H. This iterative projection process is continued until the actual intersection point is reached. The vector $x^{(0)}$ represents the initial guess, $x^{(1)}$ and $x^{(2)}$ the current estimations after the first, respectively the second projection.

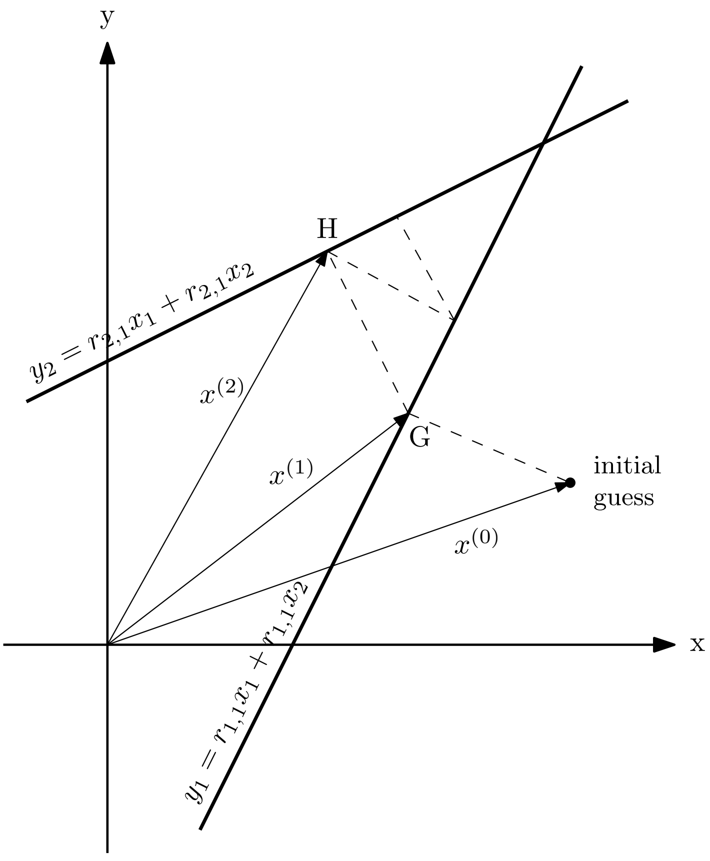

y
x
H
G
initial
guess
$x^{(0)}$
$x^{(1)}$
$x^{(2)}$
$y_2 = r_{2,1}x_1 + r_{2,1}x_2$
$y_1 = r_{1,1}x_1 + r_{1,1}x_2$

Figure 4.1.: Kaczmarz method of solving algebraic equations (adapted from [9, p. 278])

Mathematically, the projection onto the i-th hyperplane can be described by

$$x^{(k+1)} = \begin{cases} x^{(k)} + r_i \frac{y_i - \langle x^{(k)}, r_i \rangle}{\langle r_i, r_i \rangle} & , \langle r_i, r_i \rangle \neq 0 \\ x^{(k)} & , \langle r_i, r_i \rangle = 0 \end{cases} , \tag{4.3}$$

where r_i is the i-th line $(r_{i,1}, r_{i,2}, \dots, r_{i,J})$ of the projection matrix R and $\langle *, * \rangle$ is the inner product. For more details on (4.3) see [9, p. 278ff]. Tanabe proofed in [19] that if there exists a unique solution for the system of equations, then the series $x^{(kZ)}$ converges to x^*.

$$\lim_{k \to \infty} x^{(kZ)} = x^* \tag{4.4}$$

For further details confer [5], [6] and [9].

4.1 Algebraic Reconstruction Technique

A variety of different suggestions for the implementation exists for the ART algorithm. The following form shows a version of ART that is better suited for implementations than (4.3).

Definition 4.1. *The iterative step of ART is defined as*

$$x_j^{(k+1)} = x_j^{(k)} + r_{i,j} \frac{y_i - y_i^{(k)}}{\sum\limits_{j'=1}^{J} r_{i,j'}^2}, \tag{4.5}$$

where y_i is the measured ray-sum of the i-th ray, $y_i^{(k)}$ the calculated ray-sum of the i-th ray after the k-th iteration and $r_{i,j}$ is the (i,j)-th component of the projection matrix R [6].

The calculated ray-sum can be written as

$$\begin{aligned} y_i^{(k)} &= \left\langle x^{(k)}, r_i \right\rangle & (4.6) \\ &= \sum_{j=1}^{J} x_j^{(k)} r_{i,j} & (4.7) \end{aligned}$$

The equation (4.5) can be interpreted as an iteration step $k+1$ adding a correction term $\Delta x_j^{(k+1)}$ to the output of the previous (k-th) step. The correction term

$$\Delta x_j^{(k+1)} = x_j^{(k+1)} - x_j^{(k)} = r_{i,j} \frac{y_i - y_i^{(k)}}{\sum\limits_{j'=1}^{J} r_{i,j'}^2}. \tag{4.8}$$

is the difference of the measured ray-sum y_i and the current computed ray-sum $y_i^{(k)}$

normalized by $\sum_{j'=1}^{J} r_{i,j'}^2$.

Figure 4.2 visualizes the fundamental operation principle of ART. All projection rays are represented as planes in the Z-dimensional space (M projections with $2N+1$ rays each). Their intersection represents the required estimate x^*. For a better understanding, the hyperplanes are visualized as straight lines in $\mathbb{R}^2$.

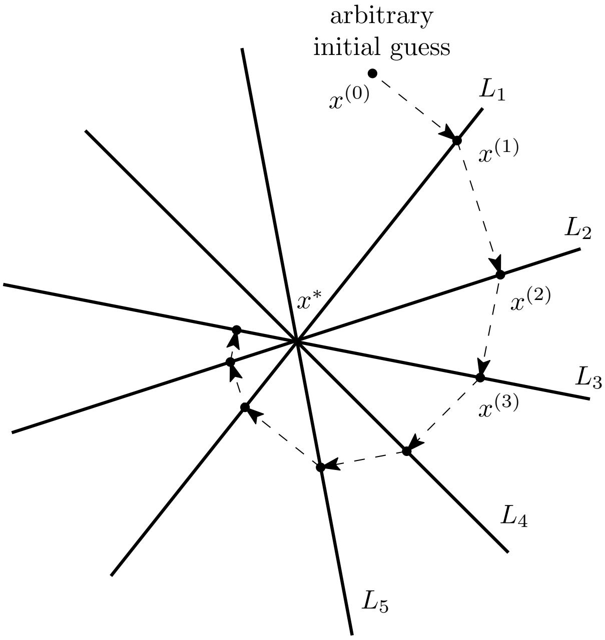


Figure 4.2.: Update step for ART

Starting with an arbitrary initial guess $x^{(0)}$, the current image vector is projected onto the next selected ray. With every projection-step, the current estimate is getting closer to the required estimate x^*. This can be seen by the progressive reduction of the distance from $x^{(1)}$, $x^{(2)}$ and $x^{(3)}$ to x^*. The iterative process is continued until a specified stopping criterion is reached. It is easy to see, that the order in which the beams are selected is decisive for the speed at which the algorithm converges. The closer the angle between two consecutive used rays is to 90°, the faster the result improves. If the two rays are exactly perpendicular to each other, the exact result can be achieved with only two projections-steps. In practical applications, the selection of the order of the rays is random.

A common technique to improve the performance of an ART algorithm is to introduce a so called relaxation parameter $\lambda^{(k)}$ to control the amount of the error correction. After introducing relaxation into the iterative step of (4.5) combined with the definition of

Value	**Movement regarding to the projection hyperplane L_k**
$\lambda^{(k)} < 0$	move away from L_k
$\lambda^{(k)} = 0$	there is no movement
$0 < \lambda^{(k)} < 1$	movement towards L_k, L_k is not reached
$\lambda^{(k)} = 1$	projection onto L_k, new estimate $x^{(k+1)} \in L_k$
$1 < \lambda^{(k)} < 2$	movement past L_k, but $x^{(k+1)}$ is closer to L_k than $x^{(k)}$
$\lambda^{(k)} = 2$	mirroring, the distance of $x^{(k+1)}$ is the same as for $x^{(k)}$
$\lambda^{(k)} > 2$	movement past L_k, but $x^{(k+1)}$ is further away from L_k

Table 4.1.: Influence of the relaxation parameter $\lambda^{(k)}$

the correction term in (4.8), the iterative step of ART can be written as

$$x_j^{(k+1)} = x_j^{(k)} + \lambda^{(k)} \Delta x_j^{(k)}, \tag{4.9}$$

where the correction $\Delta x_j^{(k+1)}$ is the difference between the resulting image vector $x_j^{(k+1)}$ and the previous image vector $x_j^{(k)}$. It can be interpreted as the distance of the vector $x_j^{(k)}$ to the projection plane and thus includes a right angle to it.

The influence of different relaxation parameters $\lambda^{(k)}$ for a projection of $x^{(k)}$ with respect to the hyperplane L_k is explained in Table 4.1. Values of $\lambda^{(k)} < 0$ or $\lambda^{(k)} > 2$ move the new value $x^{(k+1)}$ further away from the projection plane than $x^{(k)}$. Values of $\lambda^{(k)} = 0$ or $\lambda^{(k)} = 2$ keep the same distance to L_k. Therefore the relaxation parameter $\lambda^{(k)}$ should be restricted to $0 < \lambda^{(k)} < 2$.

Figure 4.3 visualizes the basic principle of relaxation. It shows the projection of an initial image vector $x^{(0)}$ with respect to three different planes $\{L_1, L_2, L_3\}$ where every projection uses a different relaxation parameter $\lambda^{(k)}$. The dashed gray circle represents the distance of current estimate $x^{(0)}$ to the solution x^*. If $\lambda^{(k)}$ is chosen 0 or 2 the projected points will be located on this circle. The projection on L_1 uses a relaxation parameter $\lambda^{(1)} = 1$. Therefore the projected image vector $x^{(1)}$ is on the hyperplane L_1. The projection on L_2 with $\lambda^{(2)} = 2$ results in a mirrored $x^{(2)}$ which lies on the dashed gray distance circle. The projection on L_3 uses an arbitrary $\lambda^{(3)} = 1.7$ which means that its result $x^{(3)}$ passes L_3 but is closer to it than the previous estimate $x^{(0)}$.

A well-chosen relaxation parameter can lead to a faster convergence of the algorithm and will make it more robust against noise.

Detailed information about this chapter can be found in [4], [5], [6] and [9].

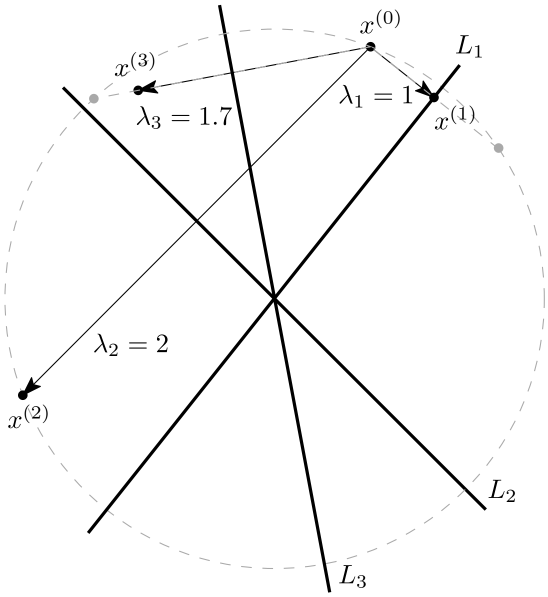


Figure 4.3.: Impact of the relaxation parameter

4.2 Simultaneous Algebraic Reconstruction Technique

The SART algorithm is based on the ART algorithm. The main difference is that instead of projecting the current estimate on only one hyperplane, it is projected (in a weighted form) onto all hyperplanes simultaneously. The result of $x^{(k)}$ of the iteration is the average of all those projections.

Definition 4.2. *The iterative step of SART is defined as*

$$x_j^{(k+1)} = x_j^{(k)} + \frac{1}{\sum\limits_{i=1}^{I} r_{i,j}} \sum_{i=1}^{I} \left\{ r_{i,j} \frac{y_i - \sum\limits_{j'=1}^{J} r_{i,j'} x_{j'}^{(k)}}{\sum\limits_{j'=1}^{J} r_{i,j'}} \right\}, \tag{4.10}$$

where y_i is the measured ray-sum of the i-th ray, $y_i^{(k)}$ the calculated ray-sum of the i-th ray after the k-th iteration and $r_{i,j}$ is the (i,j)-th component of the projection matrix R [9].

Figure 4.4 shows one iterative step of the SART algorithm. Starting with an arbitrary initial guess, the current estimation of the image will be projected on every hyperplane simultaneously. The result of the current iteration x^k is the average of all projections. Please be aware, that - for a better understanding - the algorithm has been simplified and therefore the visualization is not 100% accurate. The simplification is that the

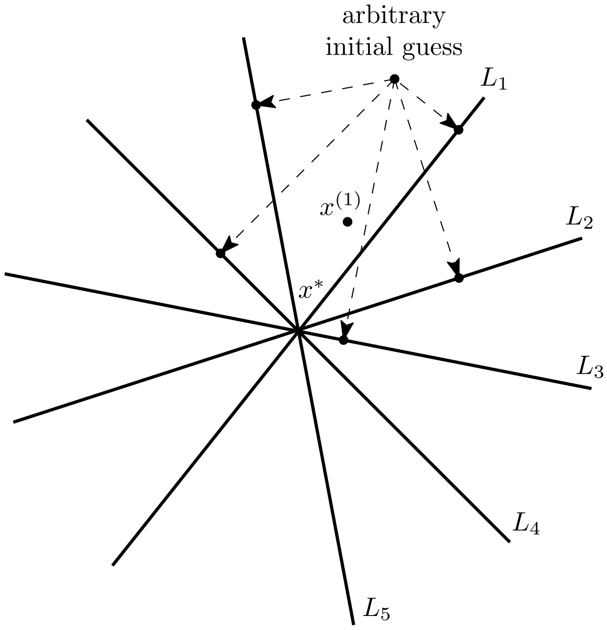


Figure 4.4.: Update step for SART

unweighted projection onto the hyperplane is visualized. The projection actually used in the visualization is stated in (4.11). The weight factor can be seen when comparing (4.10), (4.11) and (4.12).

$$x_j^{(k+1)} = x_j^{(k)} + \frac{1}{\sum\limits_{i=1}^{I} r_{i,j}} \sum_{i=1}^{I} \left\{ r_{i,j} \, \frac{y_i - \sum\limits_{j'=1}^{J} r_{i,j'} x_{j'}^{(k)}}{\sum\limits_{j'=1}^{J} r_{i,j'}^{2}} \right\} \tag{4.11}$$

$$x_j^{(k+1)} = x_j^{(k)} + \frac{1}{\sum\limits_{i=1}^{I} r_{i,j}} \sum_{i=1}^{I} \left\{ r_{i,j} \, \frac{\sum\limits_{j'=1}^{J} r_{i,j'}}{\sum\limits_{j'=1}^{J} r_{i,j'}^{2}} \, \frac{y_i - \sum\limits_{j'=1}^{J} r_{i,j'} x_{j'}^{(k)}}{\sum\limits_{j'=1}^{J} r_{i,j'}} \right\} \tag{4.12}$$

As for ART, the performance of the SART algorithm can be improved by introducing a relaxation parameter $\lambda^{(k)}$. The new update step is then

$$x_j^{(k+1)} = x_j^{(k)} + \lambda^{(k)} \frac{1}{\sum\limits_{i=1}^{I} r_{i,j}} \sum_{i=1}^{I} \left\{ r_{i,j} \frac{y_i - \sum\limits_{j'=1}^{J} r_{i,j'} x_{j'}^{(k)}}{\sum\limits_{j'=1}^{J} r_{i,j'}} \right\}. \tag{4.13}$$

The value of $\lambda^{(k)}$ is again restricted to $0 < \lambda^{(k)} < 2$.

All details on SART can be found in [1], [8] and [9].

5 Superiorization

The superiorization methodology is a heuristic approach to optimization. The basic idea is, that in many existing applications, there exist computationally-efficient iterative algorithms which produce constraint compatible solutions for some given constraints. These constraints might be physical properties of the object of interest or any other constraints obtained from any other source. In computerized tomography the constraints come from the detector readings of the CT scanner. Furthermore, a lot of these algorithms are perturbation resilient, which means that even if some kinds of changes are made at the end of each iteration, the result is still constraint compatible. This property is used in superiorization to steer an algorithm to a solution that is not only constraint compatible, but is also superior according to a specified optimization criterion.

Superiorization is a general approach, which provides a totally automatic procedure that turns an iterative algorithm into its superiorized version. It is applicable to many iterative procedures and optimization criteria. The methodology is called heuristic, as it is not guaranteed, that it will lead to the optimal solution of the problem.

The approach of designing an algorithm in a way that it uses alternating steps of different nature to get better results is well established. But none of the existing solutions provides a framework that automatically incorporates an optimization criterion into an existing iterative algorithm. Therefore superiorization is having the potential of saving a lot of time and effort for researchers.

In this chapter all declarations and definitions, that are required for the superiorization methodology, are provided. As this thesis deals with image reconstruction from projections in the field of computerized tomography, all examples are meant to be for image reconstruction of CT scans [7].

5.1 Problem sets, proximity function and ε-compatibility

In medical physics optimization is usually performed in a Euclidian space $\mathbb{R}^J$. In practice, the solution space is further restricted to be an nonempty subset Ω of $\mathbb{R}^J$. In the field of image reconstruction the result of an optimization is the image vector

x (see Definition 3.10). As images normally only contain positive values, Ω might be restricted to $\Omega = \mathbb{R}^J_+$.

Definition 5.1. *The* ***problem set*** $\mathbb{T}$ *is the set of all problems* $T \in \mathbb{T}$ *where each* T *is the description of the constraints of one particular problem [7].*

In computerized tomography each T is the problem of reconstructing the picture for a particular patient at a particular time. $\mathbb{T}$ is the set of all possible reconstruction problems.

Definition 5.2. *The* ***proximity function*** $\mathcal{P}r$ *on the problem set* $\mathbb{T}$ *is defined as function from the solution space into the positive real numbers* $\mathcal{P}r_T(x) : \Omega \mapsto \mathbb{R}_+$*. It indicates how incompatible the solution* x *is with the given constraints of* $T \in \mathbb{T}$*.* x *is said to be perfectly constraint compatible with the problem* T *if* $\mathcal{P}r_T(x) = 0$ *[7].*

When reconstructing CT images, $\mathcal{P}r_T(x)$ could be the norm-distance $\|y - Rx\|$ of the reconstruction problem in (4.1), where y is the measurement vector of the CT scan (see Definition 3.5), R is the projection matrix (see Definition 3.11) and x is the image vector. Another legitimate choice for the proximity function is the weighted squared distance (see Section 7.5).

Definition 5.3. *A* ***problem structure*** $\langle \mathbb{T}, \mathcal{P}r \rangle$ *is defined as the combination of a nonempty problem set* $\mathbb{T}$ *and a proximity function* $\mathcal{P}r$ *[7].*

Definition 5.4. *The solution* $x \in \Omega$ *for a problem* $T \in \mathbb{T}$ *is* ε***-compatible*** *regarding a problem structure* $\langle \mathbb{T}, \mathcal{P}r \rangle$ *when* $\mathcal{P}r_T(x) < \varepsilon$*, where* ε *is a non-negative number [7].*

The ε-compatibility is needed, as in practical applications all measurements are noisy. Therefore it is unlikely that a perfect reconstruction x for the problem $T \in \mathbb{T}$ exists. ε is the threshold for the proximity function $\mathcal{P}_r$, below which a image reconstruction x is acceptable ($\mathcal{P}r_T(x) < \varepsilon$). The proximity function can be interpreted as the stopping criterion of an iterative algorithm, confer [7].

5.2 Algorithms and Output

The concept of algorithms is defined in the general context of problem structures. For technical reasons which will be explained in Section 5.4, an additional set Δ is introduced such that $\Omega \subseteq \Delta \subseteq \mathbb{R}^J$. Both Ω and Δ are assumed to be known and fixed for any particular problem structure $\langle \mathbb{T}, \mathcal{P}r \rangle$.

Definition 5.5. *An* ***algorithm*** *P assigns to each problem $T \in \mathbb{T}$ an operator $P_T : \Delta \mapsto \Omega$ [7].*

Definition 5.6. *A* ***iterative process*** *(or iterative algorithm) produces a (potential) infinite sequence*

$$\left(\left(P_T\right)^k (x) \right)_{k=0}^{\infty} = x,\ P_T\left(x\right),\ P_T\left(P_T\left(x\right)\right),\ \ldots \tag{5.1}$$

where x is any initial point $x \in \Omega$ and P_T is an algorithm as defined in Definition 5.5 [7].

The two reconstruction algorithms ART and SART described in chapter 4 are iterative processes as defined above. The updatestep is the algorithm P and the iterative execution of the updatestep is the iterative process.

Definition 5.7. *For a problem structure $\langle \mathbb{T}, \mathcal{P}r \rangle$, a $T \in \mathbb{T}$, an $\varepsilon \in \mathbb{R}_+$ and a sequence $RS = \left(x^k\right)_{k=0}^{\infty}$, the* ***output*** *$O\left(T, \varepsilon, RS\right)$ is used to denote the $x \in \Omega$ that has the following properties:*

- *$\mathcal{P}r_T(x) \leq \varepsilon$,*
- *there is a non-negative integer K such that $x^K = x$, $x^K \in RS$ and*
- *for all non-negative integers $k < K$, the proximity function $\mathcal{P}r_T(x) > \varepsilon$.*

If there exists such an x then $O\left(T, \varepsilon, RS\right)$ is defined, otherwise it is undefined [7].

If RS is a (infinite) sequence of points that is produced by an algorithm that solves the problem T without termination criterion, then $O\left(T, \varepsilon, RS\right)$ is the output produced by that algorithm when the termination criterion $\mathcal{P}r_T(x) \leq \varepsilon$ is added to it. The point x donates the first ε-compatible point within the sequence RS, which was reached after K iterations, see [7].

5.3 Bounded Perturbation Resilience

An iterative algorithm P is bounded perturbation resilient for a problem structure $\langle \mathbb{T}, \mathcal{P}r \rangle$ if the sequence $\left(\left(P_T\right)^k (x) \right)_{k=0}^{\infty}$ generated by it still converges - even if the result of every step is perturbed. For real applications bounded perturbation resilience is not sufficient, as it can only be used for problems $T \in \mathbb{T}$, for which a perfectly constraint compatible solution - which means that the proximity function $\mathcal{P}r_T(x) = 0$ - exists. As the data of real applications is mostly noisy, this is not the case. Therefore bounded perturbation resilience has been extended to strongly perturbation resilience.

Definition 5.8. *An algorithm P for a problem structure $\langle \mathbb{T}, \mathcal{P}r \rangle$ is **strongly perturbation resilient** if, for all $T \in \mathbb{T}$*

- *there exists an $\varepsilon \in \mathbb{R}_+$, such that $O\left(T, \varepsilon, \left((P_T)^k(x)\right)_{k=0}^{\infty}\right)$ is defined for every $x \in \Omega$ and*
- *for all $\varepsilon \in \mathbb{R}_+$, such that $O\left(T, \varepsilon, \left((P_T)^k(x)\right)_{k=0}^{\infty}\right)$ is defined for every $x \in \Omega$, also $O(T, \varepsilon', RS)$ is defined for every $\varepsilon' > \varepsilon$ and for every sequence $RS = \left(x^k\right)_{k=0}^{\infty}$ of points in Ω generated by*

$$x^{k+1} = P_T\left(x^k + \beta_k v^k\right), \text{ for all } k \geq 0\,, \tag{5.2}$$

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 $\left(v^k\right)_{k=0}^{\infty}$ of vectors in $\mathbb{R}^J$ is bounded and, for all $k > 0, x^k + \beta_k v^k \in \Delta$ [7].

The properties of Definition 5.8 states that all perturbed sequences contains a ε'-compatible point, when for every problem T and any non-negative number ε an ε-compatible solution for every initial point $x \in \Omega$ (and $\varepsilon' > \varepsilon$) exists. This means that the perturbed version of the algorithm produces a ε'-compatible output $O(T, \varepsilon', RS)$, confer [3] and [7].

5.4 Optimization criterion and non-ascending vector

Additionally to the constraint optimization problem of the previous sections, it might be desirable, to improve the reconstruction towards a secondary criterion which is denoted as optimization criterion ϕ.

Definition 5.9. *An **optimization criterion** ϕ is a function $\phi : \Delta \mapsto \mathbb{R}$ which indicates how good a point $x \in \Delta$ fulfills the desired conditions. A point $x_1 \in \Delta$ is considered superior to another point $x_2 \in \Delta$ when $\phi(x_1) < \phi(x_2)$ [7].*

Some valid optimization criteria are listed in Section 3.6. In the context of superiorization ϕ_1 are the given constraints of $T \in \mathbb{T}$ and ϕ_2 is the optimization criterion ϕ.

The main idea of superiorization is to use the perturbations of (5.2) to steer a strong perturbation resilient algorithm, which produces constraint-compatible solutions, into a superiorized algorithm. The new algorithm not only produces a constraint-compatible output, the output is also expected to be superior regarding the secondary optimization criterion ϕ. This is achieved by adding bounded perturbations $\beta_k v^k$ to the solution vector x^k. The optimization criterion ϕ will have the property $\phi(x + \beta_k v^k) \leq \phi(x)$. In order to achieve this, v^k must be a non-ascending vector.

Definition 5.10. *A vector $v \in \mathbb{R}^J$ for a given function $\phi : \Delta \mapsto \mathbb{R}$ and a point $x \in \Delta$ is called a **non-ascending vector** for ϕ at x if*

- $\|v\| \leq 1$ *and*
- *there exists a $\delta > 0$ such that for all $\beta \in [0, \delta]$*
  $(x + \beta v) \in \Delta$ *and* $\phi(x + \beta v) \leq \phi(x)$ *[7].*

As $(x + \beta v) \in \Delta$ can be outside of Ω, it is important, that the algorithm P is defined as $P : \Delta \mapsto \Omega$. This guarantees, that the final result of the superiorized version of the algorithm is still in the defined solution space Ω.

Note that independent of the optimization criterion ϕ and the point x, the zero-vector (all components are 0) is always a non-ascending vector. This is a useful property for proving the convergence of the algorithm, but in order to have an improvement regarding the optimization criterion, the vector v should have the property $\phi(x+\beta v) < \phi(x)$ rather than $\phi(x + \beta v) \leq \phi(x)$, see [7].

5.5 Superior Version of an Algorithm

In this section it is described how an iterative algorithm P can be automatically turned into its superiorized version. The original, unsuperiorized version of the algorithm is stated in algorithm 1.

In lines 1-2 the variables are initialized. k is the number of the current iterations and $\overline{x}$ is the initialization of the result vector. The loop in line 3-6 is executed until the proximity function $\mathcal{P}r(x^k)$ reaches a value that is smaller than a defined value ε. The loop includes the execution of the iterative step $P_T(x^k)$ (line 4).

Algorithm 1 Algorithm P

```
1: set $k = 0$
2: set $x^k = \overline{x}$
3: while $\mathcal{P}r(x^k) > \varepsilon$ do
4:     set $x^{k+1} = P_T(x^k)$
5:     set $k = k + 1$
6: end while
```

Theorem 3. *Let Ω and Δ be the underlying sets for a problem structure $\langle \mathbb{T}, \mathcal{P}r \rangle$ where $\Omega \subseteq \Delta \subseteq \mathbb{R}^J$, $P : \Delta \mapsto \Omega$ is an iterative, strong perturbation resilient algorithm for $\langle \mathbb{T}, \mathcal{P}r \rangle$ and $\phi : \Omega \mapsto \mathbb{R}$. Algorithm 2 is called the superiorized version of the algorithm P. It produces for any problem $T \in \mathbb{T}$ and any point $x \in \Omega$ a sequence $RS_T = \left(x^k\right)_{k=0}^{\infty}$. The result sequence RS of the superiorized algorithm is ε'-compatible and expected to be superior to the original algorithm P with regard to the optimization criterion ϕ [7].*

The proof of the theorem can be found in [7].

The superiorized version shown in algorithm 2 depends on the initialization of the initial vector $\overline{x} \in \Omega$, a positive integer N, which defines the number of executions of the superiorization step, and it needs a summable sequence $(\gamma_l)_{l=0}^{\infty}$ of positive real numbers. One example of such a sequence could be $\gamma_l = a^l$ where $0 < a < 1$.

In the lines 1-3 the variables are initialized. k is the number of the current iteration, l is the integer sequence for picking values from $(\gamma_l)_{l=0}^{\infty}$, and x^0 is the initial result vector. Every run of the loop started in line 4 represents one iteration. It is performed as long as the proximity function $\mathcal{P}r$ is bigger than a defined threshold ε. During one iterative step, there is one application of the original algorithm P_T but N executions of the inner loop (line 7-20) where superiorization is performed.

In the superiorization part n is the counter for the inner superiorization loop (line 10-19), whereas $x^{k,n}$ is the current solution, $v^{k,n}$ is the non-ascending vector and $\beta_{k,n}$ is a positive real number picked from $(\gamma_l)_{l=0}^{\infty}$ (l is increased in every iteration of the superiorization loop). In that innermost loop, increasingly smaller perturbations are added to the previous solution until the conditions $z \in \Delta$ and $\phi(z) \leq \phi(x^k)$, where x^k is the solution of the previous iteration, are satisfied.

Algorithm 2 Superiorized Version of Algorithm P (adapted from [7, 14])

```
 1: set $k = 0$
 2: set $x^k = \overline{x}$
 3: set $l = -1$
 4: while $\mathcal{P}r(x^k) > \varepsilon$ 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 non-ascending vector for $\phi$ at $x^{k,n}$
 9:         set $loop = true$
10:         while loop do
11:             set $l = l + 1$
12:             set $\beta_{k,n} = \gamma_l$
13:             set $z = x^{k,n} + \beta_{k,n} v^{k,n}$
14:             if $z \in \Delta$ and $\phi(z) \leq \phi(x^k)$ then
15:                 set $n = n + 1$
16:                 set $x^{k,n} = z$
17:                 set $loop = false$
18:             end if
19:         end while
20:     end while
21:     set $x^{k+1} = P_T x^{k,N}$
22:     set $k = k + 1$
23: end while
```

After N superiorization steps have been performed, the original algorithm P_T is applied on $x^{k,N}$. This produces a constraint-compatible output which can be expected to be superior to the output of the algorithm without superiorization regarding to the optimization criterion ϕ. This means that $\phi(x^{k,N}) \leq \phi(x^k)$. A more detailed description of the superiorization methodology can be found in [7].

5.6 Adapted Version of the Superiorized Algorithm

The superiorization algorithm 2 is a general approach to improve any iterative algorithm by performing perturbations in the form of $x^{k,n+1} = x^{k,n} + \beta_{k,n} v^{k,n}$ within every iteration. As long as the perturbation term $\beta_{k,n} v^{k,n}$ does not result in an acceptable value, the inner while loop (10-19) is executed and the variable l - which controls $\beta_{k,n}$ - will be increased. If l is very large, β_k will be very small. As a result of this, the perturbations $\beta_{k,n} v^{k,n}$ will be small as well and they will not have an significant impact on the reconstruction anymore. As the mathematical proof of superiorization relies on the fact that $(\beta_k)_{k=0}^{\infty}$ is a summable sequence, which is in the proof of algorithm 2 in

[7] only the case when $(\beta_k)_{k=0}^{\infty}$ is a subsequence of $(\gamma_l)_{l=0}^{\infty}$, it is not possible to reset l. If it would be possible to reset l, the influence of superiorization might be held on a higher level.

The adoption of the superiorization algorithm proposed in algorithm 3 offers the possibility to reset the l-value after every iteration and still fulfills all mathematical conditions for its convergence.

Line 5 holds the main change: the integer index l for picking the next element from the sequence $(\gamma_l)_{l=0}^{\infty}$ is reset to $l = k$, where k is the number of the current iteration. Another minor change is that the incrementation of l is moved back to line 18. This change is just of cosmetic nature and is not related to the modification of the algorithm itself. Therefore it will not be mentioned in the further discussion.

Algorithm 3 Adapted version of the superiorized algorithm

```
 1: set $k = 0$
 2: set $x^k = \overline{x}$
 3: while $\mathcal{P}r(x^k) > \varepsilon$ do
 4:     set $n = 0$
 5:     set $l = k$
 6:     set $x^{k,n} = x^k$
 7:     while $n < N$ do
 8:         set $v^{k,n}$ to be a non-ascending 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:             set $l = l + 1$
19:         end while
20:     end while
21:     set $x^{k+1} = P_T x^{k,N}$
22:     set $k = k + 1$
23: end while
```

According to [7], for the proof, that the new algorithm with its changes still fulfills theorem 3 (and therefore still converges), it is sufficient to show that x^{k+1} of every iteration can be written in the form $x^{k+1} = x^k + \beta_k v^k$ where $(\beta_k)_{k=0}^{\infty}$ is a summable sequence of positive real numbers and $\left(\upsilon^k\right)_{k=0}^{\infty}$ is bounded.

Proof. In comparison to the proof of the existing algorithm in [7], the proposed algorithm requires a sequence $(\gamma_l)_{l=0}^{\infty}$ of positive real numbers which is not only summable but also monotonously decreasing. Although this is a stricter restriction it does not change anything in practice as the proposed sequence of [7] $\gamma_l = a^l$ where $0 < a < 1$ fulfills that condition anyway.

The results of the superiorization steps of each iteration (loop in line 7-20) can be written as

$$\begin{array}{ccccc} x^{k,1} & = & x^k & + & \beta_{k,0}\upsilon^{k,0} \\ x^{k,2} & = & x^{k,1} & + & \beta_{k,1}\upsilon^{k,1} \\ \vdots & & \vdots & & \vdots \\ x^{k,N} & = & x^{k,N-1} & + & \beta_{k,N-1}\upsilon^{k,N-1} \end{array} \tag{5.3}$$

which can be accumulated to

$$x^{k,N} = x^k + \sum_{n=0}^{N-1} \beta_{k,n}\upsilon^{k,n}\,. \tag{5.4}$$

If β_k is defined as

$$\beta_k = \beta_{k,0} = \gamma_k \tag{5.5}$$

this results in the required monotonously decreasing summable sequence $(\beta_k)_{k=0}^{\infty}$ of positive real numbers by the definition of $(\gamma_l)_{l=0}^{\infty}$.

By using β_k as defined in (5.5), the iterative step of (5.4) can be rewritten to:

$$\begin{aligned} x^{k+1} = x^{k,N} &= x^k + \sum_{n=0}^{N-1} \beta_{k,n}\upsilon^{k,n} \\ &= x^k + \sum_{n=0}^{N-1} \frac{\beta_k}{\beta_k}\beta_{k,n}\upsilon^{k,n} \\ &= x^k + \beta_k \sum_{n=0}^{N-1} \frac{\beta_{k,n}}{\beta_k}\upsilon^{k,n} \end{aligned} \tag{5.6}$$

As a result of this υ^k must be

$$\upsilon^k = \sum_{n=0}^{N-1} \frac{\beta_{k,n}}{\beta_k}\upsilon^{k,n}\,. \tag{5.7}$$

As l is increased whenever the algorithm passes line 18, it is obvious, that $\beta_{k,n} \leq \beta_k$ for every $0 < n < N$, and therefore $\frac{\beta_{k,n}}{\beta_k} \leq 1$. Each $\left\| v^{k,n} \right\| \leq 1$ by the definition of the non-ascending vector. As a result of those two limitation $\left\| v^k \right\| \leq N$ and therefore the sequence $\left(v^k\right)_{k=0}^{\infty}$ is bounded. ■

It is important to mention, that the new version of superiorization presented in algorithm 3 is not necessarily an improvement of the existing algorithm 2, but is just an additional option how superiorization can be applied. As superiorization is a general approach which works independent of the choice of algorithm P and optimization criterion ϕ will be used, it is not possible to make a general statement on which of the two algorithms is better. Both versions will work for all suitable algorithms P optimization criterion ϕ. The advantage of the new version is, that it emphasizes superiorization, a drawback is that it tends to longer execution times. At the end, the operator needs to decide which of the versions is better suited for his particular problem.

A good compromise to the two algorithms might be to set l at the beginning of every iteration (line 5) to a random number between the current iteration k and the value of l from the last iteration ($l_k = rand(k, l_{k-1})$). This should reduce the execution time while still keeping the effect of superiorization on a higher level than in algorithm 2. The proof of the convergence is the same as the one for setting $l = k$.

6 Implementation

6.1 SNARK

SNARK is a software package for reconstruction of 2D images from their 1D projections. SNARK14 - the current version of it - provides a total framework for reconstruction from projections for both, simulated and real data. It provides the possibility to create projection data for parallel and divergent projection geometry and offers frequently used reconstruction algorithms. The source code is freely available at www.dig.cs.gc.cuny.edu and can easily be extended with user defined functionality as described in [4] and [10].

All features implemented during the work for this thesis are needed in order to be able to perform the experiment described in chapter 7.

6.2 Simultaneous Algebraic Reconstruction Technique

The simultaneous algebraic reconstruction technique is a series expansion reconstruction algorithm (see Section 3.5). As described in Chapter 4, it tries to solve the reconstruction problem $y = Rx + e$ (3.25) by performing a projection of an estimated result image onto the system of equations. This projection is executed iteratively until a defined stopping criterion is reached.

As SART was not available in SNARK, it has been implemented and incorporated into the latest version of it. The implementation has been done according to the descriptions in Section 4.2. SART is available for pixel and blobs basis functions. The only configuration parameter available is the relaxation parameter λ of (4.13).

6.3 Weighted Squared Distance

The weighted squared distance (WSQD) (see Section 7.5) is a quadratic distance measure. As it is proven that SART minimizes the WSQD (see [8]), it has been chosen as distance measure for the experiment.

For the experiment WSQD is needed as a stopping criterion and as a figure of merit. As both options have not been available in the existing SNARK version, they have been implemented and incorporated into SNARK14.

7 Comparison of ART and SART

This chapter deals with the comparison of iterative algorithms with and without superiorization for image reconstruction from projections. It compares the algebraic reconstruction technique (ART) with the simultaneous algebraic reconstruction technique (SART) using pixel basis functions as defined in (3.18). Both algorithms are frequently used in CT image reconstruction. As a figure of merit (FOM) the average value of the image-wise region of interest (IROI) (see Section 7.2) is used.

As a gold standard, the ART version using generalized Kaiser-Bessel window functions (blobs) as described in [6] is used, refer to Section 7.7.2 for details.

The creation of the projection data, the reconstruction of the images and the evaluation is done in SNARK14.

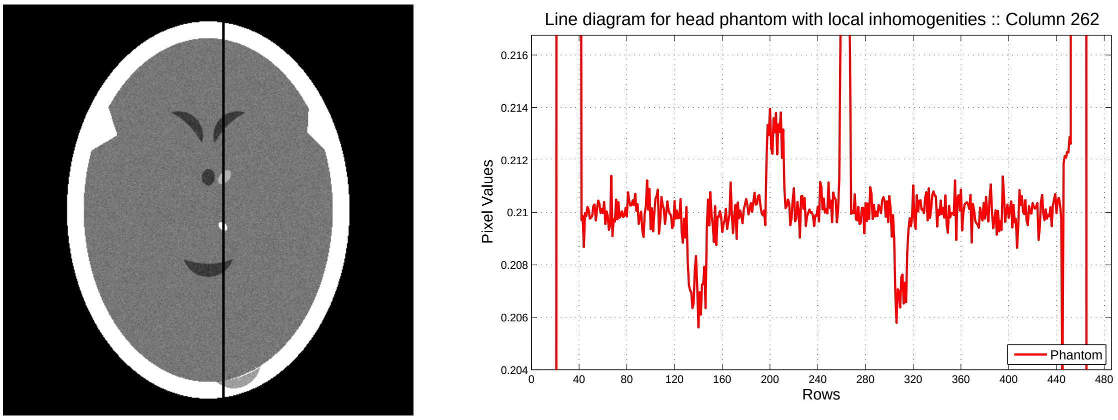
7.1 Test Image Set

In order to be able to evaluate the relative efficacy of two or more reconstruction algorithms against each other, it is necessary to have a large enough sample set of statistical independent projection data. SNARK with its Experimenter (see Section 7.6 and [10]) offers the possibility to generate as much random test images as needed for a sample set. In medical imaging such specially designed images are called phantoms. A commonly used phantom - the head phantom - is shown in Figure 7.1a [10].

7.2 Methods of Image Comparison

When reconstructing an image from simulated projection data, the quality of the reconstructed image can be measured by comparing it to the digitization of the generated projection data. Both, the reconstructed image and the digitization of the phantom must be of the same size.

Visual evaluation is subjective and therefore not suited for academic purposes. A more accurate technique is to compare sequences of pixels (rows or columns) which pass through interesting features of the image. One way to evaluate the quality of a reconstruction is to compare the densities of the original and the reconstructed phantom. Figure 7.1 shows how the values for such a comparison are taken: Figure 7.1a shows



(a) A head phantom with local inhomogeneities

(b) The densities along the column indicated in the phantom

Figure 7.1.: A head phantom and its density distribution at a specified column CT images

the phantom. In order to clearly see the features in the interior of the skull, all values below 0.204 are represented as black and all values above 0.21675 are represented as white. This way, the density values of the plot are stretched to the interesting region and the small features are made visible. The black line indicates along which column the density values have been taken. Figure 7.1b visualizes these density values as line graph. The actual comparison is done by comparing the graph of the original phantom with the graph of the reconstructed image.

Distance measures can be used to measure how close a reconstruction is to its original phantom. As single numbers cannot take care of all ways in which two images may differ, such values have to be used carefully. Nevertheless, if the values are used carefully, the numbers can provide a rough measure of the closeness of the reconstruction to the original phantom [6].

Definition 7.1. *The* ***normalized root mean square distance*** *d of two n x n images is defined as*

$$d = \sqrt{\frac{\sum_{u=1}^{n}\sum_{v=1}^{n}(t_{u,v} - x_{u,v})^2}{\sum_{u=1}^{n}\sum_{v=1}^{n}(t_{u,v} - \bar{t})^2}}, \tag{7.1}$$

where $t_{u,v}$ is the v-th pixel of the u-th row from the digitization of the phantom, and $x_{u,v}$ the reconstruction of the same pixel. $\bar{t}$ is the average density of the digitized phantom.

The normalized root mean square distance d is sensitive to large differences of a small amount of pixels.

Definition 7.2. *The* ***normalized mean absolute distance*** r *of two* n *x* n *images is defined as*

$$r = \frac{\sum_{u=1}^{n} \sum_{v=1}^{n} |t_{u,v} - x_{u,v}|}{\sum_{u=1}^{n} \sum_{v=1}^{n} |t_{u,v}|}, \tag{7.2}$$

where $t_{u,v}$ *is the* v*-th pixel of the* u*-th row from the digitization of the phantom, and* $x_{u,v}$ *the reconstruction of the same pixel [6].*

In comparison to the normalized root mean square distance, the normalized mean absolute distance r is not sensitive to a large errors. It emphasizes the importance of a lot of small deviations of the density values, confer [6].

7.2.1 Task-Oriented Comparison of Algorithm Performance

Task-oriented comparison of algorithm performance is a methodology based on statistical hypothesis testing (SHT). It allows to evaluate the relative efficacy of reconstruction methods for a given task. The evaluation methodology answers the following basic question: given a specific medical problem, what is the relative merit of two (or more) image reconstruction algorithms in presenting images that are helpful for solving the problem?

The method consists of four steps:

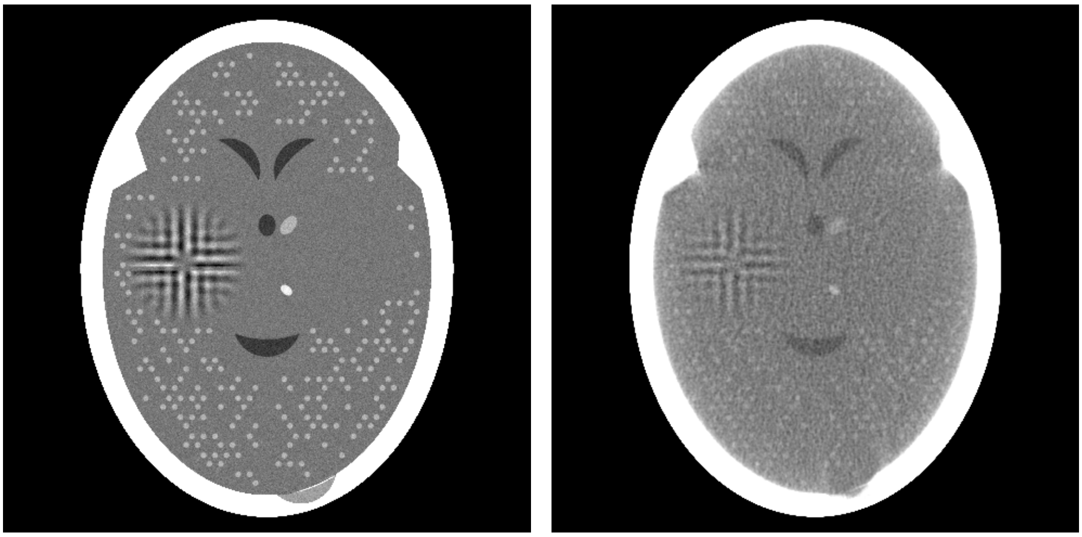
1. generation of random samples,
2. reconstruction from the data by each of the algorithms,
3. assignment of an appropriate figure of merit (FOM) to each reconstruction and the
4. calculation of statistical significance based on the assigned FOM by which the null hypothesis that the reconstructions are equally helpful for solving the problem at hand can be rejected.

In order that the comparison of algorithms is relevant to a particular task, the steps must be adopted to the actual task. Especially, the FOM must be chosen appropriately. In this thesis the image-wise region of interest (see Section 7.3 for details) is used as figure of merit. It was especially designed for comparing the performance of various CT reconstruction algorithms in detecting small low-contrast tumors in the human brain. For all details on task-oriented comparison of algorithm performance see [6].

7.3 Image-wise region of interest

The image-wise region of interest (IROI) is a figure of merit for comparing the performance of various CT reconstruction algorithms in detecting small, low-contrast tumors in the human brain. Therefore it needs specially developed test data. The test data used in this thesis is based on the head phantom shown in Figure 7.1a.

For the comparison, a large number of pairs of potential tumor sites are specified. All sites are symmetrically placed in the left and right half of the brain which means that every tumor site in the left half has an exact counterpart in the right half. Those two sites are considered as one pair. In every pair exactly one of the two possible sites holds a tumor with equal probability for either side. Figure 7.2a shows such a generated head phantom. The generated tumors are clearly visible. In the reconstruction shown in Figure 7.2b the tumors are hard to see. That behavior of the reconstruction algorithm is intended: the performance of a reconstruction algorithm can be measured best when the detection of the evaluated features is possible but difficult.



(a) A random sample for the algorithm comparison

(b) The reconstruction of the random sample

Figure 7.2.: Test images for a task-oriented comparison of algorithm performance using IROI

A set of phantoms created as described above is statistically relevant as - additionally to the introduced, randomly populated tumor-sites - Gaussian distributed noise is added as local inhomogeneities.

Definition 7.3. *The **image-wise region of interest (IROI)** is defined as*

$$IROI = \frac{\dfrac{\sum\limits_{b=1}^{B} (\alpha_t^r(b) - \alpha_n^r(b))}{\sqrt{\sum\limits_{b=1}^{B} \left(\alpha_n^r(b) - \dfrac{1}{B}\sum\limits_{b'=1}^{B} \alpha_n^r(b')\right)^2}}}{\dfrac{\sum\limits_{b=1}^{B} (\alpha_t^p(b) - \alpha_n^p(b))}{\sqrt{\sum\limits_{b=1}^{B} \left(\alpha_n^p(b) - \dfrac{1}{B}\sum\limits_{b'=1}^{B} \alpha_n^p(b')\right)^2}}}, \tag{7.3}$$

where B is the total number of tumor-site pairs. $\alpha_t^r(b)$ is the average density of the tumor site of the b-th pair actually holding the tumor, and $\alpha_n^r(b)$ for the non-tumor site. $\alpha_t^p(b)$ and $\alpha_n^p(b)$ are defined respectively for the original phantom [6].

For the IROI only pixels that belongs to the specified tumor sites are from interest. Therefor, (7.3) restricts the pixel used to those that belong to the specified tumor site pairs. α_t is the calculated average density of all pixels of the site actually holding the tumor, and α_n is the average density of the non-tumor site.

The nominator is the exact same calculation for the reconstruction as the denominator is for the phantom. Therefore the $IROI = 1$ for a perfect reconstruction.

In the reconstruction part of the equation (nominator), $\sum_{b=1}^{B} (\alpha_t^r(b) - \alpha_n^r(b))$ is the signal of the tumor and $\sqrt{\sum_{b=1}^{B} \left(\alpha_n^r(b) - \frac{1}{B}\sum_{b'=1}^{B} \alpha_n^r(b')\right)^2}$ is the standard deviation of the regions with no tumors in it. This can be interpreted as the level of noise. The division using the standard deviation can be considered as normalization. The same applies to the phantom part (denominator).

Experiments with human observers showed that the results of the image-wise region of interest correlates well with the human perception, see [6].

7.4 Total Variation Minimization

Total variation (TV) is a popular secondary optimization criterion. Using TV results in smoothing the final image while - to some extend - it preserves its edges. Total variation

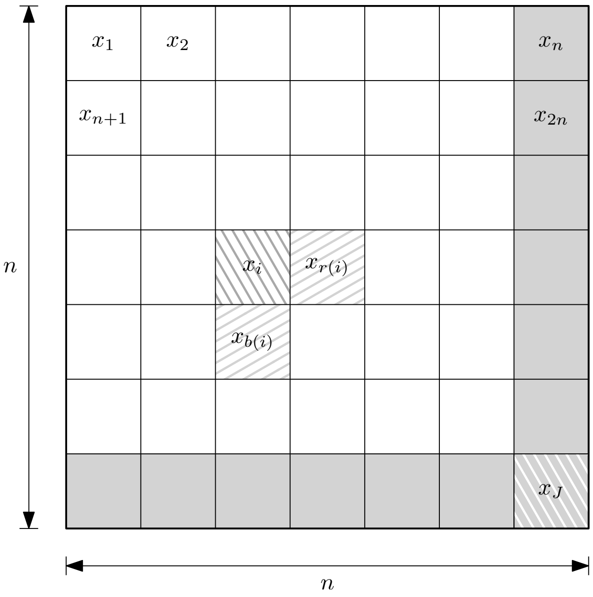


Figure 7.3.: Selection of Pixel for TV evaluation

is simple to implement, reasonably fast and delivers good results even for very noisy images. Currently it is considered as a state-of-the-art algorithm for removing noise from images.

Definition 7.4. *Let T be the set of all indices of pixel except the ones in the rightmost column and the bottom row of a square n x n pixel image. Then the* ***total variation (TV)*** *is defined as the function*

$$TV(x) = \sum_{i \in T} \sqrt{\left(x_{r(i)} - x_i\right)^2 + \left(x_{b(i)} - x_i\right)^2}, \tag{7.4}$$

where $x_{r(i)}$ is the pixel right to the current pixel and $x_{b(i)}$ the one below it [6] [17].

Figure 7.3 visualizes the position of the mentioned pixels from (7.4) within the reconstruction region. The light gray pixels at the right side and the bottom are not part of T, but they still contribute to the TV as either the right pixel $x_{r(i)}$ or the bottom pixel $x_{b(i)}$. The only pixel, which does not contribute to the TV, is pixel x_J. Further information can be found in [6], [16] and [17].

7.5 Weighted Squared Distance

In order to be able to objectively compare the reconstruction results of the different algorithms, they must all be stopped at the same quality. Therefore a criterion to measure the reconstruction quality must be available. One possible criterion is the weighted squared distance.

Definition 7.5. *Let* $y = (y_i)_{i=1}^{I}$ *be the set of all measured ray integrals of a CT data acquisition, and* $x = (x_i)_{i=1}^{J}$ *be the current estimation of the image reconstruction, then the* ***weighted squared distance (WSQD)*** *is defined as*

$$Pr(x) = \sum_{i=1}^{I} \frac{\left(y_i - \sum_{j=1}^{J} r_{i,j} x_j \right)^2}{\sum_{j=1}^{J} r_{i,j}}, \tag{7.5}$$

where $r_{i,j}$ *is the* (i,j)*-th element of the projection matrix* R *[8].*

Jiang and Wang proved in [8] that SART minimizes the weighted squared distance.

7.6 SNARK Experimenter

SNARK Experimenter is an execution mode of SNARK that allows to compare two or more reconstruction algorithms in a statistically sound manner. It supports the task-oriented comparison of algorithm performance as described in Section 7.2.1. All details on how to use SNARK Experimenter can be found in [4] and [10].

7.7 The Experiment

The experiment carried out in this thesis is a statistically rigorous comparative evaluation of the following iterative methods for image reconstruction from projections:

- ART using blobs
- ART without superiorization, using pixel basis functions
- ART with superiorization, using pixel basis functions
- SART using blobs
- SART without superiorization, using pixel basis functions
- SART with superiorization, using pixel basis functions

The experiment is executed as a task-oriented comparison between the algorithms. All algorithms are used to reconstruct 30 different, randomly generated projection data sets. The results are the basis of the comparison. As SNARK with its Experimenter offers a framework, that is capable of doing this, there is very little manual work involved. The main time factor is the execution time of the reconstruction runs.

As described in Section 7.2.1 the steps of the experiment are the

1. generation of random phantoms, the
2. reconstruction from the data by each of the algorithms, the
3. calculation of the figures of merit and the
4. statistical evaluation of the results.

All SNARK input files can be found in the Appendix.

7.7.1 Data generation

In order to obtain statistically significant results, we need to obtain a large number of generated phantoms. The number of phantoms generated for this experiment is 30. The samples are fully automatically generated by the SNARK Experimenter according to the description in Section 7.3.

The data is generated using the standard (divergent) projection geometry as shown in Figure 7.4. All line rays of one projection start at a common point (the source position). The source has a fixed distance to the origin. The $2N + 1$ detectors are equally spread on the detector strip where the detector strip is wide enough, that the whole reconstruction region is covered for all projections (source positions). The needed area is intimated as dashed circle. Source and detector strip are rotated around the image region by an angle of Δ. This is repeated for a total of M views where $M\Delta = 2\pi$, confer [6] and [10].

7.7.2 Image reconstruction

The image reconstruction is done based on the data of the phantoms generated in the previous step. In [6], a particular version of ART using blobs - namely that with relaxation parameter $\lambda^k = 0.05$ and efficient ordering[1] - provided a better reconstruction performance than all other settings against which it has been compared. Therefore it has been defined as reference (gold standard) for the experiment in this thesis.

The reference algorithm will be reconstructed in five iterations through the data. All other algorithms will be stopped when they reach a proximity function smaller or

---

[1] A ray selection method provided by SNARK (see [6, p. 209] for details)

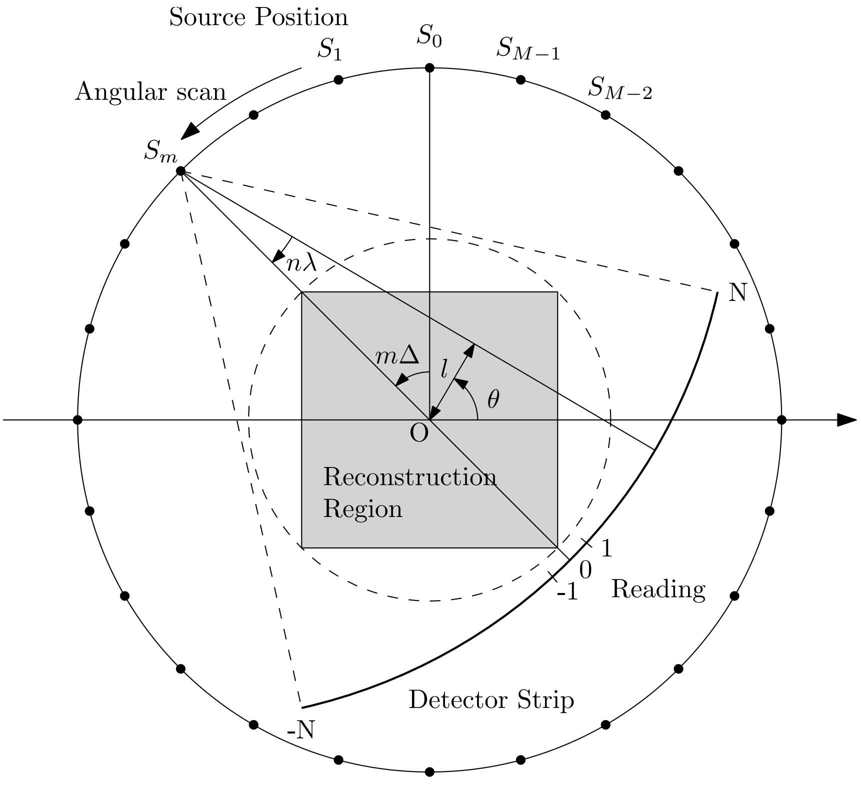


Figure 7.4.: Standard (divergent) projection geometry [6, p. 78]

even to a fixed ε. As proximity function the weighted squared distance has been defined. ε is determined by averaging the WSQD from the 30 runs of ART using blobs. Using a defined ε as stopping criterion will ensure, that the reconstruction of the four other reconstructions have a proximity value equal or slightly better than the reference algorithm, which makes the results comparable.

In analogy to the book [6], the relaxation parameter for the ART algorithms has been chosen as constant value $\lambda^k = 0.05$. ART uses efficient ordering. Our experiments showed that for SART a choice of a constant $\lambda^k = 1.8$ provides good results. Both algorithms use a non-negativity restriction.

The reconstruction region will be initialized for all algorithms with an estimate of the average density of the image based on the projection data (see [4] for further details). In the algorithms the initialization is referred to as $\overline{x}$.

The superiorized versions of the algorithms depend on the choice of the secondary optimization criterion ϕ, the positive integer N (which defines the number of superiorization steps performed before every execution of the iterative step), the choice of the summable sequence γ_l and the handling of l (see Section 5.6).

Some of the parameters depend on the implementation and are therefore predetermined by SNARK:

- the summable sequence used in SNARK is $\gamma_l = b \cdot a^l$ where $0 < a < 1$ and $0 < b \leq 1$.
- SNARK offers three ways of handling l which are explained ins Sections 5.5 and 5.6.

Algorithm	**handling of l**
superiorized	no reset of l (algorithm 2)
ATL1	reset to $l = k$ (algorithm 3)
ATL2	reset to $l_k = rand(k, l_{k-1})$

Table 7.1.: Possible handlings of l

The secondary optimization criterion ϕ for all algorithms is the total variation. The superiorization parameters for ART and SART have been designated in tests executed prior to the actual experiment. The used settings are:

Parameter	**ART**	**SART**
N	3	2
a	0.4	0.6
b	1.0	1.0

Table 7.2.: Superiorization parameter used in experiment

7.7.3 Figures of Merit

The algorithms will be evaluated using the following figures of merit

- number of iterations needed
- runtime
- image-wise region of interest
- total variation
- weighted squared distance.

IROI and TV are evaluated using task-oriented comparison of algorithm performance, the other FOMs are taken from one of the 30 executions of the experiment.

7.7.4 Statistical Hypothesis Testing

Statistical hypothesis testing as described in [6] is the calculation of the statistical significance that the null-hypothesis of equal performance can be rejected. Here, we employ one-sided tests. The null hypothesis is, that the expected average values (for a defined number of reconstructions) of the two algorithms are the same. The alternative hypothesis is, that the average FOM value of the algorithm, for which the experimentally observed average is higher, is in fact higher. Based on the results of the FOMs of the different reconstruction runs, the p-value can be calculated. The p-value is the probability of observing a difference between the reconstruction performances (according to the FOM) of the two algorithms that is as high or higher than the observed difference if the null hypothesis that the two algorithms are equally efficacious were true. The smallness of the p-value measures the significance, by which we can reject the null hypothesis in favor of the alternative.

In our case the figure of merit is the image-wise region of interest and the number of reconstruction runs is defined with 30. See [6], [7] and [13] for further details.

7.8 Evaluation

As explained in Section 7.7.2, the weighted squared distance reference value ε for stopping the reconstruction algorithms was taken as the average from 30 runs of ART with blobs reconstructions - each with 5 iterations. This resulted in a stopping value of $\varepsilon = 2.24496$. In order to ensure that every algorithm - even if it does not reach the defined stopping value - terminates, the maximum number of iterations was set to 200.

The resulting data will be evaluated in the following ways:

- ART with blobs against all other algorithms
- ART vs SART using the same superiorization settings
- ART/SART unsuperiorized vs superiorized
- ART/SART superiorized vs superiorized

All experiments have been performed on a system with the following configuration:

Parameter	**Value**
Linux Distribution	Red Hat Enterprise Linux Workstation release 6.5 (Santiago)
Architecture	x86_64
CPU model	Intel(R) Core(TM) i7-2600S CPU @ 2.80GHz
CPU op-mode	64-bit
CPU(s)	4 (8 Threads)
CPU MHz	2.80 GHz
Total memory	3.799.548 kB
Hard disk	Toshiba MK5061GSYN 500GB SATA 3GB/s 7200rpm

Table 7.3.: Execution System

7.9 Results

Table 7.4 shows the reconstruction results for all algorithms. It is obvious that the SART algorithms require a lot more iterations in order to reach ε than the ART algorithms. This is particular evident for SART with blobs. This algorithm was stopped after the maximum number of 200 iterations and not due to reaching ε. This can also be seen on the WSQD value of 2.6329 which is higher than the above defined stopping value of $\varepsilon = 2.24496$. SART with pixel basis functions needed approximately 17 times more iterations than ART with pixel.

Algorithm	**Iterations**	**Runtime**	**IROI**	**ϕ (TV)**	**WSQD**
Phantom	-	-	-	454.087	-
ART with blobs	6	42.512	0.18067	525.746	1.9952
ART unsuperiorized	6	14.905	0.15406	497.495	2.0413
ART superiorized	6	15.945	0.15532	486.841	2.0398
ART ATL1	6	15.267	0.15362	451.211	2.0373
ART ATL2	6	15.783	0.15615	467.743	2.0373
SART with blobs	200	1351.224	0.18446	432.626	2.6329
SART unsuperiorized	104	253.440	0.17542	468.527	2.1906
SART superiorized	104	243.992	0.17509	466.536	2.1909
SART ATL1	104	245.160	0.17449	462.711	2.1902
SART ATL2	104	247.003	0.17464	463.357	2.1903

Table 7.4.: Reconstruction Results for all Algorithms

As a result of the higher number of iterations also the runtime of the SART algorithms is higher than the one for ART. Table 7.5 shows the detailed comparison of the execution times. The duration of one SART iteration is slightly better than for ART for every algorithm type. The column "Relation" holds the value of how long one SART iteration needs compared to one ART iteration in percent (the duration of ART equals to 100%).

Type	**ART**			**SART**			**Relation**
	Iter	**Total**	**Per Iter**	**Iter**	**Total**	**Per Iter**	
with blobs	6	42.512	7.085	200	1351.224	6.75612	95.3 %
unsuperiorized	6	14.905	2.484	104	253.440	2.437	98.1 %
superiorized	6	15.945	2.658	104	243.992	2.346	88.3 %
ATL1	6	15.267	2.545	104	245.160	2.357	92.6 %
ATL2	6	15.783	2.631	104	247.003	2.375	90.1 %

Table 7.5.: Comparison of runtimes: ART vs SART

More interesting are the results for IROI and TV. All SART reconstructions result in better IROI values than their ART counterparts. If well-chosen parameters for superiorization are used, it improves the performance with regards to IROI and TV for ART, whereas SART is very resistant to improvements by superiorization with TV as secondary optimization criterion. If the influence of total variation optimization is too high, the IROI values deteriorate noticeably.

The influence of superiorization can be steered by modifying either of the parameters N, a or b. As described before, N is the number of superiorization steps performed in between every iterative step. If N is increased, more superiorization steps will be executed, and therefor the effect of superiorization is higher.

a and b are the parameters of the sequence $\gamma_l = b \cdot a^l$. As β_k is picked from γ_l, the length of the non-ascending vector $\beta_k v^k$ is controlled by γ_l. The closer a is to its supremum 1, the slower γ_l decreases and therefore also the length of the non-ascending vector $\left\|\beta_k v^k\right\|$ decreases slower. This increases the impact of each superiorization step. The influence of b has not been studied in this work.

The behavior described above has been verified in tests using all possible combinations of the following parameter values:

- $N = \{2, 3, 4, 6, 8, 10, 15\}$
- $a = \{0.400, 0.600, 0.800, 0.950\}$
- $b = 1$

As an example the results for parameters $N = 6$, $a = \{0.400, 0.600, 0.800, 0.950\}$ and $b = 1$ are shown in Figure 7.5. The left chart shows the IROI values relative to the result for ART with blobs, the right one shows the total variation relative to the value of the phantom. Higher values of a result in more optimization regarding to TV but

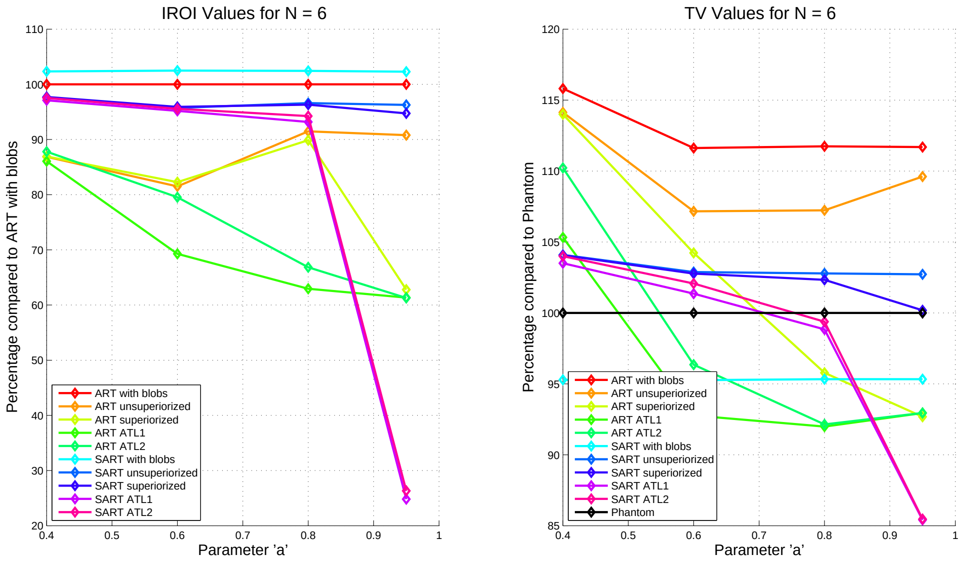


Figure 7.5.: Relative IROI and TV values for $N = 6$

also in a decrease of the IROI performance. This is especially evident for superiorized SART using ATL1 and ATL2. The fluctuations in the curves for the unsuperiorized algorithms arise from the fact that for each parameter combination new phantoms are generated. This changes the distribution of the randomly-generated tumors, and thus the projection data. Due to the changing input data, the reconstruction performance varies slightly as well.

In Figure 7.6 the reconstruction results for superiorized SART using ATL1 and $N = 6$ are displayed. The higher the parameter a is chosen, the higher is the impact of TV superiorization. A more dominant superiorization results in more uniform reconstructions images. This is especially noticeable in a very blurry image for $a = 0.950$.

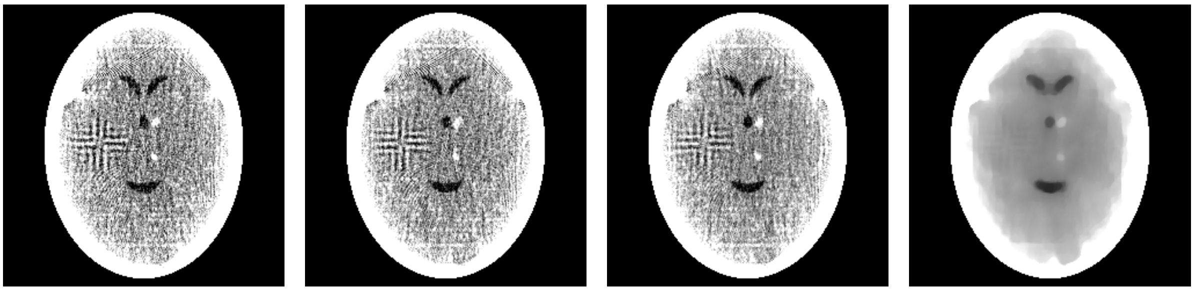

(a) $a = 0.400$ (b) $a = 0.600$ (c) $a = 0.800$ (d) $a = 0.950$

Figure 7.6.: Reconstruction results for superiorized SART using ATL1, $N = 6$

Figure 7.7 is a visualization of the reconstruction results shown in Table 7.4 using the parameters specified in Section 7.7.2. As stated above, superiorization slightly improves the IROI performance and TV is decreased noticeable. Although the improvement of IROI is only two percent, it may very well have a positive effect on medical diagnoses (as mentioned in Section 7.3, the image-wise region of interest correlates well with the perception of human observers). The assessment on how significant the improvement of the IROI with regards to medical diagnoses is, is not subject of this work.

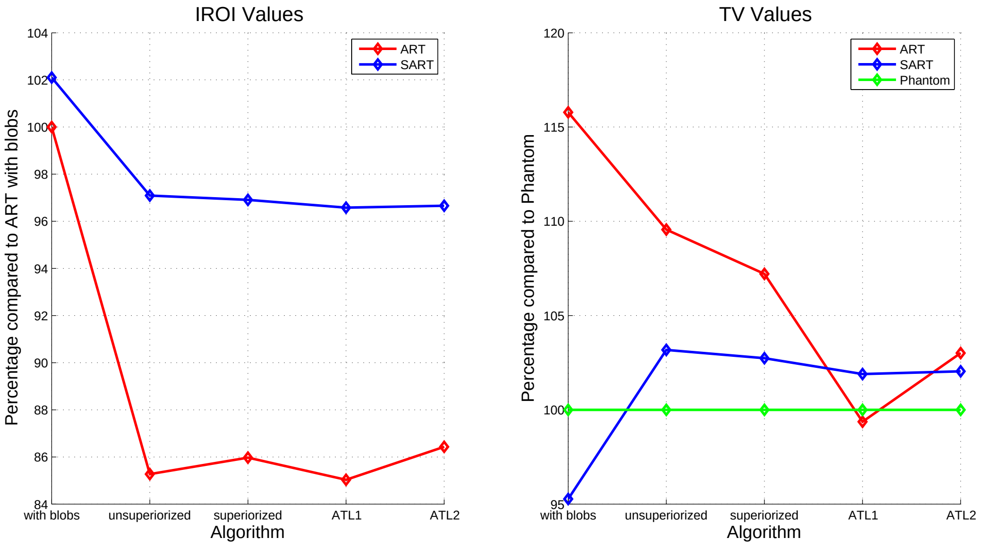
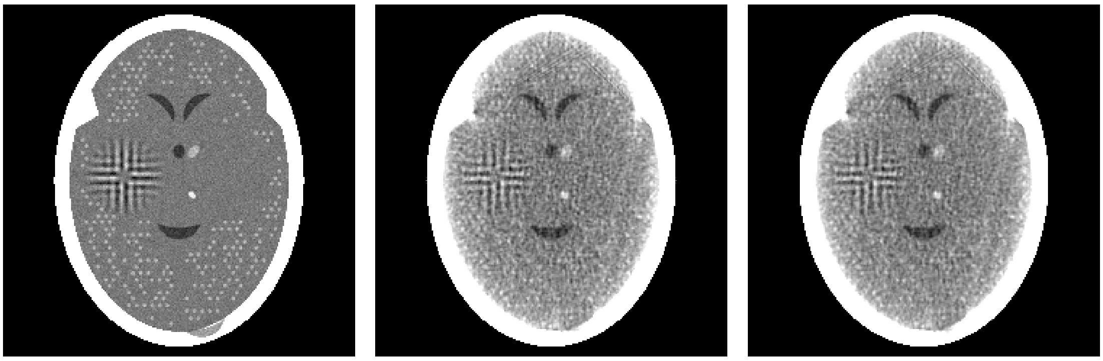


Figure 7.7.: Relative IROI and TV values

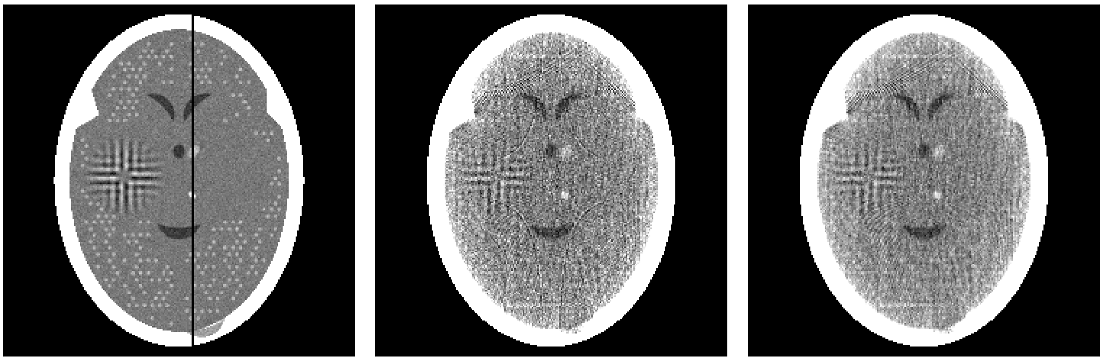
For a visual comparison of the reconstruction results, Figure 7.8 shows the reconstructed image of all algorithms after the last iteration. In order to clearly see the features in the interior of the skull, all values below 0.204 are represented as black and all values above 0.21675 are represented as white. The remaining values are contrast stretched on the interval $[0, 1]$.

Sub-image 7.8a shows the basis phantom with the randomly placed tumors and local inhomogeneities. Sub-image 7.8d shows the same phantom with the highlighted 131^{st} column which is used for creating all following line graphs. The pixel values in the line diagrams are the gray levels along this column. The pixel values represent the density values of the different tissues. All other sub-images are results of the reconstruction of the different reconstruction algorithms.

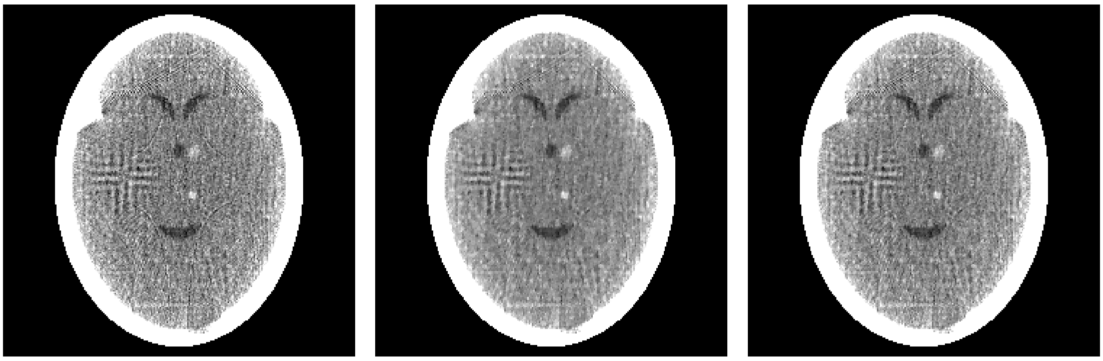
In general, the visual evaluation of the reconstructions is difficult. All reconstructions have in common that the tumors are difficult to detect. Also some introduced artifacts are visible. Clearly noticeable is the quality difference between the reconstructions



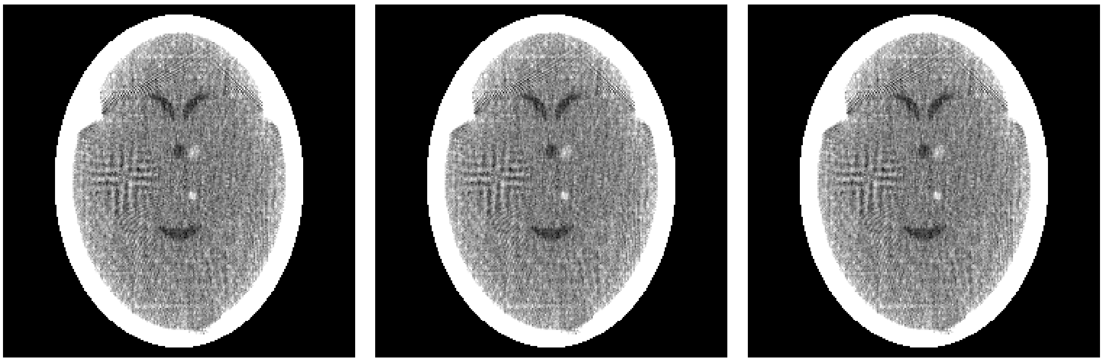
(a) Phantom (b) ART with blobs (c) SART with blobs



(d) Evaluation Column 131 (e) ART unsuperiorized (f) SART unsuperiorized



(g) ART superiorized (h) ART - ATL1 (i) ART - ATL2



(j) SART superiorized (k) SART - ATL1 (l) SART - ATL2

Figure 7.8.: Reconstruction results for all algorithms

using blobs and those using pixel basis functions. The reconstructions using blobs contain fewer artifacts and the tumors are more visible. For reconstructions using pixel basis functions, the superiorized versions are much smoother. This can be seen particularly when comparing the unsuperiorized version of ART (sub-image 7.8e) with the superiorized ART - ATL1 reconstruction (sub-image 7.8h).

7.9.1 ART with BLOB against all other algorithms

Table 7.6 shows the results of the algorithm comparison of ART with blobs against all other algorithms. As already demonstrated in the experiments in [6] the performance of ART with blobs is promising. Only the reconstruction using SART with blobs shows statistically better result. For all algorithms using pixel basis functions the results are worse. The values of this comparison are visualized in Figure 7.7.

Algorithm 1			**Algorithm 2**			**Significance**
Name	**IROI**	**ϕ (TV)**	**Name**	**IROI**	**ϕ (TV)**	
ART blobs	0.18067	525.746	SART blobs	0.18446	432.626	1.577e-06
ART blobs	0.18067	525.746	ART	0.15406	497.495	1.021e-05
ART blobs	0.18067	525.746	ART sup.	0.15532	486.841	1.802e-05
ART blobs	0.18067	525.746	ART ATL1	0.15362	451.211	7.569e-06
ART blobs	0.18067	525.746	ART ATL2	0.15615	467.743	2.650e-05
ART blobs	0.18067	525.746	SART	0.17542	468.527	0.00014
ART blobs	0.18067	525.746	SART sup.	0.17509	466.536	8.709e-05
ART blobs	0.18067	525.746	SART ATL1	0.17449	462.711	3.343e-05
ART blobs	0.18067	525.746	SART ATL2	0.17464	463.357	4.121e-05

Table 7.6.: Comparison of ART with blobs against all other algorithms

The line diagram of column 131 (see sub-image 7.8d for the location of column 131) in Figure 7.9 compares the gray levels of the phantom with ART and SART using blobs. The small spikes at lines 27, 45, 176, 195 and 217 of the phantom curve are tumor sites. Both algorithms also show peaks at these sites which indicates that the tumors can be recognized in the reconstructed images. The deviations at the boarders of the phantom result from beam hardening (see Section 2.1.2). Although SNARK supports beam hardening correction, the effect can never be completely eliminated.

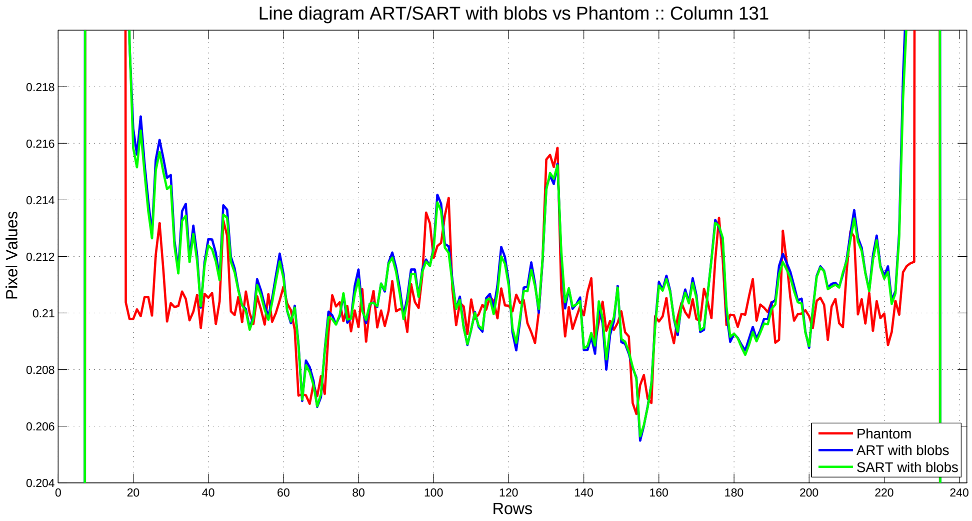


Figure 7.9.: Line Diagram: Phantom vs ART with blobs vs SART with blobs

7.9.2 ART vs SART

As Table 7.7 states, all SART algorithms have a better performance than their ART counterparts. The difference of the IROI value is larger for algorithms which uses pixel basis functions. Figure 7.10 shows the line diagram of unsuperiorized ART and SART. Generally, the reconstructions are noisier than with blobs. The peaks at the tumor spots (rows 27, 45, 176, 195 and 217) are surrounded with other peaks which make a (visual) detection more difficult.

Algorithm 1			**Algorithm 2**			**Significance**
Name	**IROI**	**ϕ (TV)**	**Name**	**IROI**	**ϕ (TV)**	
ART blobs	0.18067	525.746	SART blobs	0.18446	432.626	1.577e-06
ART	0.15406	497.495	SART	0.17542	468.527	2.698e-05
ART sup.	0.15532	486.841	SART sup.	0.17509	466.536	6.442e-05
ART ATL1	0.15362	451.211	SART ATL1	0.17449	462.711	3.177e-05
ART ATL2	0.15615	467.743	SART ATL2	0.17464	463.357	0.00014

Table 7.7.: Comparison of ART vs SART

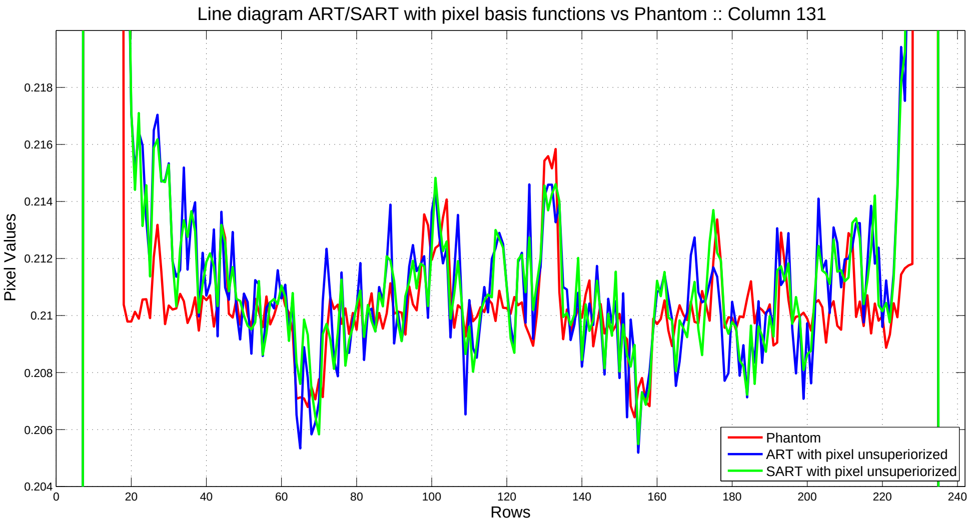


Figure 7.10.: Line Diagram: Phantom vs unsuperiorized ART/SART with pixel basis functions

7.9.3 Superiorized vs. Unsuperiorized Algorithms

When comparing the superiorized and unsuperiorized versions of the same algorithm, the values are much closer than in the previous comparisons. Table 7.8 holds the results of the comparison. Also the difference in the line diagram 7.11 is smaller. Due to the TV optimization, the superiorized reconstruction is smoother but the overall result is very similar.

Algorithm 1			**Algorithm 2**			**Significance**
Name	**IROI**	ϕ **(TV)**	**Name**	**IROI**	ϕ **(TV)**	
ART	0.15406	497.495	ART sup.	0.15532	486.841	8.872e-06
ART	0.15406	497.495	ART ATL1	0.15362	451.211	0.16163
ART	0.15406	497.495	ART ATL2	0.15615	467.743	1.728e-05
SART	0.17542	468.527	SART sup.	0.17509	466.536	0.02714
SART	0.17542	468.527	SART ATL1	0.17449	462.711	3.941e-05
SART	0.17542	468.527	SART ATL2	0.17464	463.357	0.00023

Table 7.8.: Comparison of unsuperiorized vs superiorized Algorithms

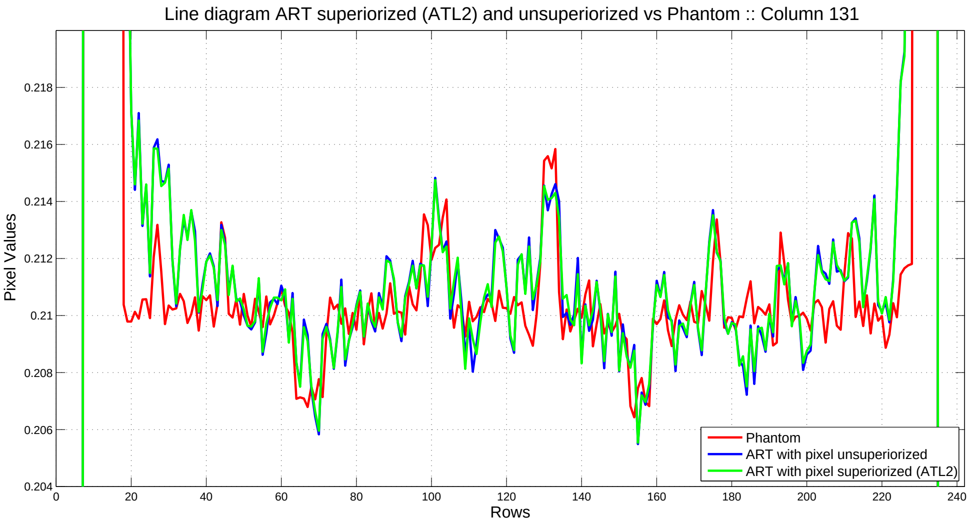
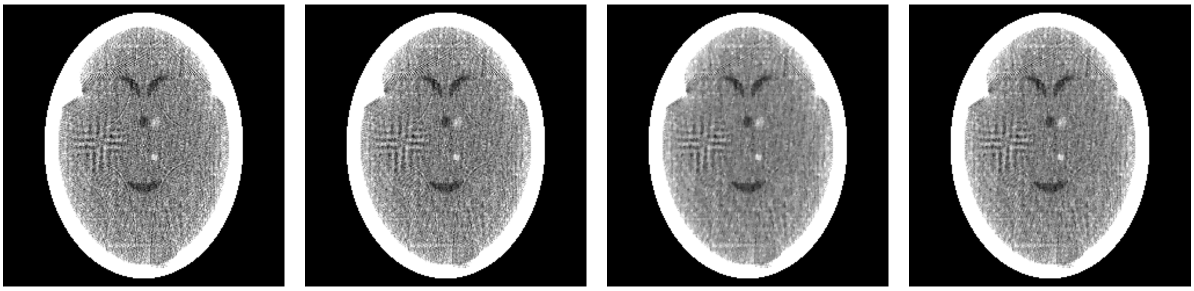
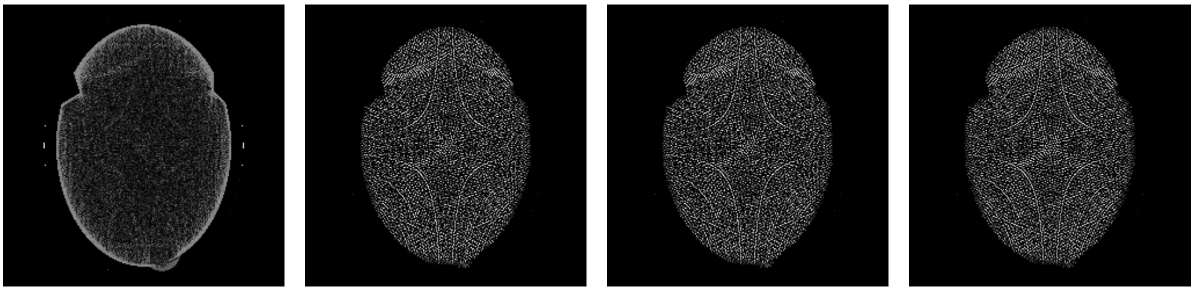


Figure 7.11.: Line Diagram: Phantom vs superiorized and unsuperiorized ART

Figure 7.12 and 7.13 show the difference of the unsuperiorized reconstruction to the three superiorized versions of ART and SART respectively. Since the reconstructions are very similar, the differences between the difference images are visually almost unrecognizable. The sub-images (a)-(d) show the reconstruction results, sub-image (e) shows the difference of the unsuperiorized version and the phantom and the sub-images (f)-(h) show the difference of the superiorized algorithms to the unsuperiorized version.

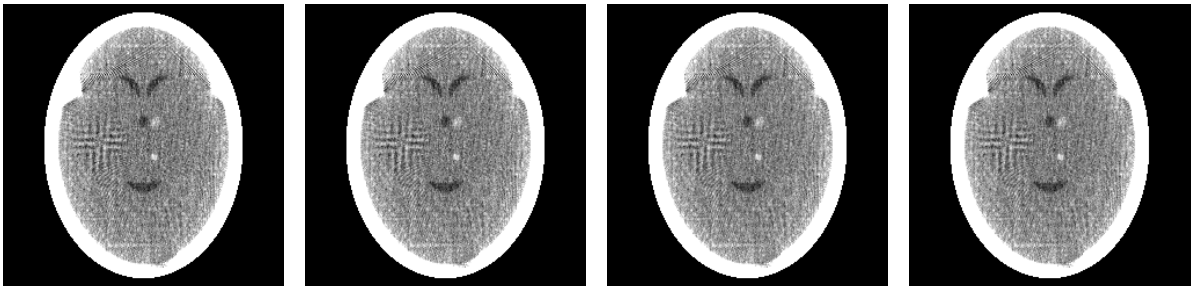


(a) ART unsuperiorized (b) ART superiorized (c) ART - ATL1 (d) ART - ATL2

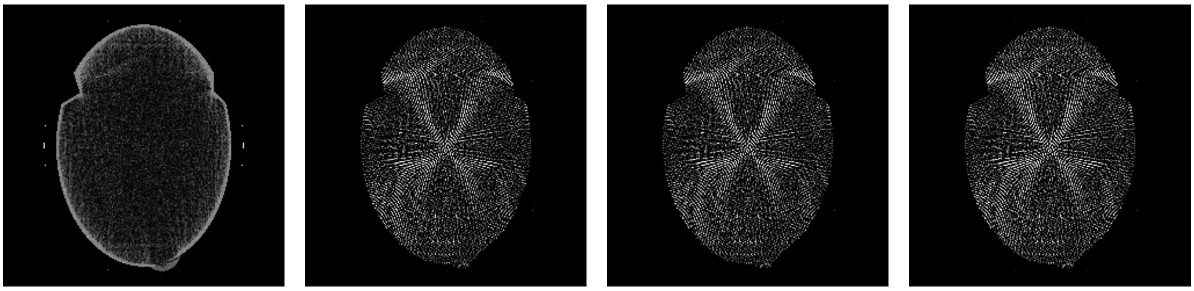


(e) Difference of ART unsuperiorized to the phantom (f) Difference of ART superiorized to ART unsuperiorized (g) Difference of ART - ATL1 to ART unsuperiorized (h) Difference of ART - ATL2 to ART unsuperiorized

Figure 7.12.: Reconstruction results ART with pixels



(a) SART unsuperiorized (b) SART superiorized (c) SART - ATL1 (d) SART - ATL2



(e) Difference of SART unsuperiorized to the phantom (f) Difference of SART superiorized to SART unsuperiorized (g) Difference of SART - ATL1 to SART unsuperiorized (h) Difference of SART - ATL2 to SART unsuperiorized

Figure 7.13.: Reconstruction results SART with pixels

7.9.4 Superiorized vs. Superiorized Algorithms

Finally Table 7.9 shows the comparison of the different superiorization algorithms. They only differ in the handling of l (see Section 5.6 for further details). It is not clearly obvious which version of the l handling is the best. In case of SART there is no big difference between all three versions, whereas for ART, ATL2 seems to deliver the best results. Please note that the statement just made is not statistically proven and only reflects a trend that was observed during the executed experiments. Therefore this statement is only valid for this particular experiment with the specific parameters used.

Algorithm 1			**Algorithm 2**			**Significance**
Name	**IROI**	**ϕ (TV)**	**Name**	**IROI**	**ϕ (TV)**	
ART sup.	0.15532	486.841	ART ATL1	0.15362	451.211	0.00106
ART sup.	0.15532	486.841	ART ATL2	0.15615	467.743	0.00067
ART ATL1	0.15362	451.211	ART ATL2	0.15615	467.743	2.119e-05
SART sup.	0.17509	466.536	SART ATL1	0.17449	462.711	4.738e-05
SART sup.	0.17509	466.536	SART ATL2	0.17464	463.357	0.00062
SART ATL1	0.17449	462.711	SART ATL2	0.17464	463.357	0.01186

Table 7.9.: Comparison of superiorized vs superiorized Algorithms

8 Conclusion

Generally speaking, the experiments performed for this thesis showed that the reconstruction performance (with respect to the image-wise region of interest figure of merit) of SART is better than the one for ART. The drawbacks of SART are more iterations and longer execution times (factor approximately 17).

The impact of superiorization was different for the two algorithms. For ART, superiorization had a positive effect. Not only that the TV could be decreased, also the reconstruction quality with respect to the IROI was increased. SART did not show that behavior. There superiorization had only a minor impact.

The main aim of the comparison in this thesis was not the minimization of the total variation of the reconstructed image, but a good performance with respect to the image-wise region of interest. For other parameters than the ones chosen in Table 7.2, the influence of superiorization could have easily been increased, but this resulted in a worse performance with respect to the image-wise region of interest (see Figure 7.6). This shows that the choice of the superiorization parameter is crucial for the quality the reconstruction results. Incorrect selected parameters can lead to poor results.

It is important to state that there is no general rule on how the parameters should be selected. The choice depends on the particular problem. The results of the algorithm comparison demonstrated that an overly intense optimization regarding the secondary optimization criterion can lead to worse results for other FOMs.

It can be discussed whether the choice of TV was a good decision for this particular experiment. An optimization with respect to total variation results in smoothing of the reconstructed image. Thus, the recognition of the small and already difficult to detect tumors gets even more difficult, and as a result of this the IROI performance drops.

An open task for further work is the evaluation of the different handlings of l. l controls the selection of $\beta_{k,n}$ which itself controls the length of the non-ascending vector $v^{k,n}$ and therefore the impact of every superiorization step.

List of Figures

2.1 Typical CT image 2

2.2 CT data acquisition 3

2.3 Projection and backprojection of CT images 3

2.4 CT data collection 4

3.1 Definition of an Image 10

3.2 Relationship (part I) between (r, ϕ) and (l, θ) space 11

3.3 Relationship (part II) between (r, ϕ) and (l, θ) space 12

3.4 Equivalents of the Radon transform 13

3.5 Location of points in the (l, θ) space 14

3.6 Numbering of the components of the image vector 18

4.1 Kaczmarz method of solving algebraic equations 22

4.2 Update step for ART 24

4.3 Impact of the relaxation parameter 26

4.4 Update step for SART 27

7.1 A head phantom and its density distribution at a specified column CT images 41

7.2 Test images for a task-oriented comparison of algorithm performance using IROI 43

7.3 Selection of Pixel for TV evaluation 45

7.4 Standard (divergent) projection geometry 48

7.5 Relative IROI and TV values for $N = 6$ 53

7.6 Reconstruction results for superiorized SART using ATL1, $N = 6$ . . . 53

7.7 Relative IROI and TV values 54

7.8 Reconstruction results for all algorithms 55

7.9 Line Diagram: Phantom vs ART with blobs vs SART with blobs 57

7.10 Line Diagram: Phantom vs unsuperiorized ART/SART with pixel basis functions 58
7.11 Line Diagram: Phantom vs superiorized and unsuperiorized ART . . . 59
7.12 Reconstruction results ART with pixels 60
7.13 Reconstruction results SART with pixels 60

List of Tables

4.1 Influence of the relaxation parameter $\lambda^{(k)}$ 25

7.1 Possible handlings of l 49
7.2 Superiorization parameter used in experiment 49
7.3 Execution System 51
7.4 Reconstruction Results for all Algorithms 51
7.5 Comparison of runtimes: ART vs SART 52
7.6 Comparison of ART with blobs against all other algorithms 56
7.7 Comparison of ART vs SART 57
7.8 Comparison of unsuperiorized vs superiorized Algorithms 58
7.9 Comparison of superiorized vs superiorized Algorithms 61

Acronyms

AISP	Applied Image and Signal Processing.
ART	algebraic reconstruction technique.
ATL1	alternative handling of l ($l_k = k$).
ATL2	alternative handling of l ($l_k = rand(k, l_{k-1})$).
blobs	generalized Kaiser-Bessel window functions.
CT	computerized tomography.
CUNY	The City University of New York.
FOM	figure of merit.
HU	Hounsfield unit.
IROI	image-wise region of interest.
pixel	picture element.
SART	simultaneous algebraic reconstruction technique.
SHT	statistical hypothesis testing.
SUAS	Salzburg University of Applied Sciences.
TV	total variation.
voxel	volume element.
WSQD	weighted squared distance.

Bibliography

[1] Andersen, A. H. and A. C. Kak: *Simultaneous Algebraic Reconstruction Technique (SART) - A Superior Implementation Of The ART Algorithm*. Ultrasonic Imaging, Vol. 6:81–94, 1984.

[2] Bates, R.H.T., Kathryn L. Garden, and Terrence M. Peters: *Overview of Computerized Tomography with Emphasis on Future Developments*. Proceedings of the IEEE, Vol. 71, No. 3:356–372, 1983.

[3] Censor, Yair, Ran Davidi, and Gabor T. Herman: *Perturbation Resilience and Superiorization of Iterative Algorithms*. Inverse Problems, Vol 26, 2010. http://arxiv.org/abs/1005.0069.

[4] Davidi, Ran, Gabor T. Herman, and Joanna Klukowska: *SNARK09: A Programming System for the Reconstruction of 2D IImage from 1D projections*. Discrete Imaging and Graphics Group - The City University of New York, November 2012. http://www.dig.cs.gc.cuny.edu/software/snark09/SNARK09.pdf, User Manual.

[5] Gordon, Richard, Robert Bender, and Gabor T. Herman: *Algebraic Reconstruction Techniques (ART) for Three-dimensional Electron Microscopy and X-ray Photography*. Journal of Theoretical Biology, Vol. 29:471–481, 1970.

[6] Herman, Gabor T.: *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Springer, 2009. http://link.springer.com/book/10.1007%2F978-1-84628-723-7.

[7] Herman, Gabor T., Edgar Garduno, Ran Davidi, and Yair Censor: *Superiorization: An optimization heuristic for medical physics*. Medical Physics, Vol. 39:5532–5546, 2012.

[8] Jiang, Ming and Ge Wang: *Convergence of the Simultaneous Algebraic Reconstruction Technique (SART)*. IEEE Transactions on Image Processing, Vol. 12, No. 8:957–961, 2003.

[9] Kak, Avinash C. and Malcolm Slaney: *Principles of Computerized Tomographic Imaging*. IEEE Press, New York, 1999. https://engineering.purdue.edu/~malcolm/pct/.

[10] Klukowska, Joanna, Ran Davidi, and Gabor T. Herman: *SNARK09: A software package for reconstruction of 2D images from 1D projections.* Computer Methods and Programs in Biomedicine, Vol. 110 Nr. 3:424–440, 2011.

[11] Langthaler, Oliver: *Incorporation of the Superiorization Methodology into Biomedical Imaging Software.* Master's thesis, Fachhochschule Salzburg, 2014.

[12] Marabini, Roberto, Gabor T. Herman, and José M. Carazo: *3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs).* Ultramicroscopy, Vol. 72:53–65, 1998.

[13] Nikazad, Touraj, Ran Davidi, and Gabor T. Herman: *Accelerated perturbation-resilient block-iterative projection methods with application to image reconstruction.* Inverse Problems, Vol. 28, 2012. `http://iopscience.iop.org/0266-5611/28/3/035005/`.

[14] Radon, Johann: *Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten.* Berichte über die Verhandlungen der Königlich-Sächsischen Gesellschaft der Wissenschaften zu Leipzig. Mathematisch-Physische Klasse., Band 69:262–277, 1917. `people.csail.mit.edu/bkph/courses/papers/Exact_Conebeam/Radon_Deutsch_1917.pdf`.

[15] Radon, Johann: *On the determination of functions from their integral values along certain manifolds.* IEEE Transactions on Medical Imaging, Volume: 5 Issue: 4:170–176, 1986. `http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=4307775`.

[16] Rudin, Leonid I., Stanley Osher, and Emad Fatemi: *Nonlinear total variation based noise removal algorithms.* Physica D, Vol. 60:259–268, 1992. `http://www.math-info.univ-paris5.fr/~lomn/Cours/ECE/PhysicaRudinOsher.pdf`.

[17] Schrapp, Michael J. and Gabor T. Herman: *Data fusion in X-ray computed tomography using a superiorization approach.* Review of Scientific Instruments, Vol. 85:10, 2014. `http://scitation.aip.org/content/aip/journal/rsi/85/5/10.1063/1.4872378`.

[18] Smith, Steven W.: *The Scientist and Engineer's Guide to Digital Signal Processing.* California Technical Publishing, San Diago, 1997.

[19] Tanabe, Kunio: *Projection Method for Solving a Singular System of Linear Equations and its Applications.* Numerische Mathematik, Vol. 17:203–214, 1971.

http://www.digizeitschriften.de/dms/img/?PPN=GDZPPN001169653.

A SNARK Input Files

A.1 experiment_ART_vs_SART.in

```
ENSEMBLE        brain.ens
EXPERIMENT      1 0 243 0.0752 11 1 30
DATA            ART_vs_SART_projection.ss
RECONSTRUCTION  ART_vs_SART_recon.ss
ANALYSIS        ART_vs_SART_compare.ss
END
```

A.2 ART_vs_SART_projection.ss

```
RAYSUM AVERAGE 11
1 1 1 1 1 1 1 1 1 1 1
*
GEOMETRY
divergent arc 78 110.735
RAYS user 345 detector spacing 0.10668
ANGLES 720 EQUAL SPACING
0.0  359.5
MEASUREMENT NOISY
QUANTUM 1000000 720 CALIBRATION 2
SCATTER 0.012 0.4445
SEED
BACKGROUND 0.0
RUN
*
PICTURE TEST
*
PROJECTION REAL   BEAM HARDENING CORRECTION
NITERS = 2 polynomial degree = 1
polynomial coefficients   0.0  1.028
nergy = 5
number of points = 2
*41KeV
0.210 0.265
0.416 0.999
*52KeV
0.210 0.226
```

```
0.416 0.595
*84KeV
0.210 0.183
0.416 0.265
*100KeV
0.210 0.174
0.416 0.208
```

A.3 ART_vs_SART_recon.ss

```
STOP TERMINATION WSQD 2.24496 RPRT



BASIS BLOBS

EXECUTE AVERAGE ART
artb: ART 0.05 efficient blobs
ART3 relaxation constant 0.05
constraint bound

EXECUTE AVERAGE SART
sarb: SART 1.80 blobs
relaxation constant 1.8



BASIS PIXEL
MODE LOWER 0


EXECUTE AVERAGE ART
art1: nonnegative ART 0.05 pixel
ART3 relaxation constant 0.05
constraint bound

EXECUTE AVERAGE SART
sar1: SART 1.80 pixel
relaxation constant 1.8


SUPERIORIZE 3 0.4 1 TVAR RPRT
```

```
EXECUTE AVERAGE ART
art2: TV STD nonnegative ART 0.05 pixel
ART3 relaxation constant 0.05
constraint bound

SUPERIORIZE 2 0.6 1 TVAR RPRT

EXECUTE AVERAGE SART
sar2: TV STD SART 1.80 pixel
relaxation constant 1.8


SUPERIORIZE 3 0.4 1 TVAR ATL1 RPRT

EXECUTE AVERAGE ART
art3: TV ATL1 nonnegative ART 0.05 pixel
ART3 relaxation constant 0.05
constraint bound

SUPERIORIZE 2 0.6 1 TVAR ATL1 RPRT

EXECUTE AVERAGE SART
sar3: TV ATL1 SART 1.80 pixel
relaxation constant 1.8


SUPERIORIZE 3 0.4 1 TVAR ATL2 RPRT

EXECUTE AVERAGE ART
art4: TV ATL2 nonnegative ART 0.05 pixel
ART3 relaxation constant 0.05
constraint bound

SUPERIORIZE 2 0.6 1 TVAR ATL2 RPRT

EXECUTE AVERAGE SART
sar4: TV ATL2 SART 1.80 pixel
relaxation constant 1.8



MODE
```

A.4 ART_vs_SART_compare.ss

```
results_ART_vs_SART

COMPARE artb  artb   IROI   WSQD
0     0

COMPARE artb  art1   IROI   WSQD
0     0
COMPARE artb  art2   IROI   WSQD
0     0
COMPARE artb  art3   IROI   WSQD
0     0
COMPARE artb  art4   IROI   WSQD
0     0
COMPARE artb  sar1   IROI   WSQD
0     0
COMPARE artb  sar2   IROI   WSQD
0     0
COMPARE artb  sar3   IROI   WSQD
0     0
COMPARE artb  sar4   IROI   WSQD
0     0

COMPARE art1  sar1   IROI   WSQD
0     0
COMPARE art2  sar2   IROI   WSQD
0     0
COMPARE art3  sar3   IROI   WSQD
0     0
COMPARE art4  sar4   IROI   WSQD
0     0

COMPARE art1  art1   IROI   WSQD
0     0
COMPARE art1  art2   IROI   WSQD
0     0
COMPARE art1  art3   IROI   WSQD
0     0
COMPARE art1  art4   IROI   WSQD
0     0

COMPARE art2  art2   IROI   WSQD
0     0
COMPARE art2  art3   IROI   WSQD
0     0
```

```
COMPARE art2  art4   IROI   WSQD
0      0

COMPARE art3  art4   IROI   WSQD
0      0

COMPARE sar1  sar1   IROI   WSQD
0      0
COMPARE sar1  sar2   IROI   WSQD
0      0
COMPARE sar1  sar3   IROI   WSQD
0      0
COMPARE sar1  sar4   IROI   WSQD
0      0

COMPARE sar2  sar2   IROI   WSQD
0      0
COMPARE sar2  sar3   IROI   WSQD
0      0
COMPARE sar2  sar4   IROI   WSQD
0      0

COMPARE sar3  sar4   IROI   WSQD
0      0

COMPARE artb  sarb   IROI   WSQD
0      0

END
```

B Mathematical Details

B.1 Proof of Theorem 2

The following proof has been taken from [7, p. 261ff].

It is assumed that the picture function f is continuous and bounded and that $f(r,\phi)=0$ if $r>E$.

For any point (r,ϕ) in the picture region (in particular, $|r|\leq E$), the function $\overline{F}_{(r,\phi)}$ of one variable is defined as

$$\overline{F}_{(r,\phi)}(q)=\frac{1}{2\pi}\int_0^{2\pi}\mathscr{R}f\left(r\cos\left(\theta-\phi\right)+q,\theta\right)\,d\theta \tag{B.1}$$

Since $\mathscr{R}f(l,\theta)=0$, if $|l|\geq E$, it is ensured that, for $|r|\leq E$,

$$\overline{F}_{(r,\phi)}(q)=0,\ \text{if } q\geq 2E. \tag{B.2}$$

Radon proved in [14] that

$$f(r,\phi)=\frac{1}{\pi}\lim_{\varepsilon\to 0}\left(\frac{1}{\varepsilon}\overline{F}_{(r,\phi)}(\varepsilon)-\int_\varepsilon^\infty\frac{1}{q^2}\overline{F}_{(r,\phi)}(q)\,dq\right). \tag{B.3}$$

Proof of Theorem 2. The next steps show that (2.7) can be derived from (B.3). For this, the additional assumption that $\mathscr{R}f$ has a continuous first derivate; i.e., that $\mathscr{D}_Y\mathscr{R}f$ exists and is continuous in its first variable. $\mathscr{D}_Y$ has been defined in (3.11).

$$\begin{aligned}\int_\varepsilon^\infty\frac{1}{q^2}\overline{F}_{(r,\phi)}(q)\,dq&=\int_\varepsilon^{2E}\frac{1}{q^2}\left[\frac{1}{2\pi}\int_0^{2\pi}\mathscr{R}f\left(r\cos\left(\theta-\phi\right)+q,\theta\right)\,d\theta\right]\,dq\\&=\frac{1}{2\pi}\int_0^{2\pi}\left[\int_\varepsilon^{2E}\frac{1}{q^2}\mathscr{R}f\left(r\cos\left(\theta-\phi\right)+q,\theta\right)\,dq\right]\,d\theta\end{aligned} \tag{B.4}$$

Using integration by parts, we get

$$\begin{aligned}\mathscr{R}f\left(r\cos\left(\theta-\phi\right)+q,\theta\right)\,dq=&\left[-\frac{1}{q}\mathscr{R}f\left(r\cos\left(\theta-\phi\right)+q,\theta\right)\right]_{q=\varepsilon}^{q=2E}\\&-\int_\varepsilon^{2E}-\frac{1}{q}\,\mathscr{D}_Y\mathscr{R}f\left(r\cos\left(\theta-\phi\right)+q,\theta\right)dq.\end{aligned} \tag{B.5}$$

Substituting this into (B.4) results in

$$\begin{aligned}\int_{\varepsilon}^{\infty} \frac{1}{q^2}\overline{F}_{(r,\phi)}(q)\,dq =& \frac{1}{\varepsilon}\overline{F}_{(r,\phi)}(\varepsilon) \\ &+ \frac{1}{2\pi}\int_0^{2\pi}\left[\int_{\varepsilon}^{2E} -\frac{1}{q}\,\mathscr{D}_Y\mathscr{R}f\left(r\cos\left(\theta-\phi\right)+q,\theta\right)\,dq\right]d\theta\end{aligned} \tag{B.6}$$

Changing the order of integration in the second term and substituting into (B.3) results in

$$f(r,\phi) = -\frac{1}{2\pi^2}\lim_{\varepsilon\to 0}\int_{\varepsilon}^{\infty}\frac{1}{q}\int_0^{2\pi}\mathscr{D}_Y\mathscr{R}f\left(r\cos\left(\theta-\phi\right)+q,\theta\right)\,d\theta\,dq\,, \tag{B.7}$$

which is (2.7). ■