Towards a Variational Approach to Regularized Tomographic Reconstruction

Master's Thesis in Computational Engineering

submitted
by

Stefan Schneider

born April 15, 1979 in Siegen.

Written at

Lehrstuhl für Mustererkennung (Informatik 5)
and
Lehrstuhl für Systemsimulation (Informatik 10),
Institut für Informatik,
Friedrich-Alexander-Universität Erlangen-Nürnberg.

Advisors:
Dipl. Inf. Marcus Prümmer (IMMD5), Dipl. Inf. Harald Köstler (IMMD10), Prof. J. Hornegger (IMMD5), Prof. U. Rüde (IMMD10) and Dr. med. M. Lell (Radiologie)

Started: November 21, 2005

Finished: May 19, 2006

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien des Lehrstuhls für Examensarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts in Punkt 2.4.

Erlangen, den 19. Mai 2006

Übersicht

Die klassische Methode von Kaczmarz, die die Grundlage für viele algebraische Rekonstruktionstechniken (ART) bildet, ist im Bereich der Bildrekonstruktion weit verbreitet. Allerdings liefert der Algorithmus nur im Fall von konsistenten Daten zufriedenstellende Resultate. Bei inkonsistenten Daten, wie es in der Praxis der Fall ist, hat der Algorithmus einige Konvergenzprobleme.

In der vorliegenden Masterarbeit formulierten wir das Rekonstruktionsproblem im Rahmen der Variationsrechnung. Neben dem Rekonstruktionsanteil führten wir einen Regularisierungsteil ein, der die allgemeine Glattheit im Bild verstärkt. Wir untersuchten fünf verschiedene Regularisierungsterme; zunächst den linearen Regularisierer den wir Quadratische Funktion nennen, dann zwei konvexe Regularisierer, zum einen die Hypersurface Minimal und zum anderen die Green Funktion, und abschließend zwei nicht konvexe Terme, die wir Lorentzian und Tukey Funktion nennen. Bei korrekter Wahl der Parameter verstärken die vier nicht linearen Regularisierer die Glattheit innerhalb homogener Bereiche, während Kanten erhalten bleiben.

Ausgehend vom Variationsansatz entwickelten wir drei Erweiterungen des klassischen Algorithmus' von Kaczmarz; zunächst den regularisierten, erweiterten Kaczmarz Algorithmus mit Relaxationsparameter (RKERP), als Zweites eine vereinfachte Version (SRKERP), und schließlich den neuen regularisierten, algebraischen Rekonstruktionstechnik Algorithmus (RART).

Untersucht wurden die Algorithmen für die Rekonstruktion von dreidimensionalen (3D), medizinischen Bilddaten aus nur wenigen zweidimensionalen (2D) Röntgenbildern. Unsere Experimente zeigten, dass nur der SRKERP Algorithmus zufriedenstellende Resultate liefert.

Abstract

The classical Kaczmarz's method, which is the basis for many Algebraic Reconstruction Techniques (ART), is very popular in the field of image reconstruction. However, this algorithm gives only satisfactory results for consistent data. For inconsistent data, which is the case in practice, the algorithm has some problems with respect to convergence.

In this master's thesis, we formulated the reconstruction problem in the framework of variational calculus. Additional to the reconstruction part, we introduced a regularization term which enforces the overall smoothness in the image. We examined five different regularization terms; first of all, the linear regularizer which we call Quadratic function, then two convex regularizers which we call Hypersurface minimal and Green function, and finally, two non-convex terms which we denominate Lorentzian and Tukey function. If the right parameters are chosen the four non-linear regularizers intensify smoothness within homogenous areas while edges are preserved.

Out of the variational approach, we developed three extensions of the Kaczmarz's algorithm; first of all, the Regularized Kaczmarz Extended with Relaxation Parameters algorithm (RKERP), secondly, a simplified version of it (SRKERP), and finally, the new Regularized Algebraic Reconstruction Technique algorithm (RART).

The algorithms were examined for the reconstruction of three-dimensional (3-D) medical image data out of only a few two-dimensional (2-D) X-ray images. Our experiments showed that only the SRKERP algorithm yields satisfactory results.

Contents

Chapter 1

Introduction

In the last decade 3D image reconstruction became very important in medical imaging. In contrary to the 2D reconstruction which is used e.g. in traditional computed tomography (CT) and where the volume is reconstructed slice by slice[1], the whole three-dimensional volume is computed out of two-dimensional projection images. This means that the projections are nothing else than normal X-ray images which can also be read by the physician.

One important example for 3D reconstruction is surgery supported with mobile C-Arm devices. This technique is frequently used in bone-surgery of the spine or extremities. During the surgery the device is positioned around the patient, images are taken and reconstructed and finally the device is removed. So it is possible to keep an eye on changes of bone or tissue during the operation. In practice the acquisition of 100 images with a size of 1024×1024 pixels and the reconstruction of the volume with a size of $256 \times 256 \times 256$ voxels takes approximately one minute. These devices are supplied with electric current out of a normal socket which means that they can be used in nearly every operating room. Figure 1.1 shows a C-Arm with these capabilities.

Another example is the reconstruction of 4D heart images.[2] One has to synchronize the image acquisition with the different states of the heart or else the result would be very blurred. As a consequence one has only a few images of a specific state of the heart available for image reconstruction. Naturally the more projections are available the better reconstruction results can be achieved.

Another reason for acquiring as few images as possible is the radioactive contamination of

[1] In this master's thesis we use the term "2D reconstruction" for the reconstruction of a two-dimensional slice out of one-dimensional projections. The term "3D reconstruction" is used for computing a three-dimensional volume out of two-dimensional projection images.

[2] Here the fourth dimension is time.

Figure 1.1: *ARCADIS Orbic 3D* from SIEMENS Medical Solutions.

the patient. Additionally the dose is reduced which leads to noisier projection images. A way to compensate the resulting artifacts could be the introduction of a-priori knowledge into the reconstruction algorithm. This leads to regularized reconstruction and is the topic of this master's thesis.

We formulate the mathematical problem in a variational approach and show that the discretization of the solution leads to the well-known *algebraic reconstruction technique* (ART). Instead of using the *filtered back-projection* (FBP) which is used commonly in practice we extend the algebraic technique by a regularization term. The main drawback is that the numerical computation of ARTs is much slower than the FBP but efficiency is not a topic in this thesis. The advantage is that ARTs yield better results than the FBP in many cases, e.g. if only a few images are used. Mainly we keep track of two approaches, one complete new one which we call *Regularized algebraic reconstruction technique* (RART) and the *Regularized Kaczmarz's algorithm with relaxation parameter* (RKERP), developed by C. Popa and R. Zdunek.[3] According to the authors this approach is not implemented in three dimensions up to now.

1.1 Overview

In the following we give a brief description of the structure of this thesis.

In Chapter 2, **Overview of 3D Tomographic Reconstruction**, we describe the basics of 3D

[3] See [Pop05a], [Pop05b] and [Pop04].

image reconstruction. A short introduction of the Radon transform is given which is the basis of every reconstruction algorithm. Additionally, the fundamentals of the underlying geometry are described. Finally, we present the idea of algebraic reconstruction techniques. We do not go too much into detail since there are many books and publications about these topics. See e.g. [Eps03] and [Kak01].

In Chapter 3, **Variational Formulation**, we describe the reconstruction problem by a minimization term of a functional. The introduced regularization terms are meaningful in this variational formulation.

In Chapter 4, **Regularized Algebraic Reconstruction Technique**, the connection between the variational term and the well-known algebraic techniques is established. Basically, this leads to the discretization of the problem. The resulting matrix is very large, fortunately sparsely populated but no band-matrix. Consequently, there is no chance to solve the problem with a direct approach. One has to use an iterative approach which is the content of Chapter 5,

Iterative Solvers for the RART. We give a detailed description of the above mentioned algorithms, RART and RKERP which are both extensions of Kaczmarz's algorithm (see e.g. [Ros82] and [Kak01]. Additionally, we introduce a simplified version of the RKERP algorithm, which we call the SRKERP method.

In Chapter 6, **Evaluation and Results**, we compare the results of these approaches with standard algebraic techniques on noisy synthetic data and on real image data. The comparison is done visually and with the help of mathematical tools.

In Chapter 7, **Further Research**, and Chapter 8, **Summary and Conclusion** we summarize the results and give an outlook over future research topics.

Chapter 2

Overview of 3D Tomographic Reconstruction

In this chapter we take a brief look on the fundamentals of 3-D reconstruction. The idea is to visualize inner parts of the human body which are not visible from the outside without performing invasive interactions. X-ray imaging has this property but provides no depth information. The gray value of a specific point in the image corresponds to an attenuated ray which goes from the X-ray source through the body to the detector. One can compare such an image with a shadow generated by an object located between a light source and a projection wall.

In today's approaches and implementations the following assumptions for the interaction of X-rays with matter are used:[1]

1. NO REFRACTION OR DIFFRACTION: X-ray beams travel along straight lines that are not "bent" by the objects they pass through.

2. THE X-RAYS USED ARE MONOCHROMATIC: The waves making up the x-ray beam are all of the same frequency.

3. BEER'S LAW: Each material encountered has a characteristic linear attenuation coefficient μ for x-rays of a given energy. The intensity, I of the x-ray beam satisfies Beer's law

$$\frac{dI}{ds} = -\mu(x)I. \tag{2.1}$$

   Here s is the arc-length along the straight-line trajectory of the x-ray beam.

[1] From [Eps03, page 57].

According to Beer's law the measured energy on the detector is

$$I = I_0 e^{-\int_l \mu(x) ds}, \tag{2.2}$$

where l is the path along a certain ray.

The energy I_0 which is measured on the detector if no object is hit by the ray follows from the integration of equation (2.1). Taking the logarithm of (2.2) yields

$$\int_l \mu(x) ds = -\log \frac{I}{I_0}. \tag{2.3}$$

2.1 The Radon Transform

The integral in (2.3) is nothing else than the Radon transform which is defined for parallel projections. It was introduced by Johann Radon in 1917, more than 50 years before the first CT scanner was built. The Radon transform describes the integrals along projection rays perpendicular to a given line in 2-D space (or perpendicular to a given plane in 3-D space, respectively). We define the two-dimensional Radon transform as follows:

$$\begin{aligned} \mathcal{R}_\theta(t)u &:= \int_{(\theta,t)} u(x,y) ds \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} u(x,y)\delta(x\cos\theta + y\sin\theta - t) dx\, dy, \end{aligned} \tag{2.4}$$

where $u : \mathbb{R}^2 \mapsto \mathbb{R}$ is the density function and δ is the delta distribution. Figure 2.1 shows the Radon transform for two different angles θ_1 and θ_2.

In this definition the dimensions are preserved, we have

$$\begin{aligned} \mathcal{R} : \quad \mathbb{R}^2 &\mapsto \mathbb{R}^2 \\ (x,y) &\mapsto (\theta, t). \end{aligned}$$

In three dimensions $u : \mathbb{R}^3 \mapsto \mathbb{R}$ is a three-dimensional density function, the object is projected onto a plane and the plane can be described by two angles. This means that the 3-D Radon transform has four degrees of freedom. In practice, we only have a rotation around one axis which means we have three degrees of freedom and the dimensions are preserved as well. If we restrict ourselves on a rotation around the z-axis only, the three-dimensional Radon transform

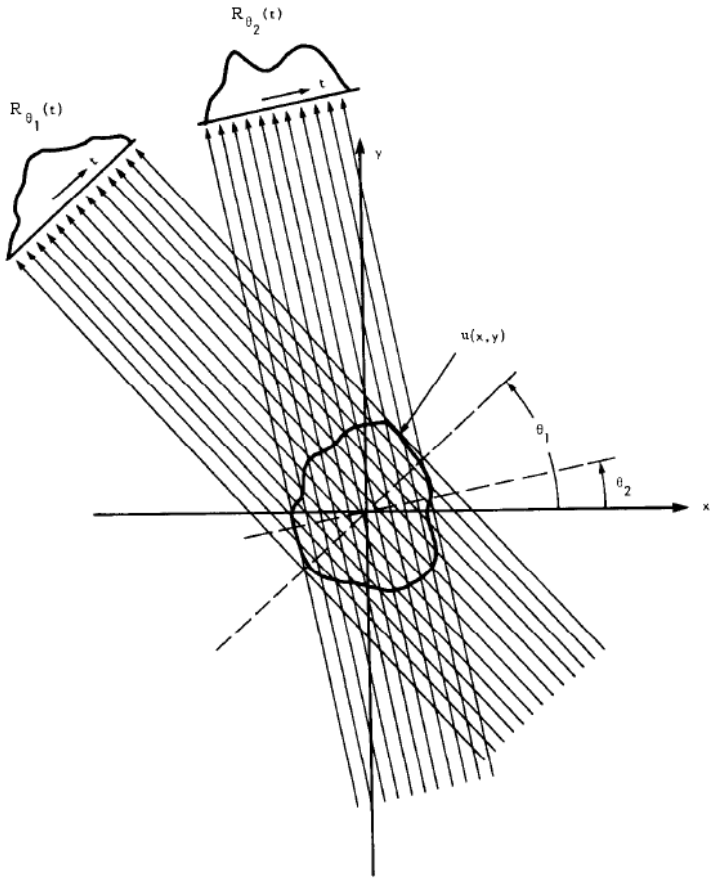


Figure 2.1: Parallel projections of different angles. (Image is taken from [Kak01, page 51].)

looks as follows:

$$\begin{aligned} \mathcal{R}_\theta(t,z)u &:= \int_{(\theta,t)} u(x,y,z)ds \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} u(x,y,z)\delta(x\cos\theta + y\sin\theta - t)dx\,dy, \end{aligned} \tag{2.5}$$

and the mapping looks as

$$\begin{aligned} \mathcal{R}: \quad \mathbb{R}^3 &\mapsto \mathbb{R}^3 \\ (x,y,z) &\mapsto (\theta,t,z). \end{aligned}$$

For further details about the Radon transform we refer to [Kak01] and [Eps03]; in the following, we only mention some basic properties.

Proposition 1 (Linearity) *The Radon transform is linear. That means that for every* $a, b \in \mathbb{R}$

$$\mathcal{R}(au + bv) = a\mathcal{R}u + b\mathcal{R}v \tag{2.6}$$

for every u *and* v *in the natural domain of the Radon transform.*

Proposition 2 (Monotonicity) *The Radon transform is monotone. That means that if* u *is non-negative on the whole natural domain of the Radon transform then*

$$\mathcal{R}u \geq 0 \tag{2.7}$$

in the whole range of $\mathcal{R}$*.*

Proposition 3 (Periodicity) *The Radon transform is periodic with respect to* θ*:*

$$\mathcal{R}_{\theta+2\pi}u = \mathcal{R}_\theta u. \tag{2.8}$$

Additionally, it has the following property:

$$\mathcal{R}_{\theta+\pi}(t)u = \mathcal{R}_\theta(-t)u. \tag{2.9}$$

Consequently, the Radon transform is completely described if it is given for $\theta \in [0; \pi)$*.*

The problem we want to solve is the inverse Radon transform: Find $\mathcal{R}^{-1}$, such that

$$\mathcal{R}^{-1} \circ \mathcal{R}f = f. \tag{2.10}$$

Solving this equation analytically is only possible for model problems and is useful for understanding the theory behind reconstruction issues. Mathematical tools like the Hilbert transform and Fourier transform are required and we refer the interested reader to [Ste05] and [Eps03].

2.2 Cone Beam Reconstruction

Using parallel beam reconstruction means that one has to move the source and **one** detector element for each measurement of a single ray. Another possibility would be to ensure that each detector element is hit by a ray from a single source. The effort for practical realization would be too big, so this technique is not used. Instead fan beam reconstruction is used for 2-D and

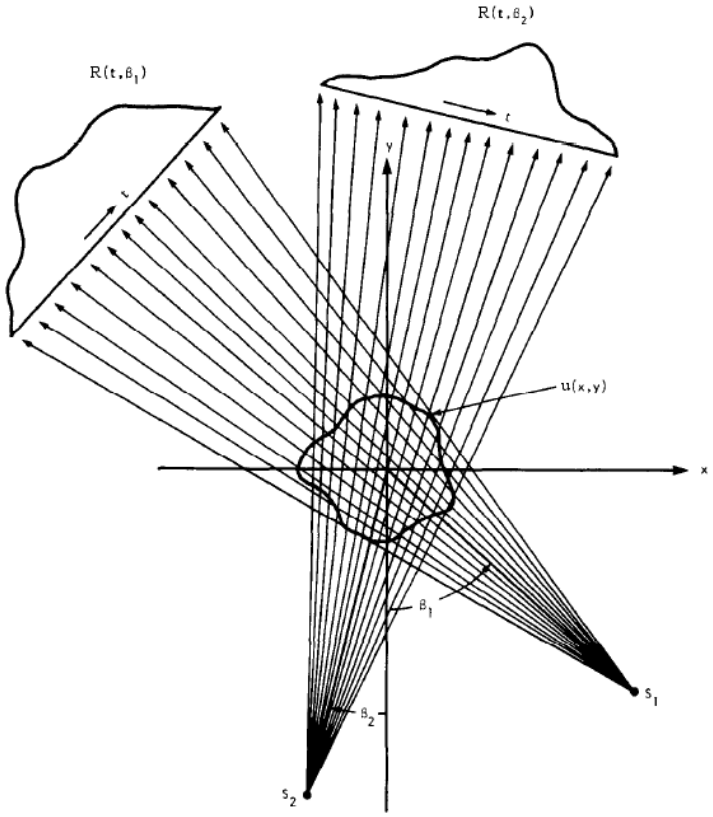


Figure 2.2: Fan beam projections of different angles. (Image is taken from [Kak01, page 52].)

cone beam reconstruction is used for 3-D reconstruction. This means that the rays from a single point source hit many detector elements on a line (or a plane, respectively).[2] Figure 2.2 shows this method for two dimensions schematically.

Obviously, it can be seen that it is not sufficient anymore to know the angle and position of the line (or plane) a ray is projected onto. Additionally, one has to know the exact position of the source and the detector line (or plane) to establish a connection between a measured ray and the geometry of the object. In three-dimensional reconstruction we use a model that uses eleven degrees of freedom which are known as extrinsic and intrinsic camera parameters. Extrinsic parameters describe the geometry which changes for every projection. The intrinsic parameters are bind to the source/detector hardware. (Notice that the directions of the rays are reversed with respect to a classical camera environment). The parameters are scheduled in the following list:

- Extrinsic parameters

[2]The fact that in real systems the X-rays leave the tube not from a single point but from a small region is neglected here.

  - Translation of X-ray source, 3 parameters.
  - Rotation of X-ray source, 3 parameters.

- Intrinsic parameters
  - Translation of image coordinate system to pixel coordinate system, 2 parameters.
  - Angle between axes of pixel coordinate system, 1 parameter.
  - Units of axes in the pixel coordinate system with respect to the image coordinate system, 2 parameters.[3]

Using homogenous coordinates this results in a 3×4 matrix by which each point in 3-D can be projected onto the 2-D plane. Details for calibrating a C-Arm system and computing these projection matrices can be found e.g. in [Hor05]. Unfortunately, the calibration matrix can only be computed up to a certain precision in real systems which results in one of the major sources of errors in the reconstruction.

2.3 Algebraic Reconstruction Techniques

In this section we briefly describe the idea of algebraic reconstruction. As mentioned above, it is quite complicated to solve the reconstruction problem in the continuous domain. To enter the discrete domain, we take equation (2.3) and replace the integral by a sum which yields

$$\sum_{j=1}^{N} a_{i,j} u_j = b_i. \tag{2.11}$$

Here $b_i = -\log \frac{I_i}{I_0}$ is the i-th pixel value of the acquired images, u_j is the j-th density value of the volume[4] and $a_{i,j}$ is the weight by which the j-th density value is met by the i-th ray. Figure 2.3 shows the interaction between volume and projection image.

Let us denote the number of rays or pixels in **all** projection images by M and the number of all unknown density values or voxels by N. Then equation (2.11) can be written in matrix form as

$$Au = b, \tag{2.12}$$

[3]The focal length of the camera, which depends on the distance between the X-ray source and the detector plane is included here.

[4]From now on we restrict ourselves on three-dimensional reconstruction.

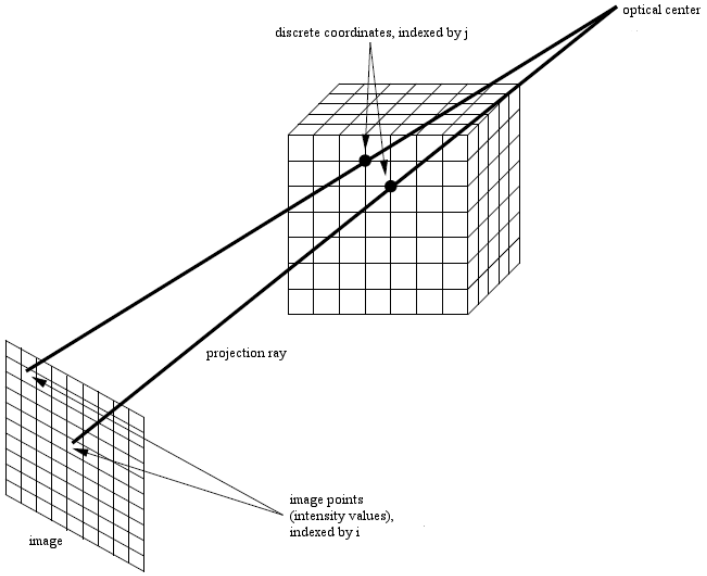


Figure 2.3: 3D Projection. (Image is taken from [Hor05].)

with $A \in \mathbb{R}^{M,N}$, $u \in \mathbb{R}^N$ and $b \in \mathbb{R}^M$.

There are many possibilities to define the weights of A, the simplest idea is to define the values in a binary way. If the ray i touches the voxel j then $a_{i,j} = 1$, otherwise $a_{i,j} = 0$. Another possibility is to assign to each ray an artificial thickness. The value $a_{i,j}$ is then the volume of the voxel j sliced by the ray i. Figure 2.4 shows this method in two dimensions.

The last method we mention is to assign the length of the ray i within the voxel j to the corresponding entry in the matrix A. Figure 2.5 explains this idea visually.

Notice that in most of the cases the matrix A is too large to hold every non-zero value in memory at one point in time. Instead the values of one row which correspond to one ray have to be computed for each usage time and again. How to solve such a large system? An additional problem is that the linear system is inconsistent in real applications since the measured pixel values are distorted by noise. A common technique used in algebraic reconstruction is the Kaczmarz algorithm which we introduce in the following.

2.3.1 The Kaczmarz Method

The idea of the Kaczmarz method is to treat each equation of the linear system $Wf = p$, with $W \in \mathbb{R}^{M,N}$, $f \in \mathbb{R}^N$ and $p \in \mathbb{R}^M$, as an $N-1$-dimensional hyperplane in an N-dimensional

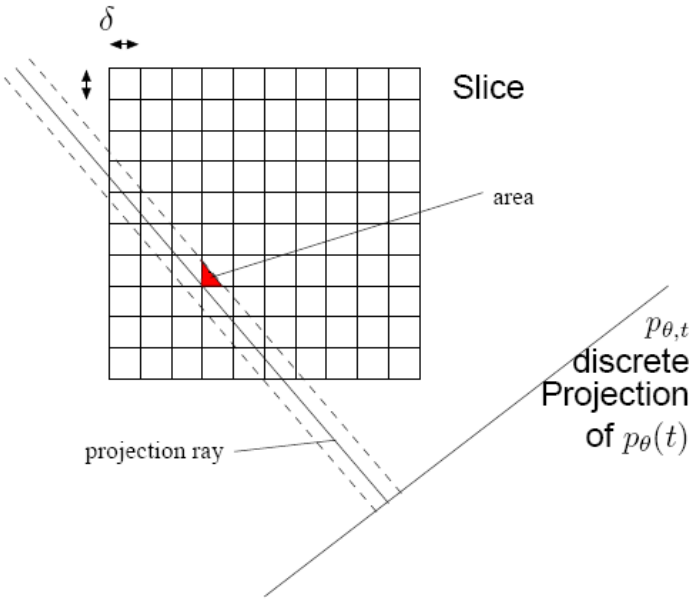


Figure 2.4: Weighting matrix defined by rays with artificial width. (Image is taken from [Hor05].)

space. If, for example $N = 2$, then

$$w_{i,1}f_1 + w_{i,2}f_2 = p_i$$

defines a line in a two-dimensional space. For $N = 3$,

$$w_{i,1}f_1 + w_{i,2}f_2 + w_{i,3}f_3 = p_i$$

defines a plane in a three-dimensional space, and so on.

If $M \geq N$ and the system is consistent then all hyperplanes intersect in one common sub-hyperplane. (If they all intersect in one point, there is only one solution of the system.) Obviously, the points of this sub-hyperplane are solutions of all linear equations and thus solutions of the linear system. In the distorted and inconsistent case not all hyperplanes intersect in one common sub-hyperplane but in a relative small area. Somewhere in this area one can find a least-squares solution.

The Kaczmarz method takes a given starting vector and successively projects it on all hyperplanes. This can be done several times. Figure 2.6 shows the idea for $N = 2$, $M = 4$ and two iterations.

Let us denote the linear system again as $Au = b$, with $A \in \mathbb{R}^{M,N}$, $u \in \mathbb{R}^N$ and $b \in \mathbb{R}^M$. By

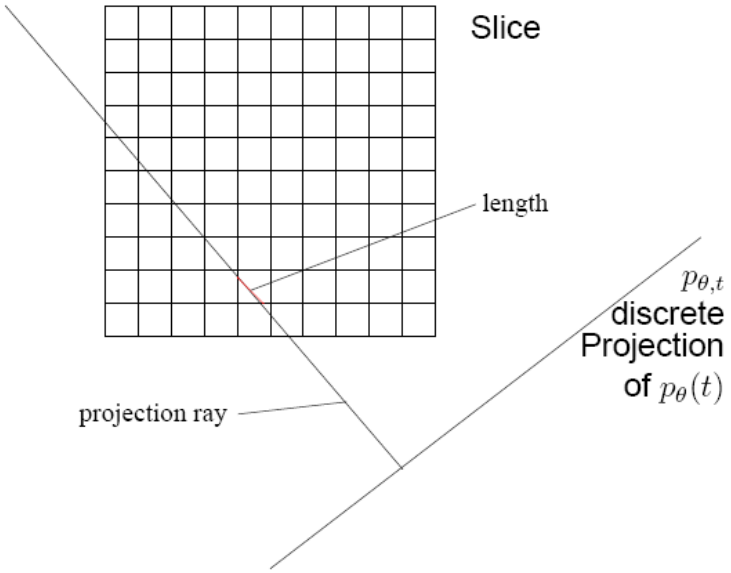


Figure 2.5: Weighting matrix defined by rays according to length. (Image is taken from [Hor05].)

$a_i \in \mathbb{R}^N$, we denote the i-th row of A and we suppose that

$$a_i \neq 0, \forall i = 1, \ldots, M. \tag{2.13}$$

All vectors that will appear are considered as column vectors. Let us define

$$f_i(b; u) := u - \frac{\langle u, a_i \rangle - b_i}{\|a_i\|^2} a_i, \tag{2.14}$$

for $i = 1, \ldots, M$ and

$$F(b; u) := (f_1 \circ \ldots \circ f_M)(b; u). \tag{2.15}$$

Now the classical **K**aczmarz algorithm (for short **K**) is defined as follows:[5]

Algorithm 1 (K) *Let $u^0 \in \mathbb{R}^N$; for $k = 0, 1, 2, \ldots$ do*

$$u^{k+1} = F(b; u^k). \tag{2.16}$$

In [Nat86] it was proved that the sequence $(u^k)_{k \geq 0}$ from (2.16) always converges but not necessarily to a least-squares solution if the linear system is inconsistent. As a consequence, one often gets "rough" results by reconstructing real image data.

A first improvement is to use the **K**aczmarz with **R**elaxation **P**arameter algorithm (for short

[5] According to [Pop05a].

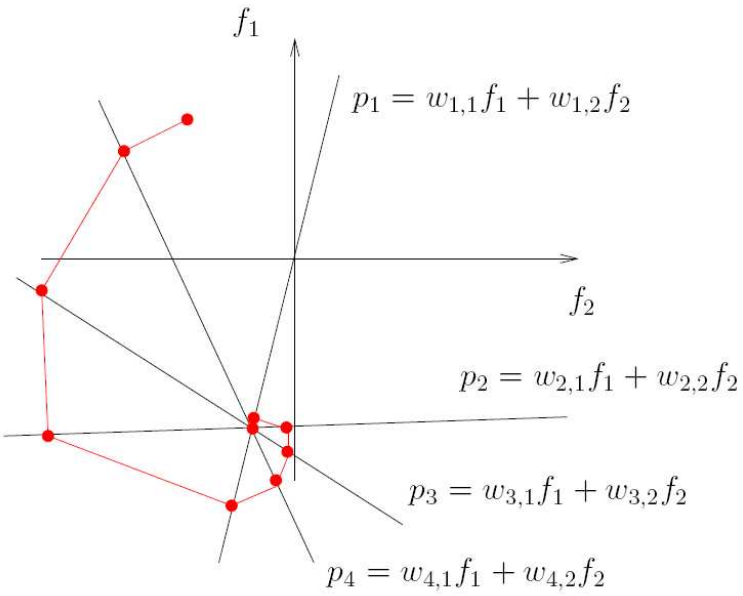


Figure 2.6: Kaczmarz method. (Image is taken from [Hor05].)

KRP). For this we have to alter equations (2.14) and (2.15) as follows: Let $\lambda \in (0, 2)$. Then we define

$$f_i(\lambda; b; u) := (1 - \lambda)u + \lambda f_i(b; u), \tag{2.17}$$

for $i = 1, \ldots, M$,

$$F(\lambda; b; u) := (f_1 \circ \ldots \circ f_M)(\lambda; b; u) \tag{2.18}$$

and the **KRP** algorithm we define as

Algorithm 2 (KRP) *Let $u^0 \in \mathbb{R}^N$; for $k = 0, 1, 2, \ldots$ do*

$$u^{k+1} = F(\lambda; b; u^k). \tag{2.19}$$

It turns out that a relaxation parameter of $\lambda \in [0.1, 0.2]$ is a good choice.

In this chapter we have presented a brief overview of 3-D image reconstruction and the basic idea of algebraic reconstruction techniques. All these methods are state of the art and we refer again to [Kak01] and [Eps03] for a more detailed survey about the topic. In the next chapter we introduce a new point of view of the reconstruction problem.

Chapter 3

Variational Formulation

In this chapter we formulate the reconstruction problem in a variational setting. The idea of using variational calculus for tomographic reconstruction was already followed by P. Milanfar et al. in [Mil96] and J. Kybic et al. in [Kyb01]. We use a different formulation which is more related to algebraic reconstruction techniques.

The chapter is partitioned as follows: Firstly, we present a brief and general setting in which the reconstruction problem is introduced. Secondly, we formulate the variational approach to tomographic reconstruction in the continuous domain. The reformulation of the variational approach in a semi-discrete setting to establish the connection to algebraic reconstruction techniques is part of the next chapter.

3.1 Introduction to Variational Calculus

In differential calculus one important task is the following question: For which x-values has the function $y(x)$ an extreme value? The corresponding question in variational calculus is as follows: For which functions of a specific function class does an integral, whose integrand depends on this function and its derivatives, have an extreme value?[1] Additionally, boundary and side conditions could be defined for the function and its derivatives. The following equation describes the one-dimensional, general case:

$$I[y] = \int_a^b F(x, y(x), y'(x), \ldots, y^{(n)}(x))dx. \tag{3.1}$$

A famous introductory example for variational calculus is to find a curve $y(x)$, spanned by a

[1] Cf. [Bro01] and [Els70].

rope with given length l, which yields the maximum area $I[y]$. A special case is to find a junction between two points $A(a,0)$ and $B(b,0)$ with length l: Find a one times differentiable function $y(x)$, for which

$$I[y] = \int_a^b y(x)dx = \max, \tag{3.2}$$

the side condition

$$G[y] = \int_a^b \sqrt{1 + y'^2(x)}dx = l \tag{3.3}$$

and

$$y(a) = y(b) = 0 \tag{3.4}$$

holds.[2]

3.1.1 Eulerian Differential Equation in Variational Calculus

In the following we do a short derivation of the simple, one-dimensional variational problem. Then we extend the formulation to three dimensions which is the basis of our reconstruction setting. The derivation is geared to [Bro01, page 574 et sqq.].

As a *simple variational problem* we denote the following task: Find extreme values of the integral

$$\boldsymbol{I[y] = \int_a^b F(x, y(x), y'(x))dx,} \tag{3.5}$$

with $y(x)$ two times differentiable and the side conditions $y(a) = A$ and $y(b) = B$. The values a, b and A, B and the function F are given. The integral in (3.5) is called a *functional*. If the functional $I[y]$ has a relative maximum for the function $y_0(x)$, then

$$I[y_0] \geq I[y] \tag{3.6}$$

for every other function y for which the side conditions hold. The curve $y = y_0(x)$ is called an *extremal*.

To solve the variational problem we first construct the function

$$y(x) = y_0(x) + \epsilon\eta(x), \tag{3.7}$$

with the two times differentiable function $\eta(x)$ and the special side conditions $\eta(a) = \eta(b) = 0$.

[2] From [Bro01, page 573].

The parameter ϵ is a real parameter. Inserting (3.7) into (3.5) yields

$$I(\epsilon) = \int_a^b F(x, y_0 + \epsilon\eta, y_0' + \epsilon\eta')dx. \tag{3.8}$$

The requirement that $y(x)$ makes $I[y]$ to an extreme value is equivalent to the following condition that $I(\epsilon)$ has an extreme value for $\epsilon = 0$:

$$\frac{dI}{d\epsilon} = 0 \quad \text{for} \quad \epsilon = 0. \tag{3.9}$$

According to the assumption that the integrand F is sufficiently often partial differentiable, we get with help of its Taylor expansion

$$I(\epsilon) = \int_a^b \left(F(x, y_0, y_0') + \frac{\partial F}{\partial y}(x, y_0, y_0')\epsilon\eta + \frac{\partial F}{\partial y'}(x, y_0, y_0')\epsilon\eta' + O(\epsilon^2) \right) dx. \tag{3.10}$$

The necessary condition (3.9) yields

$$\int_a^b \eta \frac{\partial F}{\partial y} dx + \int_a^b \eta' \frac{\partial F}{\partial y'} dx = 0, \tag{3.11}$$

and by integration by parts with respect to η we finally get

$$\int_a^b \eta \left(\frac{\partial F}{\partial y} - \frac{d}{dx} \left(\frac{\partial F}{\partial y'} \right) \right) dx = 0. \tag{3.12}$$

Since the integral in (3.12) should vanish for every η, the necessary condition

$$\boldsymbol{\frac{\partial F}{\partial y} - \frac{d}{dx} \left(\frac{\partial F}{\partial y'} \right) = 0} \tag{3.13}$$

must hold. Equation (3.13) is called *Eulerian differential equation of variational calculus* and can also be written as

$$\frac{\partial F}{\partial y} - \frac{\partial^2 F}{\partial x \partial y'} - \frac{\partial^2 F}{\partial y \partial y'} y' - \frac{\partial^2 F}{\partial y'^2} y'' = 0. \tag{3.14}$$

The simple variational problem in three dimensions is very similar. Let $\Omega \subset \mathbb{R}^3$, $x \in \Omega$ and $u : \Omega \mapsto \mathbb{R}$. Furthermore, we assume that u is two times partial differentiable. Then the simple variational problem is denoted as follows: Find u such that

$$\boldsymbol{I[u] = \int_{\Omega} F(x, u(x), \nabla u(x)) d\Omega = \text{extreme!}} \tag{3.15}$$

holds. The integral in (3.15) is a triple integral. In components (3.15) can also be written as

$$I[u] = \iiint_{\Omega} F(x_1, x_2, x_3, u(x_1, x_2, x_3), u_{x_1}, u_{x_2}, u_{x_3}) dx_1\, dx_2\, dx_3 = \text{extreme!}, \tag{3.16}$$

where $x = (x_1, x_2, x_3)^T$ and $u_{x_n} = \frac{\partial u}{\partial x_n}$ for $n = 1, 2, 3$. Similar to the one-dimensional case, $u(x)$ has to fulfill the Eulerian differential equation

$$\frac{\partial F}{\partial u} - \frac{\partial}{\partial x_1}\left(\frac{\partial F}{\partial u_{x_1}}\right) - \frac{\partial}{\partial x_2}\left(\frac{\partial F}{\partial u_{x_2}}\right) - \frac{\partial}{\partial x_3}\left(\frac{\partial F}{\partial u_{x_3}}\right) = 0 \tag{3.17}$$

to be a solution of (3.16). In vector notation (3.17) can be written as

$$\boldsymbol{\frac{\partial F}{\partial u} - \operatorname{div}(\nabla_{u'} F) = 0,} \tag{3.18}$$

with $\nabla_{u'} = \left(\frac{\partial}{\partial u_{x_1}}, \frac{\partial}{\partial u_{x_2}}, \frac{\partial}{\partial u_{x_3}}\right)^t$.

3.2 Variational Approach to Tomographic Reconstruction

How do we formulate the reconstruction problem in a variational setting? The basic idea is to find an extremal, namely the minimum, of the difference of measured projection images and the forward projections of the reconstructed volume. Since the volume, which is used for forward projection, is the searched object, we have an inverse problem. In the following we define by $\Omega \subset \mathbb{R}^3$ the open, connected and convex domain which contains the reconstructed volume. By $\Gamma \subset \mathbb{R}^2$ we define the area which contains a projection image. Reconsider the definition of the three-dimensional Radon transform in Section 2.1 with $\theta \in [0, \pi)$ the rotation angle with respect to the z-axis. Let

$$\vartheta_i \in [0, \pi) \quad \text{with} \quad i = 1, \ldots, K$$

the angles for which projection images are acquired. By

$$p_{\vartheta_i} : \Gamma \mapsto \mathbb{R}$$

we denote these projection images. Finally, we define an operator

$$\mathrm{I} : \Gamma \times \{\vartheta_1, \ldots \vartheta_K\} \mapsto \Gamma \times [0, \pi)$$

which interpolates projection images corresponding to angles that are not part of the acquisition. Obviously,

$$\mathrm{I}_\theta p(t,z) = p_\theta(t,z) \quad \text{if} \quad \theta \in \{\vartheta_1, \ldots, \vartheta_K\}. \tag{3.19}$$

Then the variational approach looks as follows: Find u such that

$$\iiint\limits_{\Gamma\times[0,\pi)} \left(\mathcal{R}_\theta(t,z)u - \mathrm{I}_\theta p(t,z)\right)^2 dt\, dz\, d\theta = \min. \tag{3.20}$$

As mentioned earlier we also want to include a-priori knowledge into the functional. For that purpose we introduce a *roughness function* that measures the "roughness" of the reconstructed volume and that should be penalized. For example, this function may depend on the length of the gradient; if the gradient is large then the volume is rough. Consequently,

$$\phi : \mathbb{R} \mapsto \mathbb{R}$$

and the penalization term looks as

$$\int_\Omega \phi(\|\nabla u\|) d\Omega = \min. \tag{3.21}$$

The integral in (3.21) is a triple integral again. Adding (3.20) and (3.21) and weighting (3.21) by a real parameter α yields the final variational approach

$$\iiint\limits_{\Gamma\times[0,\pi)} \left(\mathcal{R}_\theta(t,z)u - \mathrm{I}_\theta p(t,z)\right)^2 dt\, dz\, d\theta + \alpha \iiint\limits_{\Omega} \phi(\|\nabla u\|) dx\, dy\, dz = \min, \tag{3.22}$$

or with an abbreviated notation

$$\int_{\tilde{\Gamma}} \left(\boldsymbol{\mathcal{R}u - \mathrm{I}p}\right)^2 \boldsymbol{dx} + \boldsymbol{\alpha} \int_{\boldsymbol{\Omega}} \boldsymbol{\phi}(\|\boldsymbol{\nabla u}\|) \boldsymbol{dx} = \mathbf{min}\,. \tag{3.23}$$

Finally, the application of the Eulerian differential equation to (3.23) yields

$$\boldsymbol{\mathcal{R}^*\mathcal{R}u} - \frac{\boldsymbol{\alpha}}{\mathbf{2}} \mathbf{div}\left(\frac{\boldsymbol{\phi'}(\|\boldsymbol{\nabla u}\|)}{\|\boldsymbol{\nabla u}\|} \boldsymbol{\nabla u}\right) = \boldsymbol{\mathcal{R}^*\mathrm{I}p}, \tag{3.24}$$

whose solution is the sought reconstruction volume. Here $\mathcal{R}^*$ is some kind of adjoint Radon transform which we do not specify further. The introduction of $\mathcal{R}^*$ is possible since the Radon transform is linear.

Chapter 4

Regularized Algebraic Reconstruction Technique

In the following, we reformulate the variational approach in a semi-discrete setting. Solving this with a "discrete version" of (3.18) yields an extension to the algebraic reconstruction techniques. Secondly, we introduce some properties which the regularization term should have. In the last part of the chapter we give some examples of regularization terms which we examine later in this master's thesis.

4.1 Discrete Variational Approach

From Section 2.3, we know that the Radon transform (or an adapted version with respect to geometry) can be written in a discrete form by a simple matrix notation. The integral in (3.20) becomes a sum and we get

$$\|Au - b\|^2 = \min, \tag{4.1}$$

with $A \in \mathbb{R}^{M,N}$ the matrix constructed by one of the methods of Section 2.3, or more sophisticated methods, $u \in \mathbb{R}^N$ the vector of the discretized volume to be reconstructed and $b \in \mathbb{R}^M$ the vector of all pixels from the projection images. The norm in (4.1) is the *Euclidean norm* and is defined as

$$\|v\| := \sqrt{\sum_{j=1}^{N} v_j^2}, \tag{4.2}$$

for a vector $v \in \mathbb{R}^N$. Notice that we can neglect the interpolation operator since we just use the discrete values of the measurement. Obviously, a u which is a least-squares solution of (4.1) is

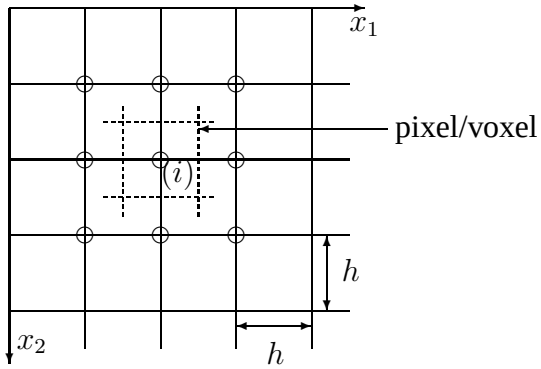


Figure 4.1: Grid on the space domain. (Picture is adapted from [Aub02, page 257].)

obtained by solving

$$A^t A u = A^t b, \tag{4.3}$$

which is the discretized version of the data term in (3.24).

From a numerical point of view, the discretization of the regularization term in (3.24) has some difficulties. A common mistake is to rewrite the divergence term in components and perform the discretization afterwards. A better and easier solution can be achieved by separate discretization of divergence and gradient. By rewriting the regularization part as

$$g(s) := \frac{\phi'(s)}{2s}, \tag{4.4}$$

where

$$g : \mathbb{R} \mapsto \mathbb{R},$$

the term that remains to discretize is $\mathrm{div}(g(\|\nabla u\|)\nabla u)$. Applying the finite difference method in an appropriate way the divergence term can be discretized as follows:[1]

$$\begin{aligned} \mathbf{div}(\boldsymbol{g}(\|\boldsymbol{\nabla u}\|)\boldsymbol{\nabla u})|_{\boldsymbol{i}} &\approx \widetilde{\mathbf{div}}(\boldsymbol{g}(\|(\widetilde{\boldsymbol{\nabla}}\boldsymbol{u})_{\boldsymbol{i}}\|)(\widetilde{\boldsymbol{\nabla}}\boldsymbol{u})_{\boldsymbol{i}}) \\ &= \sum_{\boldsymbol{p}\in\boldsymbol{\eta_i}} \boldsymbol{\lambda_{i,p}} \boldsymbol{g}(|\boldsymbol{u_p} - \boldsymbol{u_i}|)(\boldsymbol{u_p} - \boldsymbol{u_i}). \end{aligned} \tag{4.5}$$

Here η_i is the set of neighboring voxels with respect to voxel i. In the three-dimensional case the simplest possibilities are 6-, 18- or 26-neighborhoods. Figure 4.1 shows the 8-neighborhood in the 2D case. Notice that we normalize the distance h between two neighboring voxels to 1. The

[1] For a detailed derivation, we refer the interested reader to [Aub02], [Per90] and [Sch04].

	$\lvert x_p - x_i \rvert$	$\lambda_{i,p}$	$\lambda_{i,p}$, num. approx.
6-neighborhood	1	$\frac{1}{6}$	0.1667
18-neighborhood	1	$\frac{1}{6(1+\sqrt{2})}$	0.0690
	$\sqrt{2}$	$\frac{1}{6\sqrt{2}(1+\sqrt{2})}$	0.0488
26-neighborhood	1	$\frac{3}{2(9+9\sqrt{2}+4\sqrt{3})}$	0.0523
	$\sqrt{2}$	$\frac{3}{2\sqrt{2}(9+9\sqrt{2}+4\sqrt{3})}$	0.0370
	$\sqrt{3}$	$\frac{\sqrt{3}}{2(9+9\sqrt{2}+4\sqrt{3})}$	0.0302

Table 4.1: Weights $\lambda_{i,p}$ in regularization term.

scalars $\lambda_{i,p}$ are weights to be chosen. In this thesis we chose the following two conditions:

$$\begin{cases} \lambda_{i,p} \sim \frac{1}{|x_p - x_i|}, \\ \sum\limits_{p \in \eta_i} \lambda_{i,p} = 1. \end{cases}$$

With respect to neighborhood type and distance to voxel i, the weights has to be chosen as shown in Table 4.1.

There are other possibilities for choosing the weights $\lambda_{i,p}$, e.g. with a more accurate mathematical interpretation or with dependency on the direction of the gradient.

With the final abbreviation

$$L(u)u = \widetilde{\operatorname{div}}(g(\|(\widetilde{\nabla}u)_i\|)(\widetilde{\nabla}u)_i)_{i=1}^N, \tag{4.6}$$

where $L(u)u \in \mathbb{R}^N$, and $L(u) \in \mathbb{R}^{N,N}$ is defined as

$$L(u)_{i,j} := \begin{cases} \sum\limits_{p \in \eta_i} \lambda_{i,p} g(|u_p - u_i|), & i = j, \\ -\lambda_{i,j} g(|u_j - u_i|), & j \in \eta_i, \\ 0, & \text{else}, \end{cases} \tag{4.7}$$

the complete discrete scheme looks as

$$\boldsymbol{A^t A u - \alpha L(u) u = A^t b.} \tag{4.8}$$

4.2 Properties of the Regularization Term

In the following we study more precisely the influence of the smoothing term. According to [Aub02, page 64] our goal is to

> [...] find the properties of ϕ so that the solution of the minimization problem is close to a piecewise constant image that is formed by homogenous regions separated by sharp edges.

In the first step we rewrite the divergence term in components. In three dimensions and for $\nabla u = (u_{x_1}, u_{x_2}, u_{x_3})^t$, the decomposition yields after some calculation

$$\begin{aligned}
&\operatorname{div}\left(\frac{\phi'(\|\nabla u\|)}{\|\nabla u\|}\nabla u\right)\\
&= \frac{\phi'(\|\nabla u\|)}{\|\nabla u\|}\frac{1}{\|\nabla u\|^2}(u_{x_2}^2 u_{x_1x_1} + u_{x_3}^2 u_{x_1x_1} + u_{x_1}^2 u_{x_2x_2} + u_{x_3}^2 u_{x_2x_2} + u_{x_1}^2 u_{x_3x_3} + u_{x_2}^2 u_{x_3x_3}\\
&\qquad - 2u_{x_1}u_{x_2}u_{x_1x_2} - 2u_{x_1}u_{x_3}u_{x_1x_3} - 2u_{x_2}u_{x_3}u_{x_2x_3})\\
&\qquad + \phi''(\|\nabla u\|)\frac{1}{\|\nabla u\|}(u_{x_1}^2 u_{x_1x_1} + u_{x_2}^2 u_{x_2x_2} + u_{x_3}^2 u_{x_3x_3}\\
&\qquad\qquad + 2u_{x_1}u_{x_2}u_{x_1x_2} + 2u_{x_1}u_{x_3}u_{x_1x_3} + 2u_{x_2}u_{x_3}u_{x_2x_3}).
\end{aligned}$$

According to the local image structure, we introduce the vectors $n(x) = \frac{\nabla u(x)}{\|\nabla u(x)\|}$, $t_1(x) = \frac{(u_{x_2}, -u_{x_1}, 0)^t}{\|\nabla u\|}$, $t_2(x) = \frac{(-u_{x_3}, 0, u_{x_1})^t}{\|\nabla u\|}$ and $t_3(x) = \frac{(0, u_{x_3}, -u_{x_2})^t}{\|\nabla u\|}$. Obviously, these vectors are only defined for $\|\nabla u(x)\| \neq 0$ and have the properties

$$\begin{cases}
\|n(x)\| = 1,\\
\sum_{i=1}^{3} \|t_i(x)\|^2 = 2 \quad \text{and}\\
\langle t_i(x), n(x)\rangle = 0 \quad \text{for} \quad i = 1, 2, 3.
\end{cases}$$

Figure 4.2 shows the normal direction and one tangent direction to the isophote line (or surface, respectively) schematically. With the second directional derivatives

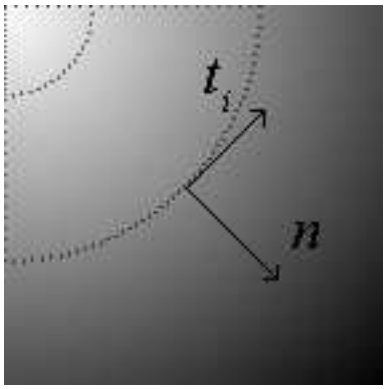


Figure 4.2: Normal and tangent directions to isophote lines. (Image is adapted from [Aub02, page 64].)

$$
\begin{aligned}
u_{t_1t_1} = \quad t_1^t\,\nabla^2 u\,t_1 \quad &= \frac{1}{\|\nabla u\|}(u_{x_1}^2 u_{x_2x_2} + u_{x_2}^2 u_{x_1x_1} - 2u_{x_1}u_{x_2}u_{x_1x_2}),\\
u_{t_2t_2} = \quad t_2^t\,\nabla^2 u\,t_2 \quad &= \frac{1}{\|\nabla u\|}(u_{x_1}^2 u_{x_3x_3} + u_{x_3}^2 u_{x_1x_1} - 2u_{x_1}u_{x_3}u_{x_1x_3}),\\
u_{t_3t_3} = \quad t_3^t\,\nabla^2 u\,t_3 \quad &= \frac{1}{\|\nabla u\|}(u_{x_2}^2 u_{x_3x_3} + u_{x_3}^2 u_{x_2x_2} - 2u_{x_2}u_{x_3}u_{x_2x_3}) \quad \text{and}\\
u_{nn} = \quad n^t\,\nabla^2 u\,n \quad &= \frac{1}{\|\nabla u\|}(u_{x_1}^2 u_{x_1x_1} + u_{x_2}^2 u_{x_2x_2} + u_{x_3}^2 u_{x_3x_3}\\
&\quad +2u_{x_1}u_{x_2}u_{x_1x_2} + 2u_{x_1}u_{x_3}u_{x_1x_3} + 2u_{x_2}u_{x_3}u_{x_2x_3}),
\end{aligned}
$$

where $\nabla^2 u$ is the Hessian matrix of u with

$$(\nabla^2 u)_{i,j} = \frac{\partial u}{\partial x_i \partial x_j}, \tag{4.9}$$

we can rewrite (3.24) as

$$\mathcal{R}^*\mathcal{R}u - \frac{\alpha}{2}\left(\frac{\phi'(\|\nabla u\|)}{\|\nabla u\|}(u_{t_1t_1} + u_{t_2t_2} + u_{t_3t_3}) + \phi''(\|\nabla u\|)u_{nn}\right) = \mathcal{R}^*\mathrm{I}p. \tag{4.10}$$

This notation allows us to see clearly the influence of the regularization term along the directions n and t_i. The function ϕ should be chosen as follows:

- At locations where the gradient is low, we would like to encourage smoothing. These areas are likely disturbed by noise. Assuming that the function ϕ is regular, this isotropic

smoothing condition may be achieved by imposing[2]

$$\phi'(0) = 0, \quad \lim_{s\to 0^+} \frac{\phi'(s)}{s} = \lim_{s\to 0^+} \phi''(s) = \phi''(0) > 0. \tag{4.11}$$

Therefore, at locations where $\|\nabla u\|$ is small, (4.10) becomes

$$\mathcal{R}^*\mathcal{R}u - \frac{\alpha}{2}\phi''(0)(u_{t_1t_1} + u_{t_2t_2} + u_{t_3t_3} + u_{nn}) = \mathcal{R}^*\mathrm{I}p,$$

and since $u_{t_1t_1} + u_{t_2t_2} + u_{t_3t_3} + u_{nn} = \triangle u$,

$$\mathcal{R}^*\mathcal{R}u - \frac{\alpha}{2}\phi''(0)\triangle u = \mathcal{R}^*\mathrm{I}p. \tag{4.12}$$

Obviously, a function that satisfies these conditions is $\phi(s) = s^2$.

- In the neighborhood of edges, the image presents strong gradients. In order to preserve these edges, it is preferable to diffuse along the t_i-directions and not across the edge. Looking again at (4.10), we see that this can be achieved by

$$\lim_{s\to+\infty} \phi''(s) = 0, \quad \lim_{s\to+\infty} \frac{\phi'(s)}{s} = c > 0. \tag{4.13}$$

Unfortunately, these conditions are incompatible. A compromise is to let both terms, $\phi''(s)$ and $\phi'(s)/s$, converge to zero but the first one faster than the second one. This property can be accomplished by

$$\begin{aligned} \lim_{s\to+\infty} \phi''(s) = \lim_{s\to+\infty} \frac{\phi'(s)}{s} &= 0 \quad \text{and} \\ \lim_{s\to+\infty} \frac{\phi''(s)}{\phi'(s)/s} &= 0. \end{aligned} \tag{4.14}$$

All these assumptions on ϕ are qualitative. They are not a guarantee that equation (3.24), or its discrete derivation, are mathematically well posed.

An extension to this approach would be to make the roughness function ϕ more general, i.e. to extend the dependency of ϕ on the direction of the gradient. An idea can be found in [Wei98] or [Wei96]. Another possibility to design the regularization term is to use Markov Random Fields (MRF), which leads to similar results with respect to our diffusion approach (see eg. [Pop05a]).

[2] See [Aub02, page 65].

A detailed mathematical analysis of existence and uniqueness of a solution can be found in [Aub02, page 66 et sqq.] and [Cha97]. The authors introduce the so-called ARTUR algorithm, which minimizes data and regularization term iteratively. This concept is also used in the algorithms we introduce in the next chapter.

4.3 Examples for Regularizers

In this section we present some examples for regularization terms. For each function ϕ we show graphs for the *roughness function* $\phi(s)$, *influence function* $f(s) := \phi'(s)/2$ and *penalty function* $g(s) = f(s)/s = \phi'(s)/(2s)$. As mentioned above the roughness function measures the "roughness" of the image. The influence function measures the impact of the gradient length with respect to the regularization term. The penalty function shows how gradients are penalized.

Note that the convexity of ϕ has an influence on existence and uniqueness of the solution, so this property is also mentioned for the different terms. Additionally, we introduce a *variance parameter* σ which allows us to scale the regularization term with respect to the area of reconstructed values. Finally, we scale the functions such that

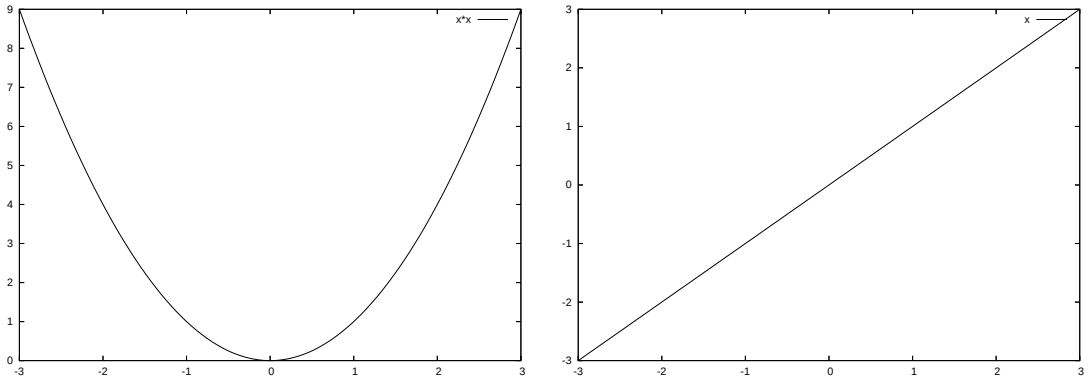
$$\max g(s) = 1. \tag{4.15}$$

- The first function we introduce is the quadratic one which yields the well-known Laplacian regularizer denoted in (4.12). Unfortunately, with this regularizer all gradients are penalized equally, regardless of their lengths. This is also the regularizer with the highest steepness. All other introduced smoothing terms do not penalize high gradients as much as this quadratic function. It is defined as follows:

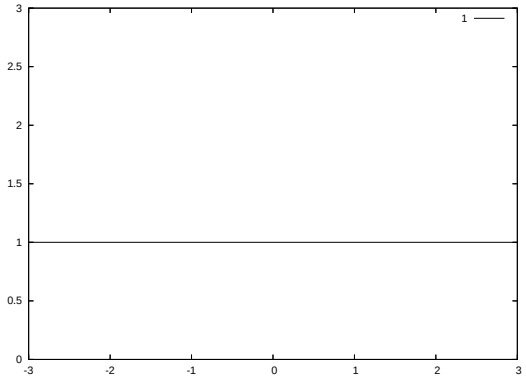
$$\begin{aligned} \phi_q(s) &:= s^2, && (4.16)\\ f_q(s) &:= s \quad \text{and} && (4.17)\\ g_q(s) &:= 1. && (4.18) \end{aligned}$$

  All three functions are shown in Figure 4.3.

- The next regularizer is the so-called *hypersurface minimal function* (see e.g. [Aub02, page



(a) $\phi_q(x)$ (b) $f_q(x)$



(c) $g_q(x)$

Figure 4.3: *Quadratic* roughness, influence and penalty functions.

66]). It has good properties according to the above mentioned conditions and is defined as

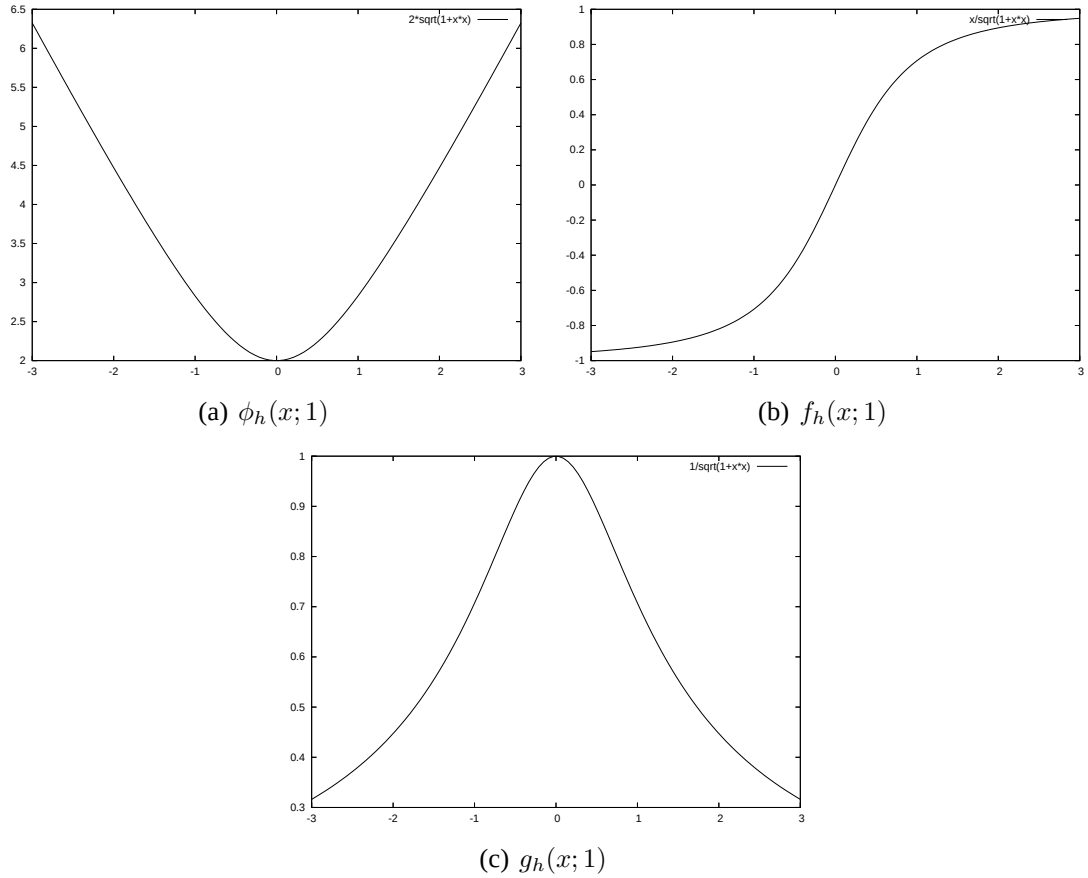
$$\phi_h(s;\sigma) := 2\sigma^2\sqrt{1+\left(\frac{s}{\sigma}\right)^2}, \tag{4.19}$$

$$f_h(s;\sigma) := \frac{s}{\sqrt{1+\left(\frac{s}{\sigma}\right)^2}} \quad \text{and} \tag{4.20}$$

$$g_h(s;\sigma) := \frac{1}{\sqrt{1+\left(\frac{s}{\sigma}\right)^2}}. \tag{4.21}$$

The functions are plotted in Figure 4.4.

- The third smoothing term we introduce, penalizes high gradients even less but therefore loses the convexity property. We call it *Lorentzian function* due to its connection to the Lorentzian error norm (see e.g. [Sch04]). It is also mentioned in [Pop05a], where it is



(a) $\phi_h(x;1)$ (b) $f_h(x;1)$

(c) $g_h(x;1)$

Figure 4.4: *Hypersurface minimal* roughness, influence and penalty functions.

called *Hebert and Leahy*. We define the functions as

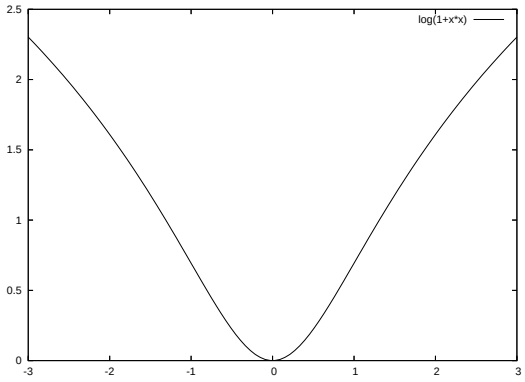
$$\phi_l(s;\sigma) := \sigma^2 \log\left(1+\left(\frac{s}{\sigma}\right)^2\right), \tag{4.22}$$

$$f_l(s;\sigma) := \frac{s}{1+\left(\frac{s}{\sigma}\right)^2}, \tag{4.23}$$

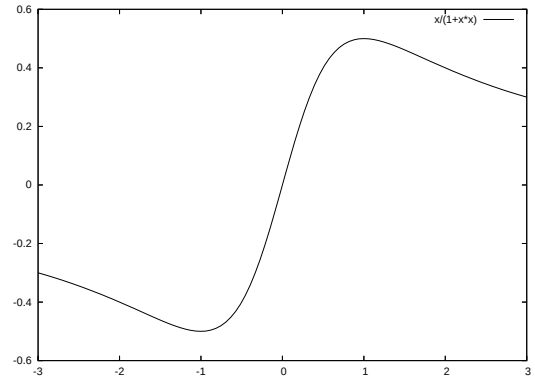
$$g_l(s;\sigma) := \frac{1}{1+\left(\frac{s}{\sigma}\right)^2}, \tag{4.24}$$

and present them in Figure 4.5.

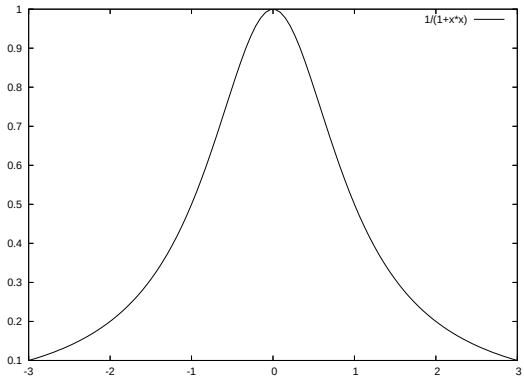
- The fourth regularizer is convex again and is known as the *Green function*. We have



(a) $\phi_l(x;1)$



(b) $f_l(x;1)$



(c) $g_l(x;1)$

Figure 4.5: *Lorentzian* roughness, influence and penalty functions.

adapted it from [Pop05a] and define the regularizer as

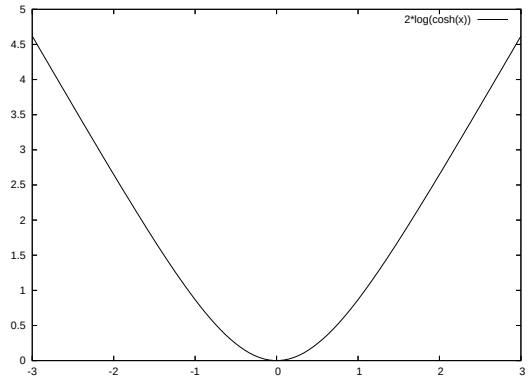
$$\phi_g(s;\sigma) := 2\sigma^2 \log\left(\cosh\left(\frac{s}{\sigma}\right)\right), \tag{4.25}$$

$$f_g(s;\sigma) := \sigma \tanh\left(\frac{s}{\sigma}\right) \quad \text{and} \tag{4.26}$$

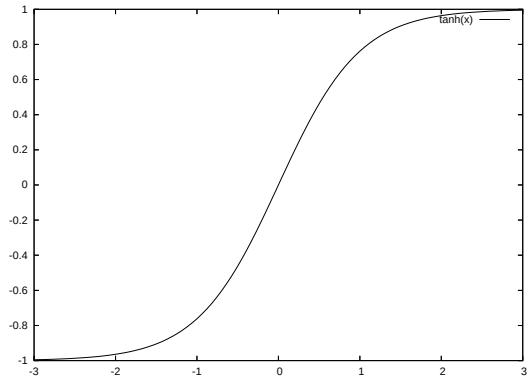
$$g_g(s;\sigma) := \begin{cases} \sigma \tanh\left(\frac{s}{\sigma}\right)/s, & s \neq 0, \\ 1, & s = 0. \end{cases} \tag{4.27}$$

The regularizer is shown in Figure 4.6.

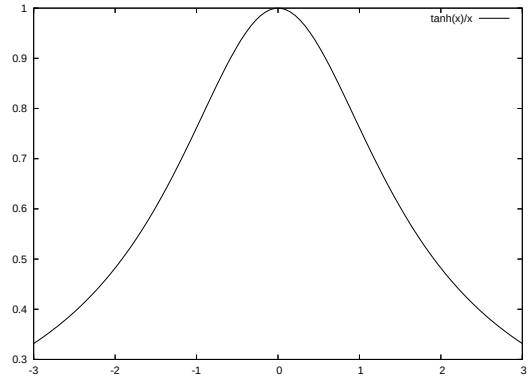
- The last example we present is adapted from Tukey's biwight (see [Bla98] and [Sch04]); we call it *Tukey function*. Instead of being convex, it assigns to each $\|\nabla u\| \geq \sigma$ the same "roughness value". This means that gradients whose length is greater than a certain value,



(a) $\phi_g(x;1)$



(b) $f_g(x;1)$



(c) $g_g(x;1)$

Figure 4.6: *Green* roughness, influence and penalty functions.

are not penalized at all. The corresponding functions are defined as follows:

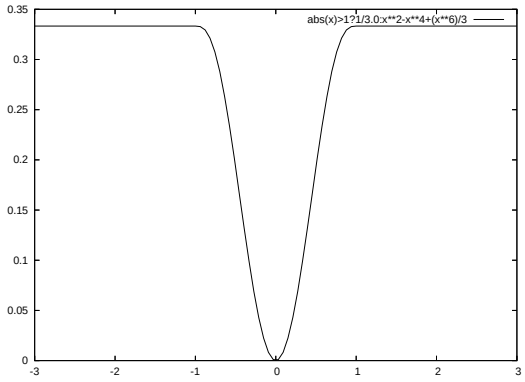
$$\phi_t(s;\sigma) := \begin{cases} \sigma^2\left(\left(\frac{s}{\sigma}\right)^2 - \left(\frac{s}{\sigma}\right)^4 + \frac{1}{3}\left(\frac{s}{\sigma}\right)^6\right), & |s| \le \sigma, \\ \frac{\sigma^2}{3}, & |s| > \sigma, \end{cases} \tag{4.28}$$

$$f_t(s;\sigma) := \begin{cases} s\left(1 - \left(\frac{s}{\sigma}\right)^2\right)^2, & |s| \le \sigma, \\ 0, & |s| > \sigma, \end{cases} \tag{4.29}$$

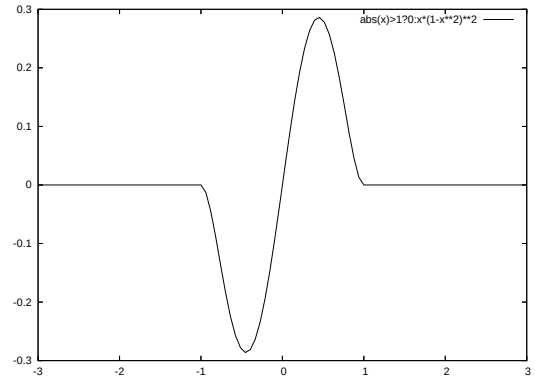
$$g_t(s;\sigma) := \begin{cases} \left(1 - \left(\frac{s}{\sigma}\right)^2\right)^2, & |s| \le \sigma, \\ 0, & |s| > \sigma, \end{cases} \tag{4.30}$$

and are presented in Figure 4.7.

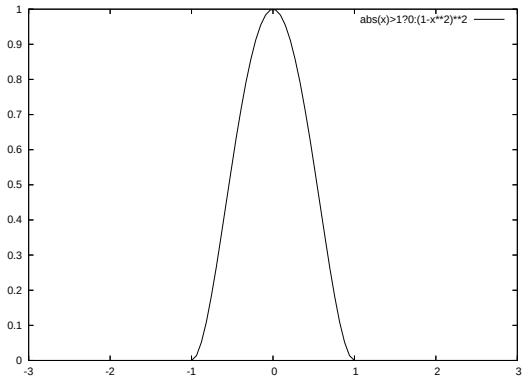
Obviously, there are many more possibilities to define regularization terms, we restrict ourselves on the five functions introduced above. As previously mentioned, convexity of the regu-



(a) $\phi_t(x;1)$



(b) $f_t(x;1)$



(c) $g_t(x;1)$

Figure 4.7: *Tukey* roughness, influence and penalty functions.

larizer is a measurement for existence and uniqueness of a solution. According to [Aub02] it can only be shown for convex smoothing functions that there exist a unique solution. Nevertheless, many solvers work also with non-convex regularization terms.

In Table 4.2 we summarize some important properties of the introduced regularizers.

The question that remains is how to adjust the parameters α and σ? It is clear that for convex roughness functions the reconstructed volume u converges to a constant for $\alpha \to +\infty$ if the solution algorithm is stable and consistent. Consequently, one has to find a compromise between effectiveness of the regularizer and blurring of the reconstructed volume. There are also automatic approaches to adjusting the weighting parameter α, we refer the interested reader to [Tit85]. In this master's thesis we restrict ourselves on heuristic adjustment.

The scale parameter σ should be chosen according to the inserted noise. If the variance of the noise is large then large gradients should be penalized. If the variance is low then large gradients should be treated as edges.

Regularizer	convex?	$\lim_{s\to 0^+} \frac{\phi'(s)}{2s}$	$\lim_{s\to 0^+} \frac{\phi''(s)}{2}$	$\lim_{s\to +\infty} \frac{\phi'(s)}{2s}$	$\lim_{s\to +\infty} \frac{\phi''(s)}{2}$	$\lim_{s\to +\infty} \frac{\phi''(s)}{\phi'(s)/s}$
Quadratic	Yes	1	1	1	1	1
Hypersurface	Yes	1	1	0	0	0
Lorentzian	No	1	1	0	0	0
Green	Yes	1	1	0	0	0
Tukey	No	1	1	0	0	undef.

Table 4.2: Summary of regularizer properties.

Chapter 5

Iterative Solvers for the RART

In this chapter we examine the above mentioned reconstruction techniques. The first one is the ***R**egularized **K**aczmarz **E**xtended with **R**elaxation **P**arameter algorithm* (RKERP)[1] which was introduced by Constantin Popa and Rafal Zdunek.[2] Additionally, we present a simplified version of it which we call ***S**implified **R**egularized **K**aczmarz **E**xtended with **R**elaxation **P**arameter algorithm* (SRKERP).

The third method we discuss is the ***R**egularized **A**lgebraic **R**econstruction **T**echnique algorithm* (RART).[3] The RART algorithm is a new approach and besides the mathematical analysis shown in this thesis it contains some more details to be investigated further.

In the third and last section of this chapter we present the implementation of the introduced methods.

5.1 The RKERP Algorithm

Let us start again with the minimization problem

$$\|Au - b\|^2 = \min, \tag{5.1}$$

where $A \in \mathbb{R}^{M,N}$ is the algebraic reconstruction matrix, $u \in \mathbb{R}^N$ the vector of the volume to be reconstructed and $b \in \mathbb{R}^M$ the vector of all pixels from the projection images. Let $\mathrm{LSS}(A; b)$ be the set of all least-squares solutions of (5.1) and u_{LS} its minimal norm one. By B^t, $\mathrm{R}(B)$ and

[1] The algorithm is also known as *regularized Kaczmarz's algorithm with relaxation parameter*.

[2] See [Pop04], [Pop05a] and [Pop05b].

[3] In this master's thesis the term RART is used for the general idea of regularized algebraic reconstruction techniques as well as for the presented algorithm.

$\mathrm{N}(B)$, we denote the transpose, range and null space of a matrix B, respectively, P_S denotes the orthogonal projection onto a closed convex set $S \subset \mathbb{R}^q$. By $a_i \in \mathbb{R}^N$ and $\alpha_j \in \mathbb{R}^M$, we denote the i-th row and j-th column of A, respectively. Furthermore, we suppose that

$$a_i \neq 0,\ i = 1, \ldots, M, \quad \alpha_j \neq 0,\ j = 1, \ldots, N. \tag{5.2}$$

Moreover, all vectors that appear are considered as column vectors. Obviously, the equality

$$\mathrm{LSS}(A; b) = u_{LS} + \mathrm{N}(A) \tag{5.3}$$

holds. Finally, the decomposition of b as

$$b = b_A + b_A^*, \quad b_A = \mathrm{P}_{\mathrm{R}(A)}(b), \quad b_A^* = \mathrm{P}_{\mathrm{N}(A^t)}(b) \tag{5.4}$$

yields the equivalent formulation of (5.1)

$$u \in \mathrm{LSS}(A; b) \Leftrightarrow Au = b_A \Leftrightarrow A^t Au = A^t b, \tag{5.5}$$

which is the solution of the discrete variational approach without regularization term.[4]

Let us recall Algorithm 1 from page 13. As alluded above the classical Kaczmarz algorithm always converges but not necessarily to a least-squares solution. As was proved in [Nat86] the convergence result is always as follows:

$$\lim_{k \to \infty} u^k = \mathrm{P}_{\mathrm{N}(A)}(u^0) + A^\dagger b, \tag{5.6}$$

where $A^\dagger \in \mathbb{R}^{N,N}$ is a general inverse of A. If

$$b \in \mathrm{R}(A) \Leftrightarrow b = b_A, \tag{5.7}$$

then the limit in (5.6) is a solution of (5.1) and we have

$$u_{LS} = A^\dagger b. \tag{5.8}$$

However, if (5.7) is not fulfilled, i.e. the problem is inconsistent, then the following result holds:

[4]The denotations are taken from [Pop05a] and other papers from C. Popa.

For any limit point $u^*(u^0)$ in (5.6) we have the equality

$$d(u^*(u^0), \mathrm{LSS}(A;b)) \geq \|A^\dagger b_A^*\|, \tag{5.9}$$

where d denotes the Euclidean distance.[5]

As mentioned previously, Kaczmarz's algorithm with relaxation parameter (Algorithm 2) yields some better convergence results. A further improvement was proposed in [Pop05a]. For this, let us use $f_i(\lambda;b;u)$ from (2.17) and $F(\lambda;b;u)$ from (2.18). Furthermore, we define

$$\varphi_j(v) := v - \frac{\langle v, \alpha_j\rangle}{\|\alpha_j\|^2}\alpha_j, \tag{5.10}$$

$$\Phi(v) := (\varphi_1 \circ \ldots \circ \varphi_N)(v) \tag{5.11}$$

and finally, the relaxed versions

$$\varphi_j(\kappa; v) := (1-\kappa)v + \kappa\varphi_j(v) \tag{5.12}$$

and

$$\Phi(\kappa; v) := (\varphi_1 \circ \ldots \circ \varphi_N)(\kappa; v), \tag{5.13}$$

where $\kappa \in (0,2)$ is an additional relaxation parameter. With all these definitions we define the **K**aczmarz **E**xtended with **R**elaxation **P**arameter algorithm as follows:

Algorithm 3 (KERP) *Let $u^0 \in \mathbb{R}^N$, $v^0 = b$; for $k = 0, 1, 2, \ldots$ do*

$$v^{k+1} = \Phi(\kappa; v^k), \tag{5.14}$$

$$b^{k+1} = b - v^{k+1}, \tag{5.15}$$

$$u^{k+1} = F(\lambda; b^{k+1}; u^k). \tag{5.16}$$

Notice that the algorithm does nothing else than to solve the two equations

$$A^t v^{k+1} = 0 \tag{5.17}$$

and

$$Au^{k+1} = b - v^{k+1} \tag{5.18}$$

[5] Result was proved in [Pop04].

with one classical Kaczmarz step, alternately. The idea is to start with a very small right hand side ($b-v$ is near to 0 at the beginning) and then approach a right hand side which is in the range of A. In other words

$$b - v = b_A \Leftrightarrow \text{(5.18) is consistent.}$$

This is true since the limit of v in (5.17), calculated by Kaczmarz's algorithm, converges to

$$v = \mathrm{P}_{\mathrm{N}(A^t)}(b) = b_A^*.$$

Moreover, Popa proved in [Pop98] that for any $u^0 \in \mathbb{R}^N$, any $\lambda, \kappa \in (0, 2)$ and any right hand side b in (5.1), the sequence $(u^k)_{k\geq 0}$ generated by the above algorithm **KERP** converges always to one of the least-squares solutions of (5.1).

Unfortunately, a least-squares solution of (5.1) is not necessarily the sought volume since the projections are distorted by noise. To overcome this problem we have introduced the regularization term in the chapters above. Let us recall the regularized minimization problem

$$\|Au - b\|^2 + \alpha R(u) = min, \tag{5.19}$$

where $R(u)$ is a discrete version of $\int_\Omega \phi(\|\nabla u\|)d\Omega$ and $\phi : \mathbb{R} \mapsto \mathbb{R}$ denotes the roughness term. As shown in (4.8), the following equation is the solution of the minimization problem:

$$A^t Au - \alpha L(u)u = A^t b, \tag{5.20}$$

where $L(u)$ is defined in (4.7). The idea of the **R**egularized **K**aczmarz **E**xtended with **R**elaxation **P**arameter algorithm (**RKERP**) is to adjust the solution after each iteration step by the regularization term. Thus, the regularized KERP (**RKERP**) algorithm can be written as

Algorithm 4 (RKERP) *Let $u^0 \in \mathbb{R}^N$, $v^0 = b$; for $k = 0, 1, 2, \ldots$ do*

$$v^{k+1} = \Phi(\kappa; v^k), \tag{5.21}$$

$$b^{k+1} = b - v^{k+1}, \tag{5.22}$$

$$u^{k+1} = F(\lambda; b^{k+1}; u^k) - \alpha L(u^k)u^k, \tag{5.23}$$

where $F(\lambda; b^{k+1}; u^k)$ and $\Phi(\kappa; v^k)$ are given by (2.18) and (5.13), respectively.[6]

[6]The shown algorithm is a generalized version of the appropriate algorithm introduced in [Pop05a] and [Pop05b]. In our version the derivation is more general and not restricted to Gibbs priors and Markov Random Fields.

5.1.1 The SRKERP Algorithm

It can be seen easily, that the temporal complexity of the RKERP algorithm is at least twice as large as the complexity of the classical Kaczmarz's algorithm since one has to solve two linear systems in one iteration step. Because of that and because of some convergence issues we discuss later, we introduce a simplified version of the RKERP algorithm which we call **S**implified **R**egularized **K**aczmarz **E**xtended with **R**elaxation **P**arameter algorithm (**SRKERP**).[7] The idea is just to omit the extension step. Thus, the simplified RKERP can be written as follows:

Algorithm 5 (SRKERP) *Let $u^0 \in \mathbb{R}^N$; for $k = 0, 1, 2, \ldots$ do*

$$u^{k+1} = F(\lambda; b; u^k) - \alpha L(u^k)u^k, \tag{5.24}$$

with $F(\lambda; b^{k+1}; u^k)$ again given by (2.18).

In the next chapter we show that this simplest approach out of the here introduced ones, yields the best results.

5.2 The RART Algorithm

The final method we introduce is a complete new approach. Let us recall again the solution of the regularized minimization problem

$$A^tAu - \alpha L(u)u = A^tb. \tag{5.25}$$

Now the idea is to introduce a $w \in \mathbb{R}^M$, such that

$$A^tw = -\alpha L(u)u. \tag{5.26}$$

With this introduction we can reformulate (5.25) as

$$A^tAu = A^t(b - w). \tag{5.27}$$

These two equations are solved alternately, (5.26) as it is defined and (5.27) without the transposed of A. This results in the following new **R**egularized **A**lgebraic **R**econstruction **T**echnique

[7] We also could have named the algorithm according to another extension of the classical Kaczmarz's algorithm but since it was developed out of the RKERP algorithm we decided to call it this way.

algorithm (**RART**): Let $u^0 \in \mathbb{R}^N$, $w^0 \in \mathbb{R}^M$, for $k = 0, 1, 2, \ldots$ solve the following two equations:

$$Au^{k+1} = b - w^k \tag{5.28}$$

and

$$A^t w^{k+1} = -\alpha L(u^{k+1})u^{k+1}. \tag{5.29}$$

This is the general idea of the RART algorithm. How equations (5.28) and (5.29) are solved is not specified yet. However, to fit the RART algorithm into the framework of the other shown algorithms, let us introduce the following applications:

$$\psi_j(s; w) = w - \frac{\langle w, \alpha_j \rangle - s_j}{\|\alpha_j\|^2}\alpha_j, \tag{5.30}$$

$$\Psi(s; w) = (\psi_1 \circ \ldots \circ \psi_N)(s; w) \tag{5.31}$$

and the two relaxed versions

$$\psi_j(\kappa; s; w) = (1 - \kappa)w + \kappa\psi_j(s; w) \tag{5.32}$$

and

$$\Psi_j(\kappa; s; w) = (\psi_1 \circ \ldots \circ \psi_N)(\kappa; s; w), \tag{5.33}$$

with $\kappa \in (0, 2)$. With these equations we can write the RART algorithm as follows:

Algorithm 6 (RART) *Let $u^0 \in \mathbb{R}^N$, $w^0 \in \mathbb{R}^M$; for $k = 0, 1, 2, \ldots$ do*

$$w^{k+1} = \Psi(\kappa; -\alpha L(u^k)u^k; w^k), \tag{5.34}$$

$$b^{k+1} = b - w^{k+1}, \tag{5.35}$$

$$u^{k+1} = F(\lambda; b^{k+1}; u^k). \tag{5.36}$$

Notice that a possibility to combine the RART with the KERP algorithm is to initialize w^0 by the right hand side b. This is one topic of further research and is not examined here.

5.2.1 Convergence of the RART Algorithm

In the following mathematical analysis we restrict ourselves on the general idea of the RART algorithm. This means we take a look on what happens if we solve (5.28) and (5.29) exactly.

Furthermore, we only give a brief overview about convergence properties of the method, detailed analysis of the algorithm has to be examined further.

Consistency

To examine consistency properties of the scheme we assume that the sought vectors have already converged to $u^*(u^0)$ and $w^*(w^0)$, respectively. This means that the equations

$$Au^* = b - w^* \tag{5.37}$$

and

$$A^t w^* = -\alpha L(u^*)u^* \tag{5.38}$$

hold and that both vectors remain unchanged in the algorithm. This would be the case if equation (5.38) remains true if we insert u^* or w^* from (5.37) into it and vice versa. In the following we take a brief look into all four cases:

- Insertion of w^* from (5.37) into (5.38).
  The insertion of w^* from (5.37) into (5.38) yields

  $$A^t(b - Au^*) = -\alpha L(u^*)u^*,$$

  which is the solution of the regularized minimization problem. So this case holds.

- Insertion of u^* from (5.37) into (5.38).
  The insertion of u^* from (5.37) into (5.38) yields

  $$A^t w^* = -\alpha L(A^\dagger(b - w^*))A^\dagger(b - w^*).$$

  In this case the problem is that the operator L is not necessarily linear. Even if we assume linearity the equality depends on consistency properties of the minimization problem. Recapitulating, this case is not always true.

- Insertion of w^* from (5.38) into (5.37).
  Inserting w^* from (5.38) into (5.37) yields

  $$Au^* = b + \alpha(A^t)^\dagger L(u^*)u^*.$$

  This equation is true if the minimization problem is consistent.

- Insertion of u^* from (5.38) into (5.37).
  Finally, inserting u^* from (5.38) into (5.37) with

$$u^* = -\frac{1}{\alpha} L^{-1}(u^*) A^t w^*$$

  and the assumption that L is invertible, yields

$$-\frac{1}{\alpha} A L^{-1}(u^*) A^t w^* = b - w^*.$$

  Applying (5.26) yields

$$\frac{1}{\alpha} \alpha A L^{-1}(u^*) L(u^*) u^* = b - w^*,$$

  which means that the original equation (5.37) remains unchanged. So this case holds.

As one can see it is difficult to prove consistency of the RART algorithm even in the consistent case of the minimization problem. Further research has to be done with respect to this topic.

Stability

In the following we prove that the RART algorithm is conditionally stable with respect to the growth of the introduced variable w in an arbitrary vector norm $\|x\|_*$, with $x \in \mathbb{R}^N$ and its induced matrix norm $\|A\|_*$ with $A \in \mathbb{R}^{\cdot,N}$.

Theorem 1 (Stability of the RART Algorithm) *The RART algorithm is stable if*

$$\alpha \|L(u)\|_* < \frac{1}{\|(A^t)^\dagger\|_* \|A^\dagger\|_*} \tag{5.39}$$

holds.

For the proof let us recall the two iteration steps of the RART algorithm

$$A u^{k+1} = b - w^k \tag{5.40}$$

and

$$A^t w^{k+1} = -\alpha L(u^{k+1}) u^{k+1}. \tag{5.41}$$

If w^k is bounded for every $k \in \mathbb{N}$ then the right hand side in (5.40) is also bounded and thus the method is stable.[8] Moreover, we prove stability for a linear regularizer $M \in \mathbb{R}^{N,N}$. Nevertheless,

[8] We assume that an appropriate solver is used.

stability also holds if a nonlinear regularizer $L(u)$, that is bounded by

$$\alpha\|L(u)\|_* \leq \|M\|_* \tag{5.42}$$

is applied.

Proof 1 (Boundedness of w) *Assume that the k-th iteration step is already calculated and w^k is given. Then inserting*

$$u^{k+1} = A^\dagger b - A^\dagger w^k \tag{5.43}$$

from (5.40) *into the linearized version of* (5.41) *yields*

$$w^{k+1} = \underbrace{(A^t)^\dagger M A^\dagger}_{=:D} b - \underbrace{(A^t)^\dagger M A^\dagger}_{=D} w^k. \tag{5.44}$$

By applying the above introduced norm on both sides we get

$$\|w^{k+1}\|_* = \|Db - Dw^k\|_*. \tag{5.45}$$

The usage of the triangle inequality and the fact that $\|.\|_*$ is a natural norm[9] *yields*

$$\begin{aligned} \|w^{k+1}\|_* &\leq \|Db\|_* + \|Dw^k\|_* \\ &\leq \|D\|_*\|b\|_* + \|D\|_*\|w^k\|_*. \end{aligned} \tag{5.46}$$

By recursive insertion we get

$$\begin{aligned} \|w^{k+1}\|_* &\leq \|D\|_*\|b\|_* + \|D\|_*\|w^k\|_* \\ &\leq \|D\|_*\|b\|_* + \|D\|_*^2\|b\|_* + \|D\|_*^2\|w^{k-1}\|_* \\ &\leq \|D\|_*\|b\|_* + \|D\|_*^2\|b\|_* + \ldots + \|D\|_*^{k+1}\|b\|_* + \|D\|_*^{k+1}\|w^0\|_* \\ &= \sum_{i=1}^{k+1} \|D\|_*^i\|b\|_* + \|D\|_*^{k+1}\|w^0\|_*. \end{aligned} \tag{5.47}$$

[9] For detailed properties of vector and corresponding matrix norms we refer the interested reader to [Bur01, page 418 et sqq.]

After some index adjustment and application of the geometric series we have[10]

$$\begin{aligned}\|w^k\|_* &\leq \sum_{i=1}^{k} \|D\|_*^i \|b\|_* + \|D\|_*^k \|w^0\|_* \\ &= \sum_{i=0}^{k} \|D\|_*^i \|b\|_* - \|b\|_* + \|D\|_*^k \|w^0\|_* \\ &= \frac{\|D\|_*^{k+1} - \|D\|_*}{\|D\|_* - 1} \|b\|_* + \|D\|_*^k \|w^0\|_*. \end{aligned} \tag{5.48}$$

If $\|D\|_* < 1$ *the limit for* $k \to \infty$ *becomes*

$$\|w^*(w^0)\|_* = \|w^\infty\|_* \leq \frac{\|D\|_*}{1 - \|D\|_*} \|b\|_*. \tag{5.49}$$

According to the multiplicative triangle inequality for matrix norms the condition $\|D\|_* < 1$ *is fulfilled if*

$$\|D\|_* = \|(A^t)^\dagger M A^\dagger\|_* \leq \|(A^t)^\dagger\|_* \|M\|_* \|A^\dagger\|_* < 1, \tag{5.50}$$

which is the same as

$$\|M\|_* < \|(A^t)^\dagger\|_*^{-1} \|A^\dagger\|_*^{-1}, \tag{5.51}$$

which finally marks the end of the proof.

□

As one can see, the method is stable if the weight of the regularizing term is small enough. This makes sense since a smoothing term that is too large would fudge anyway. Unfortunately, we were not able to derive sufficient consistency properties of the method to present a meaningful conclusion about its convergence behavior.

5.3 Implementation of the Algorithms

In the last part of this chapter, we briefly describe the implementation of the introduced algorithms. The new methods are embedded in the C++ framework of Sebastian Sauer were some well-known algebraic reconstruction methods are already implemented. Thus the setting up of the reconstruction matrix A is already implemented and is based on the extended splatting algorithm introduced in [Mue99].

[10] Here we assume that $\|D\|_* \neq 1$. Otherwise $\frac{\|D\|_*^{k+1}-1}{\|D\|_*-1}$ becomes $(k+1)$.

INPUT	The projection images $b \in \mathbb{R}^M$.
OUTPUT	The iteration volume $u_{\text{iter}} \in \mathbb{R}^N$; the helper image $w \in \mathbb{R}^M$.
Step 1	Set $u_{\text{iter}} = 0$.
Step 2	Set $w = b$.

Table 5.1: Initialization (RKERP).

OUTPUT	The iteration volume $u_{\text{iter}} \in \mathbb{R}^N$; the helper image $w \in \mathbb{R}^M$.
Step 1	Set $u_{\text{iter}} = 0$.
Step 2	Set $w = 0$.

Table 5.2: Initialization (RART).

In the following we use a pseudo-code to describe the implementations of the algorithms. The parameters are only mentioned in the modules where they are really needed and are not handed over in the several methods. The RKERP and SRKERP algorithms use the same implementation, the difference is that the Block-Iteration-Regularization step is left out in the latter one. Parameter names are defined globally in the pseudo-code. The numbers M and N are again the number of projection rays and unknown voxel values, respectively.

Table 5.1 and Table 5.2 show the initialization of the two algorithms.

Table 5.3 describes the main-loop of the methods.

The Pre-Reconstruction steps are shown in Table 5.4 and Table 5.5.

The Post-Reconstruction steps are shown in Table 5.6 and Table 5.7.

The Regularization step, which is used in both methods, is described in Table 5.8.

The Block-Iteration-Reconstruction step of the RKERP method can be found in Table 5.9. There, also the helper volumes β and γ are prepared, which are used in the second block-iteration.

INPUT	Number of iterations n_{iter}; set of projection indices P, randomly ordered.
Step 1	For $i = 1 \ldots n_{\text{iter}}$ do Steps 2-5.
Step 2	Do the `Pre-Reconstruction step`.
Step 3	For every $p \in \mathrm{P}$ do the `Block-Iteration-Reconstruction step`.
Step 4	Do the `Post-Reconstruction step`.
Step 5	For every $p \in P$ do the `Block-Iteration-Regularization step`.

Table 5.3: Main-loop of the Algorithms.

OUTPUT	The helper volumes $\beta \in \mathbb{R}^N$ and $\gamma \in \mathbb{R}^N$.
Step 1	Set $\beta = 0$.
Step 2	Set $\gamma = 0$.
Step 3	If a regularizer is used do the `Regularization step`.

Table 5.4: Pre-Reconstruction step (RKERP).

OUTPUT	The helper volumes $\beta \in \mathbb{R}^N$ and $\gamma \in \mathbb{R}^N$.
Step 1	Set $\beta = 0$.
Step 2	Set $\gamma = 0$.

Table 5.5: Pre-Reconstruction step (RART).

The similar version of the Block-Iteration-Reconstruction step of the RART algorithm is shown in Table 5.10.

Finally, Table 5.11 and Table 5.12 show the Block-Iteration-Regularization step for the RKERP and RART algorithm, respectively. Notice that the implementation of the Block-Iteration-Regularization step is a little bit different from the classical Kaczmarz iteration. This comes from the fact that it is very difficult to get a complete column of the matrix A at one point in time. The shown implementation is similar to SIRT (**S**imultaneous **I**terative **R**econstruction **T**echnique) or SART (**S**imultaneous **A**lgebraic **R**econstruction **T**echnique) algorithms introduced e.g. in [Kak01] or [Eps03].

The description of the implementation concludes the theoretical part of this master's thesis. What follows is the evaluation of the introduced methods.

INPUT	The iteration volume $u_{\text{iter}} \in \mathbb{R}^N$ and the regularization volume $u_{\text{reg}} \in \mathbb{R}^N$; the weighting parameter α.
OUTPUT	The iteration volume $u_{\text{iter}} \in \mathbb{R}^N$.
Step 1	If a regularizer is used set $u_{\text{iter}} = u_{\text{iter}} - \alpha u_{\text{reg}}$.

Table 5.6: Post-Reconstruction step (RKERP).

Step 1	If a regularizer is used do the `Regularization step`.

Table 5.7: Post-Reconstruction step (RKERP).

INPUT	The iteration volume $u_{\text{iter}} \in \mathbb{R}^N$; the regularizing function L.
OUTPUT	The regularization volume $u_{\text{reg}} \in \mathbb{R}^N$.
Step 1	Set $u_{\text{reg}} = L(u_{\text{iter}})u_{\text{iter}}$.

Table 5.8: Regularization step.

INPUT	The reconstruction matrix $A \in \mathbb{R}^{M,N}$ with its elements $a_{i,j}$; the index set I with corresponding pixels to the actual image which is indexed by p; the weighting parameters λ, κ; the iteration volume $u_{\text{iter}} \in \mathbb{R}^N$; the helper volumes $\beta, \gamma \in \mathbb{R}^N$; the projection images $b \in \mathbb{R}^M$; the helper image $w \in \mathbb{R}^M$.
OUTPUT	The iteration volume $u_{\text{iter}} \in \mathbb{R}^N$; the helper volumes $\beta, \gamma \in \mathbb{R}^N$.
Step 1	For every $i \in \mathrm{I}$ do Steps 2-7.
Step 2	For every j, where $1 \leq j \leq N \wedge a_{i,j} \neq 0$ do Steps 3-7.
Step 3	Set $\beta_j = \beta_j + a_{i,j}^2$.
Step 4	If $\kappa > 0$ do Steps 5-6.
Step 5	Set $u_{\text{iter},j} = u_{\text{iter},j} + \lambda \frac{b_i - \langle a_i, u_{\text{iter}} \rangle - w_i}{\|a_i\|^2} a_{i,j}$.
Step 6	Set $\gamma_j = \gamma_j + a_{i,j} w_i$.
	Else set $u_{\text{iter},j} = u_{\text{iter},j} + \lambda \frac{b_i - \langle a_i, u_{\text{iter}} \rangle}{\|a_i\|^2} a_{i,j}$.
Step 7	If $u_{\text{iter},j} < 0$ set $u_{\text{iter},j} = 0$.

Table 5.9: Block-Iteration-Reconstruction step (RKERP).

INPUT	The reconstruction matrix $A \in \mathbb{R}^{M,N}$ with its elements $a_{i,j}$; the index set I with corresponding pixels to the actual image which is indexed by p; the weighting parameters λ, κ, α; the iteration volume $u_{\text{iter}} \in \mathbb{R}^N$; the helper volumes $\beta, \gamma \in \mathbb{R}^N$; the projection images $b \in \mathbb{R}^M$; the helper image $w \in \mathbb{R}^M$.
OUTPUT	The iteration volume $u_{\text{iter}} \in \mathbb{R}^N$; the helper volumes $\beta, \gamma \in \mathbb{R}^N$.
Step 1	For every $i \in \mathrm{I}$ do Steps 2-6.
Step 2	For every j, where $1 \leq j \leq N \wedge a_{i,j} \neq 0$ do Steps 3-6.
Step 3	Set $\beta_j = \beta_j + a_{i,j}^2$.
Step 4	Set $u_{\text{iter},j} = u_{\text{iter},j} + \lambda \frac{b_i - \langle a_i, u_{\text{iter}} \rangle - \alpha w_i}{\|a_i\|^2} a_{i,j}$.
Step 5	If $u_{\text{iter},j} < 0$ set $u_{\text{iter},j} = 0$.
Step 6	Set $\gamma_j = \gamma_j + a_{i,j} w_i$.

Table 5.10: Block-Iteration-Reconstruction step (RART).

INPUT	The reconstruction matrix $A \in \mathbb{R}^{M,N}$ with its elements $a_{i,j}$; the number of non-zero values in the i-th row of A ζ_i; the index set I with corresponding pixels to the actual image which is indexed by p; the weighting parameter κ; the helper volumes $\beta, \gamma \in \mathbb{R}^N$; the helper image $w \in \mathbb{R}^M$.
OUTPUT	The helper image $w \in \mathbb{R}^M$.
Step 1	If $\kappa > 0$ do Steps 2-5.
Step 2	For every $i \in \mathrm{I}$ do Steps 3-5.
Step 3	For every j, where $1 \leq j \leq N \wedge a_{i,j} \neq 0$ do Step 4.
Step 4	If $\beta_j > 0$ set $w_i = w_i - \kappa \frac{\gamma_j}{\zeta_i \beta_j} a_{i,j}$.
Step 5	If $w_i < 0$ set $w_i = 0$.

Table 5.11: Block-Iteration-Regularization step (RKERP).

INPUT	The reconstruction matrix $A \in \mathbb{R}^{M,N}$ with its elements $a_{i,j}$; the number of non-zero values in the i-th row of A ζ_i; the index set I with corresponding pixels to the actual image which is indexed by p; the weighting parameter κ; the helper volumes $\beta, \gamma \in \mathbb{R}^N$; the regularization volume $u_{\text{reg}} \in \mathbb{R}^N$; the helper image $w \in \mathbb{R}^M$.
OUTPUT	The helper image $w \in \mathbb{R}^M$.
Step 1	If $\kappa > 0$ do Steps 2-5.
Step 2	For every $i \in \mathrm{I}$ do Steps 3-5.
Step 3	For every j, where $1 \leq j \leq N \wedge a_{i,j} \neq 0$ do Step 4.
Step 4	If $\beta_j > 0$ set $w_i = w_i + \kappa \frac{u_{\text{reg},j} - \gamma_j}{\zeta_i \beta_j} a_{i,j}$.
Step 5	If $w_i < 0$ set $w_i = 0$.

Table 5.12: Block-Iteration-Regularization step (RART).

Chapter 6

Evaluation and Results

In this chapter we show some results of the application of the introduced algorithms. We show that the SRKERP algorithm, which is the simplest one, yields the best results, while the RKERP algorithm, which is the most consistent one in a mathematical point of view, yields the worst results due to its slow convergence properties. The fact that we use an adapted version of the RKERP method, which is more related to the SIRT algorithm,[1] amplifies this characteristic.

In the first part of this chapter we test the algorithms on the three-dimensional Shepp-Logan phantom. In the second part the reconstruction of a real data set is examined.

We forgo the comparison of algebraic and filtered back-projection methods since the main foci of the thesis are the regularized ART methods which are related to the standard algebraic schemes. Additionally, it is well known that for many projections, acquired within a range of $> 180°$, the FBP methods yield better results, while ART methods lead to better images if only a few projections are used.[2]

6.1 Reconstruction of the Shepp-Logan Phantom

The Shepp-Logan phantom is an analytical model of the human brain. It is commonly used for testing new reconstruction algorithms. The three-dimensional phantom consists of several filled ellipsoids which are defined in Table 6.1. The phantom is defined in a unit cube where $(x, y, z)^t \in [-1, 1]^3$. The rotation angle describes the rotation of the ellipsoid around the z-axis. If the ellipsoids overlap the gray values are added up, so the overall gray value of the phantom is

[1] According to [Kak01, page 284] the SIRT algorithm leads to better looking images at the expense of slower convergence.

[2] Cf. [Kak01] and [Eps03].

Ellipsoid	Coordinates of the Center (x, y, z)	Axis Lengths (A, B, C)	Rotation Angle β (deg)	Gray Level ρ
a	$(0, 0, 0)$	$(0.69, 0.92, 0.9)$	0	2.0
b	$(0, 0, 0)$	$(0.6624, 0.874, 0.88)$	0	-0.98
c	$(-0.22, 0, -0.25)$	$(0.41, 0.16, 0.21)$	108	-0.02
d	$(0.22, 0, -0.25)$	$(0.31, 0.11, 0.22)$	72	-0.02
e	$(0, 0.35, -0.25)$	$(0.21, 0.25, 0.5)$	0	0.02
f	$(0, 0.1, -0.25)$	$(0.046, 0.046, 0.046)$	0	0.02
g	$(-0.08, -0.65, -0.25)$	$(0.046, 0.023, 0.02)$	0	0.01
h	$(0.06, -0.65, -0.25)$	$(0.046, 0.023, 0.02)$	90	0.01
i	$(0.06, -0.105, 0.625)$	$(0.056, 0.04, 0.1)$	90	0.02
j	$(0, 0.1, 0.625)$	$(0.056, 0.056, 0.1)$	0	-0.02

Table 6.1: Parameters of the 3D-Shepp-Logan phantom according to [Kak01].

$\widetilde{\rho} \in [0, 2]$.

In the following all images are shown according to a semi-linear transfer function introduced in Figure 6.1. Assumed that the gray values are in $[0, 2]$, they are mapped onto the same interval. The function has two degrees of freedom, center and width. This transfer function is used to show e.g. a specific type of tissue or bone in the three-dimensional image and is commonly used in medical imaging.

Figure 6.2 shows three slices of the Shepp-Logan phantom for $x = 0$, $y = 0.1$ and $z = -0.25$ with a center of $C = 1.04$ and a width of $W = 0.11$.

6.1.1 Concept of Evaluation

The concept of evaluating the introduced algorithms is as follows: The model of the Shepp-Logan phantom is used for creating a discrete voxel set $\widetilde{u} \in \mathbb{R}^N$. This set is then forward-projected onto $\widetilde{b} \in \mathbb{R}^M$ according to a set of projection matrices and its corresponding reconstruction matrix A described in the previous chapter. From a medical imaging point of view it would be better to forward-project the phantom analytically. However, the numerical forward projection has the advantage of resulting in a consistent linear system. Optionally, these projection images are then distorted by multiplicative white Gaussian noise as

$$b_i = (1 + n_{\sigma,i})\widetilde{b}_i, \quad i = 1, \ldots, M, \tag{6.1}$$

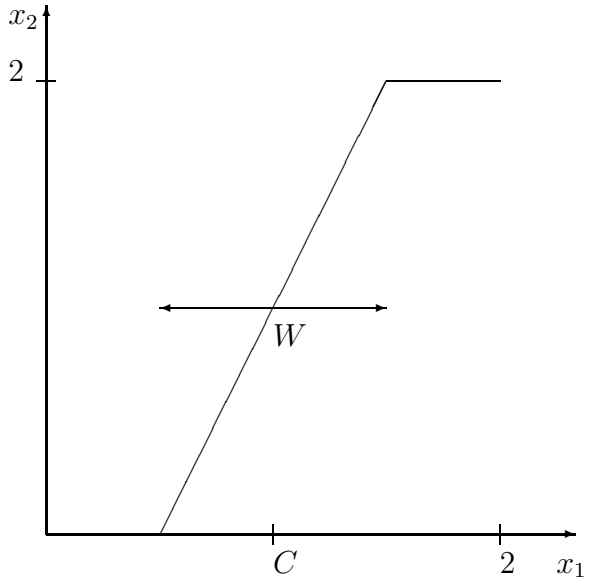


Figure 6.1: Center/width transfer function.

with $b \in \mathbb{R}^M$ and $n_{\sigma,i} \in \mathbb{R}$ a Gaussian distributed random value with zero mean and variance σ. Most likely, this step leads to a non-consistent linear system. This pixel set is then back-projected, which yields the reconstruction volume $u \in \mathbb{R}^N$.

For comparison of the reconstructed volume u and the model $\widetilde{u}$ we use two different measurements. The first one,

$$\mathrm{re}(u;\widetilde{u}) := \frac{\sum_{j=1}^{N} |u_j - \widetilde{u}_j|}{\sum_{j=1}^{N} |\widetilde{u}_j|} \tag{6.2}$$

(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.2: Slices of the Shepp-Logan phantom in x-, y- and z-direction.

is the relative error, which is a very similar measurement to the signal-to-noise ratio, and the second one

$$\mathrm{cc}(u;\widetilde{u}) := \frac{\sum_{j=1}^{N}(\widetilde{u}_j - \mu(\widetilde{u}))(u_j - \mu(u))}{\sqrt{\sum_{j=1}^{N}(\widetilde{u}_j - \mu(\widetilde{u}))^2 \sum_{j=1}^{N}(u_j - \mu(u))^2}} \tag{6.3}$$

is called the correlation coefficient, where

$$\mu(u) := \frac{1}{N}\sum_{j=1}^{N} u_j \tag{6.4}$$

is the mean value of u. Obviously, the results are good if the correlation coefficient is near to one and the relative error is near to zero. Additionally, we examine the reconstruction error

$$\mathrm{rn}_{\mathrm{RKERP}}(u; b-v) := \|Au - b + v\|^2, \tag{6.5}$$

$$\mathrm{rn}_{\mathrm{SRKERP}}(u; b) := \|Au - b\|^2 \quad \text{and} \tag{6.6}$$

$$\mathrm{rn}_{\mathrm{RART}}(u; b-w) := \|Au - b + w\|^2 \tag{6.7}$$

for the RKERP, SRKERP and RART algorithms, respectively, and the variance of u

$$\mathrm{va}(u) := \frac{1}{N}\sum_{j=1}^{N}(u_j - \mu(u))^2. \tag{6.8}$$

Types of Evaluation

The evaluation of the variance, correlation coefficient, relative error and reconstruction error is performed according to the following points:

- Comparison of the SRKERP algorithm in the non-distorted case with respect to the number of iterations and the Quadratic, Hypersurface, Lorentzian, Green and Tukey regularizers.

- Comparison of the SRKERP algorithm in the distorted case with respect to the number of iterations and the Quadratic, Hypersurface, Lorentzian, Green and Tukey regularizers.

- Comparison of the RART algorithm in the non-distorted case with respect to the number of iterations and the Quadratic, Hypersurface, Lorentzian, Green and Tukey regularizers.

- Comparison of the RART algorithm in the distorted case with respect to the number of iterations and the Quadratic, Hypersurface, Lorentzian, Green and Tukey regularizers.
- Convergence properties of the RKERP algorithm.
- Influence of the weighting parameter α on the SRKERP algorithm with respect to the Quadratic, Hypersurface and Tukey regularizers.
- Influence of the scale parameter σ on the SRKERP algorithm with respect to the Quadratic, Hypersurface and Tukey regularizers.

Parameters of Evaluation

The parameters of the reconstructions are as follows if not mentioned otherwise:

- The size of the reconstruction volume is $128 \times 128 \times 128$; so $N = 2097152$.
- The number of projections is 33. They are acquired within a range of $195°$ in an equidistant way. The size of the projections is 256×256; so $M = 2162688$.
- The weighting parameter for the reconstruction step is $\lambda = 0.1$.
- The weighting parameter for the second block-iteration in the RKERP and RART algorithm is chosen as $\kappa = 1$.
- The weighting parameter for the regularization term is $\alpha = 0.2$ for the SRKERP algorithm and $\alpha = 50000$ for the RART algorithm. The helper image w in the RART algorithm is very small in our implementation; so we chose a large weighting parameter to see the effect that this term has on the reconstruction volume.
- The scale parameter in the regularization functions is $\sigma = 0.1$.

6.1.2 SRKERP in the Non-Distorted Case

In this section we examine the SRKERP algorithm in the non-distorted case with respect to the number of iterations and the above introduced regularizers. Additionally, we compare the results with the classical Kaczmarz algorithm, i.e. where no regularizer is used.

The values of the correlation coefficient $\mathrm{cc}(u; \widetilde{u})$ is shown in Table 6.2.

Table 6.3 shows the relative error $\mathrm{re}(u; \widetilde{u})$.

Table 6.4 explains the variance $\mathrm{va}(u)$. Note that the variance of the Shepp-Logan phantom is

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.974682	0.974682	0.974682	0.974682	0.974682	0.974682
2	0.987632	0.987065	0.987564	0.987635	0.987564	0.98764
3	0.992202	0.991244	0.99212	0.992217	0.992121	0.992212
4	0.994543	0.993313	0.99447	0.994576	0.994474	0.994556
5	0.995909	0.994472	0.995852	0.995958	0.995857	0.995925
6	0.996796	0.995196	0.996756	0.996858	0.996763	0.996815
7	0.997411	0.995685	0.99739	0.997487	0.997398	0.997434
8	0.997849	0.996014	0.997845	0.997937	0.997854	0.997875

Table 6.2: Correlation coefficient of SRKERP in the non-distorted case.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.129989	0.129989	0.129989	0.129989	0.129989	0.129989
2	0.08054	0.0830453	0.0812201	0.0808671	0.0812447	0.0806932
3	0.0603865	0.0647391	0.0610317	0.0604822	0.0610463	0.060376
4	0.0484755	0.0543975	0.0489341	0.0482704	0.0489336	0.0482751
5	0.0407926	0.0478756	0.0409142	0.0402104	0.040895	0.040354
6	0.0362887	0.0445145	0.0361353	0.0353952	0.0361005	0.035648
7	0.0324463	0.0416413	0.0319943	0.0312466	0.0319443	0.0316138
8	0.029486	0.0396119	0.028736	0.0279818	0.0286722	0.0284572

Table 6.3: Relative error of SRKERP in the non-distorted case.

$\mathrm{va}(\widetilde{u}) = 0.271718$.

Finally, the reconstruction error $\mathrm{rn}(u; b)$ can be seen in Table 6.5.

All these tables are summarized in Figure 6.3. At the correlation coefficient one can see that most of the regularized versions are better than the classical Kaczmarz reconstruction. Only the Hypersurface reconstruction is worse but this may be caused by wrong parameters. The Quadratic regularized reconstruction is worst. The same result holds with respect to the relative error. Here even the Hypersurface regularized reconstruction is better than the classical one. Obviously, the Quadratic version has the lowest variance. The only reconstructed volume that has a larger variance than the classical result is the Tukey regularized volume. After the final iteration the reconstruction error of the Lorentzian and the Tukey version is even better than the error of the non-regularized volume. Here, the introduction of a-priori knowledge into the reconstruction pays off.

Recapitulating, the usage of regularizers can improve the classical ARTs a little bit if the right parameters are chosen.

Slices of the reconstructed volume in the non-regularized version and the Lorentzian version with a center of $C = 0.95$ and a width of $W = 0.15$ are shown in Figure 6.4 and Figure 6.5,

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.227205	0.227205	0.227205	0.227205	0.227205	0.227205
2	0.244687	0.243695	0.244461	0.244613	0.244452	0.244679
3	0.252277	0.250381	0.251946	0.252202	0.251936	0.25228
4	0.256651	0.253892	0.256234	0.256581	0.256225	0.256665
5	0.259768	0.256244	0.259292	0.259713	0.259283	0.259793
6	0.261882	0.257673	0.261353	0.261838	0.261346	0.261912
7	0.263368	0.258518	0.262787	0.263332	0.262781	0.263401
8	0.264659	0.259259	0.264043	0.264634	0.264038	0.264696

Table 6.4: Variance of SRKERP in the non-distorted case.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	9.20422E+08	9.20422E+08	9.20422E+08	9.20422E+08	9.20422E+08	9.20422E+08
2	1.67389E+07	1.67389E+07	1.67389E+07	1.67389E+07	1.67389E+07	1.67389E+07
3	5.98869E+06	6.38186E+06	6.06045E+06	6.00480E+06	6.06281E+06	5.98614E+06
4	3.10534E+06	3.68258E+06	3.18978E+06	3.11996E+06	3.19188E+06	3.10300E+06
5	1.83337E+06	2.49709E+06	1.91150E+06	1.84110E+06	1.91283E+06	1.82991E+06
6	1.18901E+06	1.90093E+06	1.25983E+06	1.19264E+06	1.26058E+06	1.18583E+06
7	8.24366E+05	1.56926E+06	8.88383E+05	8.25325E+05	8.88703E+05	8.21815E+05
8	5.89258E+05	1.34939E+06	6.46282E+05	5.88128E+05	6.46311E+05	5.87020E+05

Table 6.5: Reconstruction error of SRKERP in the non-distorted case.

respectively.

6.1.3 SRKERP in the Distorted Case

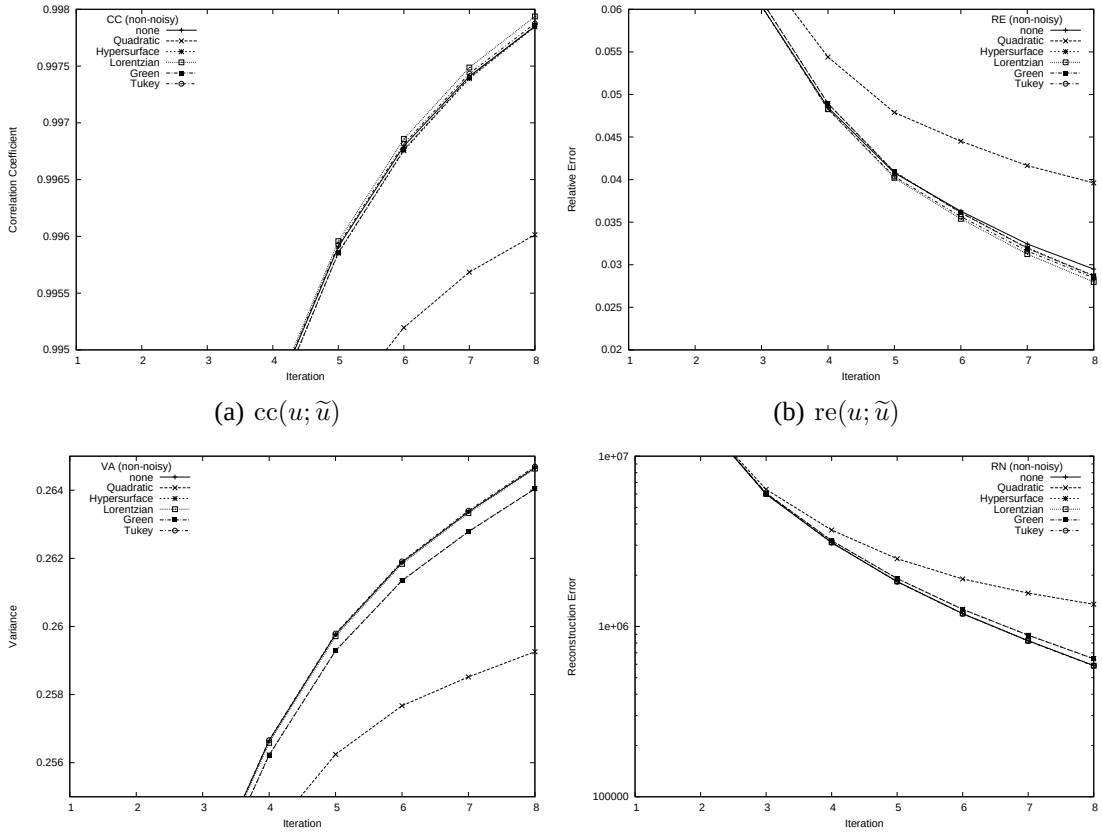
In the following we take a look at the SRKERP algorithm in the distorted case with respect to the number of iterations and the above introduced regularizers. As in the previous case, we compare the results with the classical Kaczmarz algorithm. For the multiplicative Gaussian noise a variance of $\sigma = 0.005$ is used.

The correlation coefficient $\mathrm{cc}(u; \widetilde{u})$ for eight iterations and the regularizers is shown in Table 6.6.

Table 6.7 shows the relative error $\mathrm{re}(u; \widetilde{u})$ for the distorted case.

Table 6.8 and Table 6.9 show the variance $\mathrm{va}(u)$ and the reconstruction error $\mathrm{rn}(u; b)$, respectively.

The summary of these tables is illustrated in Figure 6.6. With respect to the correlation coefficient, all regularized reconstructions are better than the classical Kaczmarz version. An exception is the Quadratic regularized one. The same result hold for the relative error. Here the difference is even more clear. For this set of parameters the Lorentzian version is best. The



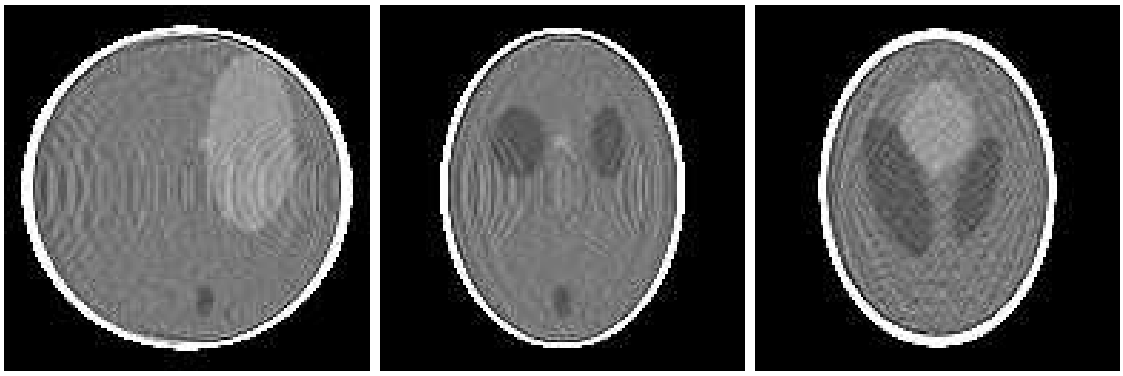
(a) $\mathrm{cc}(u;\widetilde{u})$ (b) $\mathrm{re}(u;\widetilde{u})$

(c) $\mathrm{va}(u)$ (d) $\mathrm{rn}(u;b)$

Figure 6.3: Results of SRKERP in the non-distorted case.

variance of the Lorentzian and the Tukey regularized volumes is nearly as large as the classical reconstruction. The variance of Hypersurface and Green, whose roughness measurements are both convex, is visibly lower. The Quadratic version has the lowest variance. The reconstruction error is lowest for the Kaczmarz reconstruction.

The effect of using a regularizer is even more visible in the non-consistent case as one can see in Figure 6.7 and Figure 6.8, respectively. For the images a center of $C = 0.95$ and a width of $W = 0.18$ is used. Notice the circles in the z-slices. These may yield from the fact that only the pixels of the projection images, which are all acquired around the z-axis, are distorted. In a real environment there are additional geometry errors which are caused e.g. from C-Arm oscillations, so the error is not as systematical as in our simulation.



(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.4: Slices of the reconstructed Shepp-Logan phantom, non-distorted, non-regularized, SRKERP.

(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.5: Slices of the reconstructed Shepp-Logan phantom, non-distorted, Lorentzian, SRK-ERP.

6.1.4 RART in the Non-Distorted Case

In this section we examine the RART algorithm in the non-distorted case with respect to the number of iterations and the regularizers. Additionally, we compare the results with the classical Kaczmarz algorithm, i.e. where no regularizer is used.

The values of the correlation coefficient $\mathrm{cc}(u; \widetilde{u})$ is shown in Table 6.10.

The relative error $\mathrm{cc}(u; \widetilde{u})$ can be examined in Table 6.11.

Table 6.12 shows the variance $\mathrm{va}(u)$ of the RART reconstructions in the non-distorted case.

Finally, the reconstruction error $\mathrm{rn}(u; b - w)$ is reflected in Table 6.13.

All these tables are visualized in Figure 6.9. The correlation coefficient in all six cases is nearly identical. The relative error of the regularized reconstructions is a little bit better than in the Kaczmarz reconstructed result. The variance in the classical version is the largest and

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.974661	0.974667	0.974663	0.974665	0.974665	0.974664
2	0.987586	0.987027	0.987524	0.987593	0.987526	0.987599
3	0.992129	0.991187	0.992058	0.992154	0.992062	0.992149
4	0.99444	0.993237	0.994388	0.994491	0.994396	0.994471
5	0.995777	0.994381	0.995752	0.995855	0.995762	0.995819
6	0.996631	0.99509	0.996638	0.996737	0.996651	0.996687
7	0.997217	0.99557	0.99726	0.997352	0.997274	0.997288
8	0.997625	0.995888	0.997702	0.997788	0.997718	0.997712

Table 6.6: Correlation coefficient of SRKERP in the distorted case.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.130764	0.130682	0.130722	0.130745	0.130737	0.130709
2	0.0829329	0.085222	0.0834743	0.0831484	0.0834372	0.0829148
3	0.0641527	0.0680093	0.0644182	0.0638889	0.0643457	0.0637428
4	0.0534356	0.0585597	0.0532719	0.0526205	0.0531519	0.0526294
5	0.0469178	0.0528227	0.046077	0.0454181	0.0459338	0.045608
6	0.0432958	0.0499875	0.0418773	0.0412055	0.0417076	0.0415553
7	0.0403629	0.0476008	0.0382964	0.0376295	0.0380969	0.0381579
8	0.0381546	0.0458647	0.0354396	0.0348091	0.0352275	0.0355051

Table 6.7: Relative error of SRKERP in the distorted case.

the reconstruction error is the smallest one. This may be an indicator that the scheme works. The problem of the method is the second block-iteration which converges very slowly. To see some effects in the numbers we have chosen a large regularization weight with $\alpha = 50000$, as mentioned above. Consequently, it does not make sense to show slices of the reconstructed volumes here, since the visual difference is too small.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.2272	0.227187	0.227191	0.227211	0.2272	0.227198
2	0.244638	0.243633	0.244402	0.244581	0.244405	0.244627
3	0.252197	0.250287	0.251855	0.252142	0.251855	0.252193
4	0.256553	0.253782	0.256119	0.256502	0.256124	0.256557
5	0.259665	0.256121	0.259163	0.259614	0.259166	0.25967
6	0.261824	0.257586	0.261263	0.261783	0.261268	0.26183
7	0.263302	0.258413	0.262672	0.263256	0.26268	0.263299
8	0.264599	0.259167	0.263926	0.26455	0.263932	0.264594

Table 6.8: Variance of SRKERP in the distorted case.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	9.21665E+08	9.21629E+08	9.21647E+08	9.21712E+08	9.21679E+08	9.21668E+08
2	1.78763E+07	1.78881E+07	1.78836E+07	1.78678E+07	1.78866E+07	1.78896E+07
3	7.09844E+06	7.50527E+06	7.17825E+06	7.10576E+06	7.18315E+06	7.11068E+06
4	4.19341E+06	4.78800E+06	4.28874E+06	4.20160E+06	4.29365E+06	4.20811E+06
5	2.90597E+06	3.59133E+06	2.99907E+06	2.91002E+06	3.00266E+06	2.92251E+06
6	2.24901E+06	2.98677E+06	2.33825E+06	2.25191E+06	2.34123E+06	2.26832E+06
7	1.87426E+06	2.64941E+06	1.96016E+06	1.87751E+06	1.96285E+06	1.89656E+06
8	1.63007E+06	2.42469E+06	1.71245E+06	1.63386E+06	1.71452E+06	1.65474E+06

Table 6.9: Reconstruction error of SRKERP in the distorted case.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.974839	0.974839	0.974839	0.974839	0.974839	0.974839
2	0.98753	0.987528	0.987528	0.987528	0.987528	0.987528
3	0.992129	0.992133	0.992131	0.99213	0.992131	0.992129
4	0.994478	0.994474	0.994477	0.994477	0.994477	0.994476
5	0.995869	0.995864	0.995868	0.995868	0.995868	0.995867
6	0.996776	0.996776	0.996777	0.996776	0.996777	0.996775
7	0.997394	0.997396	0.997396	0.997394	0.997396	0.997393
8	0.997833	0.997838	0.997835	0.997833	0.997835	0.997832

Table 6.10: Correlation coefficient of RART in the non-distorted case.

6.1.5 RART in the Distorted Case

In the following we analyze the RART algorithm in the distorted case with respect to the number of iterations and the regularizers. As in the other evaluation types, we compare the results with the classical Kaczmarz algorithm.

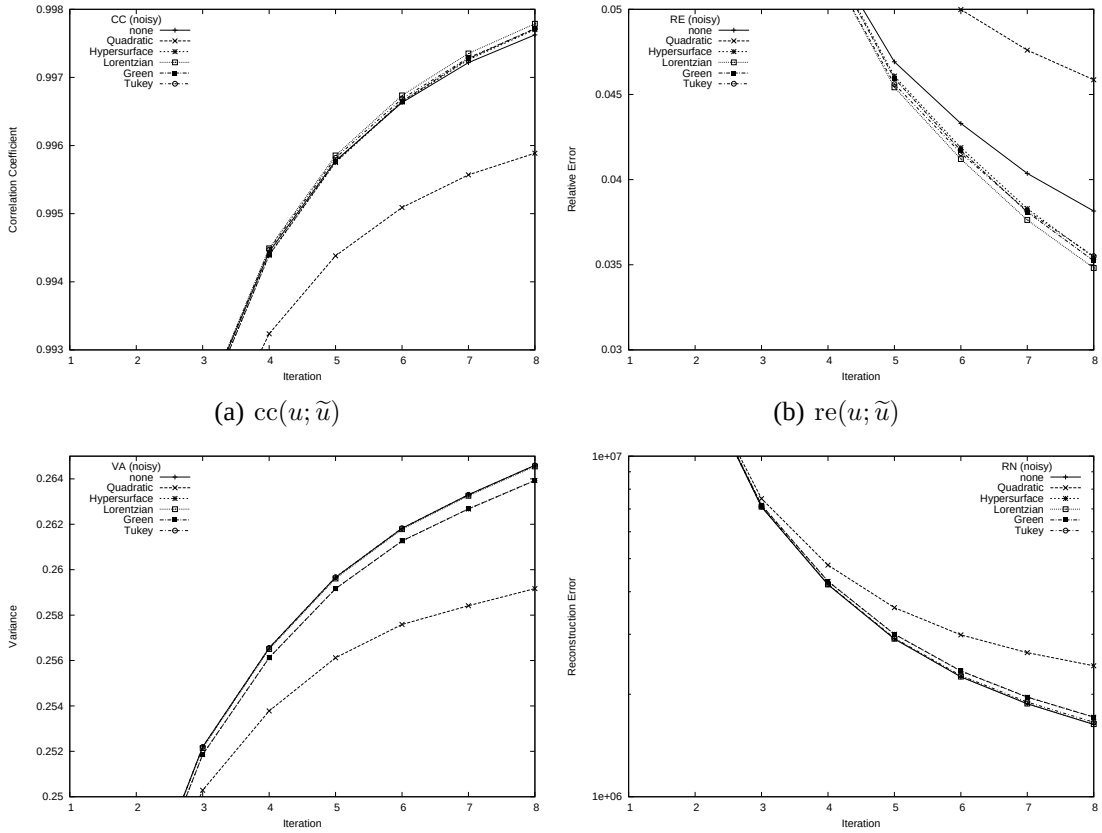
The correlation coefficient $\mathrm{cc}(u;\widetilde{u})$ can be seen in Table 6.14.

Table 6.15 explains the relative error in the distorted case of the RART algorithm.

The last two tables, Table 6.16 and Table 6.17 show the variance $\mathrm{va}(u)$ and the reconstruction

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.131622	0.131622	0.131622	0.131622	0.131622	0.131622
2	0.0807133	0.0806647	0.0806783	0.0806855	0.0806771	0.0806967
3	0.0603255	0.0601516	0.0602494	0.0602714	0.0602476	0.0602889
4	0.0486565	0.0485586	0.048599	0.0486135	0.0485978	0.0486264
5	0.0412947	0.0412108	0.0412274	0.0412455	0.041226	0.0412627
6	0.0362964	0.036185	0.0362166	0.0362427	0.0362151	0.036261
7	0.032661	0.0324991	0.0325701	0.032606	0.0325683	0.0326269
8	0.0299241	0.0297677	0.0298363	0.0298755	0.0298346	0.0298962

Table 6.11: Relative error of RART in the non-distorted case.



(a) $\mathrm{cc}(u;\widetilde{u})$ (b) $\mathrm{re}(u;\widetilde{u})$

(c) $\mathrm{va}(u)$ (d) $\mathrm{rn}(u;b)$

Figure 6.6: Results of SRKERP in the distorted case.

error $\mathrm{rn}(u;b-w)$, respectively.

Figure 6.10 illustrates the results visually. In the distorted case the difference between the methods is more clear. The correlation coefficient of the regularized versions is better than in the non-regularized case. The same holds for the relative error, where the Quadratic regularized one is best. Again, the variance of the standard Kaczmarz reconstructed volume is the largest one. The reconstruction error of the two convex cases, Hypersurface and Green, as well as of Lorentzian, is even smaller than the error of the non-regularized version.

Recapitulating, there are some indicators that the scheme works. Nevertheless, one has to do twice as many block-iterations than in the classical method and the visual difference in the result is very small after only a few iterations. In Figure 6.11 and Figure 6.12, respectively, it can be seen that the Quadratic regularized reconstruction volume is a little bit smoother than the standard Kaczmarz regularized volume. Here a center and width was chosen of $C = 0.92$ and $W = 0.22$, respectively.

(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.7: Slices of the reconstructed Shepp-Logan phantom, distorted, non-regularized, SRKERP.

(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.8: Slices of the reconstructed Shepp-Logan phantom, distorted, Lorentzian, SRKERP.

6.1.6 RKERP

In this section we take a look at the results of the RKERP algorithm. For the evaluation, we have chosen a variance of the multiplicative Gaussian noise of $\sigma = 0.001$ and have omitted the usage of a regularizer. Unfortunately, the scheme has a very slow convergence property, as can be seen in Figure 6.13, where correlation coefficient, relative error, variance and reconstruction error is shown. Obviously, the reconstruction error shrinks in a logarithmic behavior. Additionally, the figure shows the jagged convergence property of the Kaczmarz part. The reconstruction error of the second block-iteration,

$$\widehat{\mathrm{rn}}(v) := \|A^t v\|^2, \tag{6.9}$$

is illustrated in Figure 6.14 The error shrinks very slowly but much more smoothly, since the second block-iteration is implemented very similar to the SIRT algorithm.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.226901	0.226901	0.226901	0.226901	0.226901	0.226901
2	0.2446	0.244559	0.244585	0.24459	0.244584	0.244593
3	0.252638	0.252596	0.252624	0.252629	0.252623	0.252632
4	0.256805	0.25667	0.256766	0.256782	0.256765	0.25679
5	0.259786	0.259591	0.259734	0.259756	0.259732	0.259766
6	0.261612	0.261348	0.261547	0.261576	0.261545	0.261588
7	0.263202	0.262941	0.263141	0.263168	0.263139	0.263179
8	0.26461	0.264391	0.264557	0.26458	0.264556	0.264589

Table 6.12: Variance of RART in the non-distorted case.

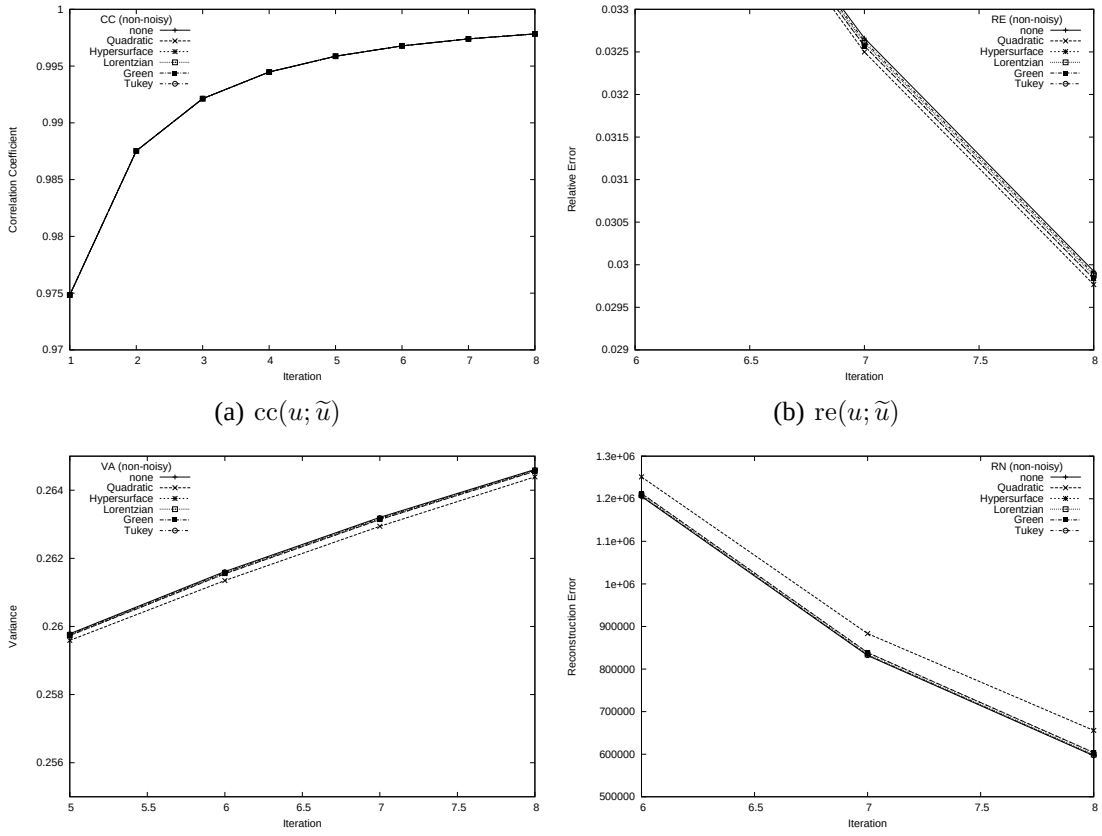
Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	8.71684E+08	8.71684E+08	8.71684E+08	8.71684E+08	8.71684E+08	8.71684E+08
2	1.64909E+07	1.64917E+07	1.64913E+07	1.64899E+07	1.64917E+07	1.64872E+07
3	6.07626E+06	6.12133E+06	6.09160E+06	6.08593E+06	6.09231E+06	6.08135E+06
4	3.09149E+06	3.08662E+06	3.08741E+06	3.08869E+06	3.08756E+06	3.08835E+06
5	1.85592E+06	1.88190E+06	1.86034E+06	1.85798E+06	1.86068E+06	1.85621E+06
6	1.20539E+06	1.25126E+06	1.21219E+06	1.20836E+06	1.21252E+06	1.20648E+06
7	8.31742E+05	8.83310E+05	8.38348E+05	8.34285E+05	8.38639E+05	8.32592E+05
8	5.96246E+05	6.55833E+05	6.03583E+05	5.98939E+05	6.03877E+05	5.97162E+05

Table 6.13: Reconstruction error of RART in the non-distorted case.

Although the RKERP algorithm works and converges to a least-squares solution, we assert that the scheme has no practical relevance for 3-D reconstruction in medical imaging since too many iterations are needed to obtain some useful images. Even after 100 iterations the details of the reconstructed Shepp-Logan phantom can only be anticipated, as can be seen in Figure 6.15.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.974823	0.974824	0.974826	0.974818	0.974823	0.974824
2	0.987487	0.987487	0.987489	0.987484	0.987488	0.987487
3	0.992054	0.992062	0.992061	0.992056	0.992062	0.992056
4	0.994371	0.994375	0.994378	0.994375	0.994379	0.994374
5	0.995731	0.995737	0.99574	0.995737	0.995741	0.995735
6	0.99661	0.996623	0.996622	0.996618	0.996623	0.996615
7	0.997197	0.997216	0.997212	0.997208	0.997213	0.997204
8	0.997605	0.99763	0.997623	0.997618	0.997623	0.997613

Table 6.14: Correlation coefficient of RART in the distorted case.



(a) $\mathrm{cc}(u;\widetilde{u})$ (b) $\mathrm{re}(u;\widetilde{u})$

(c) $\mathrm{va}(u)$ (d) $\mathrm{rn}(u;b-w)$

Figure 6.9: Results of RART in the non-distorted case.

6.1.7 SRKERP and α

In the following we examine the usage of the standard Quadratic, the convex Hypersurface and the non-convex Tukey regularizers with respect to the weighting parameter α. As shown in the previous sections, the RART and RKERP algorithms do not yield as good results as the SRKERP method; so we restrict ourselves on the latter one.

All reconstructions are done with 5 iterations and the projections are distorted by multiplicative Gaussian noise with a variance of $\sigma = 0.002$. Table 6.18 and Table 6.19 show the correlation coefficient, relative error, variance and reconstruction error, respectively.

A visualization of the tables is shown in Figure 6.16. One can see that the Quadratic regularized version gets worse the larger the weighting parameter α is. For this set of parameters the same holds for the Hypersurface version. Only the correlation coefficient and relative error of the Tukey reconstruction get better up to $\alpha = 1$. Additionally, we observe that for $\alpha > 2$, the Quadratic scheme gets unstable. Obviously, the "correction" of the regularization term is

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.132311	0.132295	0.132242	0.132276	0.132276	0.132299
2	0.0830336	0.0829154	0.0829618	0.0829263	0.0829388	0.0829976
3	0.0640401	0.0637581	0.0638888	0.063882	0.0638402	0.0639589
4	0.0535995	0.0533137	0.0534256	0.053396	0.053375	0.053504
5	0.0473358	0.0469879	0.0471101	0.0471042	0.0470783	0.0472174
6	0.0432809	0.0428344	0.0429873	0.0430003	0.0429795	0.0431289
7	0.0405388	0.0399724	0.0401875	0.0402408	0.040182	0.0403627
8	0.0384438	0.037797	0.0380293	0.0381246	0.0380421	0.0382279

Table 6.15: Relative error of RART in the distorted case.

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	0.226869	0.226893	0.226875	0.226894	0.226883	0.226863
2	0.244529	0.244517	0.244518	0.244538	0.244522	0.244517
3	0.252581	0.252562	0.252565	0.252585	0.252569	0.252562
4	0.256721	0.256607	0.256675	0.256705	0.256679	0.256692
5	0.259704	0.259525	0.259641	0.259677	0.259644	0.259665
6	0.261476	0.261215	0.261401	0.261439	0.261401	0.261436
7	0.26307	0.262812	0.262994	0.263032	0.262995	0.263024
8	0.264569	0.264346	0.264493	0.26453	0.264497	0.264522

Table 6.16: Variance of RART in the distorted case.

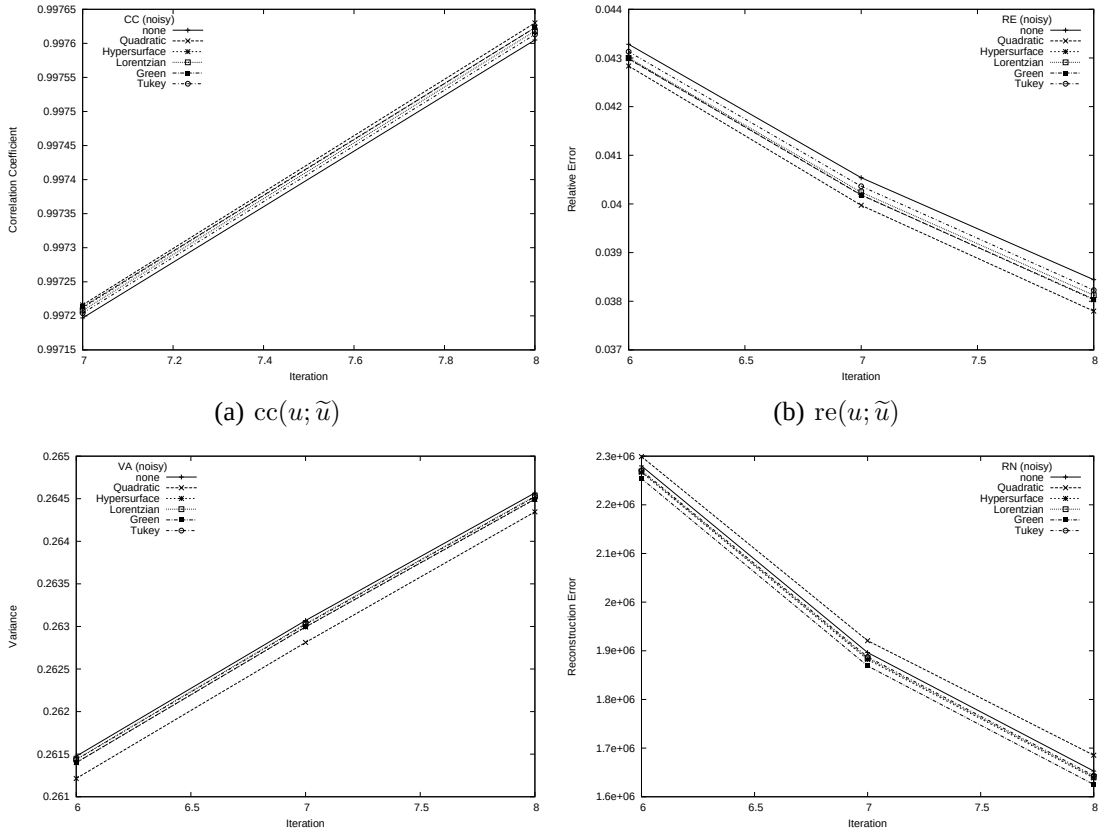
then larger than the adjustment of the block-iteration. The change of the intermediate solution by the regularizer cannot be compensated by the next Kaczmarz step. The limit for stability also depends on the weighting parameter λ; for a large λ, the regularization part can also be large.

The non-linear regularized versions get also unstable for a large weighting parameter; the effect is not as visible as in the linear reconstructed volume, since the Quadratic version is an upper boundary.

Reconstructed slices of the Shepp-Logan phantom with $\alpha = 0$ and $\alpha = 1$, regularized by Tukey, can be beheld in Figure 6.17 and Figure 6.18, respectively. For the images, center and

Iteration	No regularizer	Quadratic	Hypersurface	Lorentzian	Green	Tukey
1	8.72870E+08	8.72942E+08	8.72880E+08	8.72915E+08	8.72883E+08	8.72845E+08
2	1.76453E+07	1.76253E+07	1.76305E+07	1.76260E+07	1.76171E+07	1.76322E+07
3	7.19945E+06	7.22165E+06	7.19972E+06	7.19511E+06	7.18731E+06	7.19668E+06
4	4.19262E+06	4.16279E+06	4.17111E+06	4.17609E+06	4.15844E+06	4.18051E+06
5	2.94265E+06	2.94253E+06	2.92850E+06	2.93066E+06	2.91536E+06	2.93311E+06
6	2.27995E+06	2.29907E+06	2.26696E+06	2.26852E+06	2.25388E+06	2.27079E+06
7	1.89633E+06	1.92079E+06	1.88214E+06	1.88476E+06	1.86887E+06	1.88735E+06
8	1.65290E+06	1.68533E+06	1.63863E+06	1.64089E+06	1.62529E+06	1.64358E+06

Table 6.17: Reconstruction error of RART in the distorted case.



(a) $\mathrm{cc}(u;\widetilde{u})$ (b) $\mathrm{re}(u;\widetilde{u})$

(c) $\mathrm{va}(u)$ (d) $\mathrm{rn}(u;b-w)$

Figure 6.10: Results of RART in the distorted case.

width are chosen as $C = 0.98$ and $W = 0.24$.

6.1.8 SRKERP and σ

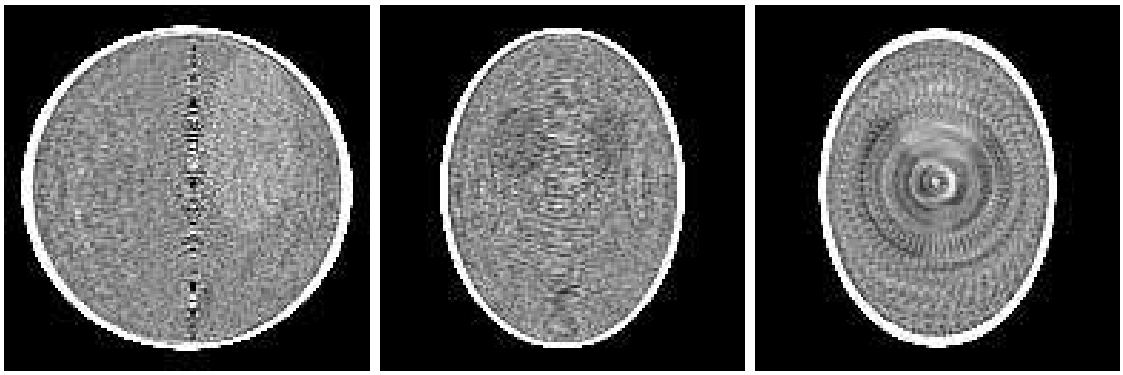
In the final section about the Shepp-Logan phantom we analyze the reconstruction behavior with respect to the scale parameter σ. The projections are again distorted with a multiplicative Gaussian noise with variance $\sigma = 0.002$.[3] Again, we examine the representative regularizers Quadratic, Hypersurface and Tukey. All reconstructed volumes are computed with 5 iterations and the weighting parameter of the regularization term is chosen again as $\alpha = 0.2$. Table 6.20 and Table 6.21 show the correlation coefficient, relative error, variance and reconstruction error, respectively.

All four tables are visualized again in Figure 6.19 Obviously, the Quadratic values are equal

[3]Note that the variance of the noise is another σ than the scale parameter. In the following σ denotes this scale parameter of the regularization function.

(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.11: Slices of the reconstructed Shepp-Logan phantom, distorted, non-regularized, RART.



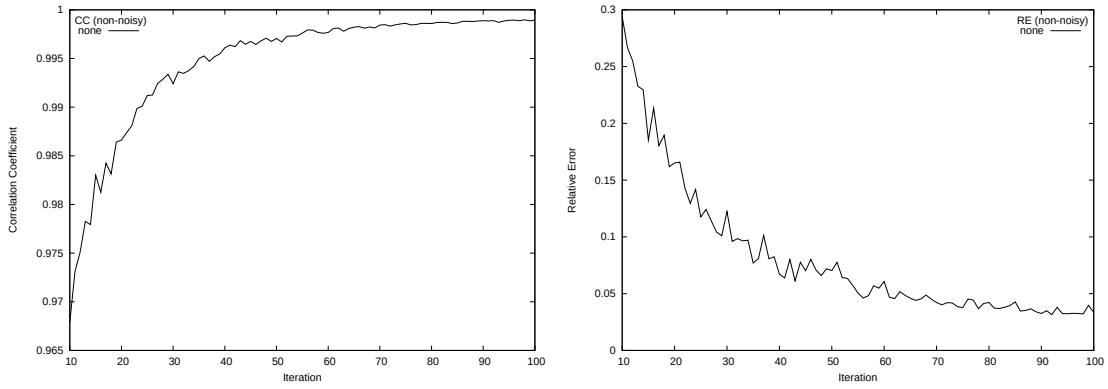
(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.12: Slices of the reconstructed Shepp-Logan phantom, distorted, Quadratic, RART.

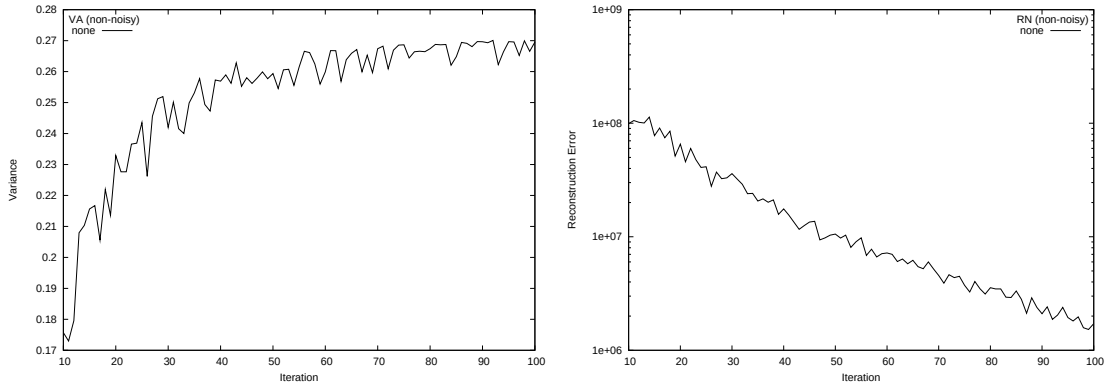
for every σ except where no regularizer is used; they are included in the tables for the sake of completeness.

For this set of parameters, the quality of the Hypersurface regularized volume shrinks with increasing σ, while the correlation coefficient and relative error of the Tukey version get better. Beginning at a certain threshold of σ the variances of Hypersurface and Tukey shrink.

Recapitulating, it is difficult to fix a set of parameters while adjusting a remaining one to obtain a conclusion about its influence. A small change of one parameter varies the effect of the others. Nevertheless, with the right set of variables, the SRKERP algorithm achieves better results than the classical Kaczmarz method.



(a) $\mathrm{cc}(u;\widetilde{u})$ (b) $\mathrm{re}(u;\widetilde{u})$



(c) $\mathrm{va}(u)$ (d) $\mathrm{rn}(u;b-v)$

Figure 6.13: Results of RKERP.

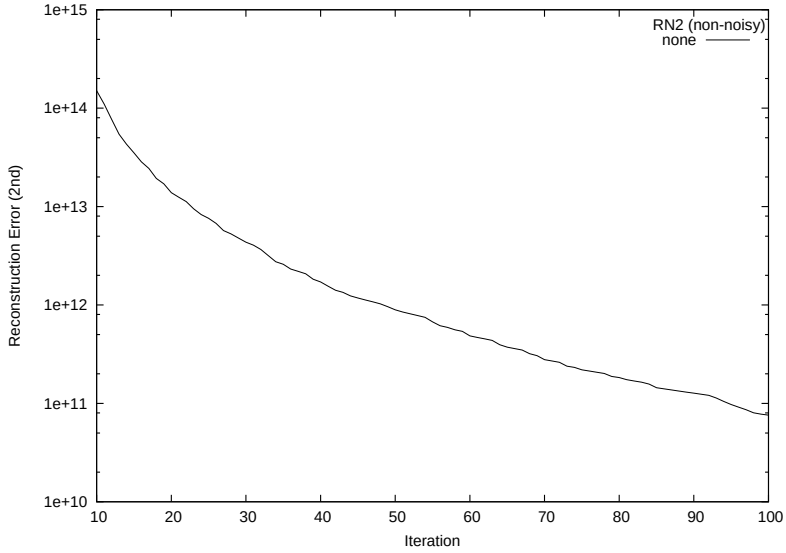
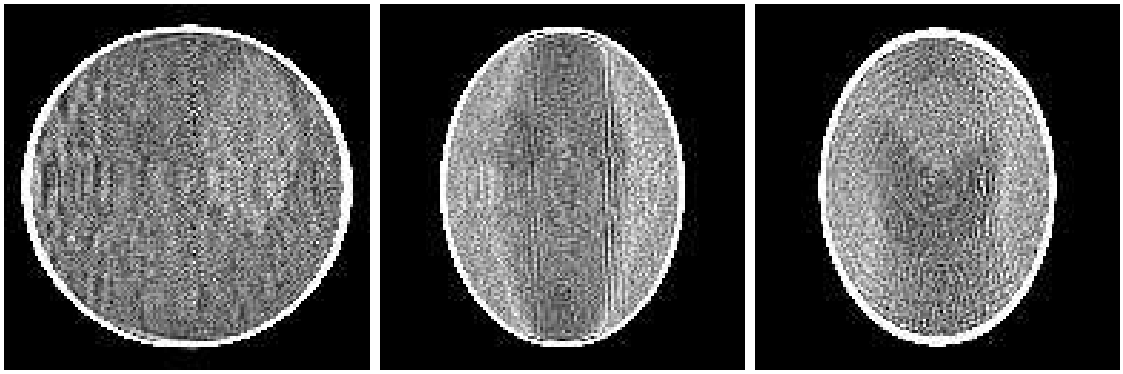


Figure 6.14: Reconstruction error of the second block-iteration (RKERP)



(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.15: Slices of the reconstructed Shepp-Logan phantom, RKERP.

(a) $\mathrm{cc}(u; \widetilde{u})$

α	Quadratic	Hypersurface	Tukey
0	0.995889	0.995889	0.995889
0.02	0.995756	0.995885	0.995891
0.05	0.995552	0.995878	0.995894
0.1	0.995199	0.995866	0.9959
0.2	0.994457	0.995837	0.99591
0.5	0.992069	0.995714	0.995933
1	0.988069	0.995405	0.99596
2	0.981165	0.994436	0.995954
5	0.75051	0.9798	0.994812
10	0.0966623	0.896322	0.992325
100	0.0939037	0.13366	0.951204

(b) $\mathrm{re}(u; \widetilde{u})$

α	Quadratic	Hypersurface	Tukey
0	0.0422884	0.0422884	0.0422884
0.02	0.0429469	0.0422563	0.0421994
0.05	0.0439136	0.042232	0.0420757
0.1	0.0456293	0.0421801	0.0418839
0.2	0.0490884	0.0421447	0.0415793
0.5	0.0594415	0.0425426	0.0409205
1	0.0751525	0.0446595	0.0407261
2	0.106578	0.057077	0.0452661
5	0.682124	0.220014	0.0951848
10	11.698	0.61174	0.140435
100	151774	7.23857	0.383853

Table 6.18: Correlation coefficient and relative error of SRKERP w.r.t. α.

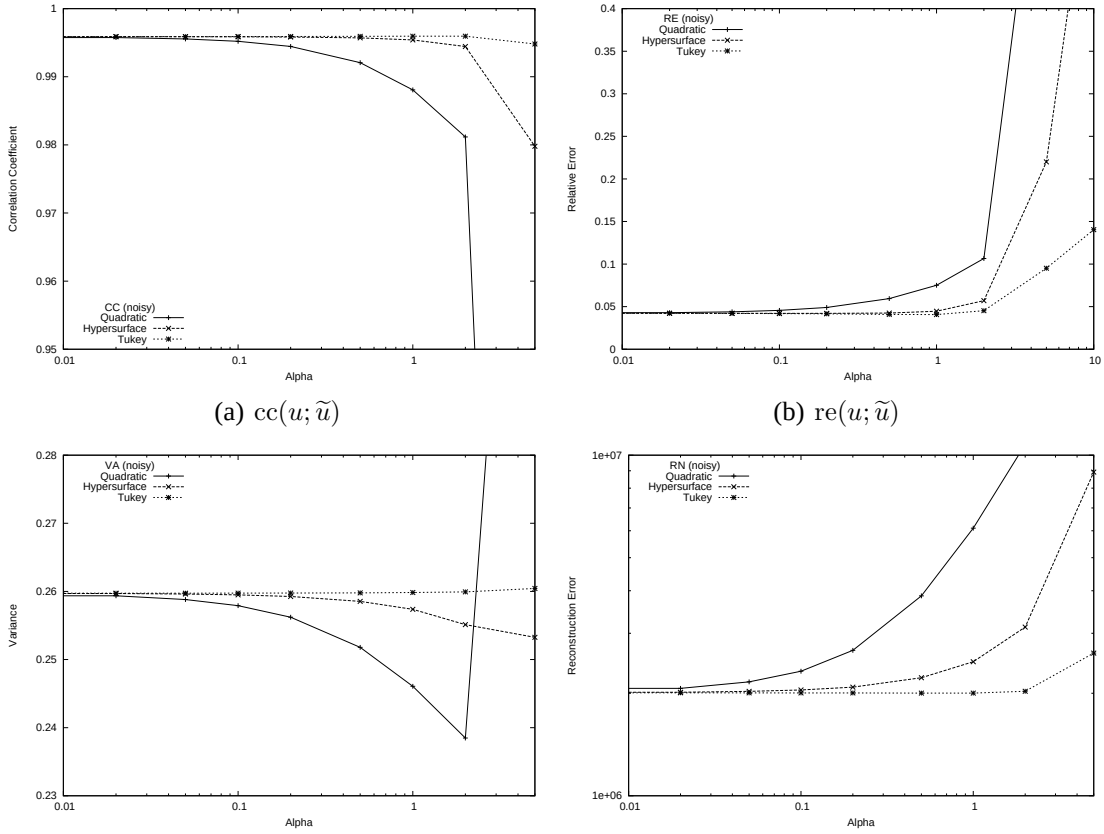
(a) $\mathrm{va}(u)$

α	Quadratic	Hypersurface	Tukey
0	0.259726	0.259726	0.259726
0.02	0.259363	0.259698	0.259744
0.05	0.258802	0.259613	0.259739
0.1	0.257913	0.259497	0.259752
0.2	0.256217	0.259255	0.259762
0.5	0.251787	0.258535	0.259789
1	0.246074	0.257372	0.259839
2	0.238474	0.255121	0.259931
5	0.377401	0.253261	0.260432
10	63.0897	0.284336	0.261397
100	1.11382E+10	7.58699	0.278772

(b) $\mathrm{rn}(u; b)$

α	Quadratic	Hypersurface	Tukey
0	2.00478E+06	2.00478E+06	2.00478E+06
0.02	2.06754E+06	2.01343E+06	2.00583E+06
0.05	2.15946E+06	2.02560E+06	2.00458E+06
0.1	2.32137E+06	2.04467E+06	2.00385E+06
0.2	2.67235E+06	2.08375E+06	2.00359E+06
0.5	3.86388E+06	2.22124E+06	2.00210E+06
1	6.10663E+06	2.47412E+06	2.00163E+06
2	1.07481E+07	3.12779E+06	2.02699E+06
5	5.25576E+07	8.92213E+06	2.62351E+06
10	2.20836E+09	4.50938E+07	3.32258E+06
100	4.89702E+15	6.07167E+09	2.77811E+07

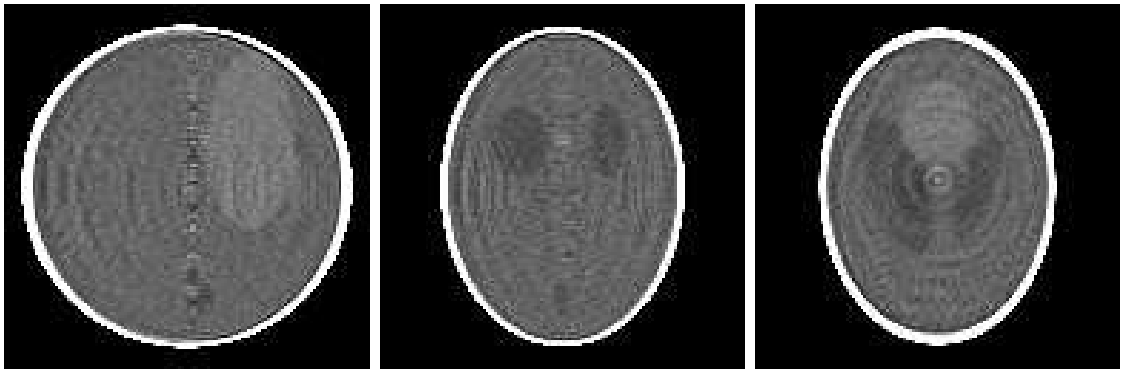
Table 6.19: Variance and reconstruction error of SRKERP w.r.t. α.



(a) $\mathrm{cc}(u;\widetilde{u})$ (b) $\mathrm{re}(u;\widetilde{u})$

(c) $\mathrm{va}(u)$ (d) $\mathrm{rn}(u;b)$

Figure 6.16: Results of SRKERP w.r.t. α.



(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.17: Slices of the reconstructed Shepp-Logan phantom, Tukey, $\alpha = 0$, SRKERP.

(a) $x = 0$ (b) $y = 0.1$ (c) $z = -0.25$

Figure 6.18: Slices of the reconstructed Shepp-Logan phantom, Tukey, $\alpha = 1$, SRKERP.

(a) $\mathrm{cc}(u; \widetilde{u})$

σ	Quadratic	Hypersurface	Tukey
-	0.995889	0.995889	0.995889
0.01	0.994457	0.995895	0.995889
0.02	0.994457	0.995895	0.995894
0.05	0.994457	0.995881	0.995901
0.1	0.994457	0.995837	0.99591
0.2	0.994457	0.995713	0.995942
0.5	0.994457	0.995324	0.996043

(b) $\mathrm{re}(u; \widetilde{u})$

σ	Quadratic	Hypersurface	Tukey
-	0.0422884	0.0422884	0.0422884
0.01	0.0490884	0.041866	0.0422041
0.02	0.0490884	0.0417753	0.0420165
0.05	0.0490884	0.0418101	0.041745
0.1	0.0490884	0.0421447	0.0415793
0.2	0.0490884	0.0429723	0.0413093
0.5	0.0490884	0.0451387	0.0410692

Table 6.20: Correlation coefficient and relative error of SRKERP w.r.t. σ.

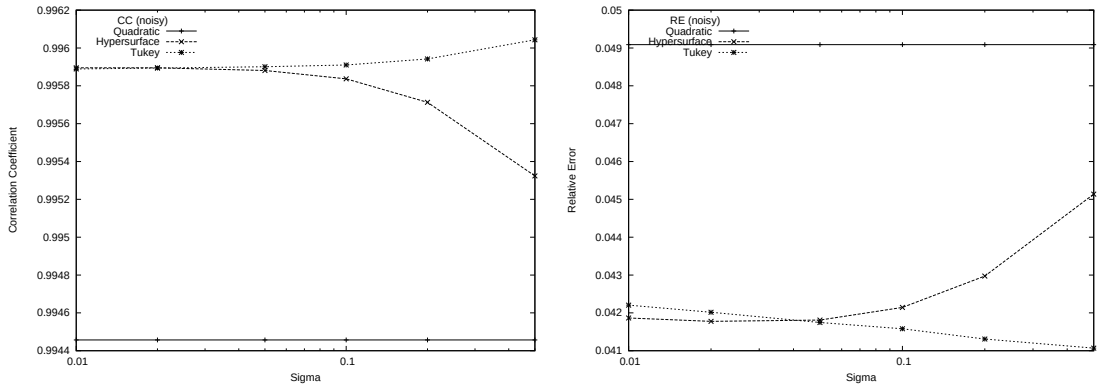
(a) $\mathrm{va}(u)$

σ	Quadratic	Hypersurface	Tukey
-	0.259726	0.259726	0.259726
0.01	0.256217	0.259684	0.259738
0.02	0.256217	0.259642	0.259732
0.05	0.256217	0.259502	0.259741
0.1	0.256217	0.259255	0.259762
0.2	0.256217	0.258799	0.259774
0.5	0.256217	0.25782	0.25964

(b) $\mathrm{rn}(u; b)$

σ	Quadratic	Hypersurface	Tukey
-	2.00478E+06	2.00478E+06	2.00478E+06
0.01	2.67235E+06	2.01295E+06	2.00329E+06
0.02	2.67235E+06	2.01924E+06	2.00486E+06
0.05	2.67235E+06	2.04353E+06	2.00689E+06
0.1	2.67235E+06	2.08375E+06	2.00359E+06
0.2	2.67235E+06	2.16759E+06	1.99911E+06
0.5	2.67235E+06	2.35044E+06	2.01728E+06

Table 6.21: Variance and reconstruction error of SRKERP w.r.t. σ.



(a) $\mathrm{cc}(u;\widetilde{u})$ (b) $\mathrm{re}(u;\widetilde{u})$

(c) $\mathrm{va}(u)$ (d) $\mathrm{rn}(u;b)$

Figure 6.19: Results of SRKERP w.r.t. σ.

6.2 Reconstruction of a Real Data Set

In the following, we take a look at the reconstruction of a human head with the SRKERP algorithm.

6.2.1 Variance and Reconstruction Error

Since we do not have a model or an ideal reconstruction of the human head we examine, we restrict ourselves on the variance and reconstruction error of the volume.

Parameters

The parameters of the reconstructions are as follows:

- The size of the reconstruction volume is $128 \times 128 \times 128$; so $N = 2097152$.
- The number of projections is 133, 66, 33 or 16, respectively. They are acquired within a range of $195°$ in an equidistant way. The size of the acquired projections is 1024×1024. They are down-sampled by a factor of 4 in each direction, so the size of the projections we reconstruct is 256×256; consequently $M_{133} = 8716288$, $M_{66} = 4325376$, $M_{33} = 2162688$ and $M_{16} = 1048576$, respectively. Note that $M_{16} < N$.
- The number of iterations is $n_{\text{iter}} = 2$ for 133 projections, $n_{\text{iter}} = 4$ for 66 projections, $n_{\text{iter}} = 8$ for 33 projections and $n_{\text{iter}} = 16$ for 16 projections, respectively.
- The weighting parameter for the reconstruction step is $\lambda = 0.1$.
- The weighting parameter for the regularization term is $\alpha = 0.2$.
- The scale parameter in the regularization functions is $\sigma = 0.1$.

Evaluation

As in the previous section, we examine the Quadratic, Hypersurface and Tukey regularized versions, as well as the classical Kaczmarz method. Table 6.22 and Table 6.23 show the variance and reconstruction error with respect to the number of projections.

Figure 6.20 visualizes the two tables; in the reconstruction error diagram the error is divided by the number of projections. As expected, the variance of the regularized reconstructed volumes is larger than the variance of the classical Kaczmarz regularized one. One can see, that for only

No. of projections	No regularizer	Quadratic	Hypersurface	Tukey
133	4.64995E-04	4.47345E-04	4.51066E-04	4.59433E-04
66	4.83391E-04	4.55300E-04	4.60976E-04	4.74605E-04
33	4.90242E-04	4.45618E-04	4.53417E-04	4.74700E-04
16	4.98438E-04	4.33767E-04	4.42366E-04	4.71776E-04

Table 6.22: Variance of SRKERP of a real data set.

No. of projections	No regularizer	Quadratic	Hypersurface	Tukey
133	3.02729E+06	3.02729E+06	3.02729E+06	3.02729E+06
66	1.56670E+06	1.57800E+06	1.57582E+06	1.57098E+06
33	8.15277E+05	8.31181E+05	8.28249E+05	8.21550E+05
16	3.41899E+05	3.57149E+05	3.54657E+05	3.48494E+05

Table 6.23: Reconstruction error of SRKERP of real data set.

a few projections, the influence of the Quadratic regularizer is very large since the output data is very distorted. The non-linear regularizers cannot distinguish between these distortions and real edges, so for only a few projections the influence of these smoothing terms is less.

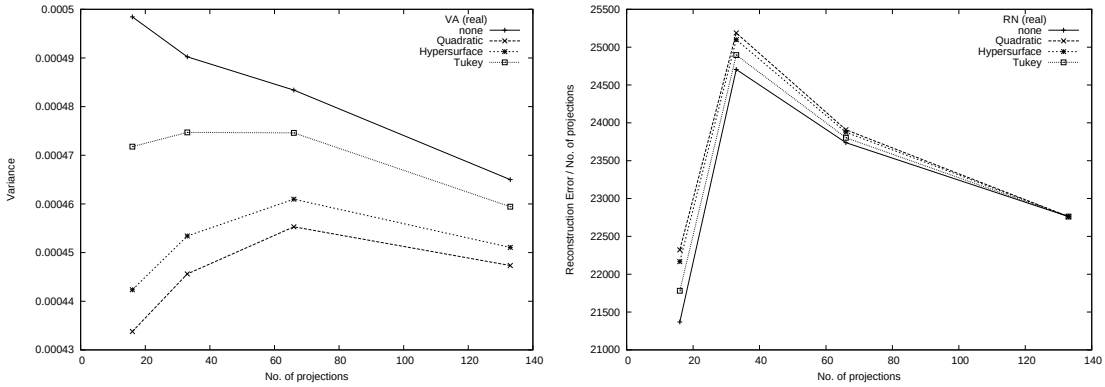
The reconstruction error is smallest for the classical Kaczmarz method. Notice that for only 16 projections the relative reconstruction error is smallest. This comes from the fact that for $M < N$ the Kaczmarz method need not satisfy as many equations as for $M \geq N$ to yield a nearly consistent solution. Nevertheless, the result is a bad reconstruction from a visual point of view, as can be seen in Figure 6.21 for the classical Kaczmarz reconstruction and Figure 6.22 for the Quadratic regularized version, respectively.

6.2.2 Visual Comparison

Finally, we present some coronal, sagittal and axial slices of the reconstructed head, computed with the classical Kaczmarz and the Quadratic, Hypersurface and Tukey SRKERP algorithms. To see some visual difference, the parameters are increased.

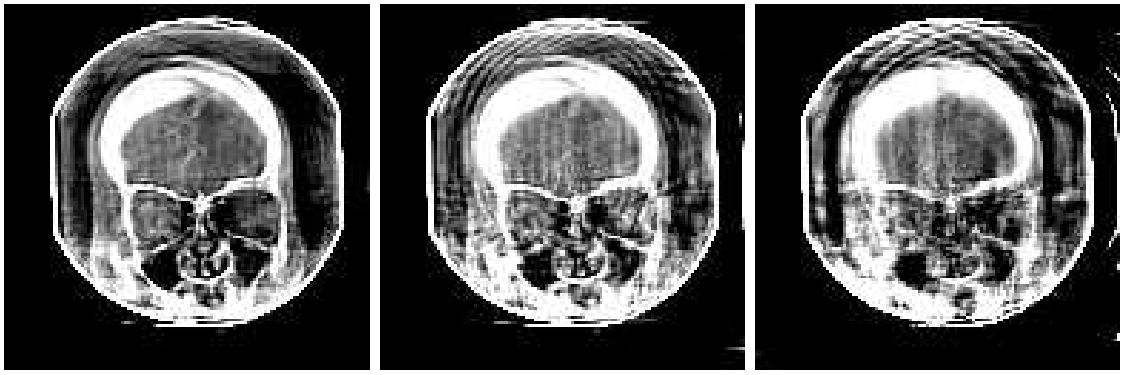
Parameters

- The size of the reconstruction volume is $128 \times 128 \times 128$; so $N = 2097152$.
- The number of projections is 66. They are acquired within a range of $195°$ in an equidistant way. The size of the acquired projections is 1024×1024. They are down-sampled by a factor of 4 in each direction, so the size of the projections we reconstruct is 256×256; consequently $M = 4325376$.



(a) $\mathrm{va}(u)$ (b) $\mathrm{rn}(u; b)$

Figure 6.20: Results of SRKERP of a real data set.



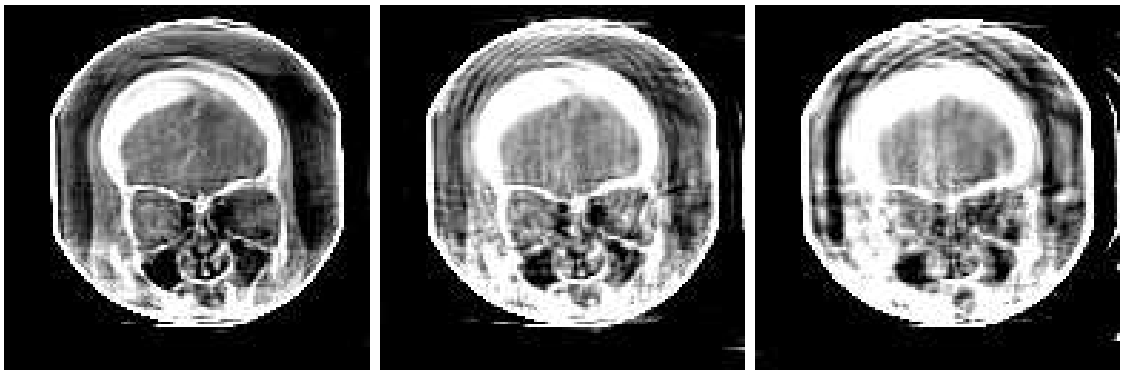
(a) 66 projs. (b) 33 projs. (c) 16 projs.

Figure 6.21: Coronal slices of the reconstructed head, non-regularized, SRKERP.

- The number of iterations is $n_{\mathrm{iter}} = 8$.
- The weighting parameter for the reconstruction step is $\lambda = 0.1$.
- The weighting parameter for the regularization term is $\alpha = 1$.
- The scale parameter in the Hypersurface regularization functions is $\sigma = 0.33$ and in the Tukey regularization function $\sigma = 1$.

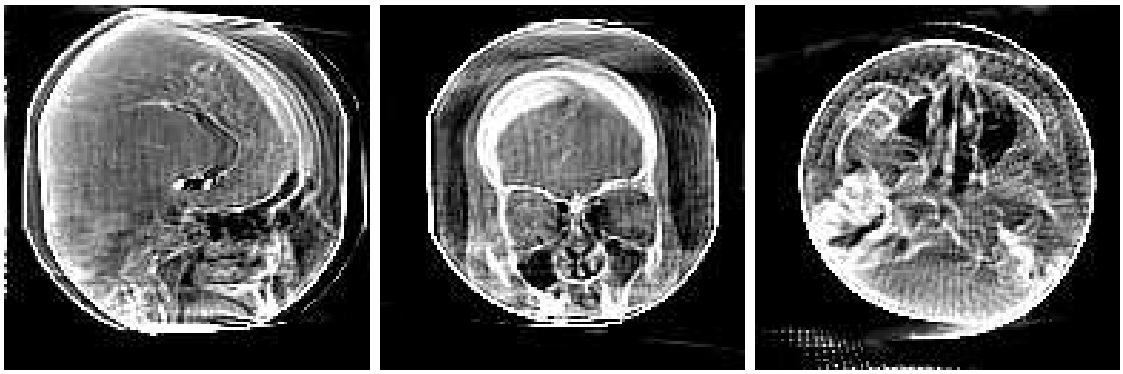
Visualization

The images can be examined in Figure 6.23, Figure 6.24, Figure 6.25 and Figure 6.26.



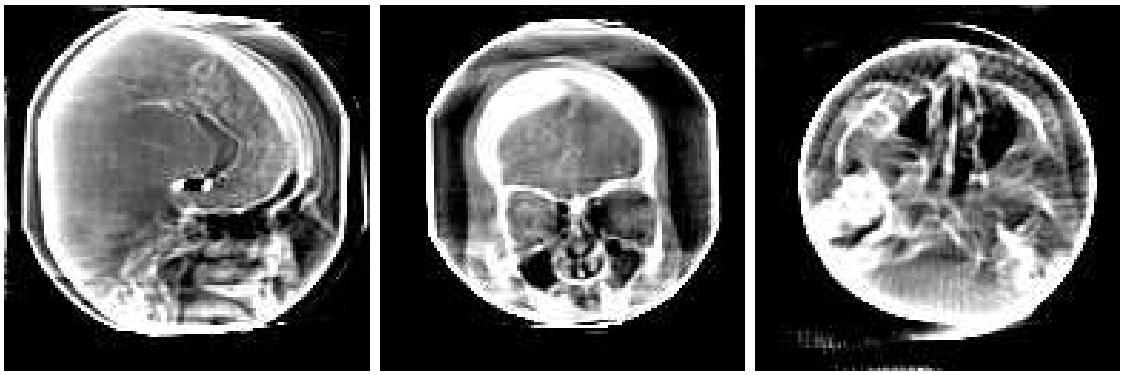
(a) 66 projs. (b) 33 projs. (c) 16 projs.

Figure 6.22: Coronal slices of the reconstructed head, Quadratic, SRKERP.



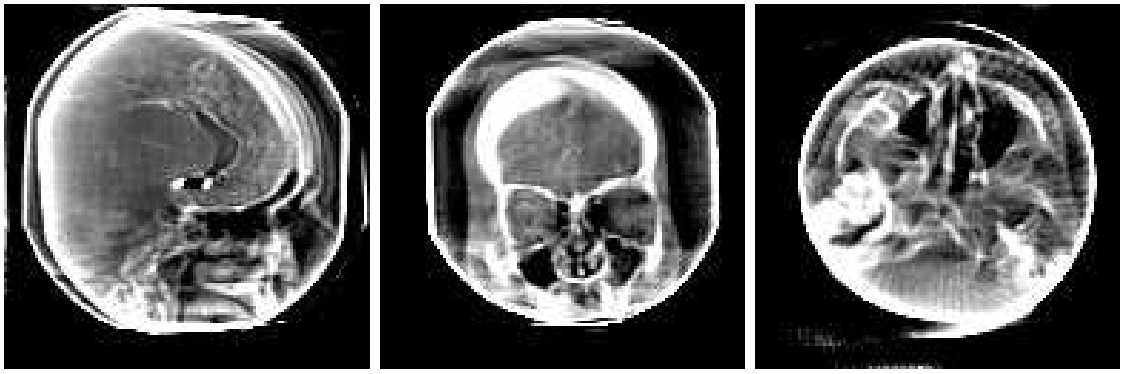
(a) sagittal (b) coronal (c) axial

Figure 6.23: Slices of the reconstructed head, non-regularized, SRKERP.



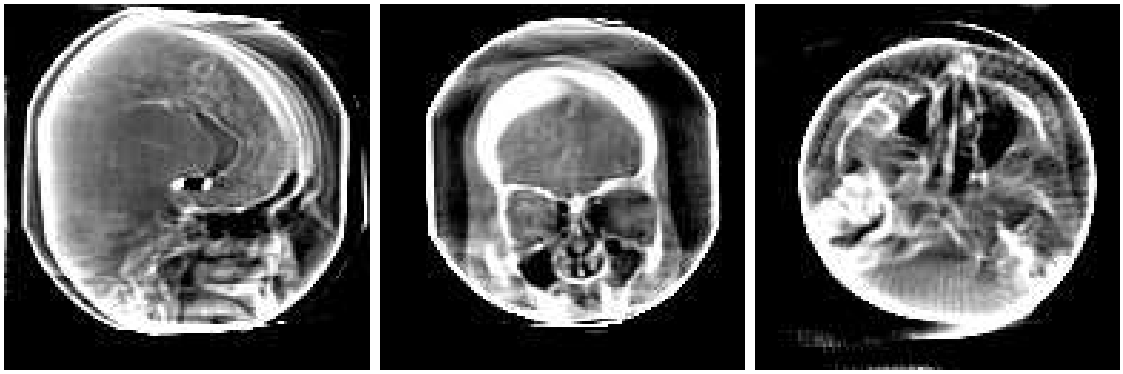
(a) sagittal (b) coronal (c) axial

Figure 6.24: Slices of the reconstructed head, Quadratic, SRKERP.



(a) sagittal (b) coronal (c) axial

Figure 6.25: Slices of the reconstructed head, Hypersurface, SRKERP.



(a) sagittal (b) coronal (c) axial

Figure 6.26: Slices of the reconstructed head, Tukey, SRKERP.

Chapter 7

Further Research

In the following, we itemize some recommendations which may be part of further investigation.

- The convergence properties of the RART algorithm should be examined in more detail. Up to now it is not evident if the RART algorithm is consistent or not.
- Part of further research could be to analyze the combination of the RKERP and the RART algorithm, i.e. what happens if the helper image w^0 from Algorithm 6 is initialized with the right hand side b.
- It is possible to introduce more sophisticated regularization terms, e.g. functions ϕ which do not only depend on the length of the gradient but on the gradient itself or even on higher derivatives.
- To make the schemes more feasible, in particular the SRKERP algorithm, it would be important to ascertain the parameters of the regularization term automatically, i.e. in what way do the parameters depend on the input data.
- It should be analyzed if the implementation of the second block-iteration of the RKERP algorithm (and also the RART algorithm) with the classical Kaczmarz method leads to a faster convergence than our “SIRT-like” implementation.

Chapter 8

Summary and Conclusion

In the following pages we recapitulate this master's thesis and present our conclusion of the results.

Algebraic Reconstruction Technique At the beginning, we have derived the general idea of algebraic reconstruction techniques with the aid of the attenuation law which describes how a ray, emitted by a point source, is attenuated along its path through a body. With the help of these measured intensity values and the weights that describe how a particular region in the body is affected by a certain ray, one can establish a linear system which is the basis of algebraic reconstruction.

Unfortunately, the realization of ART algorithms is problematically since the computation time of a reconstruction is very long. Additionally, ART methods yield only better results than the commonly used filtered back-projection methods if some ancillary conditions are given, e.g. only a few projection images are available. This fact makes the usage of ART methods not very vindicable.

After that, we have explained the idea of Kaczmarz's method, which treats the equations as $N-1$-dimensional hyperplanes in an N-dimensional space and successively projects a point in this N-dimensional space onto the hyperplanes. A drawback of the classical Kaczmarz algorithm is that it only yields satisfactory results in the consistent case. We have illustrated the method, since the algorithm is the basis for the more sophisticated schemes, introduced in this thesis.

Variational Formulation In Chapter 3, we have formulated the reconstruction problem with the aid of variational calculus as

$$\underbrace{\int_{\widetilde{\Gamma}} (\mathcal{R}u - \mathrm{I}p)^2 \, dx}_{\text{reconstruction part}} + \underbrace{\alpha \int_{\Omega} \phi(\|\nabla u\|) dx}_{\text{regularization part}} = \min . \tag{8.1}$$

The reconstruction part consists of the integral of the difference between the Radon transform of the volume and the corresponding projection images, which, to fill the gaps between the acquired projections, are interpolated. The regularization part consists of the integral of the roughness of the reconstructed volume which is measured by ϕ. The volume u which minimizes the sum of both parts is the sought one.

The formulation of the reconstruction problem in the context of variational calculus affords a new perception over the issue. This type of minimization problem makes the introduction of a regularization term plausible in an easy way.

Furthermore, we have seen that the application of the Eulerian differential equation yields

$$\mathcal{R}^*\mathcal{R}u - \frac{\alpha}{2}\mathrm{div}\left(\frac{\phi'(\|\nabla u\|)}{\|\nabla u\|}\nabla u\right) = \mathcal{R}^*\mathrm{I}p, \tag{8.2}$$

whose solution is the sought volume.

Discrete Formulation In the next chapter, we have established the connection between (8.2) and the algebraic reconstruction techniques. We have seen that the discretization of $\mathcal{R}^*\mathcal{R}u = \mathcal{R}^*\mathrm{I}p$ leads to a linear system which can be solved by iterative methods, e.g. by Kaczmarz's algorithm or one of the more sophisticated schemes, introduced in this thesis.

With the abbreviation $g(s) = \phi'(s)/(2s)$, we have discretized the regularization part as

$$\mathrm{div}(g(\|\nabla u\|)\nabla u)|_i \approx \sum_{p\in\eta_i} \lambda_{i,p} g(|u_p - u_i|)(u_p - u_i) =: (L(u)u)_i, \tag{8.3}$$

so that the complete discrete equation we want to solve looks as

$$A^tAu - \alpha L(u)u = A^tb. \tag{8.4}$$

Regularization Term Additionally, we have rewritten the divergence term such that it is subdivided into a part described by the normal vector, which points into the direction of the gradient, and into three other parts described by tangential vectors, which are orthogonal to the gradient.

We have identified some properties of the function ϕ, such that we get a good reconstruction. First of all, the smoothing property should be large in all directions for small gradients. Secondly, for large gradients where may be edges, the weight of diffusion along these edges should be large, whereas the diffusion across the edges should be small.

Finally, we have presented five examples for the smoothing function; the Quadratic one, which states the upper boundary for roughness measurements, the Hypersurface and Green functions, which are both convex, and the two non-convex functions, Lorentzian and Tukey. These four non-linear regularizers fulfill the above mentioned properties.

Iterative Solvers In Chapter 5, we have introduced the RKERP, SRKERP and RART algorithms, which solve the minimization problem

$$\|Au - b\| = \min, \tag{8.5}$$

with the regularization adjustment, iteratively.

RKERP The extension of the classical Kaczmarz algorithm to the RKERP algorithm consists of two steps. Firstly, the algorithm tries to obliterate the part of b which is not in the range of A, i.e. to make the problem consistent. This is done by a second block-iteration, which solves

$$A^t v^{k+1} = 0, \tag{8.6}$$

iteratively; v^0 is initialized with b. For $k \to \infty$, v^k converges to the part of b, which is not in the range of A, so the first Kaczmarz step is altered to solve

$$Au^{k+1} = b - v^{k+1}. \tag{8.7}$$

The second modification tunes the intermediate solution u^k by the regularization step

$$u^{k+1} = u^k - \alpha L(u^k)u^k. \tag{8.8}$$

SRKERP Since the RKERP algorithm has a very slow convergence property, we have modified the method by just omitting the second block-iteration which leads to the SRKERP algorithm.

RART After that, we have shown the idea of the RART algorithm, which introduces a helper vector $w \in \mathbb{R}^M$, such that

$$A^t w = -\alpha L(u)u, \tag{8.9}$$

which allows the reformulation of (8.4) as

$$A^t A u = A^t(b - w). \tag{8.10}$$

Similarly to the RKERP algorithm, it modifies the first block-iteration to

$$Au^{k+1} = b - w^k, \tag{8.11}$$

and introduces the second bock-iteration

$$A^t w^{k+1} = -\alpha L(u^k)u^k. \tag{8.12}$$

We were able to prove stability of the method but unfortunately, we have not proved if the scheme is consistent or not.

Evaluation In Chapter 6, we have presented some results of the introduced algorithms. Whereas the RKERP and RART algorithms yield rather bad reconstructions, the SRKERP method provides satisfying images, if the parameters of the regularization term are chosen in a right manner.

Concluding Remarks For the introduced RART and RKERP methods an additional block-iteration is required, which makes the computation time twice as long. Additionally, the RKERP algorithm has a very slow convergence, which makes the scheme not feasible. The RART algorithm yields slightly better results than the classical Kaczmarz method; however, the weighting parameter of the regularization term has to be very large and convergence of the method is not proved yet, so we dissuade from using the scheme as it is implemented right now.

The SRKERP algorithm yields good results without a longer computation time. The additional time of calculating $L(u)u$ is negligible with respect to the computation time of a block-iteration. However, we assent that using some post-processing for smoothing the reconstructed volume leads to similar results.

Appendix A

Abbreviations and Glossary

ART:	**A**lgebraic **R**econstruction **T**echnique.
FBP:	**F**iltered **B**ack-**P**rojection.
K:	**K**aczmarz algorithm.
KERP:	**K**aczmarz **E**xtended with **R**elaxation **P**arameter algorithm.
KRP:	**K**aczmarz with **R**elaxation **P**arameter algorithm.
MRF:	**M**arkov **R**andom **F**ields.
RART:	**R**egularized ART (abbreviation is also used for the RART algorithm).
RKERP:	**R**egularized **K**aczmarz **E**xtended with **R**elaxation **P**arameter algorithm (it is also known as the *regularized Kaczmarz's algorithm with relaxation parameter*.
SART:	**S**imultaneous **A**lgebraic **R**econstruction **T**echnique.
SIRT:	**S**imultaneous **I**terative **R**econstruction **T**echnique.
SRKERP:	**S**implified **R**egularized **K**aczmarz **E**xtended with **R**elaxation **P**arameter algorithm.

A.1 Mathematical Notations

$A := B$	The expression appearing on the right defines the symbol on the left.
$\|.\|$	Euclidean norm: $\|v\| := \sqrt{\sum_{j=1}^{N} v_j^2}$ for $v \in \mathbb{R}^N$.
$\|.\|_*$	A norm that has to be defined further. If the symbol is used on a matrix it is the induced matrix norm associated with the vector norm in the current context.
∇u	Gradient of u: $(\nabla u)_j = \frac{\partial u}{\partial x_j}$.
$\mathrm{div}(u)$	Divergence operator: $\mathrm{div}(u) = \sum_{j=1}^{N} \frac{\partial u}{\partial x_j}$.
$\nabla^2 u$	Hessian matrix of u: $(\nabla^2 u)_{i,j} = \frac{\partial^2 u}{\partial x_i \partial x_j}$.
$\triangle u$	Laplacian operator: $\triangle u = \sum_{j=1}^{N} \frac{\partial^2 u}{\partial x_j^2}$.
$\{A : P\}$	The set of elements A satisfying property P.
$\mathbb{R}$	Set of real numbers.
$\mathbb{N}$	Set of natural numbers.
(a, b)	The open interval $\{x \in \mathbb{R} : a < x < b\}$.
$[a, b)$	The half-open interval $\{x \in \mathbb{R} : a \leq x < b\}$.
$(a, b]$	The half-open interval $\{x \in \mathbb{R} : a < x \leq b\}$.
$[a, b]$	The closed interval $\{x \in \mathbb{R} : a \leq x \leq b\}$.
$\langle x, y \rangle$	The inner product of the vectors x and y.
$\mathcal{O}^*$	The adjoint operator of $\mathcal{O}$.
A^t	The transposed matrix of A. It is also used to transform a row vector into a column vector and vice versa.
$A^\dagger$	A general inverse of A.

A.2 Symbols for the Reader's Convenience

$\square$	Symbol marking the end of a proof.

Appendix B

Patent Research

Only a few patents arose from our patent research; all of them are only marginally related to the topic of this master's thesis.

The first one we allude is [DV03]. There De Villiers describes a method to successively reconstruct a volume by using the forward-projection of the intermediate result as an a-priori information for the next iteration. Similarly to the regularization terms used in this thesis, the method enforces smoothness within certain regions. A major difference to our methods is that De Villiers identifies regions of a particular tissue using estimators from probability theory.

The second and last patent we mention here is [Kal01]. In this invention the authors propose a regularization with the help of wavelets. The reconstructed data is decomposed in a wavelet vector family. After that, the resulting coefficients are regularized by applying a thresholding operation. Finally, the coefficients are transformed back into image data which can be visualized. The regularizers in this thesis work directly on the reconstruction data and not on an additional decomposition.

We have not found patents which directly rely on the usage of a regularizer within algebraic reconstruction techniques or even on a formulation of the reconstruction problem in the framework of variational calculus.

Appendix C

Included DVD

We have included a DVD which can be found at the end of this master's thesis. It contains the following directories:

- **latex**: This thesis is written in LaTeX. The sources can be compiled by using the *makedoc.sh* script.
- **doc**: This directory contains the thesis in postscript format (*da00.ps*), as well as in portable document format (*da00.pdf*).
- **doc_additional**: This directory contains some text files which were used to create the tables and diagrams of the results.
- **results**: This directory contains the reconstructed images in *unsigned short - raw* format. The result is described in its corresponding comma separated values (.csv) file. Additionally, the directory contains a Microsoft Excel™ file (*evaluation_concept.xls*), which indexes the result files.
- **bin**: This directory contains the binaries which may be executed under Microsoft Windows XP™ or higher and on an Intel Pentium 4™ with SSE2™ technology. Detailed usage is given by executing the programs without parameters. The execution file *rart.exe* is the main program which performs the reconstruction. There are some scripts (*.cmd*) which explain the usage. The program *GetSlice.exe* extracts slices out of the reconstructed volume data. The program *shepp.exe* creates a Shepp-Logan phantom and its projection images. Finally, the directory contains the projection images of the Shepp-Logan phantom and the real image data *shepp128_original_float.raw* and *WaterCorr_Float.raw*, respectively, in single precision floating point format.

- **Reconstruction**: This directory contains the C++ sources. A Microsoft Visual Studio 2005™solution file is also included (*Reconstruction.sln*) which contains the three projects corresponding to the above mentioned binaries. Additionally, a makefile is included which may be used to compile the reconstruction program within UNIX environments. Finally, the *Doxyfile* file contains information to build documentation out of the source-code with doxygen.

- **Reconstruction/doc/html**: This directory contains the html documentation of the source code, created by doxygen.

- **Reconstruction/doc/latex**: This directory contains the latex documentation of the source code, created by doxygen. It may be used to create postscript or portable document format documentations.

- **Reconstruction/doc/pdf**: This directory contains the pdf documentation of the source code (*refman.pdf*).

List of Figures

List of Tables

Bibliography

[Aub02] Aubert, G.; Kornprobst, P.: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, Springer-Verlag New York, Inc., 2002.

[Bla98] Black, M.; Sapiro, G.; Marimont, D.; Heeger, D.: *Robust Anisotropic Diffusion*, *IEEE Transactions on Image Processing*, Vol. 7, No. 3, March 1998, p. 421–432.

[Bro01] Bronstein, I.; Semendjajew, K.; Musiol, G.; Mühlig, H.: *Taschenbuch der Mathematik*, Verlag Harri Deutsch, 2001.

[Bur01] Burden, R.; Faires, J.: *Numerical Analysis*, Wadsworth Group. Brooks/Cole, 7th. ed., 2001.

[Cha97] Charbonnier, P.; Blanc-Féraud, L.; Aubert, G.; Barlaud, M.: *Deterministic Edge-Preserving Regularization in Computed Imaging*, *IEEE Transactions on Image Processing*, Vol. 6, No. 2, February 1997, p. 298–311.

[DV03] De Villiers, M.: *Verfahren zur Rekonstruktion von Tomographischen Bildern*, Trademark EP 1 356 433 A1, Ref.: 02715609, Int. Cl.: G06T 11/00, Appl.: Watkins, Eserick Tanchwel, Pr.: January 1, 2001 ZA 200100482, October 2003.

[Els70] Elsgolc, L.: *Variationsrechnung*, Bibliographisches Institut AG, Mannheim, 1970.

[Eps03] Epstein, C.: *Introduction to the Mathematics of Medical Imaging*, Pearson Education, Inc., 2003.

[Hor05] Hornegger, J.; Paulus, D.: *Medical Image Processing*, Script of Lehrstuhl für Mustererkennung, http://www5.informatik.uni-erlangen.de, Universität Erlangen-Nürnberg, WS 2004/05.

[Kak01] Kak, A.; Slaney, M.: *Principles of Computerized Tomographic Imaging*, SIAM Society for Industrial & Applied Mathematics, June 2001.

[Kal01] Kalifa, J.; Laine, A.; Esser, P.: *Method for Tomographic Reconstruction with Non-linear Diagonal Estimators*, Trademark WO 2001/084500, Ref.: PCT/US01/13737, Int. Cl.: G06T 11/00, Appl.: The Trustees of Columbia University in the City of New York, Kalifa, J., Laine, A. and Esser, P.D., Pr.: April 28, 2000 60/201,679, November 2001.

[Kyb01] Kybic, J.; Blu, T.; Unser, M.: *Variational Approach to Tomographic Reconstruction*, *Proceedings of SPIE, San Diego, CA*, February 2001.

[Mil96] Milanfar, P.; Karl, W.; Wilsky, A.: *A Moment-Based Variational Approach to Tomographic Reconstruction*, *IEEE Transactions on Image Processing*, Vol. 5, No. 3, March 1996, p. 459–470.

[Mue99] Mueller, K.; Yagel, R.; Wheller, J.: *Fast Implementations of Algebraic Methods for Three-Dimensional Reconstruction from Cone-Beam Data*, *IEEE Transactions on Medical Imaging*, Vol. 18, No. 6, June 1999, p. 538–548.

[Nat86] Natterer, F.: *The Mathematics of Computerized Tomography*, John Wiley and Sons Ltd, September 1986.

[Per90] Perona, P.; Malik, J.: *Scale-Space and Edge Detection Using Anisotropic Diffusion*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, July 1990, p. 629–639.

[Pop98] Popa, C.: *Extensions of Block-Projections Methods with Relaxation Parameters to Inconsistent and Rank-Deficient Least-Squares Problems*, *B I T*, Vol. 38, No. 1, 1998, p. 151–176.

[Pop04] Popa, C.; Zdunek, R.: *Kaczmarz Extended Algorithm for Tomographic Image Reconstruction from Limited-Data*, *Math. and Computers in Simulation*, Vol. 65, No. 6, 2004, p. 579–598.

[Pop05a] Popa, C.; Zdunek, R.: *Penalized Least-Squares Image Reconstruction for Borehole Tomography*, in *Proceedings of Algoritmy 2005 Conference on Scientific Computing*, Slovak Univ. of Tech., Bratislava, Vysoke Tatry - Podbanske, Slovakia, March 2005, p. 260–269.

[Pop05b] Popa, C.; Zdunek, R.: *Regularized ART with Gibbs Priors for Tomographic Image Reconstruction*, in *Proceedings of ASIM Conference*, SCS Publishing House e.V., Erlangen, September 2005.

[Ros82] Rosenfeld, A.; Kak, A.: *Digital Picture Processing: Volume 1 (Computer Science and Applied Mathematics)*, Academic Press Inc., 2nd. ed., September 1982.

[Sch04] Schneider, S.: *Finite Difference Method for Partial Differential Equations with Anisotropic Diffusion Behavior in Image Processing*, Bachelor Thesis, Institute of Applied Mathematics, Friedrich-Alexander University of Erlangen-Nuremberg, July 2004.

[Ste05] Steffen, P.: *Zweidimensionale Digitale Signalverarbeitung*, Script Lehrstuhl für Multimediakommunikation und Signalverarbeitung, http://www.lnt.de/lms, Universität Erlangen-Nürnberg, SS 2005.

[Tit85] Titterington, D.: *General structure of regularization procedures in image reconstruction*, *Astronomy and Astrophysics*, Vol. 144, 1985, p. 381–387.

[Wei96] Weickert, J.: *Theoretical Foundations of Anisotropic Diffusion in Image Processing*, *Computing Supplement*, Vol. 11, 1996, p. 221–236.

[Wei98] Weickert, J.: *Anisotropic Diffusion in Image Processing*, Teubner-Verlag, Stuttgart, 1998.