Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Robust Algorithms for Discrete Tomography

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfilment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

FRANK TABAK

Delft, the Netherlands
October 2012

MSc THESIS APPLIED MATHEMATICS

"Robust Algorithms for Discrete Tomography"

FRANK TABAK

Delft University of Technology

Daily supervisor

Dr.ir. M.B. van Gijzen

Responsible professor

Prof.dr.ir. C. Vuik

Other thesis committee members

Prof.dr. K.J. Batenburg

Prof.dr.ir. G. Jongbloed

October 2012

Delft, the Netherlands

Abstract

Tomography methods concentrate on reconstructing objects from multiple projections that are obtained by sending, for example, X-rays through the object. Applications of these methods are, among others, radiology (CT-, MRI- and PET scans), geophysics and material science. The tomographic problems can be formulated as a system of linear equations. Unfortunately, these systems are not symmetric nor positive (semi)definite, rank deficient and not square.

In material science one is often presented with very small objects (like crystals or nano-structures) that consist of one or a small number of different materials, each with its own density. Scanning these small objects can cause damage to the structure and thus one can only take a very limited amount of projections. Fortunately, one can use the prior knowledge about the object to arrive at a reconstruction of the original object. How to arrive at this reconstruction is studied by the field of discrete tomography (DT).

With every kind of tomography, and thus also with DT, one is faced with noisy data. Because of this noise the reconstruction process becomes more difficult since the system of linear equations becomes inconsistent. The DART (Discrete Algebraic Reconstruction Method) algorithm was developed to solve DT problems. This algorithm deals with noise in a very heuristic method. The goal of this project is to investigate how the problem can be regularized such that it deals with the noise in a more efficient and robust manner.

Preface

This thesis concludes my master Applied Mathematics at Delft University of Technology. This project was formulated by Joost Batenburg form the Centrum Wiskunde & Informatica (CWI) in Amsterdam, whom I want to thank for his enthusiastic involvement in this project and supervision. Martin van Gijzen was the daily supervisor of this project on the behalf of Delft University of Technology. Also Martin is greatly acknowledged for his supervision and entertaining conversations but above all very useful discussions.

Furthermore, I would like to thank Gemma van de Sande, Jacob de Zoete, Linda Crapts, Michiel de Reus and André Vieira for their inspiring and relaxing lunch and coffee conversations. Also Paul de Zeeuw, Folkert Bleichrodt, Daan Pelt, Linda Plantagie and Wagner Fortes are not forgotten for their help, lunch conversations and sporadic table tennis matches. Above all I want to thank my girlfriend Rebecca Jacobs for her support and for naming the test problem 'Speckled'. Lastly I want to thank prof.dr.ir. C. Vuik and prof. dr. ir. G. Jongbloed for their willingness to be a member of my graduation committee.

Contents

Chapter 1

Introduction

The study of reconstructing (slices of an) object from its projections is called *tomography*. The projections are obtained from various angles via penetrating waves, e.g. X-rays. The word tomography has it origins in the two Greek words $\tau\acute{o}\mu o\sigma$ (tomos), meaning *slice* or *part*, and $\gamma\rho\acute{a}\phi\epsilon\iota\nu$ (graphein), meaning *to write*. Tomography was first considered after the invention of X-rays by Wilhelm Röntgen in 1895 although it began flourishing in the period before the second world war [31, p. 6]. Since reconstructions are usually done by computers the term *Computerised Tomography* (CT) is often used in literature.

Figure 1.1: *'Hand mit Ringen', the first medical X-ray picture taken on December 22, 1895. It shows the hand of Wilhelm Röntgen's wife.*

Tomography is probably most famous for its applications in medicine for it provides non-invasive ways to see the internal structure of a person. Though the field of application is very broad, from nanoscopic scale where it is used to determine structure of certain nano-particles, to galactic-scale where it is used to reconstruct the X-ray structure of supernova remnants. Tomography is, among many other fields, studied in medicine, materials science, geo- and astrophysics. This motivates the need for developing better and faster reconstruction techniques.

One particular field of interest is *Discrete Tomography* (DT). The objects that are reconstructed in this field are assumed to be made up out of a small number of different materials having a homogeneous density that is known beforehand, like crystals or nano-structures. This kind of tomography is mainly used in microscopy to create 3-dimensional reconstructions of small objects, but also other fields might benefit from the results that are obtained in discrete tomography.

1.1 Motivation, Research Goals and Methodology

There are various methods to solve DT problems. One of these methods is an algorithm called DART, which is an abbreviation for Discrete Algebraic Reconstruction Technique. This iterative method, proposed by Batenburg and Sijbers in [3], consists of two loops. A discrete reconstruction is obtained by solving the reconstruction problem in the inner-loop without using the information about the densities or, equivalently, discrete grey values. Subsequently, this information is used to obtain a reconstruction that only contains the densities that are known to be in the object. This constitutes one DART iteration. The previous approximation is used in the next DART iteration to improve the reconstruction. The algorithm is applied intensively in applied fields, mainly electron microscopy, but also X-ray tomography benefits from using DART. Two of many examples of research papers in which DART is applied are [20] and [18]. This broad application motivates the following research question:

Can the DART algorithm be improved?

Improvement in the sense of accuracy of the reconstruction, that is. Of course this question is very broad and some more specific questions will be formulated along the way to help achieve this goal. This will result in an attempt to improve DART by removing heuristic constructs and introduce regularization.

There are a lot of parameters involved with DART. The experiments in this work will only change one parameter at a time, in this way the corresponding results can be attributed to the changed parameter. Consequently, DART is analysed and improved incrementally. From time to time, when some changes and optimizations are implemented, it is investigated if previous results still hold. For example, in Chapter 6 the optimal number of inner-loop iterations is derived, later on it is investigated if the results from before still hold, which actually turns out to be not entirely the case.

Some randomness is involved with DART and the simulations performed in this work. For one, the current formulation of DART contains an inherent randomness known as the *random subset*. One can easily imagine that two subsequent simulations with the same parameters will results in different results. Another factor that introduces randomness in the simulation is noise on the measurement data. Two simulations with the same parameter on (slightly) different measurement data can also produce different results. One has to keep this in mind at all times.

The improvement of DART will be based on experiments on six different test problems which have various properties. The validation of the improved DART algorithm is carried out by comparing the performance of the current DART to the improved DART on two new test problems. The new test problems are taken to make sure that the improvements are not just improvements on the six test problems and actually have an adverse effect on other problems. Lastly, the improved DART will be tested on experimental μCT data.

1.2 Outline of this Thesis

A short introduction in tomography is given in Chapter 2. For a more thorough introduction into this subject one is referred to the corresponding literature study that has been performed before realization of this thesis [29].

Next, the algorithms that can be used as a subroutine in DART are introduced and defined in Chapter 3. Also the convergence behaviour of these algorithms with and without noise in the measurement data is investigated in this chapter.

Discrete tomography is properly defined in Chapter 4. Several solutions strategies for discrete tomographic problems that are used in practice will be listed. One of these strategies is the approach DART uses. Subsequently, the DART algorithm is introduced in detail.

In Chapter 5 several research questions will be formulated to construct a framework in which the goal, i.e. to improve DART, is attempted to be achieved. The methodology of how the individual research questions will be answered is described. Finally, the test problems that will be used for the experiments are presented.

The numerical experiments are carried out in Chapter 6. First the framework of the experiments is described and some remarks about how the simulations are executed will be made. Next some parameters of DART will be investigated after which the research questions from Chapter 5 will be addressed. After that a measure for reconstruction errors is introduced which will be used to quantify the quality of a reconstruction. Finally, it is attempted to incorporate the properties of SIRT, an algorithm introduced in Chapter 3, to other algorithms.

In Chapter 7 the performance of the current DART and the improved DART algorithm will be compared. To this end they are both applied to two new test problems and to real world experimental μCT data.

Chapter 8 will conclude this work and as such the research goal will be evaluated. Furthermore, some recommendations about future research are made that might prove beneficial for the performance of DART.

Chapter 2

Tomography

There are many types of tomography. *Transmission tomography* for example is based on sending penetrating waves through an object and measuring the waves on the other end of the object with a detector. This work will focus on this type of tomography. X-ray computed tomography (CT/CATScan) uses transmission tomography for example. *Emission tomography* on the other hand is based on having the source inside the object, this source is usually a radioactive tracer which is injected into the blood stream. Two types of emission tomography can be distinguished, *Single Particle Emission Computed Tomography* (SPECT) which measures emitted particles along a half-line, and *Positron Emission Tomography* (PET) which is based on the conversion of a proton into a neutron. During this conversion a positron and neutrino are released, this positron will interact with an electron almost instantly creating two photons which travel in approximate opposite direction. These photons are measured and from the measurement one can determine the location where the annihilation took place, and hence the source was located. Another type of tomography is *reflection tomography*, typically sound waves are emitted towards an object, the sound is reflected and from that the object can be reconstructed.

For an extensive detailed report on the history of tomography and an account on the all various aspects associated with tomography one is referred to the book of Webb [31].

2.1 The Reconstruction problem

All types of tomography reconstruct an object or function from its projections, or in mathematical sense, line (or hyperplane) integrals. The object has a certain internal distribution which one wants to reconstruct. In X-ray tomography for example the density of tissue attenuates the radiation. By measuring to which extent the rays are attenuated the internal distribution of the object, and thus the distribution of the tissue, can be reconstructed. This reconstruction can be done in roughly two ways, via *analytical* methods or *algebraic* methods. The former method relies on Fourier transforms to make the reconstruction while the latter method writes the problem as a linear system of equations.

In what follows a *ray* will be referred to as the path a certain penetrating wave follows when travelling from the source to the detector. In X-ray CT this is simply the path from an X-ray tube (vacuum tube that produces X-rays) to the detector. There are many ways in which one can

emit these rays, referred to as *data acquisition techniques*. The rays can be emitted parallel to each other for each projection angle. This yields the most natural understanding of tomography. In practice, however, it is very hard to carry out such a parallel acquisition. Instead fan (in case of one-dimensional projections) or cone (two-dimensional projections) acquisition techniques are used. Luckily, some correspondence exist between the various types of techniques which enables one to use results for parallel rays for non-parallel rays. Figure 2.1 shows two different data acquisition techniques for two-dimensional reconstruction.

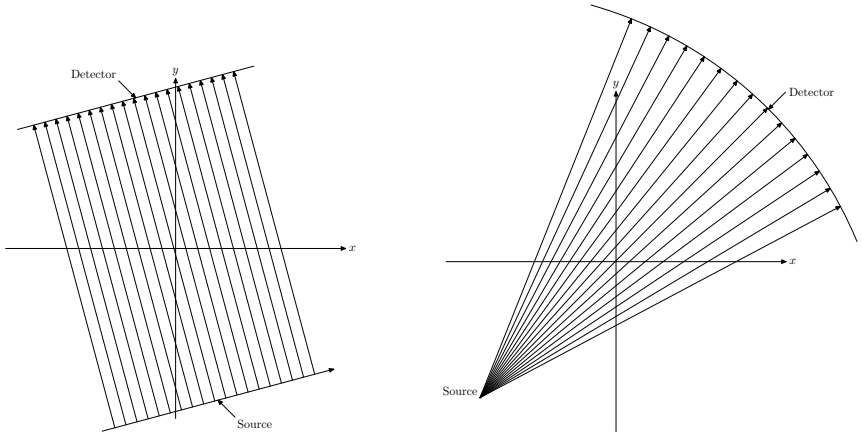


Figure 2.1: *Left: Parallel data acquisition geometry. Right: Fanbeam (non-parallel) data acquisition geometry.*

Throughout this work a parallel data acquisition technique is assumed for it provides a more natural understanding of the tomographic process. Furthermore, it is assumed that the density of the object is related to the attenuation of rays, for example caused by the energy absorption of X-rays by tissue. This attenuation or density is represented by the function $f(x, y)$. Figure 2.2 depicts the sketch of the situation. In the figure one ray is shown which is sent through the object and hits the detector at t_1. The ray subtends an angle θ with the positive y-axis which means that the projector, which moves orthogonal to the ray, subtends an angle θ with the positive x-axis. This angle is called the *projection angle*. For one projection angle a great many rays are sent through the object, each of which hits the detector at the other side resulting in measurements $P_\theta(t)$ corresponding to projection angle θ. The measurement data for one projection angle corresponds to the attenuation in the following way:

$$P_\theta(t) = \int_{x\cos(\theta)+y\sin(\theta)=t} f(x, y)\mathrm{d}s. \tag{2.1}$$

The object is *scanned* for a certain number of angles and from the resulting data the shape of the object is reconstructed. Usually there are multiple objects which yield the same projection data, and thus in general one cannot expect to reconstruct the original object. Moreover, in practice the measurements are polluted with *noise*. This noise can have a broad range of causes, the rays do not have a uniform energy level for example. Due to this contamination it might happen that there is no original object which corresponds to the measured data. In tomography one is then interested in finding a reconstruction which fits the measurements as *close as possible*. How to quantify this notion of being close to the measurements is studied in tomography.

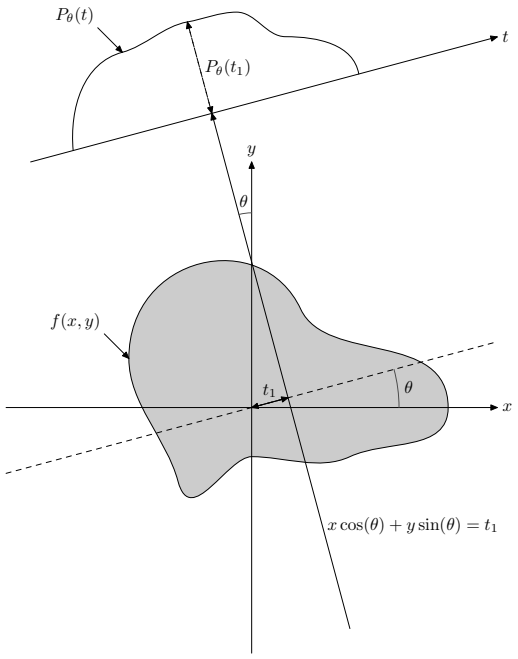


Figure 2.2: *A ray which passes through the object with attenuation $f(x, y)$, the t-axis represents the projection values of the object with projection angle θ.*

2.2 Analytical Reconstruction Methods

This section will briefly sketch the idea behind the analytical reconstruction methods. These are the methods of choice in medical imaging for they provide reasonable accurate reconstructions that require very short computation times. The analytical reconstruction methods employ the result of the *Fourier slice theorem* [16, p. 58]:

Theorem 2.1. *The Fourier transform of a parallel projection of an image $f(x, y)$ taken at angle θ gives the values of the two-dimensional Fourier transform $\hat{f}(u, v)$ along a line subtending an angle θ with the u-axis in the frequency domain.*

In Figure 2.3 one can see what the theorem entails. Taking the one-dimensional Fourier transform of the measurement data of all the projection angles will result in a star shaped area of known values in the frequency domain of $f(x, y)$, Figure 2.4. Interpolating the unknown values and taking the two-dimensional inverse Fourier transform yields a reconstruction of the original object.

The *filtered back-projection method* (FBP) is based on the Fourier slice theorem, but rather than interpolating in the frequency domain it takes the one dimensional inverse Fourier transforms of each slice separately and adds these back-projected data in the spatial domain. This implies that the interpolation is done in the spatial domain rather than the frequency domain, which, as one might suspect, leads to better results when simple interpolation techniques are used [28].

Figure 2.5 shows the workings of the FBP method. Left the original image is shown in the corresponding spatial domain. The image is scanned using 180 equally spaced projection angles in the range $[0, \pi)$. Note that taking a projection with angle 0 would yield the same data as a projection angle π (assuming no noise is present) and thus only data have to be collected

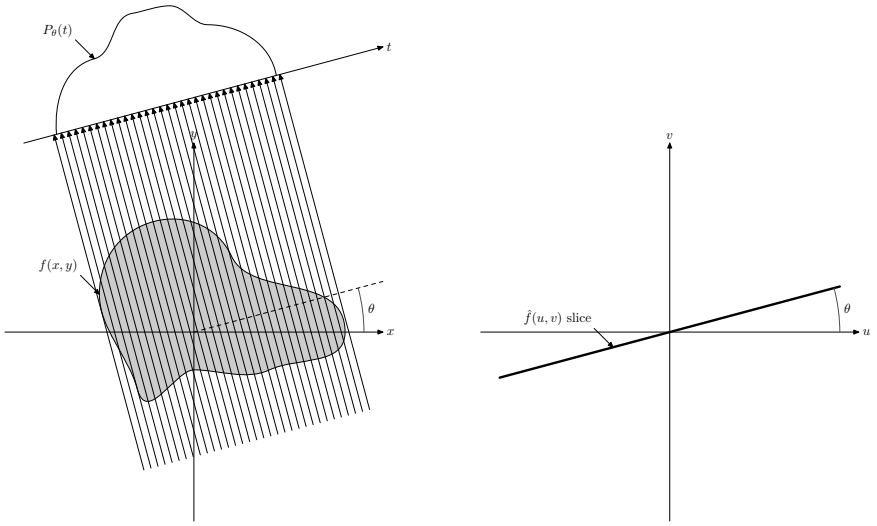


Figure 2.3: *The Fourier transform of the Projection $P_\theta(t)$ left corresponds to the line ($\hat{f}(u,v)$ slice) in the right figure.*

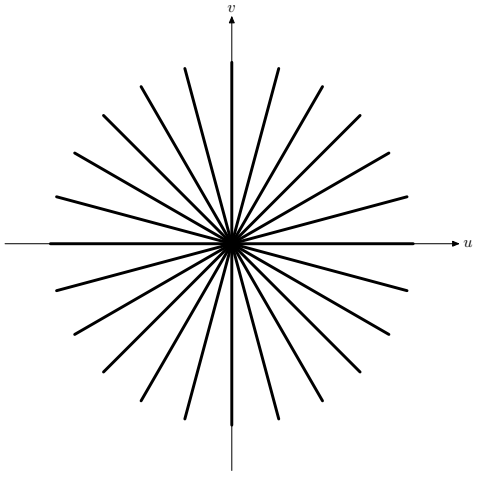


Figure 2.4: *The known frequency data from the Fourier transformed projections.*

from $[0, \pi)$. These data are called the *forward projection* of the image or simply *projection* and can be represented in a *sinogram*, the middle figure. A sinogram lists the projection data as columns, each vertical line in the image corresponds to the projection data of one particular angle θ, the variable t represents the distance on the detector, with $t = 0$ corresponding to rays going through the middle of the object (the origin). From this sinogram the reconstruction is created. In this particular example the FBP method is used with a *Ram-Lak filter* (hence *filtered* back-projection). Filters are used to suppress or amplify certain frequencies to arrive at better results.

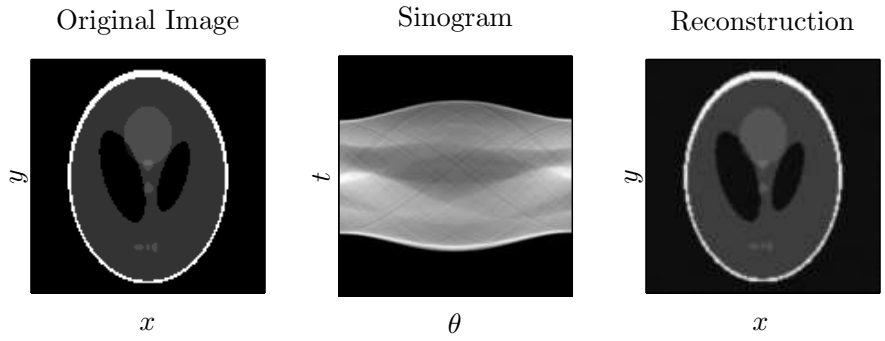


Figure 2.5: *The workings of the analytical reconstruction method FBP. Left: the original image. Middle: the sinogram of the projoction data. Right: the reconstruction obtained via FBP.*

This concludes the brief introduction into analytical methods. For a more thorough account on these methods one is referred to [29, p. 3–14], [16, p. 49–112].

2.3 Algebraic Reconstruction Methods

A different type of tomographic reconstruction is via algebraic reconstruction methods, abbreviated ARMs. In literature this class of methods is also referred to as algebraic reconstruction techniques, but one particular ARM is named the algebraic reconstruction technique (ART). Hence to avoid confusion the class is referred to as algebraic reconstruction methods. ARMs are based on the idea that the tomographic problem can be written as a system of linear equations. This system can then be solved by a huge range of methods. Although ARMs require considerable more computation time than the analytical alternatives, the algebraic methods perform better when one uses a small number of projection angles.

Consider Figure 2.6. Suppose an image $f(x, y)$ is scanned using a total of M rays. A (square) grid is superimposed on $f(x, y)$ which divides the image into $n \times n = N$ square cells, which will also be referred to as pixels. Rays $p_i, i \in \{1, \ldots, M\}$, are modelled as stripes with width τ that run through $f(x, y)$. Each cell $f_j, j \in \{1, \ldots, N\}$ has area $\delta \times \delta$, it is assumed that in each cell $f(x, y)$ is constant. The reconstruction problem is written as

$$W\mathbf{f} = \mathbf{p}. \tag{2.2}$$

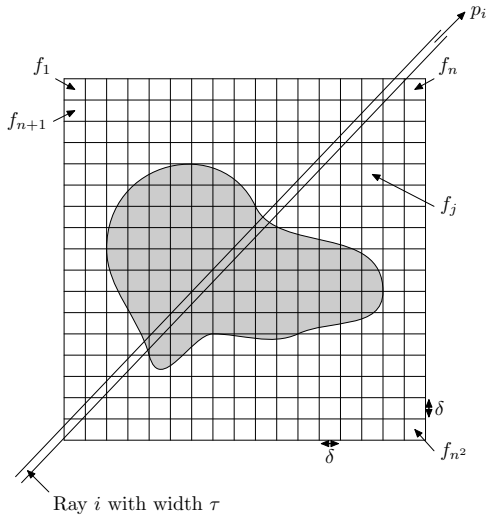


Figure 2.6: *The superimposed grid on $f(x, y)$.*

Here $\mathbf{p}$ is a vector containing all the (measured) projection values, hence the length of the vector is M. The vector $\mathbf{f}$ contains the (approximated) values of $f(x, y)$ per cell and thus is N long. The matrix W will be referred to as the weight matrix, this matrix contains per entry w_{ij} the weight cell j has corresponding to ray i, it is assumed that $w_{ij} \geq 0$. Usually w_{ij} is the length or the area of the ray running through the cell. Each of the N cells will have M weights, hence $W \in \mathbb{R}^{M \times N}$. In tomography W is often too big to construct completely due to memory limitations. For example, suppose an image is subdivided into 1024×1024 pixels and it is scanned using 52 projection angles with $1449 (= \lceil \sqrt{2 \cdot 1024^2} \rceil)$ parallel rays per angle. This would yield a matrix W of $N = 1024 \cdot 1024 = 1.048.576$ by $M = 52 \cdot 1449 = 75.348$. However, if the projection geometry is fairly structured it is possible to construct the w_{ij} on the fly during computations.

In practice $M \neq N$ and thus (2.2) is an under- or overdetermined system. In the case of an underdetermined system there are more unknowns than equations, hence if a solution exists, it is not unique. Overdetermined systems are systems with more equations than variables. An overdetermined system, if consistent and of full column rank, yields a unique solution. It might even happen that an overdetermined system has infinitely many solutions if the matrix is not of full column rank. Note that the concepts of over- and underdetermination do not provide any information about the linear (in)dependency of the equations. Sometimes overdetermined systems are, erroneously, identified as inconsistent.

Now one method to solve (2.2) will be shown. Stefan Kaczmarz was a Polish mathematician who proposed this method for solving linear systems in 1937. This method, from now the *Kaczmarz's method*, was rediscovered and first introduced in the open literature by Gordon, Bender, and Herman in 1970 who used it to solve (2.2) [14, p. 204]. The method is based on viewing the solution $\mathbf{f}$ as a point in a N dimensional space which is the intersection of M hyperplanes. The hyperplanes are described by the equations of the linear system (2.2):

$$\begin{aligned} w_{11}f_1 + w_{12}f_2 + \cdots + w_{1N}f_N &= p_1 \\ w_{21}f_1 + w_{22}f_2 + \cdots + w_{2N}f_N &= p_2 \\ &\vdots \\ w_{M1}f_1 + w_{M2}f_2 + \cdots + w_{MN}f_N &= p_M. \end{aligned} \tag{2.3}$$

Note that if $M < N$ one is faced with an underdetermined problem. Then the intersection of these planes, if it exists, is not a point but a higher dimensional object, and thus infinitely many solutions are possible. Kaczmarz's method is iterative, thus a sequence of approximate solution vectors $(\mathbf{f}^0, \mathbf{f}^1, \ldots, \mathbf{f}^k)$ is generated such that $\lim_{k\to\infty} \mathbf{f}^k = \hat{\mathbf{f}}$ assuming $\hat{\mathbf{f}}$ is the unique solution to (2.2). Starting from an initial guess $\mathbf{f}^0$ the next approximation $\mathbf{f}^1$ is obtained by projecting $\mathbf{f}^0$ orthogonal onto the hyperplane described by the first equation in (2.3). This constitutes one iteration of the method. Subsequently $\mathbf{f}^2$ is obtained by projecting the vector $\mathbf{f}^1$ orthogonal onto the hyperplane given by the second equation in (2.3). After one has projected the subsequent approximate solutions onto all the hyperplanes the result $\mathbf{f}^M$ is again projected onto the first hyperplane of the system. The process is repeated until some stopping criterion is met, e.g. the residual $\mathbf{r}^k = \mathbf{p} - W\mathbf{f}^k$ is smaller than some predetermined threshold. Figure 2.7 shows how the method works for a linear system with 2 unknowns and 2 equations.

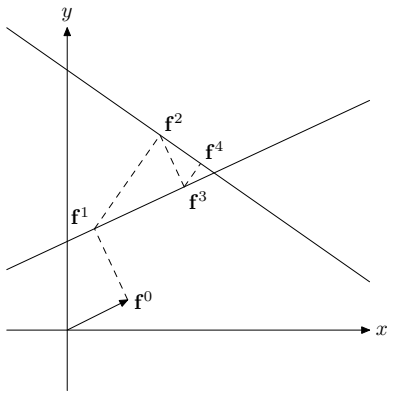


Figure 2.7: *The workings of the Kaczmarz's method in two dimensions.*

Let $\mathbf{w}_{i,:} = (w_{i1}\ w_{i2}\ \ldots\ w_{iN})^T$ be the i-th row of W such that $W = \begin{pmatrix} \mathbf{w}_{1,:}^T \\ \mathbf{w}_{2,:}^T \\ \vdots \\ \mathbf{w}_{M,:}^T \end{pmatrix}$. The i-th iteration, $i \in \{1, \ldots, M\}$, of the Kaczmarz's method can be expressed as:

$$\mathbf{f}^i = \mathbf{f}^{i-1} + \frac{p_i - \langle \mathbf{f}^{i-1}, \mathbf{w}_{i,:} \rangle}{\langle \mathbf{w}_{i,:}, \mathbf{w}_{i,:} \rangle} \mathbf{w}_{i,:}. \tag{2.4}$$

Note that this expression is composed of two vectors. The vector $\mathbf{w}_{i,:}$ is orthogonal to the hyperplane $\mathbf{w}_{i,:}^T\mathbf{f} = p_i$ given by the i-th equation of (2.3). Hence the vectors described by $\mathbf{f}^{i-1} + \alpha\mathbf{w}_{i,:}$ lie on a line which is orthogonal to that hyperplane. The quantity $\alpha = \frac{p_i - \langle \mathbf{f}^{i-1}, \mathbf{w}_{i,:} \rangle}{\langle \mathbf{w}_{i,:}, \mathbf{w}_{i,:} \rangle}$

ensures that the sum of both vectors coincides with the hyperplane, thus (2.4) projects $\mathbf{f}^{i-1}$ orthogonal onto the hyperplane $\mathbf{w}_{i,:}^T\mathbf{f} = p_i$. For a full geometric description of how this quantity α comes about one is referred to [16, p. 278–280]. Note that $i = 1, 2, \ldots, M$ and $\mathbf{f}^k$ is N long, hence, it is possible that one update alters multiple entries of $\mathbf{f}^k$. The interpretation is that the update is computed according to the projection data, thus for every ray. All the cells j are penetrated by this ray i have some weight $w_{ij} \neq 0$ and thus all those entries j in $\mathbf{f}^k$ are altered.

Note that in (2.4) $i \in \{1, \ldots, M\}$ for clarity, one can easily adapt the expression to allow for iterations with iteration number $i > M$. Set $\hat{i} = (i-1) \mod (M) + 1$, then

$$\mathbf{f}^i = \mathbf{f}^{i-1} + \frac{p_{\hat{i}} - \left\langle \mathbf{f}^{i-1}, \mathbf{w}_{\hat{i},:} \right\rangle}{\left\langle \mathbf{w}_{\hat{i},:}, \mathbf{w}_{\hat{i},:} \right\rangle} \mathbf{w}_{\hat{i},:}. \tag{2.5}$$

The denominator of (2.4) (and consequently (2.5)) can be assumed to be unequal to zero. Indeed, if $\langle \mathbf{w}_{i,:}, \mathbf{w}_{i,:} \rangle = \sum_{j=1}^{N} w_{ij}^2 = 0$ this would mean that $w_{ij} = 0$ for every j and thus equation j in (2.3) would be $\mathbf{0}^T\mathbf{f} = p_j$, with $\mathbf{0}$ the zero-vector. But this in turn represents a ray hitting no cells, consequently one can thus safely conclude that this ray is of no importance for the reconstruction. And hence removing this equation from (2.3) would not result in any loss of information.

This method only converges to a solution of $W\mathbf{f} = \mathbf{p}$ if the system has an exact solution [30, p. 59]. Unfortunately the projections obtained in tomography often contain noise which causes $\mathbf{p}$ not to lie in the range of W and thus no solution exists. In this case the sequence of approximate solutions will keep fluctuating. And even if convergence is guaranteed, it does not mean that a solution is found quickly. Consider for example a system with two unknowns (cf. Figure 2.7) where the two hyperplanes (lines) differ by a very small angle. The Kaczmarz's method would then convergence very slowly to the solution. If the two lines were perpendicular then convergence would be reached within two iterations. One can imagine that hyperplanes arising from two adjacent rays will most likely be nearly parallel because of high correlation between information, hence it would be better to project onto the hyperplanes in a different order, for example random, than in ascending order.

The range of W are all the projections arising from scanning an object with the operator W. One can easily imagine that it is possible that multiple objects result in the same projection when scanned. Imagine $\mathcal{R}(W)$, the range of W, as a plane in some three-dimensional space. The operator W, arising from the discretization of projection geometry, can only construct the projections in $\mathcal{R}(W)$. The measured projection $\mathbf{p}$, however, might not be an element of this range, due to noise for example. In that case one cannot find an object $\mathbf{f}$ for which $W\mathbf{f} = \mathbf{p}$, i.e. the problem has no exact solution. One then often resorts to finding a *least squares solution*. This solution is a vector for which the residual has minimal norm, i.e. $\|\mathbf{p} - W\mathbf{f}\|_2$ is minimal, see Figure 2.8. In the problems considered in this work one generally also has that W is not of full column rank, in that case the least squares solution is not unique. There is, however, a unique *minimal norm least squares solution*. This particular vector is the solution one is interested in. In Chapter 3 various ARM algorithms will be presented which attempt to find this solution.

Now some properties of the least squares solution will be given assuming that W is not of full rank. From Figure 2.8 it is clear that the vector $W\mathbf{f}' - \mathbf{p}$ is orthogonal to $\mathcal{R}(W)$ and, indeed, if $\mathbf{f}$ is a least squares solution of (2.2) then one has that

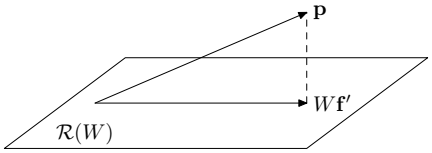


Figure 2.8: *The vector* $\mathbf{f}'$ *is a least squares solution for the problem* $W\mathbf{f} = \mathbf{p}$.

$$
\begin{aligned}
\frac{\partial}{\partial f_j} \|W\mathbf{f} - \mathbf{p})\|^2 &= 0 && \forall j \in \{1, 2, \ldots, N\} \Leftrightarrow \\
\frac{\partial}{\partial f_j} \langle W\mathbf{f} - \mathbf{p}, W\mathbf{f} - \mathbf{p}\rangle &= 0 && \forall j \in \{1, 2, \ldots, N\} \Leftrightarrow \\
\frac{\partial}{\partial f_j} \left(\langle W\mathbf{f}, W\mathbf{f}\rangle - 2\langle W\mathbf{f}, \mathbf{p}\rangle - \langle \mathbf{p}, \mathbf{p}\rangle\right) &= 0 && \forall j \in \{1, 2, \ldots, N\} \Leftrightarrow \\
2\langle \mathbf{w}_{:,j}, W\mathbf{f}\rangle - 2\langle \mathbf{w}_{:,j}, \mathbf{p}\rangle &= 0 && \forall j \in \{1, 2, \ldots, N\} \Leftrightarrow \\
\langle \mathbf{w}_{:,j}, (W\mathbf{f} - \mathbf{p})\rangle &= 0 && \forall j \in \{1, 2, \ldots, N\} \Leftrightarrow \\
W^T (W\mathbf{f} - \mathbf{p}) &= \mathbf{0} && \Leftrightarrow \qquad (2.6) \\
\langle W^T(W\mathbf{f} - \mathbf{p}), \mathbf{q}\rangle &= 0 && \forall \mathbf{q} \in \mathbb{R}^N \Leftrightarrow \\
\langle W\mathbf{f} - \mathbf{p}, W\mathbf{q}\rangle &= 0 && \forall \mathbf{q} \in \mathbb{R}^N \Leftrightarrow \\
W\mathbf{f} - \mathbf{p} &\perp \mathcal{R}(W). &&
\end{aligned}
$$

Equation (2.6) can be written as:

$$W^T(W\mathbf{f} - \mathbf{p}) = \mathbf{0} \Leftrightarrow W^T W\mathbf{f} = W^T\mathbf{p}. \tag{2.7}$$

From (2.7) it follows that $\mathbf{f}$ is a least squares solution if and only if $W^T W\mathbf{f} = W^T\mathbf{p}$. This system of equations is known as the *normal equations*.

Note that if $\mathbf{f}$ is a least squares solution of $W\mathbf{f} = \mathbf{p}$ and $\mathcal{N}(W)$ is the *null space* of W that then all other least squares solutions can be written as $\mathbf{f} + \mathbf{q}$ with $\mathbf{q} \in \mathcal{N}(W)$ since $W(\mathbf{f} + \mathbf{q}) = W\mathbf{f} + W\mathbf{q} = W\mathbf{f}$. From this one can conclude that $\mathbf{f}$ is the *minimum norm* least squares solution of (2.2) if and only if $\mathbf{f} \perp \mathcal{N}(W)$ which is equivalent to $\mathbf{f} \in \mathcal{R}(W^T)$.

2.4 Noise, Regularization and Semi-Convergence

Rays, originating from a X-ray tube for example, are modelled as uniform continuous streams of particles. The intensity of the ray can be expressed in a *number of counts*, i.e. the number of photons per time unit. In practice, however, the source does not emit the particles uniformly but rather according to some distribution, such as a Poisson distribution. This poses a problem on the detector side since now the number of incoming photons is depending on both the attenuation of the object as the number of photons emitted by the source, which is unknown. Stronger sources, with a high number of counts are more uniform than weak sources. When one sends a

ray through the object $f(x, y)$ travelling along the line l, the number of photons initially send through the object N_{in} and the measured number of photons N_{out} are related as

$$N_{out} = N_{in} e^{-\int_l f(x,y) \mathrm{d}l}. \tag{2.8}$$

From this one can conclude that

$$\int_l f(x, y) \mathrm{d}l = \log \frac{N_{in}}{N_{out}}. \tag{2.9}$$

The left-hand side represents the measured attenuation and corresponds to the measured data. In the upcoming experiments noise will be artificially added to the measured data. This is carried out by computing N_{out} for all the rays separately with equation (2.8) and adding some noise to it according to a Poisson distribution. Subsequently, the noisy sinogram is obtained using (2.9). Higher number of counts are less affected by the noise than low number of counts using this method since the quantity of the amount of added noise is independent of the magnitude of the number of counts and thus constant for all numbers of counts. In what follows noise levels will be identified with the number of counts, typically the noise levels 10^6 (high number of counts thus little noise), $10^5, 10^4$ and 10^3 (low number of counts thus much noise) are used.

There are also other phenomena such as *beam-hardening* (photons of different energy levels are not absorbed uniformly), *electronic noise* (the used electrical circuits add noise to the measurement signals) and many others that cause the measurements to deviate from what one expects from the model [13]. Many methods exist to counter these phenomena as good as possible, but unfortunately one is never able to remove all the noise from the measurements. From a mathematical point of view one can model the noise as some perturbation $\boldsymbol{\epsilon}$ in the measured projection data $\mathbf{p}$, thus if the ideal projection is $\hat{\mathbf{p}}$ then $\mathbf{p} = \hat{\mathbf{p}} + \boldsymbol{\epsilon}$.

To arrive at better reconstructions that suffer less from noise *regularization* can be used. Regularization is the use of additional information to make an ill-conditioned problem well-conditioned or at least less ill-conditioned. This entails that one finds superior reconstructions via regularization and thus it is a very broad concept. There are many types of regularization and none of these methods are superior to all the other methods. Rather, each method has its advantages, depending on the properties of the system it is applied to [11, p. 2].

Ill-conditioned problems can roughly be categorized into two classes [11, p. 2]: rank-deficient problems and discrete ill-posed problems. The former class is characterized by a system matrix W which has a cluster of small singular values and there is a well-determined gap between the larger and small singular values. The latter has singular values which, on average, decay gradually to zero. Small singular values of the system can cause the components of the approximate solution to have large errors [30, p. 54]. This can be circumvented by means of regularization. Both classes require other types of regularization. Tomographic problems are typically rank-deficient.

Tikhonov regularization was independently proposed by Phillips (1962) and Tikhonov (1963). Instead of solving the system $W\mathbf{f} = \mathbf{p}$ the system

$$\begin{pmatrix} W \\ \lambda I \end{pmatrix} \mathbf{f} = \begin{pmatrix} \mathbf{p} \\ \mathbf{0} \end{pmatrix} \tag{2.10}$$

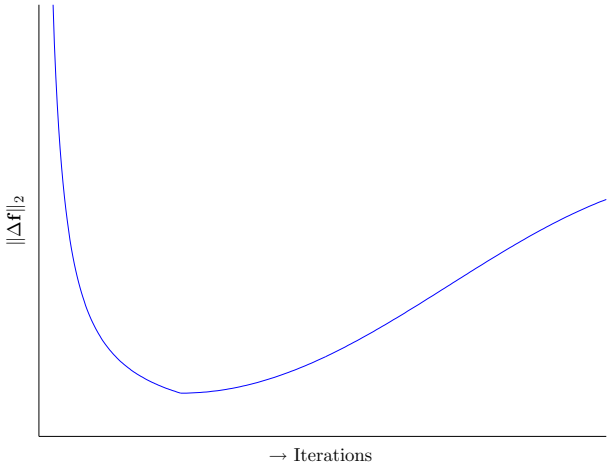


Figure 2.9: *The development of the error norm for an increasing number of iterations.*

is solved for a certain λ. Observe that this system gives preference to solutions with a smaller norm, the value of λ gives a measure for this preference. For more information about Tikhonov and other regularization methods one is referred to [11, Ch 5,6].

When solving an ill-conditioned linear system of equations one is interested in finding the best approximate solution. Perturbations, such as noise, in the right-hand side might cause the error of the approximation to explode [29, p. 32–33]. By means of regularization this can be circumvented. Suppose the model solution (the original object) is $\hat{\mathbf{f}}$ and the pure measurement data, thus without any noise, is $\hat{\mathbf{p}}$. Due to the rank deficiency and discretisation of the model one can never hope to find this model solution when solving $W\mathbf{f} = \hat{\mathbf{p}}$, rather one would find some solution $\tilde{\mathbf{f}}$. Let the right-hand side be perturbed by some noise $\boldsymbol{\epsilon}$ such that $\mathbf{p} = \hat{\mathbf{p}} = \boldsymbol{\epsilon}$. Solving the system $W\mathbf{f} = \mathbf{p}$ would then yield a solution $\mathbf{f}$ which will be different from both $\hat{\mathbf{f}}$ and $\tilde{\mathbf{f}}$. For the error $\Delta\mathbf{f} = \hat{\mathbf{f}} - \mathbf{f}$ one makes one has

$$\|\hat{\mathbf{f}} - \mathbf{f}\| \leq \|\hat{\mathbf{f}} - \tilde{\mathbf{f}}\| + \|\tilde{\mathbf{f}} - \mathbf{f}\|. \tag{2.11}$$

The first term $\hat{\mathbf{f}} - \tilde{\mathbf{f}}$ represents the *approximation error* $\Delta\mathbf{f}^{appr}$ and is ever-present, even if there is no perturbation in the data. This error depends on how the solution is obtained and thus also on the regularization method that is used. The second term $\tilde{\mathbf{f}} - \mathbf{f}$ is the *perturbation error* $\Delta\mathbf{f}^{pert}$ and is a result of the noise that was present in the system.

Typically, when solving a perturbed ill-posed problem with an iterative method, the approximation error dominates during the initial iterations and thus the perturbation error is negligible. Initially the approximation becomes more accurate, thus the approximation error decreases. But eventually the perturbation error will dominate and might even increases [12]. The norm of the error $\|\Delta\mathbf{f}\|_2$ will thus decrease for the initial iterations but will increase after a certain number of iterations. In Figure 2.9 one can see a sketch of the development of the error norm. This phenomenon is coined *semi-convergence* by Natterer in 1986.

For a more thorough analysis one is referred to [30, Ch 2.7].

Chapter 3

ARM Algorithms

The Kaczmarz's method introduced in the previous chapter is one method to solve the system $W\mathbf{f} = \mathbf{p}$. But many other iterative algorithms exist that can find an approximate solution to the problem. This chapter will present 7 different ARM algorithms. The first three methods, ART, SIRT and SART, are typically used in tomography. ART is simply Kaczmarz's method and SIRT and SART are adaptations of this method. The remaining methods, CGLS, CGNE, LSQR and LSMR, are more widely used to solve underdetermined systems. This chapter will be concluded with an analysis of the convergent behaviour of the methods.

3.1 ART

As stated earlier: ART is the same as Kaczmarz's method. Here the ART algorithm will be derived from another point of view and it will be shown that ART is actually the Gauss-Seidel method applied to system

$$WW^T\mathbf{u} = \mathbf{p} \tag{3.1}$$

of the alternative normal equations. Note that one solves this system for $\mathbf{u}$ and then finds the approximate solution $\mathbf{f} = W^T\mathbf{u}$. This derivation is due to [26, p.247–248] and will closely follow the reasoning in said source. Intuitively one can see that ART is similar to Gauss-Seidel in that it uses all the newly obtained information for the update of the i-th component.

Solving (3.1) with Gauss-Seidel yields for i-th update $\mathbf{u}^k = \mathbf{u}^{k-1} + \delta_i \mathbf{e}_i$, where $\mathbf{e}_i$ is the i-th unit vector and $i = (k-1) \mod (M) + 1$ (cf. $\tilde{i}$ in (2.5)). The scaler δ_i is chosen such that the i-th component of the residual vector $r_i^k = (\mathbf{p} - WW^T\mathbf{u}^k)_i = 0$, or equivalently

$$\begin{aligned}
\Big(\mathbf{p} - WW^T(\mathbf{u}^{k-1} + \delta_i \mathbf{e}_i)\Big)_i &= 0 && \Rightarrow \\
\Big\langle \mathbf{p} - WW^T(\mathbf{u}^{k-1} + \delta_i \mathbf{e}_i), \mathbf{e}_i \Big\rangle &= 0 && \Rightarrow \\
\langle WW^T \delta_i \mathbf{e}_i, \mathbf{e}_i \rangle &= \Big\langle \mathbf{r}^{k-1}, \mathbf{e}_i \Big\rangle && \Rightarrow \\
\delta_i \langle W^T \mathbf{e}_i, W^T \mathbf{e}_i \rangle &= \Big\langle \mathbf{r}^{k-1}, \mathbf{e}_i \Big\rangle && \Rightarrow \\
\delta_i &= \frac{\langle \mathbf{r}^{k-1}, \mathbf{e}_i \rangle}{\|W^T \mathbf{e}_i\|_2^2}
\end{aligned}$$

To analyse this, re-substitute the definition of $\mathbf{r}^{k-1}$. This gives

$$\delta_i = \frac{p_i - \langle W^T \mathbf{u}^{k-1}, W^T \mathbf{e}_i \rangle}{\|W^T \mathbf{e}_i\|_2^2}. \tag{3.2}$$

Since $\mathbf{f} = W^T \mathbf{u}$ one finds $\mathbf{f}^k = \mathbf{f}^{k-1} + \delta_i W^T \mathbf{e}_i$:

$$\mathbf{f}^k = \mathbf{f}^{k-1} + \frac{p_i - \langle \mathbf{f}^{k-1}, W^T \mathbf{e}_i \rangle}{\|W^T \mathbf{e}_i\|_2^2} W^T \mathbf{e}_i. \tag{3.3}$$

Note that the updates are performed in ascending order $i = 1, 2, \ldots, M$ and are immediately stored in the new vector $\mathbf{f}^k = W^T \mathbf{u}^k$, hence this yields a Gauss-Seidel method which is applied to the system $WW^T \mathbf{u} = \mathbf{p}$. Inspection of (3.3) confirms that this expression is actually equivalent to (2.4). Simply observe that $W^T \mathbf{e}_i = \mathbf{w}_{i,:}$, $\langle \mathbf{f}^{k-1}, W^T \mathbf{e}_i \rangle = \langle \mathbf{f}^{k-1}, \mathbf{w}_{i,:} \rangle$ and $\|W^T \mathbf{e}_i\|_2^2 = \langle \mathbf{w}_{i,:}, \mathbf{w}_{i,:} \rangle$. The ART algorithm, in pseudo code, is listed below.

ART

Choose $\mathbf{f}^0$
$k = 1$
while termination criterion is not met **do**
  for $i = 1, 2, \ldots, M$ **do**
    $\delta_i = \left(p_i - \langle \mathbf{f}^{k-1}, \mathbf{w}_{i,:} \rangle\right) / \langle \mathbf{w}_{i,:}, \mathbf{w}_{i,:} \rangle$
    $\mathbf{f}^k = \mathbf{f}^{k-1} + \delta_i \mathbf{w}_{i,:}$
    $k = k + 1$
  end for
end while

Note that it is unclear what one iteration of the ART algorithm constitutes, from now on one iteration of the ART algorithm will be regarded as executing the while-loop once (thus updating all the components). Recall from the discussion at the end of Section 2.3 that updating the components in random order order rather than ascending order, as in the algorithm above, will yield faster convergence. This adaptation is easily implemented, one only has to change to order in which the for-loop traverses the various rays.

3.2 SIRT

SIRT is differs from ART in that it updates $\mathbf{f}^{k-1}$ to $\mathbf{f}^k$ only after all the N new components have been computed, this suggests a Jacobi method. Indeed, SIRT actually stems from Jacobi applied to the alternative normal equations (3.1).

In the previous section it was established that in ART the update for the j-th component was

$$\delta_i = \frac{\langle \mathbf{r}^{k-1}, \mathbf{e}_i \rangle}{\|W^T \mathbf{e}_i\|_2^2}. \tag{3.4}$$

Instead of the Euclidean norm in the denominator, the 1-norm is used for SIRT, i.e. $\|\mathbf{x}\|_1 = \sum_i |x_i|$. This yields:

$$\delta_i = \frac{\langle \mathbf{r}^{k-1}, \mathbf{e}_i \rangle}{\|W^T \mathbf{e}_i\|_1^2}. \tag{3.5}$$

Since one deals with a Jacobi iteration the actual update is performed after all the other updates have been computed, thus for every update the same residual is used. After the updates have been computed for $i = 1, \ldots, M$ one finds the new approximation as

$$\mathbf{d}^{k-1} = \sum_{i=1}^{M} \frac{\langle \mathbf{r}^{k-1}, \mathbf{e}_i \rangle}{\sum_{h=1}^{N} |w_{ih}|} \mathbf{w}_{i,:} \tag{3.6}$$

$$\mathbf{f}^k = \mathbf{f}^{k-1} + C\mathbf{d}^{k-1} \tag{3.7}$$

Recall that $w_{ij} \geq 0$ was assumed. Note that in this case every component (pixel) j, will receive updates from multiple rays i, these are precisely the rays which are affected by that pixel. At the end of the cycle, SIRT averages over these updates proportional to total weight the cell has, thus proportional to the j-th column sum of W, i.e. C in (3.7) is a diagonal matrix containing the inverse of these sums on its diagonal: $c_j = 1/\sum_{i=1}^{M} w_{ij}$. Clearly, if such a column sum is equal to zero one would have that no ray passes through cell j and thus it would make no sense to have this variable in the system since the cell is not part of the scanned image. And indeed, if one were to incorporate this cell (and thus the zero column) into the system then one would never find a unique solution, for if the system without this variable has a unique solution, the system with this variable has infinitely many solutions, namely the unique solution found earlier where one can freely choose the extra variable. The new averaged update for component j is given by

$$f_j^k = f_j^{k-1} + \frac{1}{\sum_{i=1}^{M} w_{ij}} \sum_{i=1}^{M} \frac{w_{ij} r_i^{k-1}}{\sum_{h=1}^{N} w_{ih}}. \tag{3.8}$$

Although this expression looks messy, it is actually quite elegant. Moreover, one SIRT iteration, i.e. computing the increments for all the components and obtaining the new approximation $\mathbf{f}^k$, can be conveniently written in matrix-vector notation. Let $R \in \mathbb{R}^M, C \in \mathbb{R}^N$ be a diagonal

matrices with $r_i = 1/\sum_{j=1}^{N} w_{ij}$ and $c_j = 1/\sum_{i=1}^{M} w_{ij}$. Note that $\sum_{i=1}^{M} w_{ij} > 0$ and $\sum_{j=1}^{N} w_{ij} > 0$. With these matrices one SIRT iteration can be written as

$$\mathbf{f}^k = \mathbf{f}^{k-1} + CW^T R\mathbf{r}^{k-1}. \tag{3.9}$$

And hence the algorithm can be formulated as

SIRT

Choose $\mathbf{f}^0$
$\mathbf{r}^0 = \mathbf{p} - W\mathbf{f}^0$
$k = 1$
while termination criterion is not met **do**
  $\mathbf{f}^k = \mathbf{f}^{k-1} + CW^T R\mathbf{r}^{k-1}$
  $\mathbf{r}^k = \mathbf{p} - W\mathbf{f}^k$
  $k = k + 1$
end while

From the above algorithm it is clear that executing the while-loop once constitutes one SIRT iteration.

3.3 SART

The Simultaneous Algebraic Reconstruction Technique is combination of ART and SIRT. In ART the update of $\mathbf{f}^k$ was determined by only looking at the contribution of the i-th ray for $i = 1, 2, \ldots, M$. SIRT computed the average contribution of all the rays i, $i = 1, 2, \ldots, M$, first, using the previous approximation $\mathbf{f}^{k-1}$, and then did an update to $\mathbf{f}^k$. SART is a combination of ART and SIRT in that the update of $\mathbf{f}^k$ is carried out per projection angle. Suppose the object is scanned using the angles $P = \{\theta_1, \theta_2, \ldots, \theta_p\}$ with R rays per angle such that $p \cdot R = M$. Then the update for $\mathbf{f}^k$ is computed for all rays per projection angle, thus the updates are performed per block of R projections. Hence this method is also referred to as a *block iterative method*. For the l-th block this results into the following expression for the j-th component of the next approximation update:

$$f_j^k = f_j^{k-1} + \frac{1}{\sum_{i=R\cdot(l-1)+1}^{R\cdot l} w_{ij}} \sum_{i=R\cdot(l-1)+1}^{R\cdot l} \frac{r_i^{k-1} w_{ij}}{\sum_{h=1}^{N} w_{ih}}. \tag{3.10}$$

Let C be a diagonal matrix with diagonal elements $c_j = 1/\sum_{i=R\cdot(l-1)+1}^{R\cdot l} w_{ij}$. The following algorithm results

SART

Choose $\mathbf{f}^0$
$\mathbf{r}^0 = \mathbf{p} - W\mathbf{f}^0$
$k = 1$
while termination criterion is not met **do**
  for $l = 1, 2, \ldots, p$ **do**
    for $i = R \cdot (l-1) + 1, R \cdot (l-1) + 2, \ldots, R \cdot l$ **do**
      $\mathbf{f}^k = \mathbf{f}^{k-1} + C \frac{r_i^{k-1}}{\sum_{h=1}^{N} w_{ih}} \mathbf{w}_{i,:}$
    end for
    $\mathbf{r}^k = \mathbf{p} - W\mathbf{f}^k$
    $k = k + 1$
  end for
end while

Note that, as with the ART algorithm, it is unclear what constitutes one iteration. Also in this case one iteration is defined as executing the while-loop once, thus after all p blocks have been used.

3.4 CGLS

The conjugate gradient method is widely used to solve sparse linear systems $A\mathbf{x} = \mathbf{b}$ with A a symmetric positive definite (SPD) matrix. In essence CG uses the (shifted) Krylov subspace $\mathbf{x}^0 + \mathcal{K}_r(A, \mathbf{r}^0)$ to construct subsequent approximations, with $A \in \mathbb{R}^{n \times n}$ SPD. Here $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$, the residual corresponding to the initial guess $\mathbf{x}^0$. The method was first proposed by M. R. Hestenes and E. Stiefel in 1952 to solve SPD systems. The CG algorithm will now be presented without any further introduction or derivation. For more details about the algorithm the reader is referred to [26, Ch. 6.7].

CG

Choose $\mathbf{x}^0$
$\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$
$\mathbf{d}^0 = \mathbf{r}^0$
for $j = 1, 2, \ldots$ until convergence **do**
  $\alpha_{j-1} = \langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle / \langle A\mathbf{d}^{j-1}, \mathbf{d}^{j-1} \rangle$
  $\mathbf{x}^j = \mathbf{x}^{j-1} + \alpha_{j-1}\mathbf{d}^{j-1}$
  $\mathbf{r}^j = \mathbf{r}^{j-1} - \alpha_{j-1} A\mathbf{d}^{j-1}$
  $\beta_{j-1} = \langle \mathbf{r}^j, \mathbf{r}^j \rangle / \langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle$
  $\mathbf{d}^j = \mathbf{r}^j + \beta_{j-1}\mathbf{d}^{j-1}$
end for

CG chooses α_j and β_j in such a way the the subsequent residuals $\mathbf{r}^0, \mathbf{r}^1, \ldots$ are mutually orthogonal and hence the CG method is guaranteed to produce the exact solution to the system after at most n iterations (where n is the size of the problem). Unfortunately, due to round-off errors this may, and often will, fail in practice. Fortunately, however, it can be shown that the approx-

imate solution $\mathbf{x}^j$ minimizes the so called *energy norm* $\|\mathbf{x}^* - \mathbf{x}^j\|_A = \sqrt{\langle A(\mathbf{x}^* - \mathbf{x}^j), \mathbf{x}^* - \mathbf{x}^j\rangle}$ where $\mathbf{x}^*$ is the exact solution [26, Ch. 6.11.3]. CG thus produces a sequence of monotonically improving (in A-norm) solutions. In practice n is very big and it is very costly to perform that many iterations, but, luckily, the results after $k << n$ are often accurate enough.

The normal equations applied to the ill-conditioned system $W\mathbf{f} = \mathbf{p}$ yields $W^TW\mathbf{f} = W^T\mathbf{p}$, which is an SPD system and hence the CG method can be used to find a, possibly approximate, solution to the problem. This means that the approximate solution minimizes $\|\mathbf{f}^* - \mathbf{f}^j\|_{W^TW}$ where $\mathbf{f}^*$ is an exact solution. This term might seem puzzling since one can have multiple exact (i.e. least squares) solutions to the original system. However, for each of these $\mathbf{f}^*$ one has the same energy norm $\|\mathbf{f}^*\|_{W^TW}$ (see the discussion at the end op Section 2.3) and thus the term yields the same value for all the exact solutions. Analysis of this term results in

$$\begin{aligned}
\|\mathbf{f}^* - \mathbf{f}^j\|^2_{W^TW} &= \left\langle W^TW\left(\mathbf{f}^* - \mathbf{f}^j\right), \mathbf{f}^* - \mathbf{f}^j\right\rangle \\
&= \left\langle W\left(\mathbf{f}^* - \mathbf{f}^j\right), W\left(\mathbf{f}^* - \mathbf{f}^j\right)\right\rangle \\
&= \left\langle \mathbf{p} - W\mathbf{f}^j, \mathbf{p} - W\mathbf{f}^j\right\rangle \\
&= \|\mathbf{p} - W\mathbf{f}^j\|_2^2.
\end{aligned}$$

Thus CG applied to the normal equations of (2.2) minimizes the residual of the original problem. Hence this method is called CGLS (LS for Least Squares). Sometimes this algorithm is also called CGNR (NR for Normal Residuals). The corresponding algorithm is virtually the same as the original CG but the residual is replaced by $\mathbf{z}^j = W^T\mathbf{r}^j$. This yields the following steps for one CGLS iteration:

- $\alpha_{j-1} = \left\langle \mathbf{z}^{j-1}, \mathbf{z}^{j-1}\right\rangle / \left\langle W^TW\mathbf{d}^{j-1}, \mathbf{d}^{j-1}\right\rangle = \left\langle \mathbf{z}^{j-1}, \mathbf{z}^{j-1}\right\rangle / \left\langle W\mathbf{d}^{j-1}, W\mathbf{d}^{j-1}\right\rangle$
- $\mathbf{f}^j = \mathbf{f}^{j-1} + \alpha_{j-1}\mathbf{d}^{j-1}$
- $\mathbf{z}^j = \mathbf{z}^{j-1} - \alpha_{j-1}W^TW\mathbf{d}^{j-1}$
- $\beta_{j-1} = \left\langle \mathbf{z}^j, \mathbf{z}^j\right\rangle / \left\langle \mathbf{z}^{j-1}, \mathbf{z}^{j-1}\right\rangle$
- $\mathbf{d}^j = \mathbf{z}^j + \beta_{j-1}\mathbf{d}^j$

Note that the first term α_j is best computed using $\left\langle \mathbf{w}^{j-1}, \mathbf{w}^{j-1}\right\rangle$ with $\mathbf{w}^{j-1} = W\mathbf{d}^{j-1}$ since this vector can then be reused once computed. This new vector $\mathbf{w}^{j-1}$ also enables one the compute the residual of the normal equations $\mathbf{z}^j$ as $\mathbf{z}^j = W^T\mathbf{r}^j$ since $\mathbf{r}^j = \mathbf{r}^{j-1} - \alpha_{j-1}\mathbf{w}^{j-1}$. The numerical stability of the method is improved by this construct [5] and one has the actual residual at its disposal in every iteration. The whole algorithm can be written as:

CGLS

Choose $\mathbf{f}^0$
$\mathbf{r}^0 = \mathbf{p} - W\mathbf{f}^0$
$\mathbf{z}^0 = W^T\mathbf{r}^0$
$\mathbf{d}^0 = \mathbf{z}^0$
for $j = 1, \ldots$ until convergence **do**
  $\mathbf{w}^{j-1} = W\mathbf{d}^{j-1}$
  $\alpha_{j-1} = \langle \mathbf{z}^{j-1}, \mathbf{z}^{j-1} \rangle / \langle \mathbf{w}^{j-1}, \mathbf{w}^{j-1} \rangle$
  $\mathbf{f}^j = \mathbf{f}^{j-1} + \alpha_{j-1}\mathbf{d}^{j-1}$
  $\mathbf{r}^j = \mathbf{r}^{j-1} - \alpha_{j-1}\mathbf{w}^{j-1}$
  $\mathbf{z}^j = W^T\mathbf{r}^j$
  $\beta_{j-1} = \langle \mathbf{z}^j, \mathbf{z}^j \rangle / \langle \mathbf{z}^{j-1}, \mathbf{z}^{j-1} \rangle$
  $\mathbf{d}^j = \mathbf{z}^j + \beta_{j-1}\mathbf{d}^{j-1}$
end for

3.5 CGNE

When appling CG to the alternative normal equations $WW^T\mathbf{u} = \mathbf{p}$ an error-minimizing method known as CGNE (NE for Normal Error) or Craig's method results. This algorithm will precisely be the CG algorithm with $A = WW^T$ and $\mathbf{d}^j = W^T\mathbf{q}^j$. Thus one CGNE iteration equals:

- $\alpha_{j-1} = \dfrac{\langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle}{\langle WW^T\mathbf{q}^{j-1}, \mathbf{q}^{j-1} \rangle} = \dfrac{\langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle}{\langle W^T\mathbf{q}^j, W^T\mathbf{q}^j \rangle}$
- $\mathbf{u}^j = \mathbf{u}^{j-1} + \alpha_{j-1}\mathbf{q}^{j-1}$
- $\mathbf{r}^j = \mathbf{r}^{j-1} - \alpha_{j-1}WW^T\mathbf{q}^{j-1}$
- $\beta_{j-1} = \dfrac{\langle \mathbf{r}^j, \mathbf{r}^j \rangle}{\langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle}$
- $\mathbf{q}^j = \mathbf{r}^j + \beta_{j-1}\mathbf{q}^{j-1}$.

Note that since $\mathbf{f}^j = W^T\mathbf{u}^j$ the algorithm can also be expressed in terms of $\mathbf{f}^j$ and $\mathbf{d}^j$ instead of $\mathbf{u}^j$ and $\mathbf{q}^j$. The resulting full algorithm can then be written as:

CGNE

Choose $\mathbf{f}^0$
$\mathbf{r}^0 = \mathbf{p} - W\mathbf{f}^0$
$\mathbf{d}^0 = W^T\mathbf{r}^0$
for $j = 1, 2, \ldots$ until convergence **do**
  $\alpha_{j-1} = \langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle / \langle \mathbf{d}^{j-1}, \mathbf{d}^{j-1} \rangle$
  $\mathbf{f}^j = \mathbf{f}^{j-1} + \alpha_{j-1}\mathbf{d}^{j-1}$
  $\mathbf{r}^j = \mathbf{r}^{j-1} - \alpha_{j-1}W\mathbf{d}^{j-1}$
  $\beta_{j-1} = \langle \mathbf{r}^j, \mathbf{r}^j \rangle / \langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle$
  $\mathbf{d}^j = W^T\mathbf{r}^j + \beta_{j-1}\mathbf{d}^{j-1}$
end for

CGNE minimizes the $\|\mathbf{u}^* - \mathbf{u}^j\|_{WW^T}$ where $\mathbf{u}^*$ is the exact solution to $WW^T\mathbf{u} = \mathbf{p}$. Analysis of this term gives

$$\begin{aligned}\|\mathbf{u}^* - \mathbf{u}^j\|^2_{WW^T} &= \left\langle WW^T\left(\mathbf{u}^* - \mathbf{u}^j\right), \mathbf{u}^* - \mathbf{u}^j\right\rangle \\ &= \left\langle W^T\left(\mathbf{u}^* - \mathbf{u}^j\right), W^T\left(\mathbf{u}^* - \mathbf{u}^j\right)\right\rangle \\ &= \left\langle \mathbf{f}^* - \mathbf{f}^j, \mathbf{f}^* - \mathbf{f}^j\right\rangle \\ &= \|\mathbf{f}^* - \mathbf{f}^j\|_2^2.\end{aligned}$$

with $\mathbf{f}^*$ an exact solution to this original problem. Thus CGNE is an error minimizing method.

Note that CGLS finds a solution in the shifted Krylov subspace $\mathbf{f}^0 + \mathcal{K}_j(W^TW, W^T\mathbf{r}^0)$. CGNE finds vectors $\mathbf{u}^j$ in the shifted Krylov subspace $\mathbf{u}^0 + \mathcal{K}_j(WW^T, \mathbf{r}^0)$. Since $\mathbf{f}^j = W^T\mathbf{u}^j$ one can conclude that this subspace is equal to $\mathbf{f}^0 + W^T\mathcal{K}_j(WW^T, \mathbf{r}^0) = \mathbf{f}^0 + \mathcal{K}_j(W^TW, W^T\mathbf{r}^0)$. Thus both methods find a (in general different) solution $\mathbf{f}^j$ in the same subspace, but both solutions satisfy different optimality conditions.

3.6 LSQR

The LSQR (LS for Least Squares and QR because of a QR decomposition) algorithm was developed by C. Paige and M. Saunders in 1982 and, like all the methods considered here, is used to iteratively solve a system $A\mathbf{x} = \mathbf{b}$ for sparse matrices A that may be rectangular. Analytically the method is equivalent to CGLS but possesses better numerical properties. LSQR is celebrated for its robustness.

The method is based on the application of the Lanczos method to the auxiliary system

$$\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix}\begin{pmatrix} \mathbf{r} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \tag{3.11}$$

with starting vector $\mathbf{u}^1 = \dfrac{1}{\|\mathbf{b}\|}\begin{pmatrix} \mathbf{r} \\ \mathbf{x} \end{pmatrix}$.

Furthermore, the Golub-Kahan bidiagonalisation process and the QR-algorithm are used in LSQR. The norm of the residual $\|\mathbf{r}^k\|$ is reduced monotonically. For more information about the method one is referred to the original paper [22].

3.7 LSMR

The LSMR algorithm is very similar to the LSQR algorithm in that it uses the Golub-Kahan bidiagonalisation. Whereas LSQR is equivalent to CGLS and has monotonically reducing residuals $\|\mathbf{r}^k\|$, LSMR is equivalent to MINRES [21] applied to the normal equations and consequently the quantities $\|A^T\mathbf{r}^k\|$ are monotonically decreasing. CGLS, CGNE, LSQR and LSMR all search for solutions in the same Krylov subspace. Ultimately LSQR and LSMR converge to the least

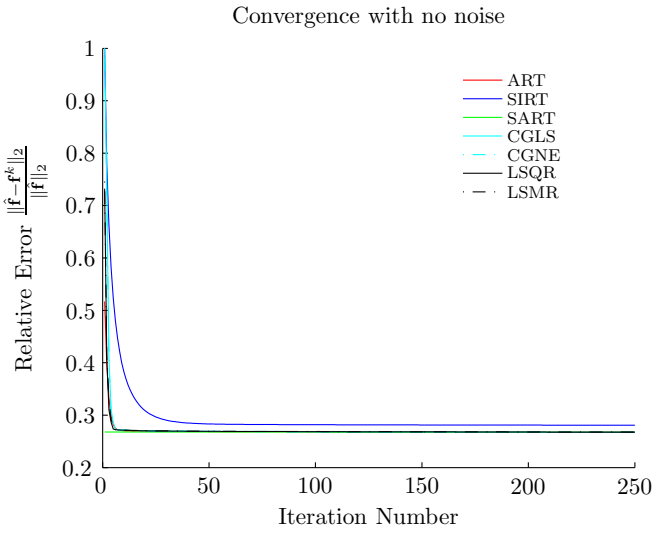


Figure 3.1: *The rate of convergence for the various ARMs when the measurement data was pure.*

squares minimum norm solution of $W\mathbf{f} = \mathbf{p}$ (as does CGLS). LSMR is claimed to be safer to terminate than LSQR. More information on LSMR can be found in [7].

3.8 Rate of Convergence

This section will be concerned with the rate of convergence of the aforementioned ARMs, both on data with and without noise. To that extent the test problem Blob (see Section 5.3 is taken. Artificial measurement data from 15 projection angles are created using the projection matrix W on the vector of the original image $\hat{\mathbf{f}}$. For the experiments with noise these data are polluted with a very high noise level; 10^3. Every algorithm will perform 250 iterations. Subsequently the relative error $\mathbf{e}^k$ of every approximation $\mathbf{f}^k$ is computed as $\mathbf{e}^k = \dfrac{\|\hat{\mathbf{f}} - \mathbf{f}^k\|_2}{\|\hat{\mathbf{f}}\|_2}$.

Figure 3.1 and 3.2 show the convergent behaviour of the ARMs acting on pure data. All ARMs seem to converge to a certain minimum solution (different per ARM). The convergence of SIRT is very slow compared to the rest of the methods. Note that it seems that SART is converged in only one iteration, and thus one could argue that SART is the fastest converging ARM. But actually SART cheats since it updates its solution per projection angle. So when SART has performed 1 iteration by the definition in Section 3.3, actually 10 updates have been performed.

More interesting results are obtained when the data is polluted with noise, see Figure 3.3 – 3.5. The concept of semi-convergence can now really be observed, especially for CGNE, CGLS, LSQR and LSMR. In the zoomed plot it can be seen that these methods converge very rapidly to a minimum error, but after that the error increases because the computed solution is fitted to the noise (cf. Section 2.4. From the last figure, which is zoomed in on SIRT, it is clear that every ARM shows semi-convergent behaviour, albeit very faint for ART, SIRT and SART. Note that the divergence after reaching a minimal error is far less for LSMR than for LSQR. CGLS & LSQR show the same behaviour, indicating that the methods indeed are equivalent.

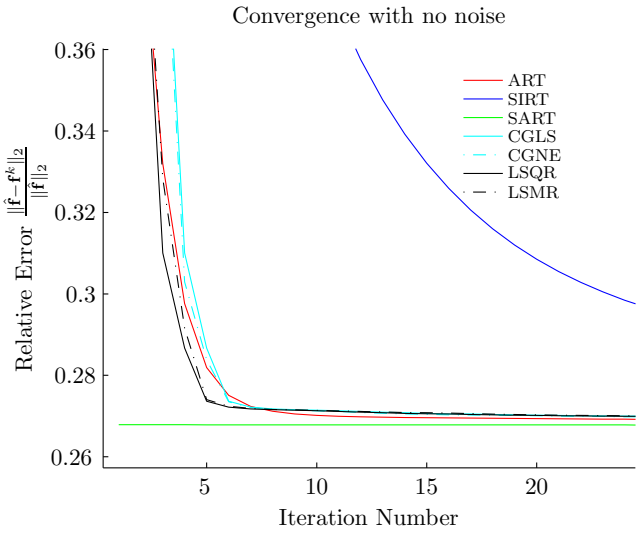


Figure 3.2: *The rate of convergence for the various ARMs when the measurement data was pure zoomed in on the first few iterations.*

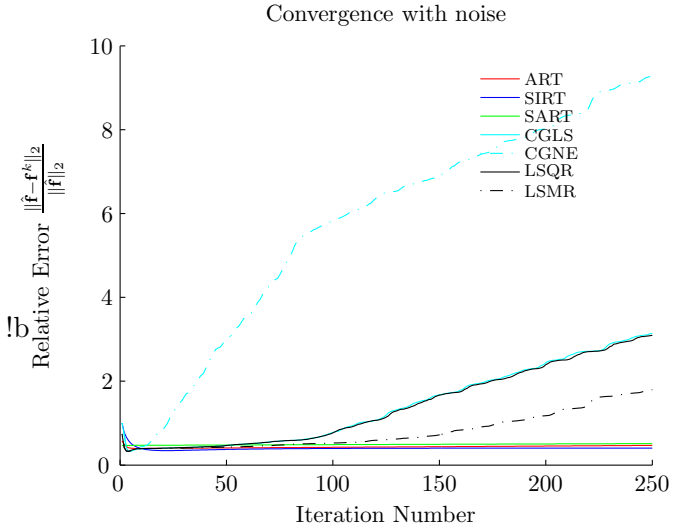


Figure 3.3: *The rate of convergence for the various ARMs when the measurement data was polluted with* 10^3 *noise.*

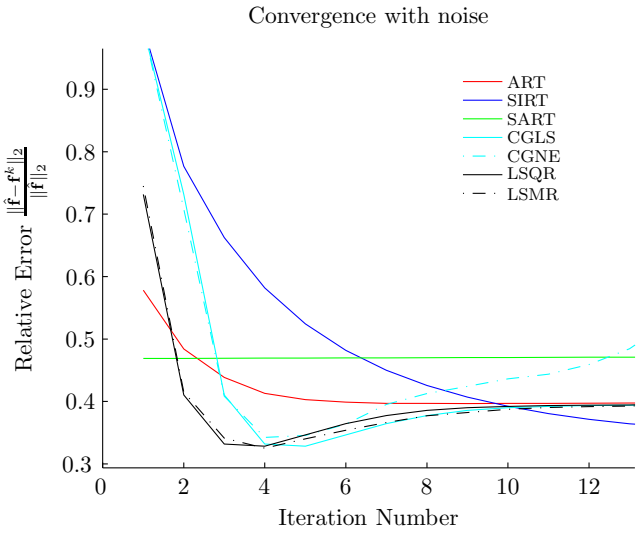


Figure 3.4: *The rate of convergence for the various ARMs when the measurement data was polluted with* 10^3 *noise zoomed in on the first few iterations.*

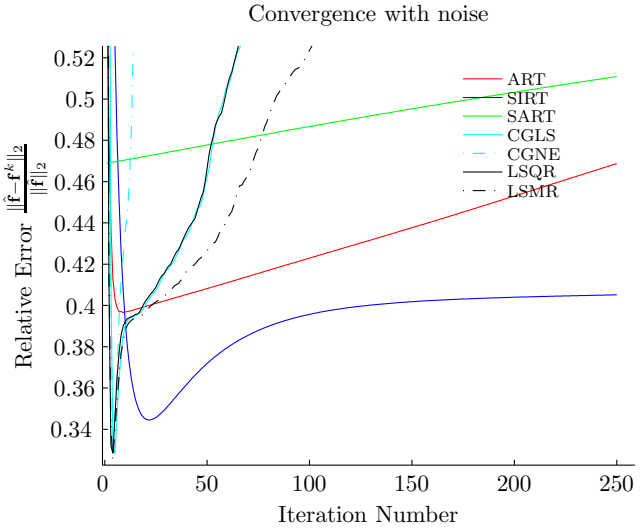


Figure 3.5: *The rate of convergence for the various ARMs when the measurement data was polluted with* 10^3 *noise zoomed in on SIRT.*

Chapter 4

Discrete Tomography

This chapter focusses on the *discrete tomography* (DT) problem. Unlike the previously considered problems, discrete tomography assumes certain properties about the image or object. Furthermore, the number of projections used in DT is far fewer than in classical (continuous) tomography. The resulting problem is more complex than for the continuous case. Four different solution strategies will be briefly discussed, one of these strategies will be of particular interest.

4.1 Description

The term discrete tomography was first coined by Larry Shepp who organised the first meeting devoted to this topic in 1994. Similar problems and a range of results were already known from earlier research. For a more detailed account on the history of this field one is referred to [17, Ch. 1].

In the foregoing chapters various methods were investigated to reconstruct a certain two-dimensional image $\mathbf{f} \in \mathbb{R}^N$ from its projections. This image could be viewed as a function $f : \mathbb{R}^2 \to \mathbb{R}$. Thus every point in the image has a corresponding function value, or grey value. Discrete tomography solves a similar problem with some extra constraints.

Roughly two kinds of discrete tomography can be identified. The first one assumes that the scanned image is defined on a lattice, i.e. the object or image is physically made up out of 'pixels'. All these pixels are assumed to have some uniform density and the density can only be one of a finite set of allowed densities. One can for example imagine that this model is suitable for scanning a diamond on nanoscopic scale. All the atoms (pixels) lie in a neat lattice and have uniform density. Hence a pixel can either represent an atom, in this case the corresponding grey value would be 1 (for simplicity) or the pixel is just void, in which case the grey value is 0. In this variant of DT one has a function with discrete domain and the range is a finite set, thus $f : \mathcal{L} \to \{\rho_1, \rho_2, \ldots, \rho_l\}$ with $\mathcal{L}$ a lattice. A lattice $\mathcal{L}$ in some d-dimensional domain is isomorphic to the integer lattice $\mathbb{Z}^d$, hence one can assume that $\mathcal{L} = \mathbb{Z}^d$. The grey values of each of these lattice points must be one of $\{\rho_1, \rho_2, \ldots, \rho_l\}$. Usually these assumed densities follow from prior knowledge about the image or object. In many cases there are only two allowed densities, then, without loss of generality, the set is denoted as $\{0, 1\}$ and one speaks of a *binary image*.

In the second kind of discrete tomography the assumption that the image is only defined on a lattice is dropped. Hence one has $f : \mathbb{R}^d \to \{\rho_1, \rho_2, \ldots, \rho_l\}$, a function with continuous domain and a finite set as range. Also for these problems it is common to only have two density values, one then speaks of *binary tomography*. The upcoming work will mainly focus on this second kind of DT.

One should note that the former DT variant is actually a special case of the latter variant. Both problems can be written, like in the continuous case, as $W\mathbf{f} = \mathbf{p}$. The weight matrix of the DT lattice problem will be a $(0, 1)$-matrix. For the latter variant the weights usually represent the length of the intersection of a ray with a cell. In continuous tomography the number of projections is often quite large compared to that in DT. On top of that, the few projections obtained in DT can originate from a small angular range. All previously proposed methods fail in solving this problem. This calls for better reconstruction techniques that incorporate the extra information known about the image.

4.2 Solution Strategies

This section will present four different strategies to solve the DT problem. The first approach will be to rewrite the problem as a combinatorial problem. The second strategy is statistical in nature. The two remaining techniques both solve the continuous problem. One of these approaches obtains a reconstruction using optimisation techniques, this reconstructed image may have different values than the allowed densities but this is penalised. The other method uses a discretisation step on the initial reconstruction, that was obtained via a continuous reconstruction method. This is also the strategy the Discrete Algebraic Reconstruction Technique (DART) algorithm uses. The algorithm that will be studied extensively in this work.

4.2.1 Combinatorial

Consider the problem of reconstructing a $(0, 1)$-matrix from its row and column sums. One can easily see that this is equivalent to the reconstruction of a binary lattice image or object from two orthogonal projections. The $(0, 1)$-reconstruction was studied by Ryser in 1957 who also proposed an algorithm for solving the corresponding problem [25]. Unfortunately, one does not have a unique solution for this problem in general. Consider for example the following two $(0, 1)$-matrices:

$$\begin{array}{cccc|c} 1 & 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 & 2 \\ 1 & 1 & 1 & 0 & 3 \\ 1 & 0 & 1 & 1 & 3 \\ \hline 3 & 1 & 4 & 2 & \end{array} \qquad \begin{array}{cccc|c} 0 & 1 & 1 & 0 & 2 \\ 1 & 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 1 & 3 \\ 1 & 0 & 1 & 1 & 3 \\ \hline 3 & 1 & 4 & 2 & \end{array} \tag{4.1}$$

Both matrices have the same column and row sums but are fairly different. Hence one cannot expect to find the correct solution in this case. More information about the reconstruction of $(0, 1)$-matrices can be found in [17, Ch. 1.2.2]. Gale modelled this reconstruction problem, also in 1957, as a network flow problem [8]. The Ford-Fulkerson algorithm for maximum flows can then be used to find the reconstruction (if a unique solution exists). The solution strategies for this problem proposed by Ryser and Gale are particularly computationally efficient. If one wants

to solve the problem for more than 2 projection angles, however, the corresponding problem becomes NP-hard [9]. Batenburg employed the idea of Gale to propose an algorithm which is able to reconstruct a *continuous* image (not defined on a lattice) [1]. The idea behind Batenburg's algorithm is to iteratively reconstruct the next image by using the previous reconstruction and two projection angles.

Another combinatorial approach for solving the DT lattice problem is from Gritzmann et al. [10]. They propose to solve two complementary problems: Best-Inner-Fit (BIF) and Best-Outer-Fit (BOF). The reconstruction is found using optimisation techniques.

4.2.2 Statistical

The statistical reconstruction method uses a *maximum a posteriori probability estimate* (MAP estimate) to approximate an optimal reconstruction based on the measured projection data. The admitted grey values are assumed to be distributed among the various pixels/cells according to some probability distribution. In 2005 Liao and Herman proposed to reconstruct discrete images using a Gibbs distribution [19].

4.2.3 Continuous Optimisation

The DT problem can also be solved using continuous optimisation. In this approach the standard continuous formulation of the problem is actually solved, i.e. the problem is relaxed so that $\mathbf{f}$ can take any real value, but function values outside the set of admitted grey values are penalised. Schüle et al. proposed a convex-concave regularization approach for the binary problem in 2003 [27]. The idea behind this approach is to use primal-dual optimisation to arrive at a solution which is steered towards a binary solution.

4.2.4 Continuous with Discretisation Step

The problem is first solved as if it were a continuous tomographic problem, thus the values of the image can attain any real value. Subsequently the reconstructed values are discretised, i.e. set to one of the admitted densities. Perhaps this is the most natural approaches to solve the DT problem. The DART algorithm uses this approach. The algorithm itself is explained in detail in Section 4.3.

4.3 DART

The Discrete Algebraic Reconstruction Technique (DART) was first proposed by Batenburg et al. in 2007 [3]. In 2011 Batenburg and Sijbers presented a more elaborate paper [4] giving a detailed description of the DART algorithm. For this section the proceedings in said paper are used to describe the DART algorithm.

4.3.1 The DART algorithm

DART is an iterative method of the type described in Section 4.2.4. It consists of reconstructing the image using a continuous update step followed by a discretisation step. The continuous step is just the reconstruction of the image using some ARM. The discretisation uses the prior knowledge, i.e. the allowed grey values $\{\rho_1, \rho_2, \ldots, \rho_l\}$. The number of allowed grey values should

not be too large, according to Batenburg et al. [4] the algorithm in general performs well for five or fewer grey values.

Initially a continuous reconstruction is computed by performing a fixed number of ARM iterations, this serves as a starting point for the DART algorithm. The reconstructed image is then *segmented* to obtain an image that consists of only the admitted grey values. The segmentation can be obtained by simply rounding the pixel values to the nearest allowed grey values. Or formally, define the threshold τ_i by

$$\tau_i = \frac{\rho_i + \rho_{i+1}}{2}, \tag{4.2}$$

for $i = 1, 2, \ldots, l-1$. Then the threshold function $r : \mathbb{R} \to \{\rho_1, \rho_2, \ldots, \rho_l\}$ is given by

$$r(v) = \begin{cases} \rho_1, & (v < \tau_1) \\ \rho_2, & (\tau_1 \leq v < \tau_2) \\ \vdots & \\ \rho_l, & (\tau_{l-1} \leq v) \end{cases}. \tag{4.3}$$

Other more intricate segmentation methods can be used that may lead to better convergence or more accurate reconstructions.

Next this segmented image is subdivided into two groups of pixels, the *free pixels* U and the *fixed pixels* F. The set U are all pixels that are adjacent to at least one pixel with a different grey value. Here one can choose the adjacent pixels of pixel j as the pixels North, East, South and West of the considered pixel, i.e. the 4-connected neighbourhood of pixel j. DART uses the 8-connected neighbourhood of pixel j which is simply the 4-connected neighbourhood augmented with the diagonally neighbouring pixels. The set F consists of the pixels that are not free, hence fixed. Note that $U \cap F = \emptyset$ and $U \cup F$ are all the pixels of the image. The set of free pixels represents the edges of the object, where an edge is the transition of one density into another. It may happen that there are edges which are not covered by the free pixels, holes in the image may for example be overlooked. The set U is therefore supplemented with some random pixels from F. Define $0 < p \leq 1$ as the *fix probability*. Each fixed pixel from F is freed with probability $1 - p$, independently of neighbouring pixels.

Subsequently DART performs some fixed number of ARM iterations, also called *intermediate iterations*, on the free pixels while, obviously, keeping the fixed pixels at their respective grey values from the set $\{\rho_1, \rho_2, ..., \rho_l\}$. The initial guess for the ARM iterations are the grey values of the free pixels from *before* the segmentation. Fixing some pixels results into a system with less variables but the same number of equations as initially, as will be shown later on. This, in combination with noise, causes heavy fluctuations in the values of the free pixels after the ARM iterations. Therefore, a *smoothing operation* is carried out. Smoothing evens out the function values and thus reduces the influence of the noise. The smoothing in DART is carried out by applying a Gaussian smoothing filter with radius 1 to the free pixels.

The process of segmenting the image, dividing it into free and fixed pixels, applying some ARM iterations to the free pixels and subsequently performing a smoothing operation constitutes one DART iteration. DART can either terminate after the total projection error is below some threshold ε, i.e.

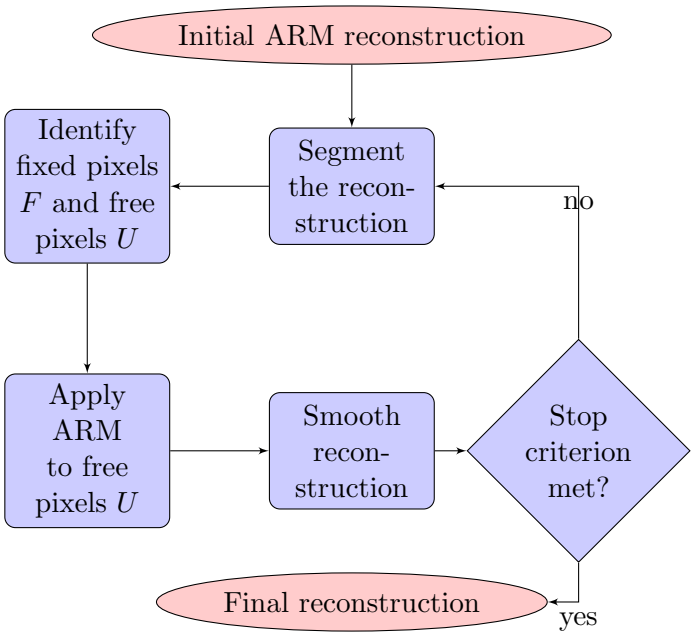


Figure 4.1: *Flowchart of the DART algorithm.*

$$\|W\mathbf{f} - \mathbf{p}\|_2 \leq \varepsilon. \tag{4.4}$$

Note that it might happen that this criterion is never satisfied. Or DART may terminate simply after a fixed number of iterations. After the last iteration the reconstruction is again segmented to obtain the final reconstruction which only contains pixels with admitted grey values. Figure 4.1 shows a flowchart of the DART algorithm. Traversing the purple boxes once constitutes one DART iteration.

Now the process of fixing pixels is investigated more thoroughly. Initially one is presented with the system:

$$\begin{pmatrix} | & & | \\ \mathbf{w}_{:,1} & \dots & \mathbf{w}_{:,N} \\ | & & | \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix} = \begin{pmatrix} p_1 \\ \vdots \\ p_M \end{pmatrix}. \tag{4.5}$$

By fixing pixel j the value of variable f_j is known. This means that $\mathbf{w}_{:,j} f_j \in \mathbb{R}^M$ can be computed beforehand and thus one may remove the variable f_j from $\mathbf{f}$ and column $\mathbf{w}_{:,j}$ from W and subtract $\mathbf{w}_{:,j} f_j$ from the right-hand-side. This results in the system

$$\begin{pmatrix} | & & | & | & & | \\ \mathbf{w}_{:,1} & \dots & \mathbf{w}_{:,j-1} & \mathbf{w}_{:,j+1} & \dots & \mathbf{w}_{:,N} \\ | & & | & | & & | \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_{j-1} \\ f_{j+1} \\ \vdots \\ f_N \end{pmatrix} = \begin{pmatrix} p_1 \\ \vdots \\ p_M \end{pmatrix} - \mathbf{w}_{:,j} f_j. \tag{4.6}$$

This system consists of the same number of equations as the initial system (4.5) but the number of variables is reduced by one. Eventually some number of ARM iterations are applied to the system

$$\tilde{W}\tilde{\mathbf{f}} = \mathbf{p} - \sum_{j \in F} \mathbf{w}_{:,f} f_j \tag{4.7}$$

which only has the free pixels as variables. This system will henceforth be referred to as the *reduced* system, the system matrix and the vector corresponding to the free pixels will have a $\tilde{\cdot}$ to indicate that they correspond to the reduced system. The effects of noise, which is ever-present, acts therefore only on the free pixels. This causes the heavy fluctuations in the values of these pixels and thus the need of the smoothing operation is more or less justified.

Below the DART algorithm is given in pseudo code. Let q be the number of initial ARM iterations that will be performed and o the number of ARM iterations during one DART iteration. If $\mathbf{f}$ is obtained by performing i ARM iterations with initial guess $\mathbf{x}$ on system W with right hand side $\mathbf{p}$ then this is denoted as $\mathbf{f} = ARM(\mathbf{x}, W, \mathbf{p}, i)$.

DART

$\mathbf{f}^0 = ARM(\mathbf{0}, W, \mathbf{p}, q)$
$\mathbf{x}^0 = \mathbf{f}^0$
$k = 1$
while termination criterion is not met **do**
  Segment image: $\mathbf{s}^k = r(\mathbf{x}^{k-1})$
  Identify boundary pixels U^k from $\mathbf{s}^k$
  Let $F^k = \{1, 2, \ldots, N\} \backslash U^k$ represent the set of fixed pixels
  Free pixels in F^k with probability $1 - p$
  Let $\tilde{W}^k$ be the reduced matrix
  $\tilde{\mathbf{p}}^k = \mathbf{p} - \sum_{i \in F^k} s_i^k \mathbf{w}_{:,i}$
  Let $\mathbf{y}^k$ be a vector containing the non-segmented values $\mathbf{x}^{k-1}$ of the free pixels U^k
  $\mathbf{f}^k = ARM(\mathbf{y}^k, \tilde{W}^k, \tilde{\mathbf{p}}^k, o)$
  Smooth all the free pixels U^k to obtain $\mathbf{x}^k$
  $k = k + 1$
end while

One should note that the initial guess $\mathbf{y}^k$ for the ARM iterations during a DART iteration consist of all the free variables that have their non-segmented values $\mathbf{x}^{k-1}$ rather than $\mathbf{s}^k$. Also observe that the smoothing is only applied to the free pixels U^k.

Chapter 5

Research Goals

This chapter will formulate the research questions and goals that will form the basis for the upcoming research. Some test cases will be described that will be involved in answering the posed questions.

5.1 Research Questions

Chapter 4 described the DART algorithm in detail. Although the performance of DART is reasonable, its approach is very heuristic. The random subset construct, for example, is needed to find any holes in the object. The reduced system (4.7) contains only the free pixels as variables, which are the boundary pixels supplemented with the random subset. The effects of the noise, which is present in the projection data $\mathbf{p}$, is distributed over these free variables only. This causes heavy fluctuations in their values. Smoothing with the Gaussian filter counters this effect, but there is no theoretical motivation which justifies the use of this blurring operator. In Chapter 3 the concept of regularization was introduced as a tool to reduce the effects of noise on the reconstructions. DART might benefit from the use of regularization on the set of free pixels U.

The main goal of the upcoming research will be to investigate if the DART algorithm can be improved. Hence the general research question will be:

Can the DART algorithm be improved?

More specific questions can be asked in the context of this general research question:

- Which algorithm should be used as ARM in DART and does it matter?
- Can better results be obtained by introducing regularization directly onto the set of free pixels U?
- Are there alternatives for the fixed-free pixels construct?

5.2 Methodology

This section will describe what approach will be used to answer each of the research questions.

Which algorithm should be used as ARM in DART and does it matter?

To answer this question further experiments have to be carried out. Analysis of the results of these experiments should yield the best ARM.

Can better results be obtained by introducing regularization directly onto the set of free pixels U?

Regularization can be interpreted in many different ways. One approach would be to use Tikhonov regularization, see (2.10), on the free pixels. One would in that case solve to following problem:

$$\begin{pmatrix} \tilde{W} \\ \lambda I \end{pmatrix} \tilde{\mathbf{f}}^U = \begin{pmatrix} \mathbf{p} - \sum_{j \in F} \mathbf{w}_{:,j} f_j \\ 0 \end{pmatrix}, \tag{5.1}$$

with $\tilde{\mathbf{f}}^U$ the vector containing the free pixels.

Are there alternatives for the fixed-free pixels construct?

DART typically attempts to solve the discrete minimization problem

$$\min_{\mathbf{f} \in \{\rho_1, \ldots, \rho_l\}^N} \|W\mathbf{f} - \mathbf{p}\|. \tag{5.2}$$

The problem with DART is that the fixing of pixels might cause holes in the object to be overlooked and that the effects of noise is distributed among the free pixels only. One can alternatively consider the following problem:

$$\begin{pmatrix} W \\ D \end{pmatrix} \mathbf{f} = \begin{pmatrix} \mathbf{p} \\ D\mathbf{v} \end{pmatrix}, \tag{5.3}$$

with D a diagonal matrix and $\mathbf{v}$ a vector containing the presumed values of the corresponding pixels. This system will steer the solution towards the presumed values in $\mathbf{v}$ but the pixels are allowed to deviate from these values. The entries d_i, $i = 1, 2, \ldots, N$, of the matrix D represent the degree to which the pixels of the solution should be steered towards the values in $\mathbf{v}$. Solving (5.3) is equivalent to the minimization problem

$$\min_{\mathbf{f} \in \mathbb{R}^{\mathbf{N}}} \|W\mathbf{f} - \mathbf{p}\| + \|D\mathbf{f} - D\mathbf{v}\|. \tag{5.4}$$

Note that deviation of the proposed values v_i is penalised proportional to d_i. Very high values of d_i will mostly likely result into a system where pixel i has grey values v_i. A smaller d_i indicate some uncertainty associated with the value v_i, the grey value of these pixels are more likely to deviate from the presumed values.

This approach gives rise to the following sub questions:

- Can this approach lead to similar behaviour as the original DART?
- How should the d_i be chosen?

5.3 Test Problems

The research questions posed in the previous section call for modifications of the DART implementation. Naturally these new implementations have to be validated, the results of experiments have to be verified and compared to the results of the original algorithm. To do this validation some *test problems* will be presented. The test problems have some hierarchical structure ranging from simple objects to more intricate shapes. The choice of problems is motivated by a range of features:

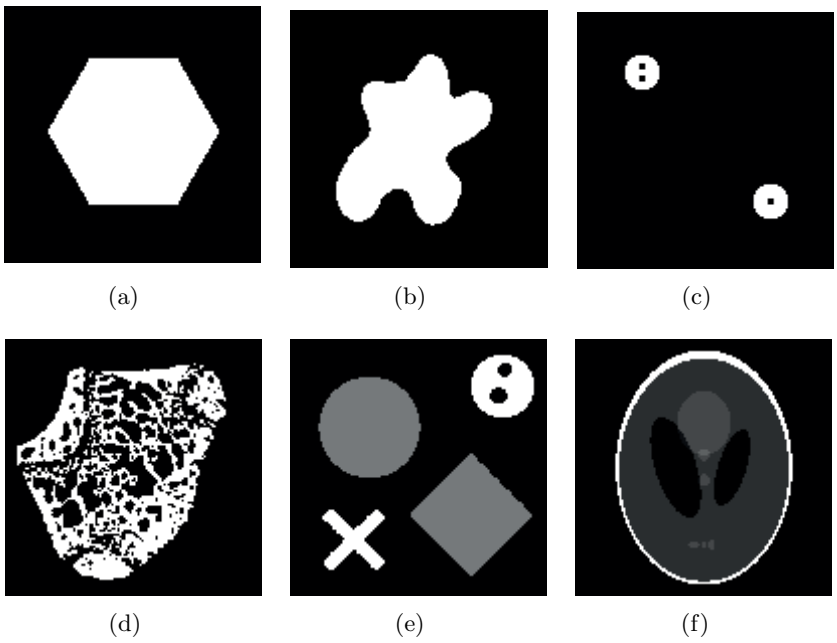
- The shape of the object;
- The ratio of edges to the total number of pixels;
- The number of grey values.

The shape of the object is a very broad notion. Objects can be very simple, e.g. symmetric, homogeneous and convex, or more complex, the image might even consist of multiple objects. Edges in the image are very important in DART since these are the only pixels that are subject to change. Images with a low ratio of edges to pixels require less computation time since a great number of pixels are fixed and thus a smaller system is considered. It is obvious why the number of different grey values is an important feature.

Figure 5.1 shows the six test problems that will be used to answer the research questions. Hexagon is a very simple symmetric homogeneous object. Blob is somewhat more intricate as it is not symmetric. Two Circles are two objects, each with a hole. Moreover, the objects are not in the center of the image. The fourth test problem is Bone which is inspired by the reconstruction of bones, the problem still contains two grey values. These two grey values in the binary images are black, or grey value 0 and white, grey value 255. The final two problems contain multiple grey values: Shapes consists of three grey levels (black, grey and white or 0, 128 and 255), Phantom is the Shepp-Logan head phantom which is a widely used test problem in tomography and consists of six different grey values: 0, 25, 51, 76, 102 and 255.

The main interest of this research is the performance of the improved implementation when one deals with noisy projection data. Therefore, the sinograms of the images will be polluted with Poisson distributed noise. Four different number of counts will be used: $10^3, 10^4, 10^5$ and 10^6. Recall that low number of counts correspond to more noise.

The experiments will reconstruct the objects from data acquired from a varying number of angles. Naturally, simple objects such as 5.1(a) will require fewer angles to obtain a perfect reconstruction than the more complex problems such as 5.1(d).



(a) (b) (c)

(d) (e) (f)

Figure 5.1: *Test problems (a) Hexagon (b) Blob (c) Two circles with holes (d) Bone (e) Shapes (f) Phantom*

Chapter 6

Numerical Experiments

In this chapter the results of the numerical experiments are listed. The experiments are carried out in MATLAB (Version R2010b, 64-bit) on a computer with an Intel Core 2 Quad Q6600 (quad core, 2.4GHz) CPU and 8GB memory. Some functionalities of the ASTRA-toolbox (All Scale Tomographic Reconstruction Antwerp) are used [23]. This toolbox is created by the ASTRA group of the vision lab, a research lab of the Department of Physics of the University of Antwerp and is free to download.

6.1 Methodology

The experiments are all performed in a similar way. First, measurement data are created from the original object (image) using parallel projections. The images considered, unless stated otherwise, are 128×128 pixels of size. The width of the detector needs to be at least as long as the diagonal of the image, for it would not be able to cover the whole image if the projection angle is $\pi/4$ for example. However, with such a detector and a reconstruction grid of 128×128 pixels, there might be very few pixels in the path of one ray, such as rays that intersect the corners of the image. This poses problems when noise is added to the measurement data. Pixels close to the corners of the reconstruction grid receive a disproportional share of this noise and the reconstructions will have extreme (positive or negative) grey values in the corners where generally no material is located. Hence the original images are zero padded to 182×182 pixels ($\lceil \sqrt{2} \cdot 128 \rceil = 182$), and thus the reconstruction grid is also this size. A detector is made up out of detector-pixels, the distance between two adjacent detector-pixels is 1 pixel and the parallel detector is thus 182 detectors wide. Projections are taken, unless stated otherwise, from equally spaced angles between 0 and π.

The quality of the reconstructions is quantified by means of the *pixel error* K which is simply the number of misclassified pixels compared to the original object. Sometimes this error is expressed in percentages. Note that the pixel error is based on the reconstruction on the slightly bigger grid of 182×182 pixels while the percentage of misclassified pixels will assume only a total of $128 \times 128 = 16384$ pixels. Hence it might happen that the percentage of misclassified pixels is more than 100%, this will only be in extreme cases and the corresponding conclusion is clear: the reconstruction is very poor. This pixel error is recorded after 100 DART iterations

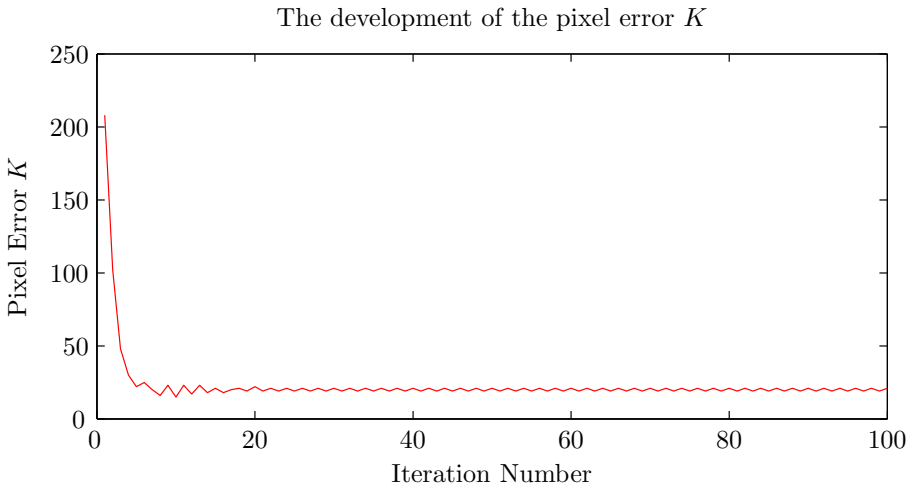


Figure 6.1: *The development of the pixel error K in DART. Note that the minimal pixel error is reached after about 10 iterations and that the pixel error after 100 iterations is slightly bigger.*

are carried out. However, generally DART does not produce reconstructions of monotonically decreasing (pixel) errors and thus one might find the best reconstruction (in terms of minimal pixel error) after 10 iterations, see Figure 6.1. Experiments have shown that DART either has converged to one solution or alternates between comparable solutions long before 100 iterations are performed. In the real world one does not know the original object and thus also not the (pixel) error of a reconstruction. Hence it might be misleading to only record the minimal pixel errors. Consequently, it has been decided to record the error after 100 iterations for all the upcoming experiments. Furthermore, one should note that the pixel error is not an absolute measure for the quality of an image, reconstructions with lower pixel errors can appear poorer by visual inspection than reconstruction with a somewhat higher pixel error. Consequently, actual reconstructions will be shown from time to time.

The ASTRA-toolbox allows one to generate the projection matrix W for a giving projection and volume geometry. In the experiments in this chapter the original test problem is passed as an argument to DART. This image is then transformed into a vector $\hat{\mathbf{f}}$ and consequently one can compute the non-polluted 'ideal' measurement data by $\hat{\mathbf{p}} = W\hat{\mathbf{f}}$. These data are then polluted with noise according to some Poisson distribution (see Section 2.4 for details) of which the magnitude is expressed in number of counts. Noise level 10^6 corresponds to very little noise while 10^3 is the highest noise level that will be considered in this work. As a result of the randomness involved by adding the noise to the data two identical simulations (same test problem, parameters, noise level etc.) will have a different right-hand-side (the measurement data) and thus will most likely converge to different solutions. Hence by this phenomenon one of the simulations can, in a very extreme case, find a near perfect reconstruction while the other generates a poor reconstruction. This has never been observed in practice however. Furthermore, all the experiments are carried out for 6 different test problems which should weed out any outliers. All the experiments show the same characteristics, for example, it will be found that LSMR will nearly always produce slightly better results than LSQR and hence the randomness as a result of the generation of noise cannot have a large influence on the reconstructions. On top of that, in the appendix the results of the experiments will be shown, from this conclusions are drawn. To strengthen these conclusions actual reconstructions will also

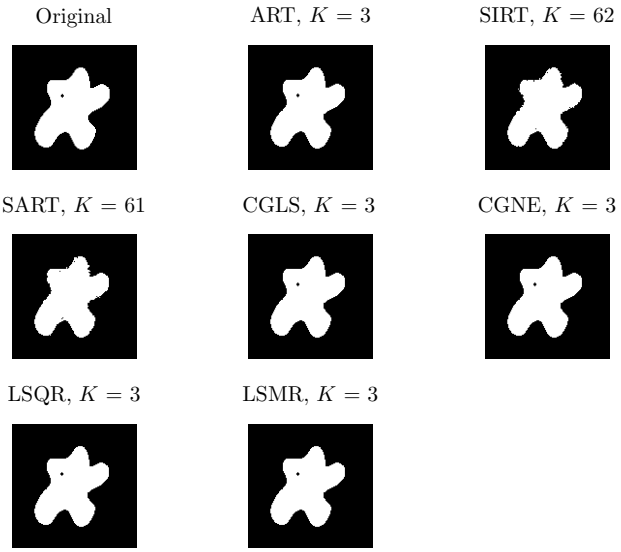


Figure 6.2: *DART reconstructions of the Blob with a hole after 100 DART iterations with different ARMs and fix probability 1*

be shown, these were obtained by repeating the experiment with the same parameters. Also in this case no significant differences were found between the simulations. Nevertheless, it is good to keep in mind that the results of the experiments, especially for the higher noise levels, can slightly change if they are run again. The conclusion of the experiments, however, will still hold.

6.2 The Fix Probability

The DART algorithm is generally used with SIRT or SART as ARM. The fix probability p, as introduced in Chapter 4, is needed for these methods since it should find holes in homogeneous areas of the reconstruction that have been overlooked. Figure 6.2 shows the DART reconstructions after 100 DART iterations using the various ARMs. The fix probability was set equal to 1, the initial reconstruction was made using 50 ARM iterations and in each DART iteration an additional 50 intermediate ARM iterations were carried out. The measurement data were not polluted with any noise. The original image of 128 by 128 pixels was scanned using 6 equally spaced angles in the range $[0, \pi)$ with 182 parallel rays per angle. As one can see all methods except SIRT and SART were able to find the hole in the image, moreover, all methods find the same near perfect reconstruction that has only 3 misclassified pixels (of the $128 \times 128 = 16384$ pixels). Repeating the experiment with $p = 0.99$ results in the reconstructions that can be found in Figure 6.3. Note that SIRT and SART are now able to find the holes. The other methods, however, seem to perform less well with this random subset construct. Consequently, because of this and the heuristic nature of the random subset it has been decided to carry out the upcoming experiments with fix probability 1.

One might wonder how some ARMs are able to find the holes in the image since DART fixes certain pixels. Figure 6.4 shows how CGLS finds the hole in the blob. First the blob is roughly reconstructed as a homogeneous object. At this stage SIRT and SART would gradually find the

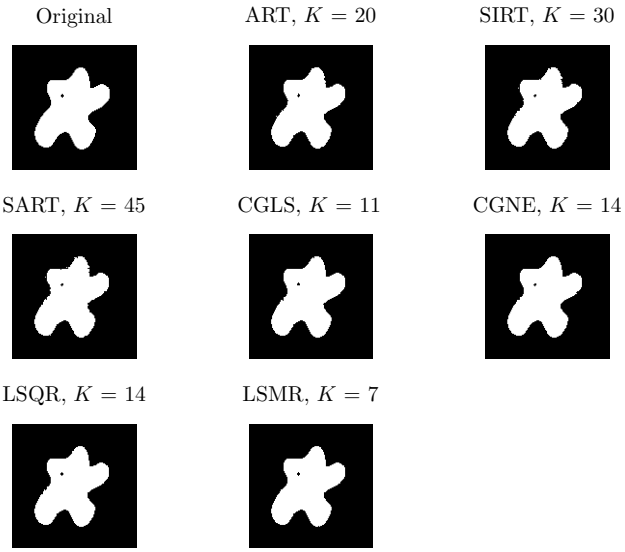


Figure 6.3: *DART reconstructions of the Blob with a hole after 100 DART iterations with different ARMs and fix probability 0.99*

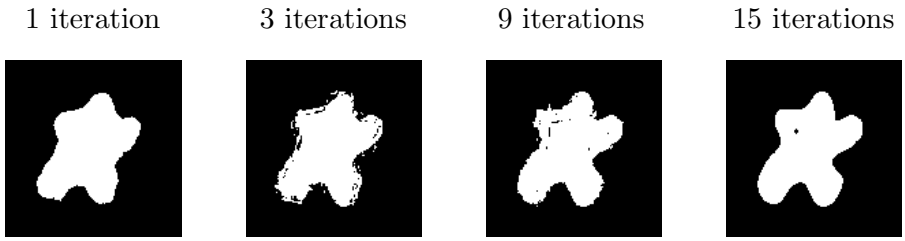


Figure 6.4: *DART with CGLS finding the hole in the image.*

edges of the blob never finding the hole. CGLS (and the other methods) on the other hand acts very unstable at the boundary, as if it knows something is not right. The number of boundary pixels increases rapidly and pixels inside the object are freed because of that. Once the free pixels arrive at the location of the hole, the boundary is reconstructed correctly, without this unstable behaviour. Recall that these methods actually solve different systems, CGLS for example solves the normal equations while SIRT, as will be seen later on, solves the 'scaled' normal equations implying that different norms are used.

One big concern needs to be expressed at this point. Choosing fix probability equal to 1 results into fewer free pixels. The noise which is present in the measurement data can have no influence on the fixed pixels, since these are not variables of the reduced system. All the effect of the noise is therefore distributed among the free pixels. When the number of free pixels is small, the impact of the noise might be greater. In practice this means that DART with SIRT, for example, might produce better results with a lower fix probability if noise increases. Indeed, this is also what is pointed out in the paper of Batenburg and Sijbers about DART [4].

6.3 Initial Guess

In Chapter 4 DART was introduced. Every DART iteration some pixels are fixed and the corresponding columns are removed from the system of linear equations. This results in the reduced system:

$$\tilde{W}\tilde{\mathbf{f}} = \mathbf{p} - \sum_{j \in F} \mathbf{w}_{:,f} f_j. \tag{6.1}$$

Subsequently, this reduced system is solved using some number of ARM iterations to find $\mathbf{f}^k_{grey}$ (subscript *grey* is to indicate that this solutions contains non-segmented grey values). As initial guess the grey values, thus before segmentation, of the free pixels from the previous DART approximation $\mathbf{f}^{k-1}_{grey}$ are used. It might, however, happen that this approximation has components in the null space $\mathcal{N}(\tilde{W})$. Let $\hat{\mathbf{f}}$ be an exact solution (the original object for example). In general the norm of the residual $\|W\mathbf{f}^k_{grey} - \mathbf{r}^k\|$ will be minimized, the norm of the approximation error, however, $\|\mathbf{f}^k_{grey} - \hat{\mathbf{f}}\|$ might never diminish to zero since $\mathbf{f}^k_{grey}$ can still have components in the null space of $\tilde{W}$, even if $\mathbf{p} \in \mathcal{R}(\tilde{W})$.

It can be proven for binary images, if the right-hand-side is not perturbed by any noise, that the minimum norm least squares solution (which has continuous grey values) is geometrically located in the middle of all binary solutions [2]. Thus the minimum norm least squares solutions, which has no components in $\mathcal{N}(\tilde{W})$, can be seen as not being biased towards any binary solution. This would motivate the choice of $\mathbf{0}$ as initial guess since it leads to finding a minimum norm least squares solution.

Now the effect of the two possible choices of the initial guess will be investigated. To this end the performance of DART with SIRT and CGLS with both initial guesses, the grey values and the zero vector, will be investigated for the noise levels 10^6 (very little noise), $10^5, 10^4$ and 10^3 (very much noise). The number of initial and intermediate ARM iterations is 50 and the fix probability is chosen equal to 1. The Blob test problem is scanned using 10 angles. In Figure 6.5 the results of the experiment are shown. It is clear that taking the zero vector as initial guess yields the best results, especially when noise increases. Although this does not prove that this choice will always perform better, it is a good indication that its results for problems with high levels of noise are better than when the grey values of the previous approximation are used. Since this work focusses on the development of robust algorithms, in the sense that the algorithm deals with noise in an efficient way, the upcoming experiments will use the zero vector $\mathbf{0}$ as initial guess for the intermediate ARM iterations.

6.4 The Best ARM

This section investigates the first research question:

Which algorithm should be used as ARM in DART and does it matter?

Chapter 3 introduced seven algebraic reconstruction methods that can be used in DART. The first three methods: ART, SIRT and SART, are widely used in the field of tomography. CGLS can also be encountered in practice for tomographic reconstruction but CGNE, LSQR and LSMR

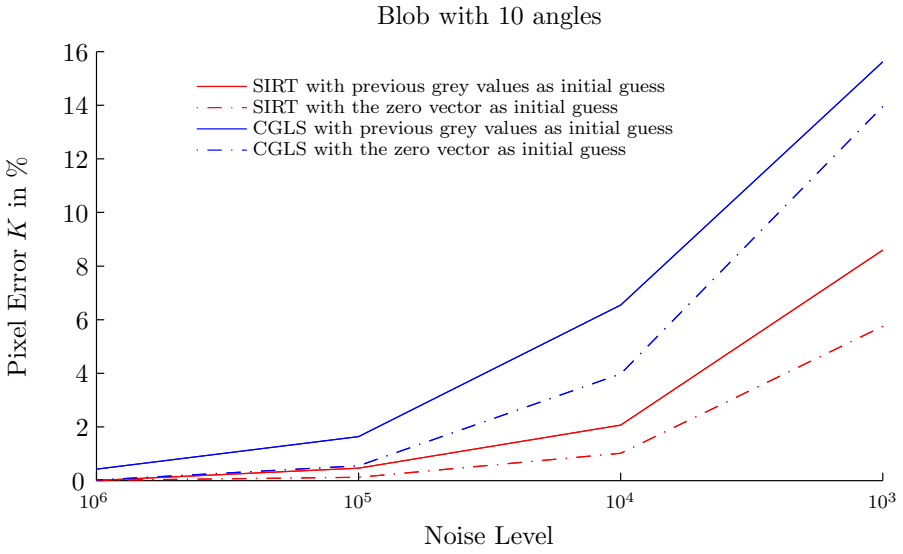


Figure 6.5: *The performance of DART with different initial guesses for the indermediate ARM iterations*

are used to a very low degree if they are used at all in tomography. It is expected that CGLS, LSQR and LSMR produce similar results since, in theory, they converge to the least squares minimum norm solution.

6.4.1 Performance of the Various ARMs

The experiments are carried out as follows. First it is investigated how many angles are needed to make a (near) perfect reconstruction with DART for each test problem, when no noise is present in the measurement data. Subsequently, a slightly larger number of angles is used to reconstruct the test problem when noise is present. The performance of DART with the various ARMs is then investigated. The number of initial ARM iterations as well as the number of intermediate ARM iterations is 50. In Table 6.1 the number of minimal angles is found (second column), for these number of angles the DART reconstruction of all the ARMs were nearly perfect. For the upcoming experiments two number of angles per test problem will be considered to investigate if more angles correspond to better results (third and fourth column). The pixel errors K of the test problems for these and fewer angles can be found in Appendix B.1: Table B.1 – B.6. It seems that SIRT and SART are outperformed by the other ARMs. Regarding the cause of this one can only speculate at this time, but since it is also observed for homogeneous objects without holes like Hexagon and Blob it cannot be attributed to the absence of the random subset. It might be that these methods need more (or fewer) initial/intermediate ARM iterations.

Next the performance of DART is investigated when the measurement data are polluted with noise. The noise levels that will be considered are 10^6 (very little noise), $10^5, 10^4$ and 10^3 (very much noise). The results of the experiments can be found in Appendix A.1: Figure A.1 – A.6. In these figures the pixel error is given in percentages, note that the total number of pixels is 16384. The performance of ART with noise was that bad that it is not even included in the plots for it distorted the vertical axis in such a way that the performance of the other methods was no longer clearly visible. In Figure 6.6 a reconstruction of DART with ART after 100 DART iterations is shown. The noise level was 10^6 which implies very little noise on the

Test Problem	Minimal	Experiment 1	Experiment 2
Hexagon	3	6	10
Blob	7	10	20
Two Circles	7	10	20
Bone	80	90	105
Shapes	20	30	45
Phantom	65	75	95

Table 6.1: *The minimal number of angles for a near perfect reconstruction and the number of angles for which the experiments with noise are performed. The different number of angles of the two experiments are to investigate if more angles lead to better results.*

measurement data. The used number of projection angles was 10. All other methods were able to reconstruct the hexagon effortlessly. This divergent behaviour of ART is observed for a great many experiments, moreover, it is very unpredictable when this divergence will occur, rendering the algorithm not trustworthy. Also CGNE seems the suffer from noise to a high degree and is practically useless for high noise levels (10^4 and up). Note that ART and CGNE solve the problem using the alternative normal equation $WW^T\mathbf{u} = \mathbf{p}$ while all the other methods except SIRT use some form of the normal equations $W^TW\mathbf{f} = W^T\mathbf{p}$. This is most likely the reason that ART and CGNE are very bad at dealing with noise. Why SIRT does not have this behaviour will be investigated later on.

Figure 6.6: *DART reconstruction with ART of Hexagon using 10 angles, noise level:* 10^6.

The remaining algorithms do seem to be able to deal with noise. For the low noise levels 10^6 and 10^5, CGLS, LSQR and LSMR produce slightly better results. As the noise increases to 10^4 and 10^3 SIRT and SART seem to outperform CGLS, LSQR and LSMR. All the algorithms have great difficulty with the reconstruction of Phantom. This is most likely because of the large number of grey values (six). Note that the pixel error of CGNE is even more than 100% which indicates that the reconstruction is exceptionally poor since the total number of misclassified pixels in the large grid is more than the number of pixels in the original grid (cf. the discussion at the beginning of this chapter). Figure 6.7 shows the reconstructions of Phantom for all the methods. The data originated from 75 angles and were polluted with a noise level of 10^5. Note that although the pixel errors are quite high, the quality of the reconstructions is still very reasonable, even for ART and CGNE.

One interesting thing that should be observed is that the performance of the algorithms is not improved when the number of angels is increased, on the contrary, in some cases (cf. A.3) the pixel errors even seem to increase. This can be attributed to the fact that although more measurement data are available, all these data are polluted with noise. Usually when solving least squares problems the noise is assumed to be distributed with mean 0 and to be uncorrelated. In this case, however, the noise is Poisson distributed and thus does note have mean 0. This

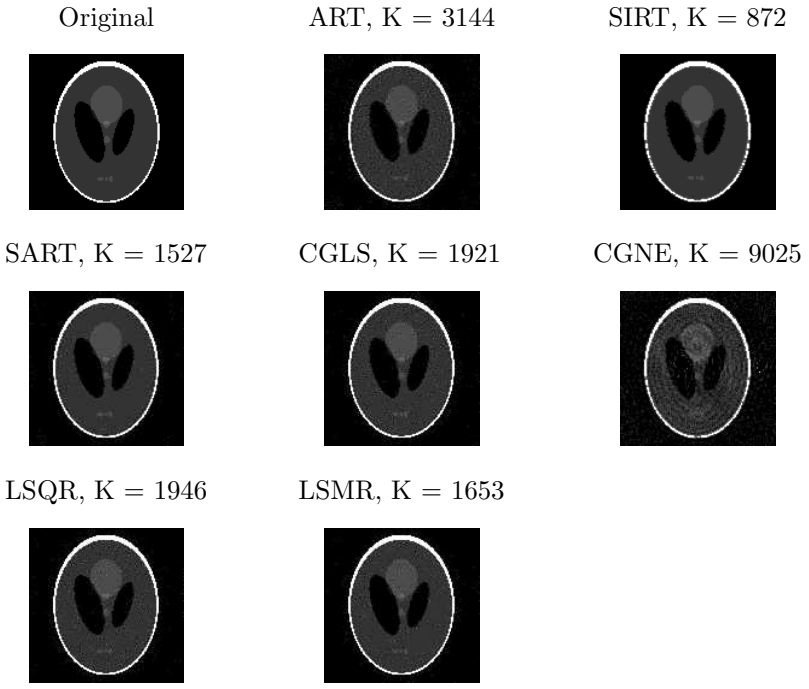


Figure 6.7: *DART reconstruction of the various methods of Phantom. Number of projection angles was 75 and the noise level* 10^5.

could be a possible reason for this phenomenon. The noise is added to the sinogram using a functionality in the ASTRA-toolbox. The method of adding noise to the sinogram is quite peculiar, since it needs to be defined in the number of counts, but also because it is dependent on the maximal value in the sinogram. Obviously the maximal detector value will not be the same if the original object is scanned using more angles and thus the noise levels are not comparable anymore. This could also play a role in the manifestation of this phenomenon.

From these experiments it is concluded that SIRT and SART are the best performing algorithms. From the remaining methods ART and CGNE immediately drop out since they are bad at dealing with noise. The performance of CGLS and LSQR is similar, as was to be expected since they are mathematically the same, LSMR seems to produce marginally better results. Since in the upcoming experiments the linear system that is solved is changed drastically, two ARMs are chosen with which the experiments will be carried out. SIRT because it was (with SART) the best performing ARM and it is faster than SART. LSQR is also chosen to have a different natured method. LSQR is chosen above CGLS because it is famous for its robustness and over LSMR since it has proven itself in practice. In some experiments, however, the performance of LSMR will also be investigated since it is mathematically different to CGLS and LSQR and quite new, so it will be interesting to see how it performs. In Table 6.2 the computation times of 100 DART iterations with the various algorithms are listed. The times originate from a reconstruction of Shapes using 30 angles, 50 initial and intermediate ARM iterations and a noise level of 10^5. Although these timings are of course very dependent on the platform on which the algorithm works it shows the relative computation times. ART and SART are both particularly slow compared to the rest, this is inherent to their definition since they cannot be formulated in a matrix-vector update for-loops need to be used, which is particularly slow in

MATLAB. It needs to be noted that these computations were carried out on a quad core CPU (on which in fact only 1 core did the computation), whereas if they would have been computed on a GPU (with the proper implementation, of course) the proportions might be very different. The high computation time of CGNE compared to SIRT, CGLS, LSQR and LSMR is mainly due to the bad performance of the algorithm with noise. Because of this a greater number of pixels is considered free by DART and therefore the corresponding reduced system that is solved is larger. If no noise were present the computation time of CGNE would be of the same magnitude as CGLS.

Method	**Time (s)**
ART	217.3882
SIRT	14.4288
SART	202.6539
CGLS	17.1253
LSQR	16.7358
CGNE	77.4953
LSMR	15.507

Table 6.2: *The computation times of 100 DART iterations per algorithm in seconds.*

6.4.2 Number of ARM iterations

In the experiments of the previous section the number of initial and intermediate ARM iterations was 50. The DART reconstructions improve if more or, because of semi-convergence (cf. Section 2.4 and 3.8), fewer ARM iterations are used. In this section it is investigated what number of ARM iterations yields the best results. Only SIRT and LSQR will be investigated as ARM due to the conclusion of the previous experiments. The upcoming experiments consist of the reconstruction of the test problems using 100 DART iterations. The chosen number of projection angles is equal to those found in the third column ('Experiment 1') of Table 6.1. The fix probability p is still chosen equal to 1 and the considered noise levels will again be 10^6 (little noise), $10^5, 10^4$ and 10^3 (much noise). The investigated number of ARM iterations (both initial and intermediate) are $20, 35, 50, 75, 100, 150$ and 200.

The results of the experiments can be found in Appendix A.2: A.7 – A.12. Overall it seems that fewer iterations yield better results. This is most likely due to semi-convergence: as the number of iterations increases, noise present in the measurement data is being fitted to the solution. SIRT outperforms LSQR on the whole but for low levels of noise LSQR seems to produce slightly better results than SIRT. In some cases (e.g. Bone and Phantom) more SIRT iterations yield better results for low levels of noise, this can also be observed for LSQR but to a lesser extent. SIRT and LSQR show similar performance regarding the number of intermediate iterations indicating that both methods have a similar convergence rate. Although the results for fewer than 20 iterations are marginally better it is not advisable to perform so few iterations since then the ARM has very little time to converge.

From the above experiments one can conclude that DART performs best when SIRT is used as ARM and every iteration 20 ARM iterations are performed. It is quite surprising that so few iterations are needed and that it is not strongly dependent on the noise level. The reduced system of DART might explain that so few iterations are needed: by fixing pixels the number of variables is reduced vastly.

6.5 Regularization on the Free Pixels

This section investigates the second research question:

Can better results be obtained by introducing regularization directly onto the set of free pixels U?

For the regularization Tikhonov regularization can be used (cf. Section 2.4). This Tikhonov regularization onto the free pixel is easily implemented as follows:

$$\begin{pmatrix} \tilde{W} \\ \lambda I \end{pmatrix} \tilde{\mathbf{f}} = \begin{pmatrix} \mathbf{p} - \sum_{j \in F} \mathbf{w}_{:,j} f_j \\ 0 \end{pmatrix}, \tag{6.2}$$

with $\tilde{\mathbf{f}}$ the vector containing the free pixels and the regularization parameter $\lambda \in \mathbb{R}_{\geq 0}$. Solving this system is equivalent to solving the minimization problem

$$\min_{\tilde{\mathbf{f}}} \left\| \tilde{W}\tilde{\mathbf{f}} - \mathbf{p} + \sum_{j \in F} \mathbf{w}_{:,j} f_j \right\|^2 + \lambda^2 \left\| \tilde{\mathbf{f}} \right\|^2. \tag{6.3}$$

From (6.2) and (6.3) one can easily see that the norm of $\tilde{\mathbf{f}}$ is kept small because of the factor λ. Hence one would expect that solutions $\mathbf{f}^k$ that result from this regularized problem have an overall smaller norm than the non-regularized problem.

For the experiments five values of λ are chosen, namely $\lambda \in \{0, 0.1, 0.5, 1, 10\}$. Note that $\lambda = 0$ corresponds to the original non-regularized version of the problem. The test problems are scanned using the same number of angles from the column 'Experiment 1' in Table 6.1 as done throughout. The performance of DART is investigated with both SIRT and LSQR as well as LSMR. The number of initial and intermediate ARM iterations is 20, in line with what was concluded from the forgoing experiments. Again noise is introduced to the problem according to the same numbers of counts as always: 10^6 (little noise), $10^5, 10^4$ and 10^3 (much noise). The results of the experiments can be found in Appendix A.3: A.13 – A.18.

The first thing one should note is that SIRT is again the overall best performing ARM, however, for the more intricate problems (Bone, Shapes and Phantom) LSQR and LSMR perform better when there is little noise. LSMR seems to perform marginally better than LSQR.

The effect of regularization on the ARMs is inconclusive. For the simple test problems (Hexagon, Blob and Two Circles) $\lambda > 0$ seem to produce better results, but $\lambda = 10$ causes the problem to tend too heavily towards the zero solution and thus produces bad results. For very high values of λ the algorithm will find, not surprisingly, the zero vector as solution. It seems that the optimal regularization parameter λ is between 1 and 10, but this optimum might be, and most likely is, different for the various test problems. For the more intricate test problems, however, the regularization seems to have a negative effect on the results.

A somewhat interesting phenomenon can observed when DART uses LSQR or LSMR on very noisy data (10^3) and regularization parameter $\lambda = 10$. In this case the results are, with the exception of Two Circles, better than any other value for λ. This is mainly due to eagerness of the solution to be close to zero because of the high value of λ. This is not the case for smaller values and then one will find speckles around the actual object. Since the regularization pulls the solution towards 0 these speckles are far less present for $\lambda = 10$, resulting in fewer

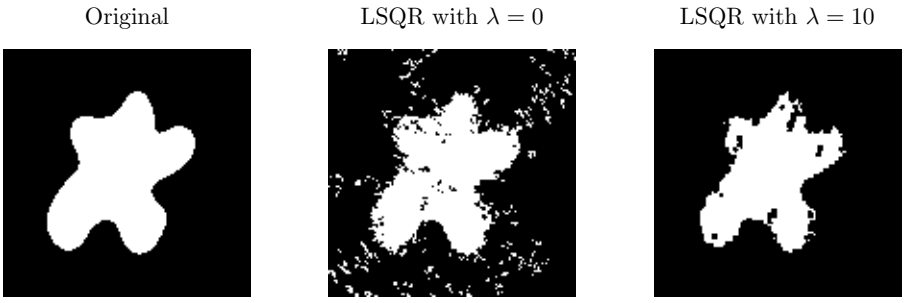


Figure 6.8: *The result after 100 DART iterations with LSQR as ARM on Blob, scanned using 10 angles. Left: the original image, middle: Using no Tikhonov regularization, right: Using Tikhonov regularization with* $\lambda = 10$.

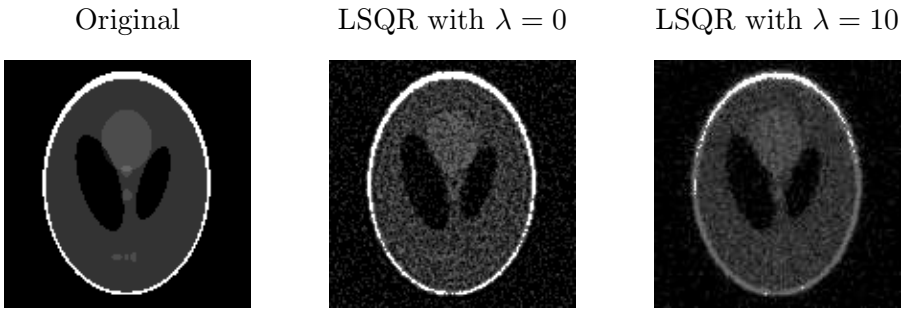


Figure 6.9: *The result after 100 DART iterations with LSQR as ARM on Phantom, scanned using 75 angles. Left: the original image, middle: Using no Tikhonov regularization, right: Using Tikhonov regularization with* $\lambda = 10$.

misclassified pixels. Figures 6.8 and 6.9 show the DART reconstructions with LSQR of Blob and Phantom from very noisy data (10^3) when no regularization was used and in the regularized case with $\lambda = 10$. It is clear that the right figures have fewer (or no) non-black pixels outside the object than the middle figures. Nonetheless one might argue that, especially for the Phantom reconstruction, the non-regularized reconstruction looks more accurate.

Better results might be obtained if the value of λ is dependent on the location of the pixel in the image. An object is usually located at the center of the reconstruction grid, whereas pixels close to the edge of the reconstruction grid are generally outside the object. Increasing the value of λ once a pixel is farther away from the center of the grid might thus decrease the speckles while keeping a superior reconstruction of the object.

During each DART iteration, after the intermediate ARM iterations, a smoothing operation is performed on the grey values (of the whole image, i.e. the new grey values for the free pixels plus the segmented values of the fixed pixels) after which the reconstruction is segmented. It might be that because of the Tikhonov regularization this smoothing has become redundant or, even worse, counter productive.

6.6 Alternative to the Fixed-Free Pixels Construct

This section investigates the final research question:

Are there alternatives for the fixed-free pixels construct?

This construct of fixing pixels can be circumvented by solving the following system:

$$\begin{pmatrix} W \\ D \end{pmatrix} \mathbf{f} = \begin{pmatrix} \mathbf{p} \\ D\mathbf{v} \end{pmatrix}. \tag{6.4}$$

Here $D \in \mathbb{R}^{N \times N}$ is a diagonal matrix with N, the number of pixels, diagonal elements d_i. The vector $\mathbf{v} \in \mathbb{R}^N$ can, for example, contain the presumed grey values of the image, i.e. the grey values after segmentation. Note that this formulation is very general and that it can also approximately formulate the original problem. Approximately, because the solution space in which the ARMs search for solutions is now different. In the original problem the ARM only had a reduced system $\tilde{W}$ at its disposal, CGLS, LSQR and CGNE for example would search for solutions $\tilde{\mathbf{f}}^j$ in the Krylov subspace $\tilde{\mathbf{f}}^0 + \mathcal{K}_j\left(\tilde{W}^T\tilde{W}, \tilde{W}^T\tilde{\mathbf{r}}^0\right)$. If one were to solve (6.4) with CGLS, LSQR or CGNE one would search for solutions $\mathbf{f}^j$ in the solution space

$$\mathbf{f}^0 + \mathcal{K}_j\left(\begin{pmatrix} W^T & D^T \end{pmatrix}\begin{pmatrix} W \\ D \end{pmatrix}, \begin{pmatrix} W^T & D^T \end{pmatrix}\mathbf{r}^0\right), \tag{6.5}$$

where

$$\mathbf{r}^0 = \begin{pmatrix} \mathbf{p} \\ D\mathbf{v} \end{pmatrix} - \begin{pmatrix} W \\ D \end{pmatrix}\mathbf{f}^0. \tag{6.6}$$

As mentioned the formulation in (6.4) is very general. If one chooses $d_i = 10^6$, or some other very high number, and v_i equal to the segmented grey value of the previous approximation for all the pixels i that would formerly be classified as fixed and $d_i = v_i = 0$ for the free pixels i, then the fixed-free construct is mimicked. Indeed, deviation from the value v_i for fixed pixels is penalised so heavily that the obtained solution will most likely have $f_i = v_i$ for these pixels. It is possible, however, for d_i to deviate from v_i, which is not possible in to classic DART. The constraints imposed by D and $\mathbf{v}$ are therefore called *weak constraints*. Tikhonov-like regularization can be carried out by choosing $d_i = \lambda$, $v_i = 0$ for the free pixels and $d_i = 10^6$, $v_i = f_i$ for the fixed pixels.

Obviously, the goal is to improve the performance of DART with this new formulation. This might be achieved by weakening the fixed-free construction. The value of the d_i can for example be dependent on the number of neighbours that have a different grey value, after segmentation, than pixel i. Or one can abandon the segmentation altogether and use the variation of grey values in the neighbourhood of pixel i as measure for the value of d_i.

During every DART iteration, after the intermediate ARM iterations and before segmentation, DART performs a smoothing operation. This smoothening has no theoretical bases but is carried out since it leads to better results. The idea behind smoothening the grey values before segmentation was that the effects of noise, which is distributed among quite a few free pixels,

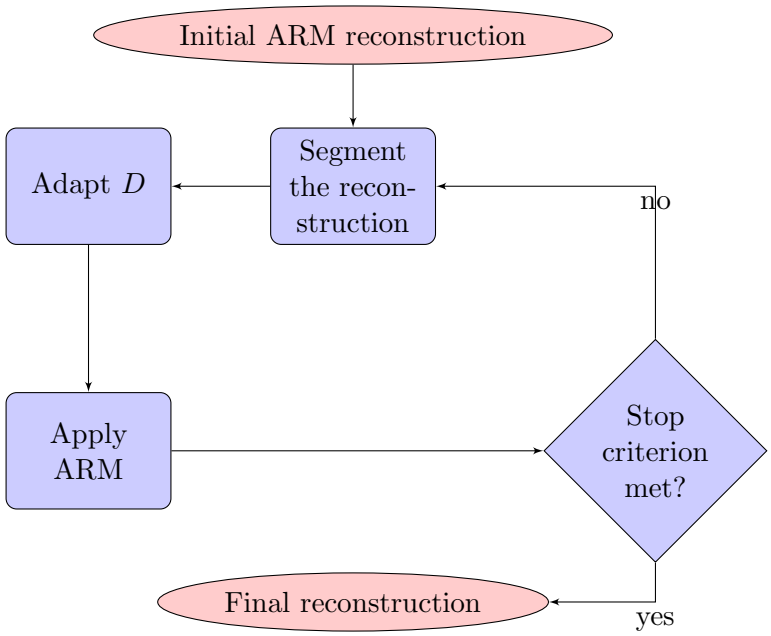


Figure 6.10: *Flowchart of the new DART algorithm.*

would be diminished. Since in the 'new DART' construct 6.4 no (truly) fixed pixels exist anymore there is no reason the do the smoothing. Hence this new DART does not perform a smoothing operation. In Figure 6.10 a flowchart of the new DART is given.

In the following sections various choices of the d_i will be proposed, subsequently the performance of this new DART is investigated.

6.6.1 Mimicking the Classical DART

First, before possible improvements are considered, the performance of the new DART is investigated when the fixed-free pixel construct is mimicked in order to see if this new formulation leads to comparable results. This is realized, as proposed in the previous section, by choosing $d_i = 10^6$ and v_i equal to the segmented grey value of the previous approximation for all the pixels i that would formerly be classified as fixed, and $d_i = v_i = 0$ for the free pixels. This penalizes the deviation of fixed pixels from their proposed grey values to such a degree that the resulting approximation will most likely still have those grey values for the fixed pixels. Experiments have shown that these grey values will vary, but to a very low degree. The maximal difference between the found grey value and the segmented grey value of fixed pixels is $\ll 1$, this has no notable effect on the outcome of the subsequent segmentation of course.

In Appendix A.4.1: Figures A.19 – A.24 the results of the experiments are shown. For every test problem the performance of the classical DART (left figure) and the new DART (middle figure) is shown (*before* and *after*). Both types of the DART algorithm performed 20 initial and intermediate ARM iterations, in line with what was concluded to be the best setup in Section 6.4. For the new DART also the minimal pixel error is included (right figure). Note that the pixel errors after 100 iterations (*final pixel error*) are higher than the minimal pixel errors confirming that also this new DART does not produce a sequence of monotonically decreasing errors. The first thing that catches the eye is the performance of SIRT. For almost every scenario

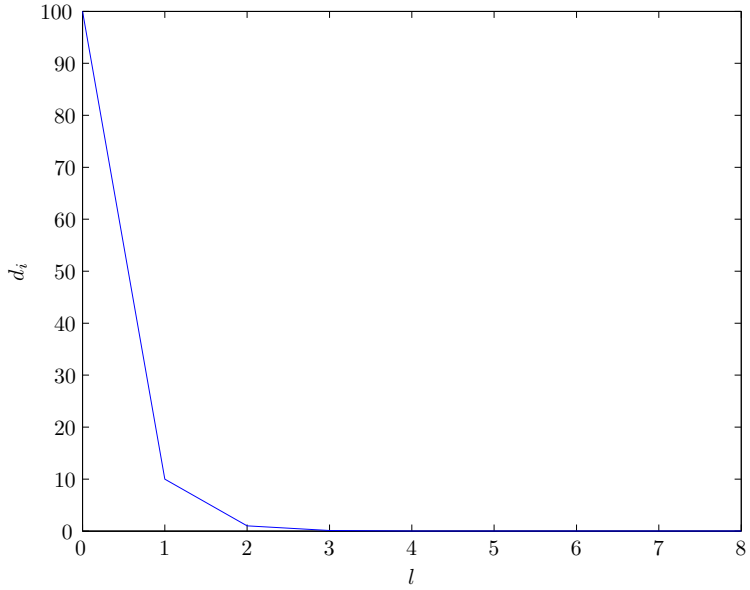


Figure 6.11: *The value of d_i corresponding to the number of neighbours l with a different segmented grey value than pixel i.*

the performance of SIRT is worse than it was before. LSQR, however, performs better in almost all the experiments. LSMR has a slightly superior reconstruction than LSQR, in accordance with what has been seen throughout. For some of the test problems, Two Circles, Shapes and Phantom, classical DART with SIRT outperforms the new DART with LSQR. The reason for the bad performance of SIRT with the new DART is most likely because it needs more ARM iterations to converge for this new formulation since the number of variables is considerably higher than in the classical DART case (recall the slow convergence of SIRT w.r.t. CGLS).

6.6.2 More Degrees of Fixedness

The advantage of the new formulation 6.4 is that one can introduce more degrees of fixedness. This can be achieved by making the value of d_i dependent on the number of neighbouring pixels that have the same segmented grey value. If this number is high the corresponding d_i should be high, lower numbers should correspond to lower values of d_i. In the right-hand side the $\mathbf{v}$ is chosen to be the previous segmented grey values of the pixels. For the following experiments the relationship

$$d_i = \frac{100}{10^l} \tag{6.7}$$

is used, with l the number of neighbours with different segmented grey values than pixel i. The 8-connected neighbourhood is used. Note that the values of the d_i are much smaller than they were in mimicking of the fixed-free construct. This allows for more flexibility in the grey values.

In Appendix A.4.2: Figure A.25 – A.30 the performance of this new construct is shown. The left figure shows the performance of the classical DART, the middle figure the performance of the new DART when the fixed-free construct is mimicked and the right figure shows the performance of the new DART with this neighbours construct. Overall this new construct seems to perform

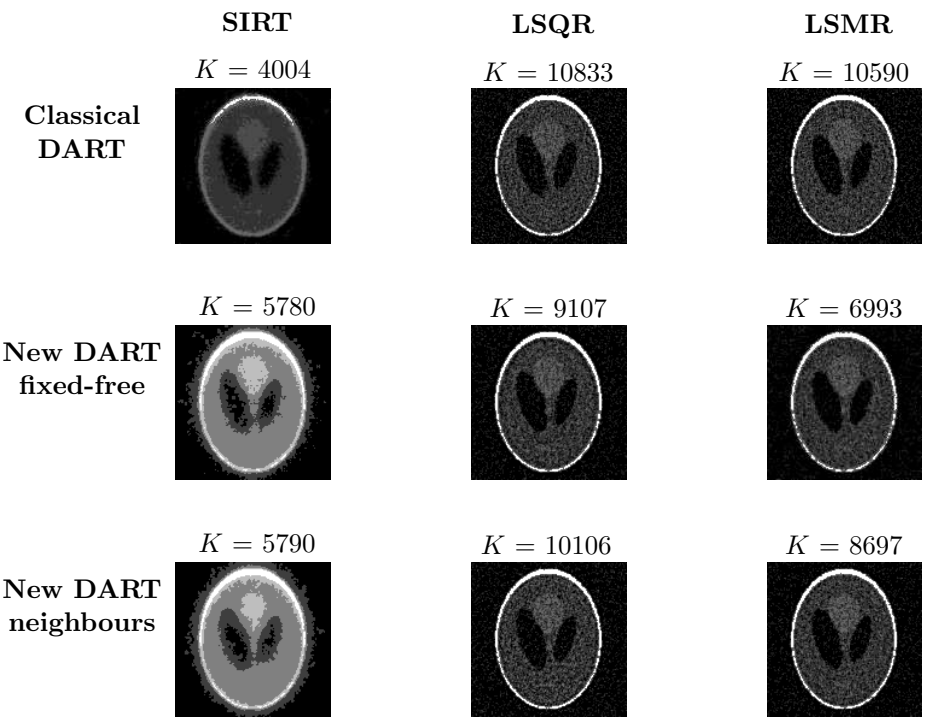


Figure 6.12: *The performance of the classical DART, new DART with fixed-free construct and new DART with the neighbours construct. Test problem Phantom, scanned using 75 angles and the data was polluted with* 10^3 *noise.*

somewhat better than the fixed-free construct, but the results of SIRT are still not what one would expect.

In this and the previous experiment it seems that SIRT is the superior ARM when it comes to dealing with very noisy data since it has the smallest pixel error. If one looks at the actual reconstructions, however, Figure 6.12, one sees that the SIRT reconstructions are actually very poor. The reason that LSQR and LSMR have higher pixel errors is because of the speckles inside and around the object (cf. the discussion about high values of λ in Section 6.5). SIRT seems to have a strong scaling property built in that causes the algorithm to favour homogeneous areas. Better results might be obtained if SIRT has more time to convergence. Recall that SIRT showed a slower convergence than LSQR and LSMR in Section 3.8.

The performance of this construct seems to depend to a certain degree upon the choice of the weight function (6.7). An other function might yield somewhat better, or poorer results.

6.6.3 Number of ARM iterations

In Section 6.4.2 it was concluded that fewer intermediate iterations produced better results for every ARM. The corresponding reduced system of this original DART contains a fraction of the variables (free pixels) of what the new DART formulation has, namely all the pixels. SIRT is known to have a slow convergence compared to ART and SART, thus maybe the poor performance of SIRT can be attributed to the number of intermediate iterations that are performed. Hence, in this section it is investigated if 20 ARM iterations is still the best option for the considered ARMs.

Figure 6.13: *The performance of the new DART with the neighbours construct and SIRT as ARM for a various number of intermediate iterations. The test problem Phantom was scanned using 90 angles and the measurement data were polluted with* 10^3 *noise.*

In Appendix A.4.3: Figure A.31 – A.36 one can find the results of the new DART formulation with a various number intermediate ARM iterations. The considered ARMs are SIRT and LSQR, the results of LSMR are not included but are similar to those of LSQR, as was to be expected. For LSQR the conclusion of Section 6.4.2 still holds, 20 iterations, especially when the data is noisy, produces the best results. SIRT, however, has the worst performance for every test problem when only 20 initial and intermediate iterations are carried out. It seems that at least 75 iterations are needed to achieve better results and 200 iterations produce the best results overall. This many iterations has a major drawback though, the computation time is increased drastically. Keeping this in mind 100 ARM iterations seems to be a good choice, it has a nice performance for all the test problems and requires roughly half the computation time of 200 iterations.

In Figure 6.13 the results of 100 new DART iterations with the neighbours construct are shown when SIRT is used as ARM for a various number of intermediate iterations. The test problem was Bone, that was scanned using 90 angles. The resulting measurement data was polluted with noise level 10^3, corresponding to very much noise. From this figure one finds that the performance of DART with SIRT improves as the number of ARM iterations increases, as was also to be expected from Figure A.34. Also the quality of the reconstructions are improving, for a small number of iterations the large outline of the object is there, but it lacks detail. Increasing the number of intermediate iterations introduces more and more detail. This same behaviour is also observed for the other test problems. In Table 6.3 the computation times of the 100 DART iterations are listed per number of intermediate SIRT iterations. The test problem Shapes was scanned using 30 angles and the resulting measurement data were polluted with 10^3 noise. Note

that indeed the times seem to increase linearly, indicating that the majority of the computation time is needed for the intermediate ARM iterations.

No. Iterations	Time (s)
20	71.0791
35	115.1104
50	160.3452
75	247.9419
100	325.7848
150	467.3016
200	618.3459

Table 6.3: *The computation times of 100 new DART iterations with SIRT for a various number of ARM iterations in seconds.*

6.7 The Reconstructed Residual Error

Throughout this section the reconstructions were obtained by running 100 DART iterations. Unfortunately, the pixel error for subsequent approximations is not monotonically decreasing meaning that in general the approximation found after 100 iterations does not have minimal pixel error. In real world problems one does not know the original image and thus cannot compute the pixel error. One does, however, know the *residual* of an approximation $\mathbf{f}^k$ which is defined as $\mathbf{r}^k = W\mathbf{f}^k - \mathbf{p}$. Unfortunately, minimizing the residual, which CGLS for example does, does not result in a minimal error in general and vice versa. In tomography one generally would like to find solutions that have minimal error since the error is more connected to the quality of the image than the residual.

An idea set forth in [24] by Roelandts et al. is to visualize the *segmentation error* using the residual of an approximation. The idea is to forward project the segmented approximation $\mathbf{f}_s$ by computing $W\mathbf{f}_s = \mathbf{p}_s$. The resulting vector $\mathbf{p}_s$ then corresponds to the projection data of the segmented approximation. Now let $\mathbf{e}_s = \mathbf{p} - \mathbf{p}_s$ be the deviation of the projection data from the measured data called the *residual projection error*. This error $\mathbf{e}_s$ can be back projected, i.e. a reconstruction of it can be found using any ARM. An approximate reconstruction $\tilde{\mathbf{y}}$ is found by solving $W\mathbf{y} = \mathbf{e}_s$. This reconstruction is referred to as the *reconstructed residual error* (RRE). The RRE is a visualization of the segmentation error, which can be very instructive and useful. In [24] it is demonstrated how the RRE can be used to adjust the allowed grey values $\{\rho_1, \rho_2, \ldots, \rho_l\}$, this is helpful if these values were not known exactly a priori. Furthermore, according to [24] it seems that the RRE resembles the real error of the approximation, i.e. $\mathbf{e} = \mathbf{f}^k - \hat{\mathbf{f}}$ where $\mathbf{f}^k$ is the approximation *before* segmentation and $\hat{\mathbf{f}}$ is the original object. If that is indeed the case, then the RRE can be used to approximate the error of the reconstruction, providing one with a quantitative error measure. Figure 6.14 shows an example of the RRE and the corresponding real error, note that it is from the outline in the picture one can see that the test problem was Phantom. The RRE does seem to resemble the real error to some extent.

Every pixel will have an individual RRE. A negative value indicates that the segmented value might be too high while a positive RRE suggests that the segmented grey value is too low. The closer the RRE is to zero, the better the quality of the segmentation. This motivates the idea

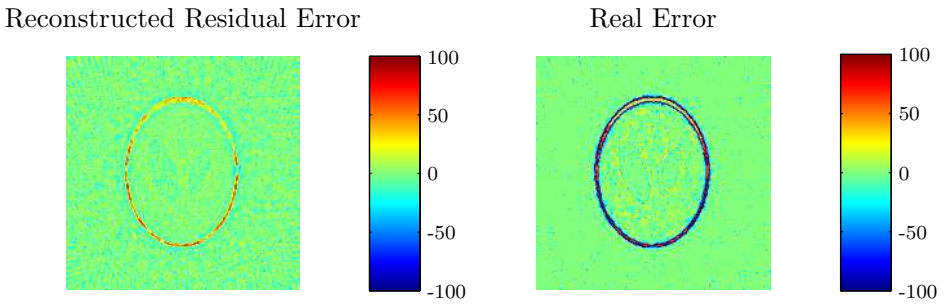


Figure 6.14: *An example of what the reconstructed residual error looks like. Left: the RRE, right: the real error.*

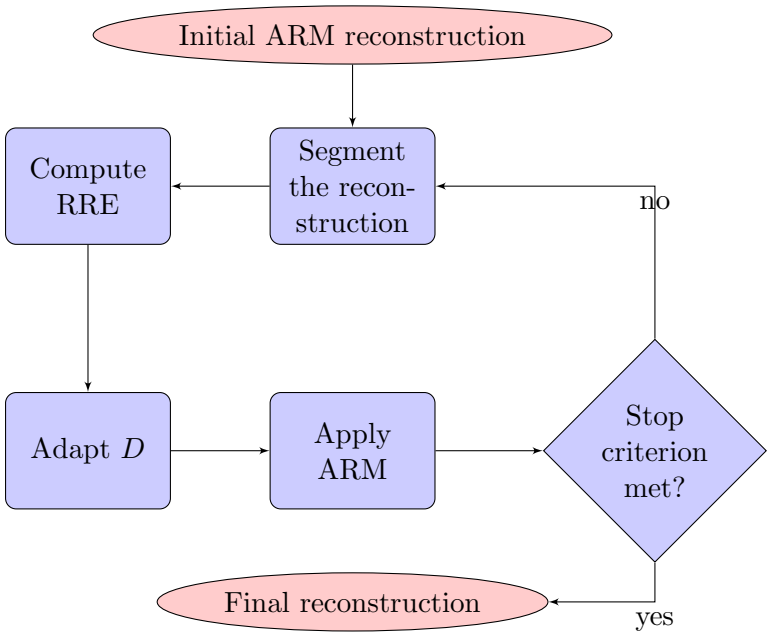


Figure 6.15: *Flowchart of the new DART algorithm with the RRE incorporated.*

for an error quantification that is based on the total deviation of RRE from zero. One simply sums up the absolute RRE values of all the pixels to arrive at a number which supposedly gives a measure for the error of the segmented reconstruction. One should note, however, that using the RRE approximately doubles the computation time for in one DART iteration two systems are solved, i.e. the actual system and $W\mathbf{y} = \mathbf{e}_s$. The latter system is solved with the same ARM and same number of ARM iterations as is chosen for DART. Figure 6.15 shows a flowchart of the new DART that also computes the RRE.

In Figure 6.16 the real error $\|\hat{\mathbf{f}} - \mathbf{f}^k\|_2$, the pixel error K and the absolute sum of the RRE are shown for various noise levels per DART iteration. The test problem was Blob which was scanned using 10 angles. SIRT was chosen as ARM and 100 ARM iterations were used to arrive at the results. Note that the asterisks on the plot lines indicate a minimum. The real error, pixel error and reconstructed residual error all seem to develop in a similar manner, indicating that the absolute RRE sum is a good error approximation. For all noise levels except 10^3 the minima of the pixel error and the RRE occur in the same iteration number. Although the minima for

10^3 are different, the pixel error of the RRE is 153 versus 151 of the minimal K. The iteration numbers in which the minima of the real error occurs do not always coincide with those of the pixel error, but the actual pixel error is (almost) the same. Table 6.4 lists the pixel errors of the various minima corresponding to the different noise levels.

Noise Level	Real Error	Pixel Error	RRE
No noise	30	29*	29*
10^6	30	30*	30*
10^5	31*	31*	31*
10^4	50*	50*	50*
10^3	152	151*	153

Table 6.4: *The pixel error K corresponding to the approximation which was minimal w.r.t. the real error (2nd column), the pixel error (3rd column) or the absolute RRE sum (4th column). Test problem Blob scanned using 10 angles and SIRT was the ARM. The '*' indicates that the iteration number of the minimum coincides with the minimal pixel error.*

Figure 6.17 shows an equivalent experiment, but now the test problem was Shapes, scanned with 30 projection angles, and LSQR was used, performing 20 iterations every DART iteration. The development of the errors seem to be similar except for the highest noise level. The iterations numbers of the minimum pixel error and minimum RRE coincide only for noise level 10^6. The minima of the real error only coincides for noise level 10^5. Table 6.5 lists the pixel errors of the various minima corresponding to the different noise levels. Note that the pixel errors of all the noise levels except 10^3 are practically the same for all the errors. For the highest noise level the RRE measure seems to produce a poor result, though this result would be comparable to what one would find if one takes the 100-th approximation.

Noise Level	Real Error	Pixel Error	RRE
No noise	117	110*	111
10^6	115	112*	112*
10^5	131*	131*	136
10^4	249	246*	263
10^3	1154	1109*	1797

Table 6.5: *The pixel error K corresponding to the approximation which was minimal w.r.t. the real error (2nd column), the pixel error (3rd column) or the absolute RRE sum (4th column). Test problem Shapes scanned using 30 angles and LSQR was the ARM. The '*' indicates that the iteration number of the minimum coincides with the minimal pixel error.*

Figure 6.18 shows the results of the same experiments as Figure 6.17 but now with SIRT as ARM. Note that the development of the pixel error and the absolute RRE sum is somewhat similar, though the pixel error seems to alternate a lot more. The minima of the real error and pixel error never coincide, for the RRE and the pixel error the first three noise levels find the same iteration number as minimum. Table 6.6 lists the pixel errors of the various minima corresponding to the different noise levels. The pixel error of the approximation that had a minimal real error is always somewhat higher, but very close to the minimal pixel error. The pixel error of the minimal RRE differs from the minimum pixel error only for noise level 10^3, and then the difference is only 3 pixels.

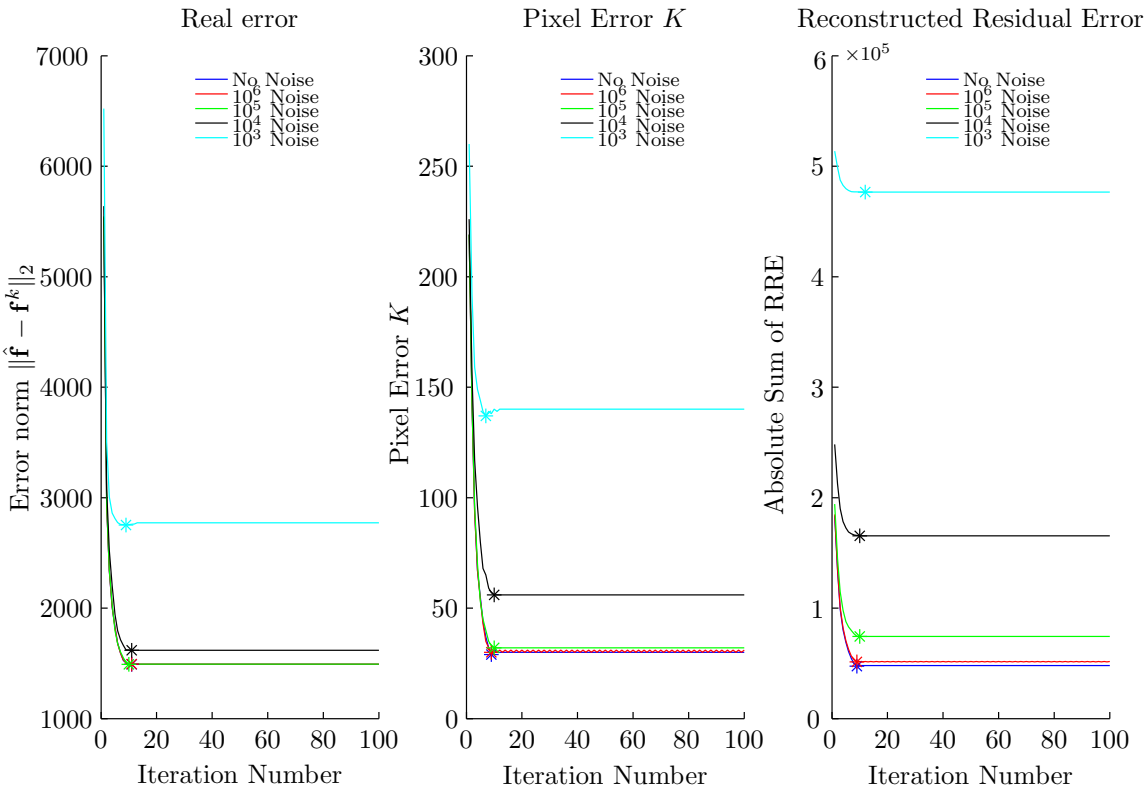


Figure 6.16: *Left: The Error norm of DART for various noise levels, middle: The Pixel Error K, right: The absolute sum of the RRE for various noise levels. The asterisks on the plot lines indicate a minimum. Test problem Blob scanned using 10 angels, chosen ARM was SIRT.*

Noise Level	**Real Error**	**Pixel Error**	**RRE**
No noise	578	568*	568*
10^6	583	582*	582*
10^5	570	564*	564*
10^4	625	616*	616
10^3	862	861*	864

Table 6.6: *The pixel error K corresponding to the approximation which was minimal w.r.t. the real error (2nd column), the pixel error (3rd column) or the absolute RRE sum (4th column). Test problem Shapes scanned using 30 angles and SIRT was the ARM. The '*' indicates that the iteration number of the minimum coincides with the minimal pixel error.*

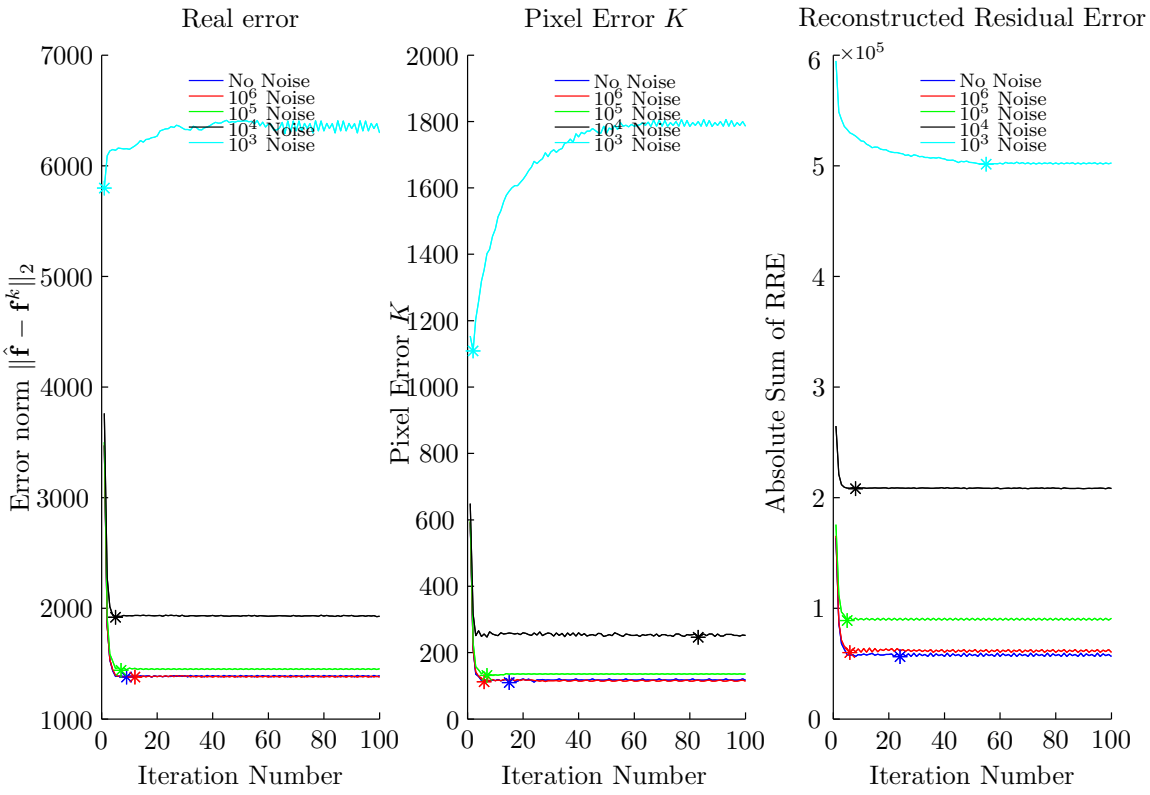


Figure 6.17: *Left: The Error norm of DART for various noise levels, middle: The Pixel Error K, right: The absolute sum of the RRE for various noise levels. The asterisks on the plot lines indicate a minimum. Test problem Shapes scanned using 30 angels, chosen ARM was LSQR.*

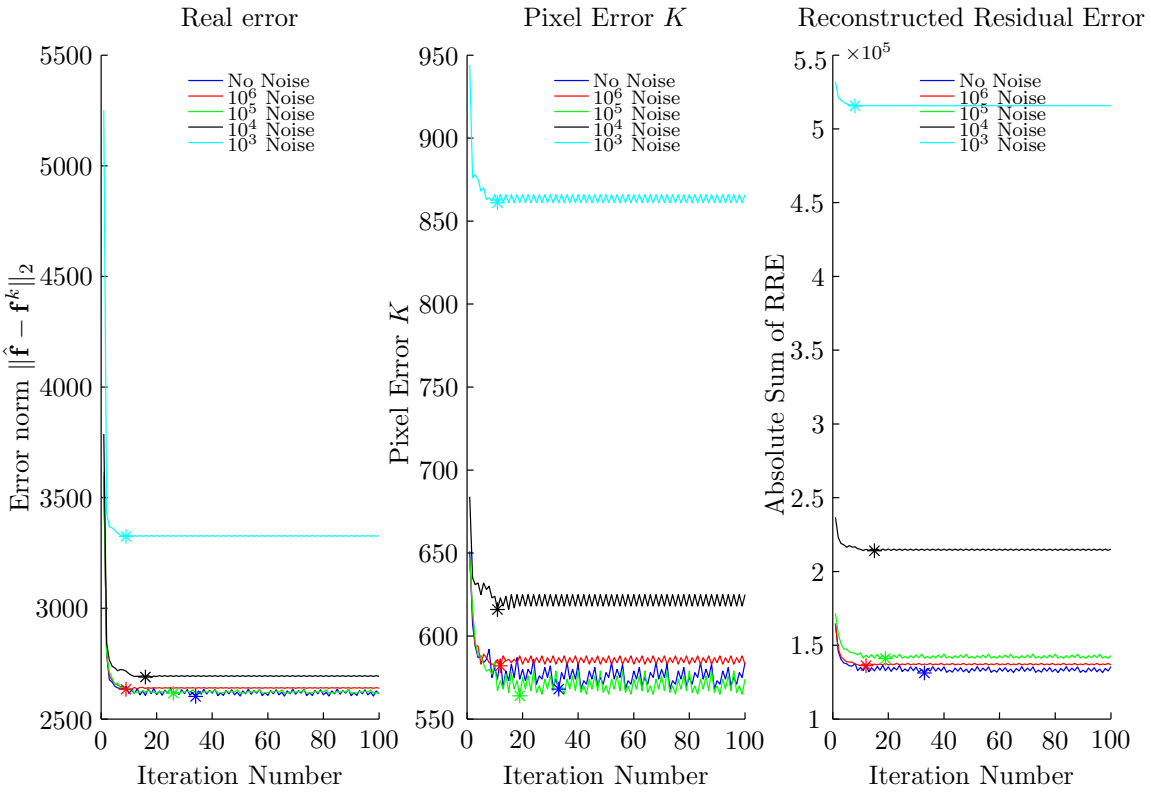


Figure 6.18: *Left: The Error norm of DART for various noise levels, middle: The Pixel Error K, right: The absolute sum of the RRE for various noise levels. The asterisks on the plot lines indicate a minimum. Test problem Shapes scanned using 30 angels, chosen ARM was SIRT.*

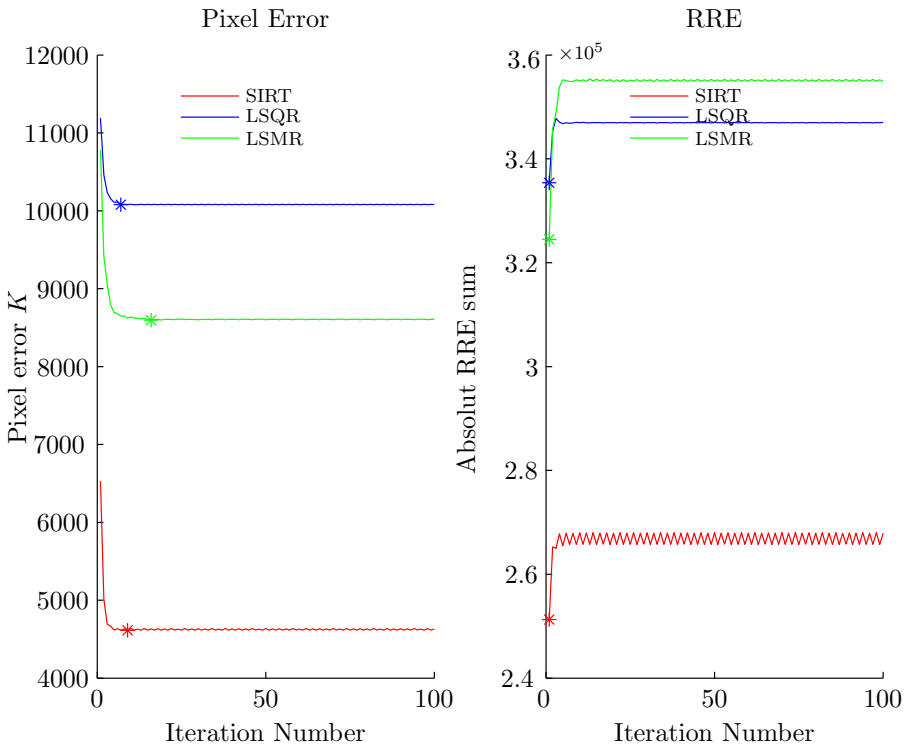


Figure 6.19: *The pixel error and absolut RRE sum for test problem Phantom scanned using 75 angles, noise level* 10^3

In Appendix A.5: Figure A.37 – A.42 the results of experiments with the absolute RRE sum as error measure are listed. In the case that SIRT was the ARM 100 intermediate iterations were used, for LSQR and LSMR 20 intermediate iterations were performed. After 100 DART iterations it was determined which iteration number had minimal absolute RRE sum and for that approximation the pixel error was recorded. The experiment was run for all the test problems, with the usual number of projection angles, and noise levels. In the figures the red line represents the pixel error K after 100 DART iterations, the blue line when the RRE was minimal and the asterisk is the actual minimal pixel error. Note that for LSQR and LSMR the new measure does not improve the results of DART, but it does not have an adverse effect either. SIRT performs somewhat better, especially for the test problem Bone where it finds a reconstruction with minimal pixel error for every noise level. The test problem Phantom seems to result in bad error approximations for all ARMs for the higher noise levels. Figure 6.19 shows the development of the pixel error and the absolute RRE sum in the scenario with noise level 10^3. Note that the minimal RRE is found as the first approximation for every ARM, but the pixel error is far from minimal for those approximations. Figure 6.20 shows the actual reconstructions for all ARMs when the pixel error is minimal or when the absolute RRE sum is minimal. Note that although the pixel error is higher for the minimal RRE approximations, the actual reconstructions are not notably poorer than when the pixel error is minimal, it contains somewhat more speckles.

The RRE does seem to give a rather accurate error measure, albeit in retrospect. Using the RRE to quantify the quality of a reconstruction corresponds to an educated guess and is thus

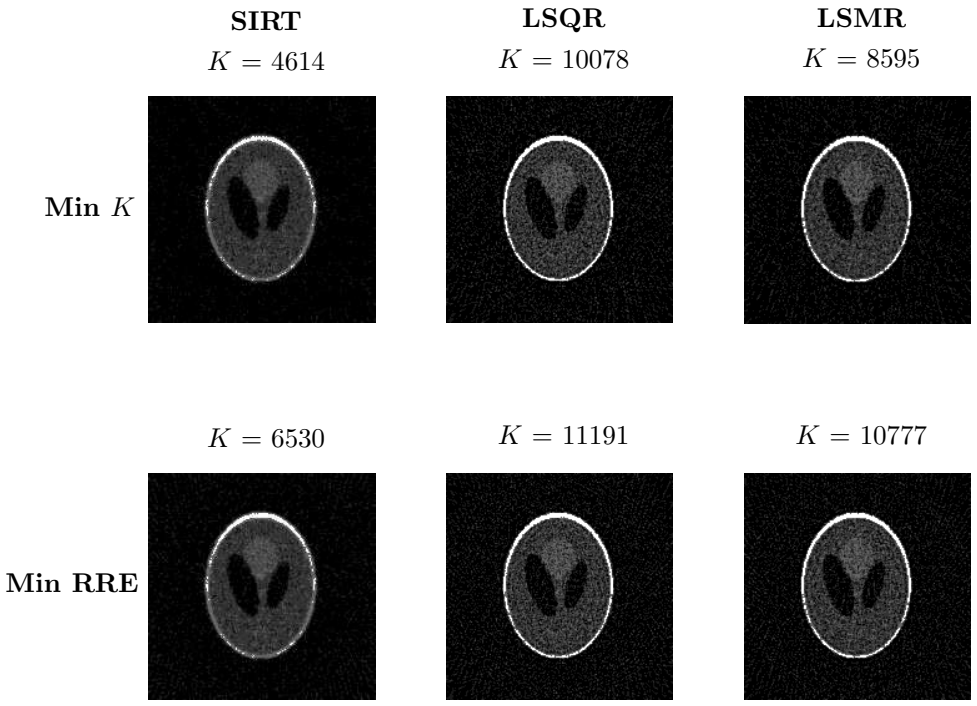


Figure 6.20: *The reconstruction for the minimal RRE and the minimal pixel error for all the three ARMs of test problem Phantom with noise level* 10^3.

better than choosing the last approximation as final reconstruction, which is actually just a shot in the dark. Also note that from the development of the pixel errors that are shown in this section one can conclude that in most cases DART will have found a minimal (in the sense of minimal pixel error) reconstruction long before 40 iterations were performed. This indicates that 50 DART iterations are sufficient in most cases to find a satisfying reconstruction.

6.8 A closer look at SIRT

In Section 3.2 SIRT was introduced. Subsequent approximations $\mathbf{f}^k$ were found as

$$\mathbf{f}^k = \mathbf{f}^{k-1} + CW^T R\mathbf{r}^{k-1}, \tag{6.8}$$

where C and R correspond to diagonal matrices containing the inverse column and row sums, respectively. According to [30, p. 63] SIRT applied to the system $W\mathbf{f} = \mathbf{p}$ with $\mathbf{f}^0 = \mathbf{0}$ actually finds the minimum norm least squares solution to the system $A\mathbf{x} = \mathbf{b}$ with $A = R^{1/2}WC^{1/2}$, $\mathbf{b} = R^{1/2}\mathbf{p}$ and then takes $\mathbf{f} = C^{1/2}\mathbf{x}$.

Solving a system $RW\mathbf{f} = R\mathbf{p}$, with R a diagonal matrix, will, in general, results in another least squares solution than the original system $W\mathbf{f} = \mathbf{p}$ since the residual $\mathbf{p} - W\mathbf{f}$ is measured in another norm. Likewise, if one finds the minimum norm least squares solution $\mathbf{x}$ for the system $WC\mathbf{x} = \mathbf{p}$, with C diagonal, and takes $\mathbf{f} = C\mathbf{x}$ one will, in general, also find another least squares solution than for the original system (which is assumed to be rank deficient) since $\mathbf{f}$ is measured in another norm. This latter case corresponds to what is called a *preconditioning*

which, in general, is used to decrease the condition number of the system matrix in order to have faster convergence. However, if the matrix is rank deficient the preconditioned system will generally lead to a different least squares solution for the reasons given before.

The altered systems discussed above actually scale the rows and the columns of the system matrix when R and C are as they are in SIRT. Of course one should be careful with doing these scalings and always keep in mind that a meaningful solution is found. In SIRT it might be argued that this choice of R scales proportional to the influence a ray has to the total system and C proportional to the number of rays that crosses a certain pixel (cf. Section 2.3).

This scaling of the rows and columns might be the reason why SIRT outperforms LSQR and LSMR in most cases. Therefore it is sensible to try to apply this scaling to LSQR and LSMR too. Since the algorithms of these methods are not given explicitly in this work it is more instructive to apply this scaling to CGLS, which, as was seen in Section 6.4, produced similar results as LSQR. Recall that mathematically CGLS and LSQR are equivalent. The CG algorithm for square systems can be written as:

CG

Choose $\mathbf{x}^0$
$\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$
$\mathbf{d}^0 = \mathbf{r}^0$
for $j = 1, 2, \ldots$ until convergence **do**
  $\alpha_{j-1} = \langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle / \langle A\mathbf{d}^{j-1}, \mathbf{d}^{j-1} \rangle$
  $\mathbf{x}^j = \mathbf{x}^{j-1} + \alpha_{j-1}\mathbf{d}^{j-1}$
  $\mathbf{r}^j = \mathbf{r}^{j-1} - \alpha_{j-1}A\mathbf{d}^{j-1}$
  $\beta_{j-1} = \langle \mathbf{r}^j, \mathbf{r}^j \rangle / \langle \mathbf{r}^{j-1}, \mathbf{r}^{j-1} \rangle$
  $\mathbf{d}^j = \mathbf{r}^j + \beta_{j-1}\mathbf{d}^{j-1}$
end for

The system SIRT solves is $R^{1/2}WC^{1/2}\mathbf{x} = R^{1/2}\mathbf{p}$, note that C and R are diagonal matrices and thus evidently symmetric. The normal equations of this system equals

$$\begin{aligned} \left(R^{1/2}WC^{1/2}\right)^T \left(R^{1/2}WC^{1/2}\right)\mathbf{x} &= \left(R^{1/2}WC^{1/2}\right)^T R^{1/2}\mathbf{p} \quad &\Leftrightarrow \\ C^{1/2}W^TRWC^{1/2}\mathbf{x} &= C^{1/2}W^TR\mathbf{p} \quad &\Leftrightarrow \\ CW^TRW\mathbf{f} &= CW^TR\mathbf{p} \end{aligned} \tag{6.9}$$

Recall that SIRT put $\mathbf{f} = C^{1/2}\mathbf{x}$. From (6.9) one can conclude that the system computes the residual in a different norm and does a left preconditioning with C. When applying CG to this system one will obtain the *scaled CGLS* algorithm. Note that while $\mathbf{r}$ corresponds to the residual of the original problem this is not the residual of the system that is being solved. The residual of this scaled system is $\tilde{\mathbf{z}} = W^TR\mathbf{r}$. The preconditioning consequently yields the preconditioned residual $\mathbf{z} = C\tilde{\mathbf{z}}$.

Multiplying with the preconditioner C might render the system matrix non-symmetric. In [26, p. 245] it is suggested that symmetry is preserved if C^{-1} is in the form of a Cholesky factorization,

i.e. $C^{-1} = LL^T$, but this is not necessary since the operator CW^TRW is self-adjoint in the C^{-1}-inner product:

$$\begin{aligned}\left\langle CW^TRW\mathbf{x}, \mathbf{y}\right\rangle_{C^{-1}} &= \left\langle W^TRW\mathbf{x}, \mathbf{y}\right\rangle \\ &= \left\langle \mathbf{x}, W^TRW\mathbf{y}\right\rangle \\ &= \left\langle \mathbf{x}, C^{-1}CW^TRW\mathbf{y}\right\rangle \\ &= \left\langle \mathbf{x}, CW^TRW\mathbf{y}\right\rangle_{C^{-1}}. \end{aligned} \tag{6.10}$$

Thus if every Euclidean inner product in CG is replaced by the C^{-1}-inner product, symmetry is preserved. This results in changes in terms α and β:

- $\alpha_{j-1} = \left\langle \mathbf{z}^{j-1}, \mathbf{z}^{j-1}\right\rangle_{C^{-1}} / \left\langle CA\mathbf{d}^{j-1}, \mathbf{d}^{j-1}\right\rangle_{C^{-1}} = \left\langle \tilde{\mathbf{z}}^{j-1}, \mathbf{z}^{j-1}\right\rangle / \left\langle A\mathbf{d}^{j-1}, \mathbf{d}^{j-1}\right\rangle$,
- $\beta_{j-1} = \left\langle \mathbf{z}^{j}, \mathbf{z}^{j}\right\rangle_{C^{-1}} / \left\langle \mathbf{z}^{j-1}, \mathbf{z}^{j-1}\right\rangle_{C^{-1}} = \left\langle \tilde{\mathbf{z}}^{j}, \mathbf{z}^{j}\right\rangle / \left\langle \tilde{\mathbf{z}}^{j-1}, \mathbf{z}^{j-1}\right\rangle$,

where $A = W^TRW$. Note that the denominator of α equals $\left\langle W^TRW\mathbf{d}^{j-1}, \mathbf{d}^{j-1}\right\rangle$ or if rewritten $\left\langle W\mathbf{d}^{j-1}, RW\mathbf{d}^{j-1}\right\rangle$ which is actually the same denominator as in the original CGLS algorithm, but now computed in the R-inner product. The resulting algorithm is given below.

Scaled CGLS

Choose $\mathbf{f}^0$
$\mathbf{r}^0 = \mathbf{p} - W\mathbf{f}^0$
$\tilde{\mathbf{z}}^0 = W^TR\mathbf{r}^0$
$\mathbf{z}^0 = C\tilde{\mathbf{z}}^0$
$\mathbf{d}^0 = \mathbf{z}^0$
for $j = 1, \ldots$ until convergence **do**
  $\alpha_{j-1} = \left\langle \tilde{\mathbf{z}}^{j-1}, \mathbf{z}^{j-1}\right\rangle / \left\langle W\mathbf{d}^{j-1}, RW\mathbf{d}^{j-1}\right\rangle$
  $\mathbf{f}^j = \mathbf{f}^{j-1} + \alpha_{j-1}\mathbf{d}^{j-1}$
  $\mathbf{r}^j = \mathbf{r}^{j-1} - \alpha_{j-1}W\mathbf{d}^{j-1}$
  $\tilde{\mathbf{z}}^j = W^TR\mathbf{r}^j$
  $\mathbf{z}^j = C\tilde{\mathbf{z}}^j$
  $\beta_{j-1} = \left\langle \tilde{\mathbf{z}}^{j}, \mathbf{z}^{j}\right\rangle / \left\langle \tilde{\mathbf{z}}^{j-1}, \mathbf{z}^{j-1}\right\rangle$
  $\mathbf{d}^j = \mathbf{z}^j + \beta_{j-1}\mathbf{d}^{j-1}$
end for

Note that scaled CGLS is equivalent to solving $R^{1/2}WC^{1/2}\mathbf{x} = R^{1/2}\mathbf{p}$ with CGLS and then taking $\mathbf{f} = C^{1/2}\mathbf{x}$. Thus this scaling can be applied to LSQR and LSMR without changing the algorithms. These algorithms will be referred to as *scaled LSQR* and *scaled LSMR*. The hope is that the fast convergence of CGLS, LSQR and LSMR will be combined with the superior reconstruction capabilities of SIRT.

The rate of convergence for data with noise of these new scaled algorithms is investigated in the same way it was done in Section 3.8, i.e. a sinogram of Blob is created using 15 projection angles. These data are polluted with noise corresponding to 10^3 number of counts. Subsequently, 250 iterations of the algorithms are performed and the corresponding relative error $\mathbf{e}^k = \dfrac{\|\hat{\mathbf{f}} - \mathbf{f}^k\|_2}{\|\hat{\mathbf{f}}\|_2}$

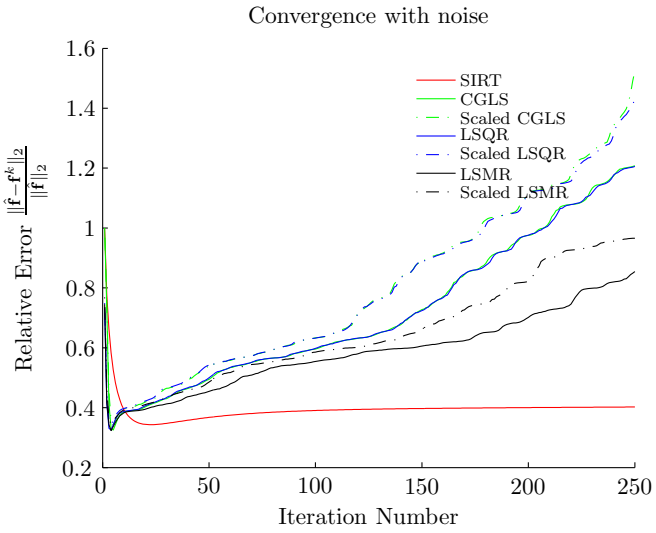


Figure 6.21: *The rate of convergence for the various scaled and non-scaled ARMs when the measurement data was polluted with* 10^3 *noise.*

is computed for each approximation $\mathbf{f}^k$, where $\hat{\mathbf{f}}$ is the vector of the original image. Figure 6.21 and 6.22 show the results of these experiments. From the figures it seems that the scaled ARMs do not perform as well as hoped; the minimal error reconstruction still has a higher error than the original ARMs and the divergence after that is greater than the original algorithms. Nonetheless, some experiments for new DART with these scaled ARMs will be carried out.

Another interesting aspect to investigate is if these scaled ARMs are, just as SIRT, not able to find holes when the fix probability is equal to 1. Running the same experiment as in Section 6.2 one finds that scaled CGLS actually still is able to find the hole in the Blob, as can be seen in Figure 6.23. The same is observed for scaled LSQR and scaled LSMR.

In Appendix A.6: Figures A.43 – A.48 the results of experiments with the scaled CGLS and scaled LSQR algorithms are shown. The new DART construct was used where the number of neighbours with a different segmented grey value determined the value of the diagonal elements of the regularization matrix D. The reconstruction with minimal absolute sum of the Reconstructed Residual Error was considered to be the best reconstruction. All ARMs except SIRT used 20 initial and intermediate iterations, SIRT used 100 iterations. First of all note that both scaled algorithms behave similarly, which was to be expected since they are mathematically equivalent. In some cases the scaled algorithms seem to behave as SIRT (Hexagon, Blob), in other cases they give practically the same results as the non-scaled algorithms (Bone, Phantom). In the case of Two Circles the results seem to be miserable, but note the scale on the left axes: 0.25% corresponds to only 40 misclassified pixels and thus the actual pixel error is marginal. This difference can be explained because of the way simulations are carried out, every simulation a new noisy sinogram is created and thus neither of the simulations use the same data (recall the discussion at the end of Section 6.1).

In Figure 6.24 the reconstructions of SIRT, LSQR and the scaled LSQR for the highest noise levels are depicted. The results of the scaled CGLS algorithm are comparable to those of scaled LSQR. Note that in almost all cases scaled LSQR gives comparable or better results. Scaled LSQR performs better than SIRT for Bone and Phantom, but for Shapes SIRT is better able

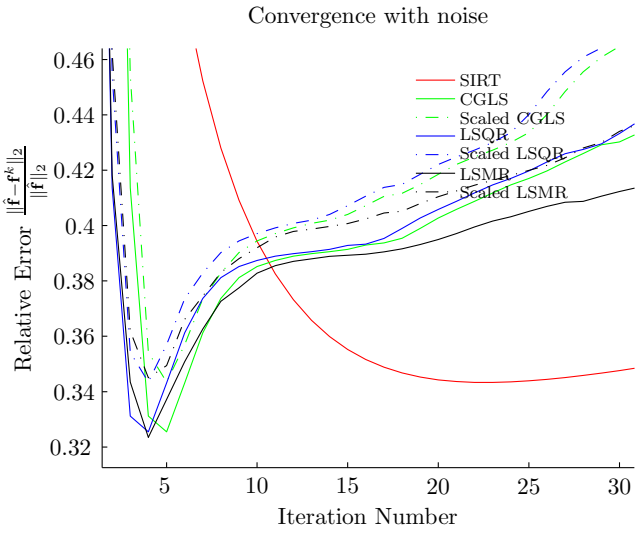


Figure 6.22: *The rate of convergence for the various scaled and non-scaled ARMs when the measurement data was polluted with* 10^3 *noise zoomed in on the first few iterations.*

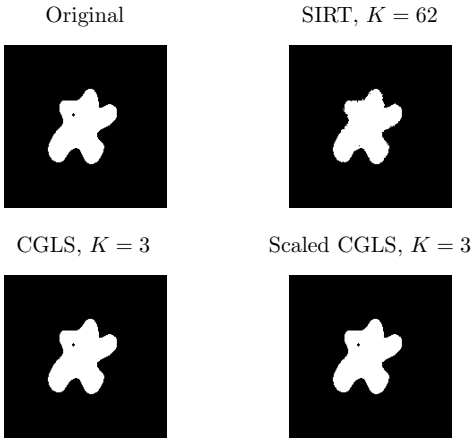


Figure 6.23: *Scaled CGLS is able, as regular CGLS, to find the hole in the Blob.*

to reconstruct the actual shapes as homogeneous regions with very little speckles on them. The grey circle and diamond for example contain very few white and black pixels in comparison to the scaled LSQR. But the white shapes have a thick grey border, something that is only observed to a low degree for scaled LSQR. The main advantage is that scaled LSQR only needed 20 intermediate iterations to arrive at these results while SIRT needed 100 iterations. For example, the new DART with 100 intermediate SIRT iterations needs about 165 seconds to solve Shapes (scanned using 30 angles) without any noise in the measurement data. For the same problem scaled LSQR needs only about 44 seconds. Scaled LSQR and scaled CGLS seem to produce the best results if a small number of intermediate iterations is used, like LSQR and CGLS. For a higher number of intermediate iterations the effects of the noise increases which is most likely because of the aforementioned semi-convergence.

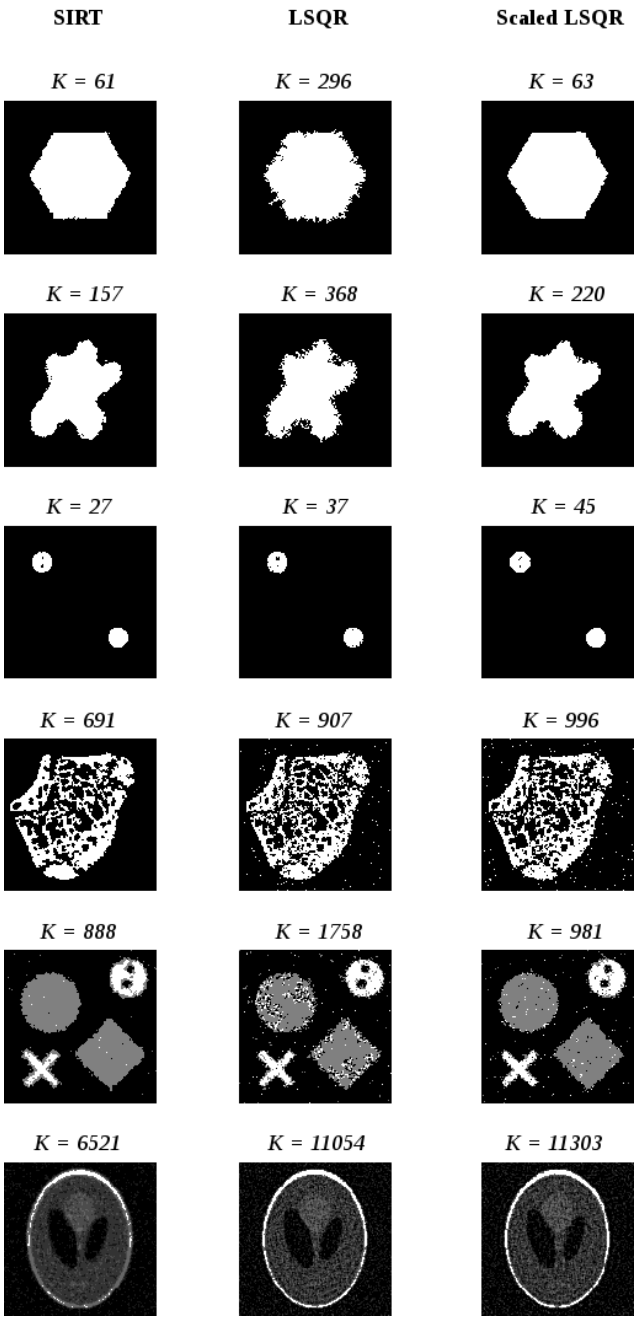


Figure 6.24: *The reconstructions of DART using SIRT (left), LSQR (middle) or scaled LSQR (right). The noise level for all test problems was* 10^3.

Chapter 7

Comparison of the Classic DART with the New DART

This chapter investigates if the new DART is an improvement on the classic DART. To that end first both methods, with 'optimal' parameters, will be applied to two new test problems. Since the new DART is trained using the six test problems from Section 5.3 it might be that it has a bias towards these problems and is actually very poor at dealing with other problems. Hence, two new test problems will be considered to be sure that none of the methods has an advantage.

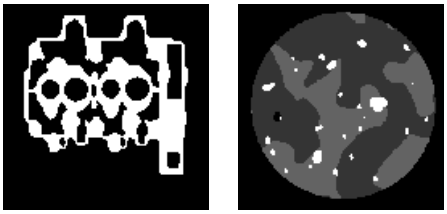
Finally, both DARTs are applied to real world experimental μCT data. These data will suffer from other typical difficulties other than the noise associated with the number of counts. In experiments the projector and detector may not rotate nicely around the object causing *alignment* issues for example. And in practice cone-beams are used (cf. Section 2.1) and these data are *rebinned*, see [15, p. 177–181] and [6, Ch. 7.7.3–7.7.4], to a parallel geometry, this may also introduce pollution in the measurement data.

7.1 Test Problems

Two new test problems will be introduced on which both types of DART will be tested such that there will be no clear bias towards either of the methods. The test problems can be found in Figure 7.1. The first test problem is called 'Cylinders' and is an binary image which is quite intricate, but not as complex as Bone was. It is based on an intersection of the cylinder head of an engine. The second test problem is referred to as 'Speckled' and is an image containing four grey values, it is more complex than Shapes was, but has fewer grey values than Phantom.

Also for these test problems it will be determined how many projection angles are needed for a near perfect reconstruction. This will be done in the same way as it was done in Section 6.4, i.e. the classic DART is used with 50 initial and intermediate ARM iterations and fix probability 1. Note that the measurement data are not polluted with noise for these experiments. Subsequently, a slightly larger number of projection angles will be used in the upcoming experiments, which will include noise.

It was found that a minimum of 25 projection angles was needed to get a near perfect reconstruc-



(a) (b)

Figure 7.1: *New Test problems (a) Cylinders (b) Speckled*

tion for Cylinders, hence for the experiments with noise 35 projection angles will be used. For Speckled 40 projections angles was sufficient and thus 50 angles will be used for the upcoming experiments. In Appendix B.1: Table B.7 – Table B.8 the pixel errors per ARM is listed for a various number of projection angles.

Artificial measurement data will be created for these test problems as usual and the corresponding data will be polluted with various levels of noise. Subsequently, both types of DART will solve the problems to the best of their abilities.

In this work it was found that classic DART performed at its best with fix probability 1, SIRT as ARM and 20 initial and intermediate iterations. However, the fix probability was set to 1 since other ARMs such as CGLS and LSMR did not fare well with the random subset. SIRT on the other hand seemed to perform somewhat better with $p < 1$. Also in [4] it is concluded that a lower fix probability results in a more robust algorithm for noisy problems, i.e. the more noise, the lower p should be. Since the noise level is not known in general beforehand a compromise of $p = 0.6$ would be reasonable. This introduces another problem, and that is that the number of variables per iteration is considerably higher than for $p = 1$, therefore more ARM iterations would perhaps result in more accurate results. From the foregoing discussion it is decided to test classic DART in three configurations: 20 iterations and $p = 1$, 20 iterations and $p = 0.6$, and 50 iterations and $p = 0.6$. Furthermore, the initial guess for classic DART was in this work set to $\mathbf{0}$. In [4], however, the non-segmented grey values of the previous approximation were used for the free pixels as initial guess. For the upcoming experiments classic DART will also use these non-segmented values rather than the zero vector as initial guess. Also the smoothing operation after each DART iteration is still performed for classic DART. The experiment consists of performing 100 classic DART iterations. The last segmented approximation will be the final reconstruction. Note that the RRE is not used as error quantification.

The new DART will perform 50 iterations with the neighbours construct as choice for D. Also for this algorithm 3 configurations will be tested in the form of three different ARMs. SIRT will use 100 initial and intermediate iterations, LSQR will perform 20 iterations, as will scaled LSMR. The absolute RRE sum is used to find a final reconstruction.

In these experiments every simulation corresponding to a certain noise level will use the same measurement data. For example, all the simulations for noise level 10^4, both for the three variants of classic DART and the three variants of new DART, will use the exact same sinogram. With this approach all the solutions can truly be compared since no randomness is left in the right hand side.

In Figure 7.2 the results of the experiments for Cylinders is depicted. Note that classic DART is very good at reconstructing from almost pure data, i.e. noise levels 10^6 and 10^5, for these

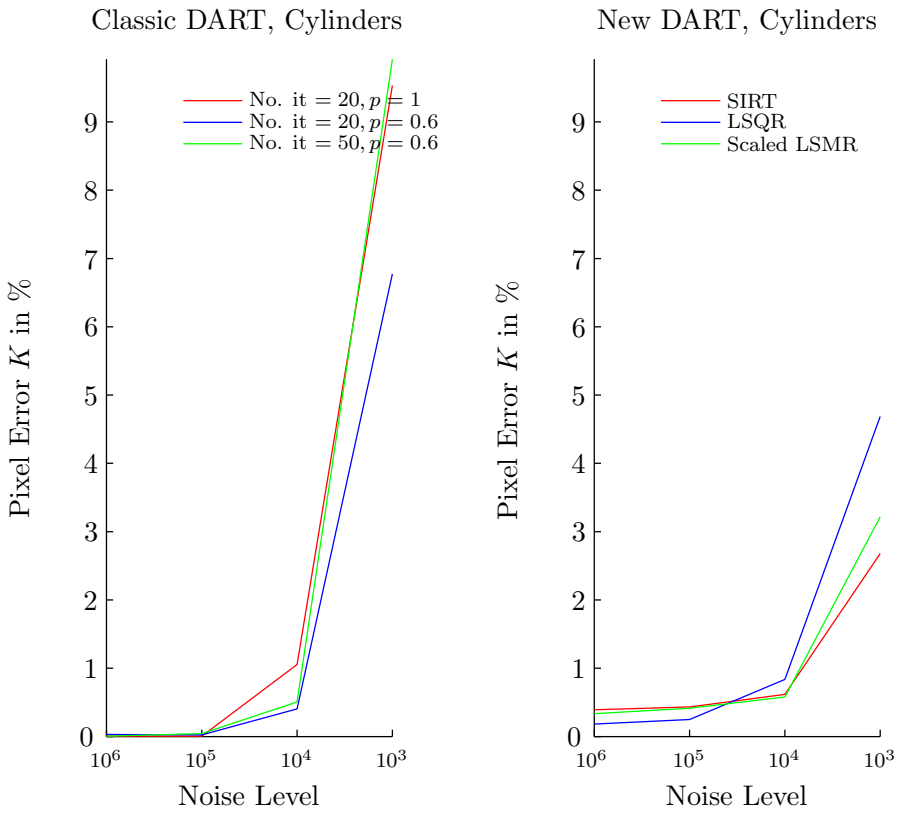


Figure 7.2: *Comparison of the performance of classic DART (left) and new DART (right). The test problem was Cylinders scanned using 35 projection angles.*

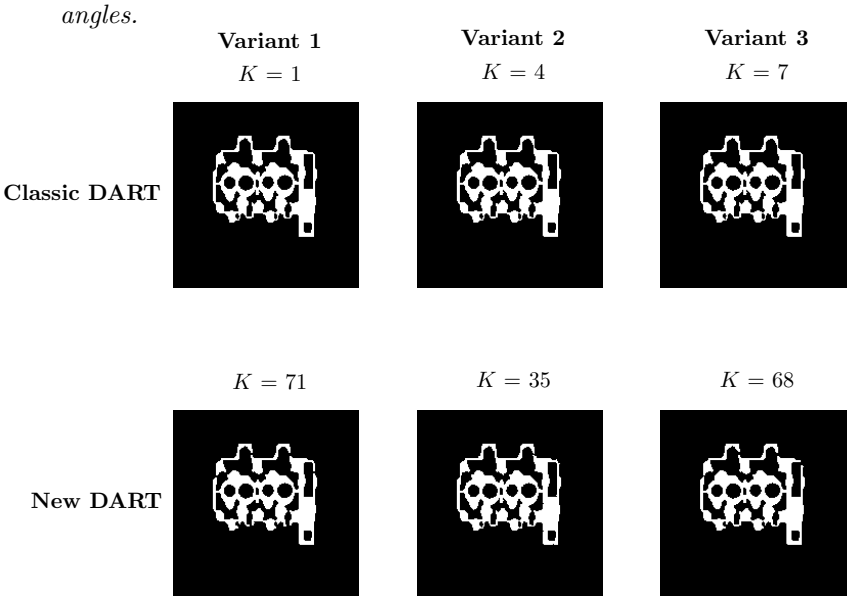


Figure 7.3: *Comparison of the actual reconstructions of classic DART (top) and new DART (bottom) for test problem was Cylinders scanned using 35 projection angles with noise level* 10^5.

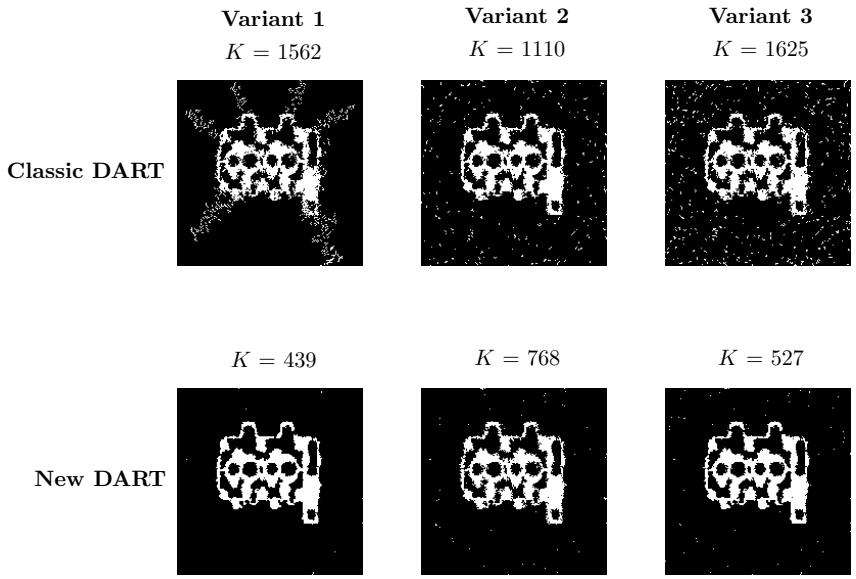


Figure 7.4: *Comparison of the actual reconstructions of classic DART (top) and new DART (bottom) for test problem was Cylinders scanned using 35 projection angles with noise level* 10^3.

simulations all classic variants find a near perfect reconstruction. Although the performance of new DART is worse, the reconstructions themselves are also very good, which can also be seen in Figure 7.3. In this figure the actual reconstructions of all the DART variants can be found for noise level 10^5. The top row corresponds to classic DART while the bottom row shows the reconstructions of new DART. The first column is the reconstruction of the first variant of the corresponding DART, likewise for the second and third column. Above each reconstruction the pixel error is given. Indeed, every reconstruction seems to be near perfect. For the higher noise levels new DART produces considerably better results. The lower fix probability seems to be beneficial in classic DART. Increasing the number of iterations does not seem to have the desired effect, on the contrary, it seems that performing 50 iterations results in poorer results than 20 iterations. This can, however, not be concluded from this single experiment. LSQR seems to perform the poorest for all new DART variants, scaled LSMR performs almost the same as SIRT and is constantly in between of SIRT and LSQR in terms of pixel error. In Figure 7.4 the actual reconstruction of noise level 10^3 are depicted. Note that new DART with SIRT not only has the minimal pixel error, it is also clearly the superior reconstruction of the six. Classic DART seems to suffer greatly from the noise, this can most likely be attributed to the reduction of variables. This causes the effects of the noise only to be distributed among the free pixels. The LSQR reconstruction has an unstable boundary, scaled LSMR has this to a lesser degree. This is most likely the result of the scaling that is taken from SIRT (cf. Section 6.2). Table 7.1 lists the computation times of all the simulations. Clearly, the superior reconstruction comes at a price. Overall the classic DART is faster, this is not surprising since it solves a smaller system than new DART. On top of that new DART solves another system, for the RRE, in each iteration. New DART using LSQR or scaled LSMR is much faster than using SIRT, of course, because every iteration only 20 intermediate ARM iterations are performed. Note that the computation times of classic DART seem to increase as the noise level increases, this is because more noise results in more free pixels and thus a larger reduced system.

Dart Variant \ Noise level	$\mathbf{10^6}$	$\mathbf{10^5}$	$\mathbf{10^4}$	$\mathbf{10^3}$
Classic, SIRT, No. It = 20, p = 1	14.90	22.85	23.51	29.44
Classic, SIRT, No. It = 20, p = 0.6	28.54	48.68	50.70	50.34
Classic, SIRT, No. It = 50, p = 0.6	62.20	107.48	110.25	130.57
New DART, SIRT	486.94	454.65	452.48	519.78
New DART, LSQR	72.35	71.18	70.09	78.83
New DART, Scaled LSMR	92.59	91.09	83.83	94.53

Table 7.1: *The computation times in seconds corresponding to the various simulations. Test problem Cylinders was scanned using 35 projection Angles.*

In Figure 7.5 the same results are depicted for test problem Speckled. Again one can conclude that new DART is superior for the higher noise levels, though classic DART is not far behind for this problem. For the lower noise levels classic DART produces results with lower pixel errors, but also for this test problem the actual reconstruction of new DART are not qualitative any worse. SIRT still is the best performing ARM but scaled LSMR also produces good results. From classic DART it can again be seen that 20 ARM iterations with $p = 0.6$ leads to the superior reconstructions. In Figure 7.6 the actual reconstructions for the highest noise levels are depicted, new DART with SIRT again seems to be the best reconstruction since it does not contain a lot of speckles as all the other reconstructions do. The other reconstructions are comparable to each other. The computation times, listed in Table 7.2, show no real surprises.

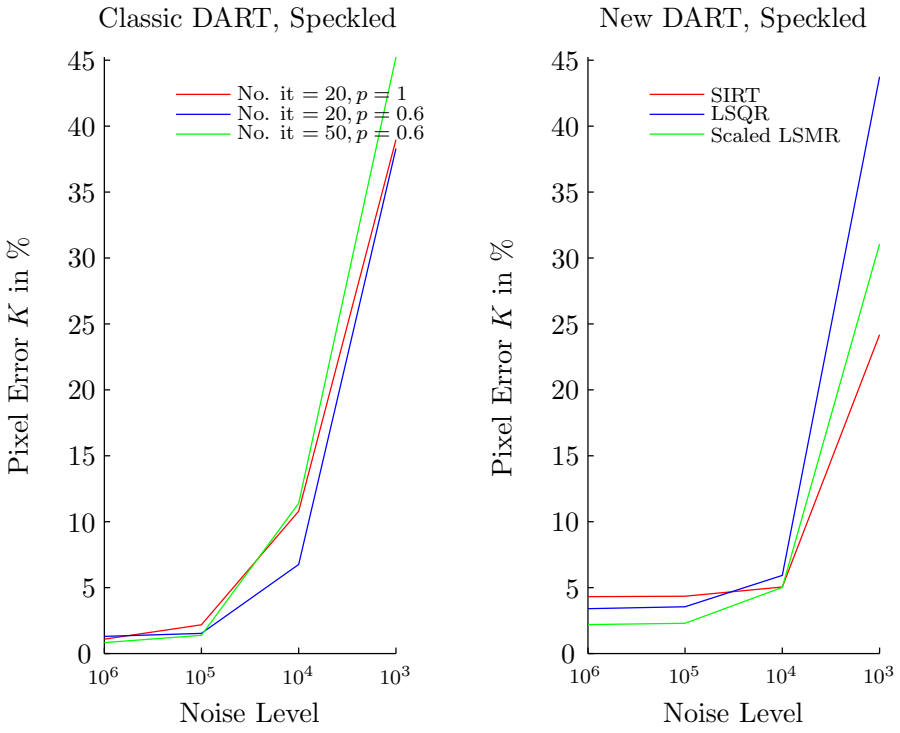


Figure 7.5: *Comparison of the performance of classic DART (left) and new DART (right). The test problem was Speckled scanned using 50 projection angles.*

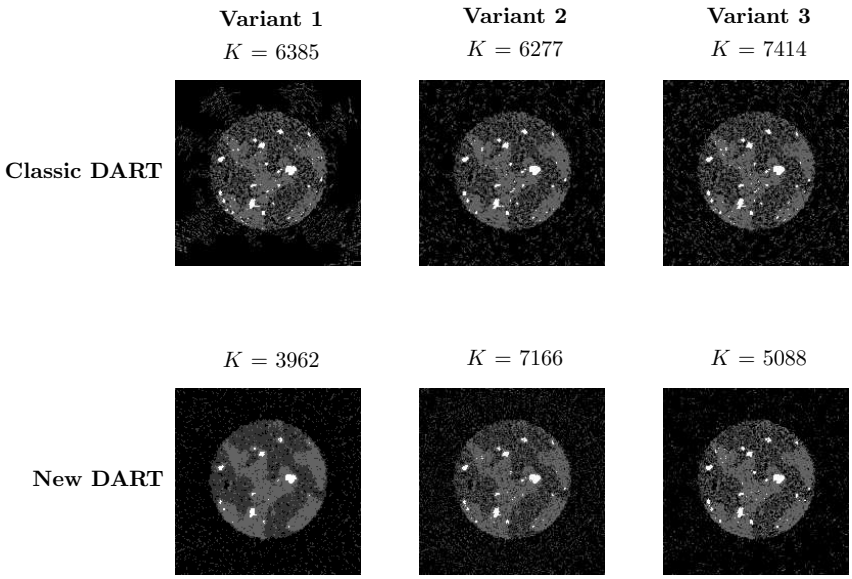


Figure 7.6: *Comparison of the actual reconstructions of classic DART (top) and new DART (bottom) for test problem was Speckled scanned using 50 projection angles.*

Dart Variant \ Noise level	10^6	10^5	10^4	10^3
Classic, SIRT, No. It $= 20, p = 1$	33.55	32.32	40.49	75.81
Classic, SIRT, No. It $= 20, p = 0.6$	69.88	67.97	65.52	58.40
Classic, SIRT, No. It $= 50, p = 0.6$	149.31	138.72	161.55	161.85
New DART, SIRT	602.34	582.38	655.63	663.83
New DART, LSQR	106.13	99.11	114.02	103.72
New DART, Scaled LSMR	117.53	121.91	115.36	122.97

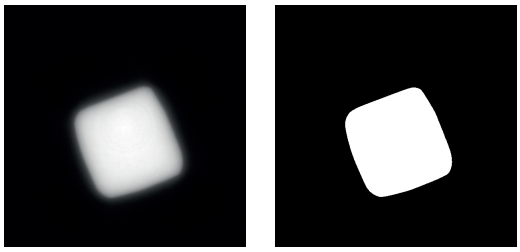
Table 7.2: *The computation times in seconds corresponding to the various simulations. Test problem Speckled was scanned using 50 projection Angles.*

7.2 Experimental Data

In this section the both types of DART will be applied to experimental μCT data. A diamond was scanned at 70 kVp in a Scanco μCT 40 X-ray scanner with a circular cone beam geometry. After the scan, the data was rebinned to a parallel beam geometry, yielding a 1024×500 sized sinogram per slice, for a total of 300 slices, with projection angles distributed equally between 0 and π. See [4] for information regarding this dataset.

For the experiments slice 156 was selected, which is located approximately in the middle of the object. This thus yields a sinogram of 1024×500, i.e. the detector was 1024 pixels wide and there were 500 projection angles. Since a large part of the sinogram contained no information a subset of 466 pixels from the middle of the detector are taken. The functionalities in the ASTRA-toolbox are applied to create a reconstruction of this sinogram. The used reconstruction algorithm was SIRT, 50 iterations were performed. This yields the reconstruction shown in Figure 7.7(a). Note that, although one knows that the reconstruction is a diamond (of one single density with no impurities), the reconstruction contains grey values in the whole spectrum. The segmentation of this reconstruction is shown in Figure 7.7(b).

In the application field of DART one is usually presented with very little data, few projection



(a) (b)

Figure 7.7: *A reconstruction of the diamond (a) Non-segmented (b) Segmented*

angles and few detector pixels. Luckily, one can use knowledge about the object to be able to make accurate reconstructions. To simulate this scenario only a small part of the original experimental data will be used. Only the data of every fourth detector pixel will be used and a number of equally spaced angles will be selected from the 500. Subsequently, classic DART and new DART with all the variants as in Section 7.1 are used to attempt to reconstruct the diamond from that data, with the prior knowledge that only 2 grey values should be present in the reconstruction.

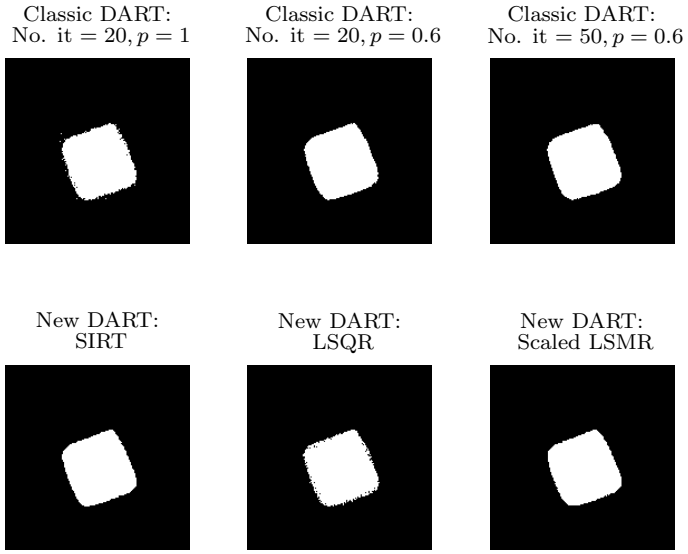


Figure 7.8: *The reconstruction of the diamond when only 10 projection angles are used.*

In Figure 7.8 the reconstructions of the various variants of DART are shown for 10 projection angles. Variant 1 of classic DART and new DART with LSQR show some unstable boundaries, but on the whole all the reconstructions seem accurate. Decreasing the number of projection angles to 5 yields the results from Figure 7.9. Clearly classic DART is having some trouble with the reconstruction and so does new DART with LSQR as ARM. New DART with SIRT and new DART with scaled LSMR produce reasonable results, though less accurate than the experiment with 10 projection angles. Also some speckles around the reconstructed objects begin to appear. Decreasing the number of projection angles further to only 3 truly shows big differences in the reconstructions for the various DART variants as can be seen in Figure 7.10. Classic DART variant 2 and 3 produce some very poor and peculiar reconstructions. The

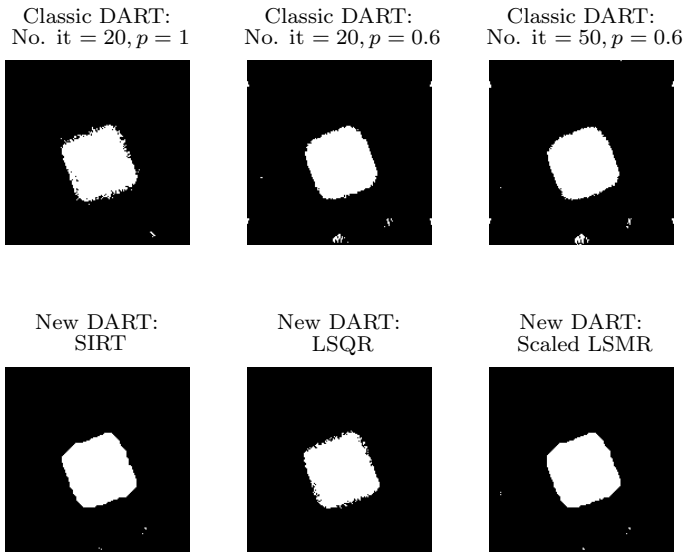


Figure 7.9: *The reconstruction of the diamond when only 5 projection angles are used.*

symmetry in the reconstructions is a result of the low number of projection angles; the lines of symmetry coincide precisely with the projection angles. Again new DART with LSQR does not perform well, but at least better than the aforementioned variants of classic DART. Classic DART with SIRT performing 20 iterations and fix probability 1 performs reasonably well but the boundary is not very distinct and some black speckles are contained inside the diamond. New DART with SIRT and scaled LSMR perform equally well and clearly produce the best reconstructions. Note the white spots on the left and right of these reconstructions, this is most likely the same phenomenon as for classic DART variant 2 and 3, but to a far lesser degree.

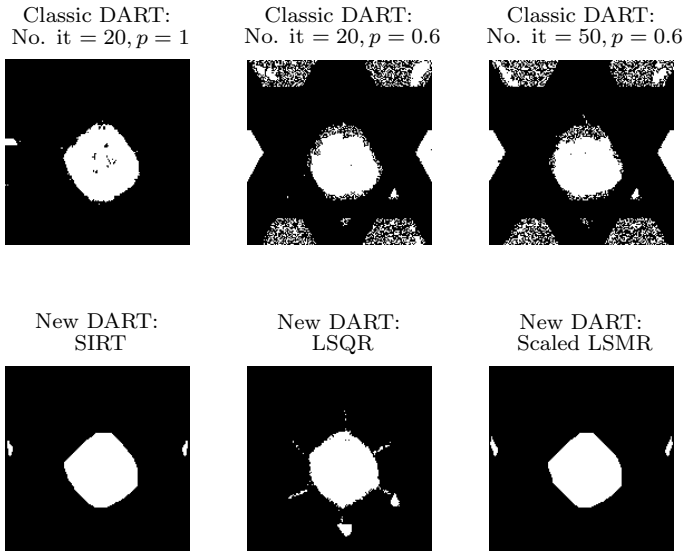


Figure 7.10: *The reconstruction of the diamond when only 3 projection angles are used.*

Chapter 8

Conclusions and Future Research

This chapter concludes this work. It will begin with a review of what experiments have been performed. Subsequently, conclusions are drawn and the research objective is evaluated. This chapter will end with some recommendations for possible future research.

8.1 Conclusions

The goal of this work was to find out if the DART algorithm could be improved. To that end several research questions were posed that should help achieve this goal. The three question read:

- Which algorithm should be used as ARM in DART and does it matter?
- Can better results be obtained by introducing regularization directly onto the set of free pixels U?
- Are there alternatives for the fixed-free pixels construct?

Chapter 6 treated several numerical experiments which were carried out to find an answer for the questions above. First of all the classic DART algorithm was studied in more detail and it was determined that the fix probability p was better to be eliminated immediately. Also two types of initial guesses for the intermediate ARM iterations were studied. It was found that better results are obtained if $\mathbf{0}$ was used as initial guess rather than the non-segmented grey values of the free pixels from the previous approximation.

Subsequently, the first research question was tackled. It turned out that SIRT and SART performed particularly well with respect to the other ARMs. ART and CGNE were immediately labelled as unreliable for they produced very bad results on noisy data. Keeping in mind that the linear system that would be solved would change considerably, it was decided to also include LSQR, and sometimes LSMR, in subsequent experiments to have different natured ARMs. A good default number of intermediate ARM iterations was found out to be 20.

Next it was investigated if regularization introduced on the free pixels leads to better results. It turned out that the effect of regularization is inconclusive. On the one hand it gives better

approximations in the sense of lower pixel error, on the other the actual reconstructions did not look accurate.

Thereafter, an alternative for the fixed-free pixels construct was introduced in the form of a new linear system. This system is able to regularize the problem in such a way that the user can steer grey values of some pixel i towards a certain value v_i. The magnitude of this steering is controlled by the value of the diagonal element d_i of the matrix D. Several choices of d_i were investigated and it was found that a value dependent on the number of neighbours with a different segmented grey value gave the best results. SIRT seemed to perform very bad with this new DART formulation, one of the possible reasons could be slow convergence; the algorithm simply needed more iterations to converge. And indeed, 100 intermediate iterations gave much better results than 20.

The stopping criterion used for DART in this work was just that it needed to perform a fixed number of iterations. Unfortunately, both the classic and new DART do not produce approximations with a monotonically decreasing error, moreover, because of the strong ill-posedness of the problem, it was impossible to compute the error. The Reconstructed Residual Error was introduced to estimate the reconstruction error. Although this did not enable one to have a different stopping criterion, the minimal absolute RRE sum can only be found in retrospect, it gave a reasonable indication of which DART iteration gave an accurate reconstruction. An additional benefit of the RRE is that it can be used to correct wrong segmentation values. Using the RRE to quantify the quality of a reconstruction corresponds to an educated guess and is thus better than choosing the last approximation as final reconstruction, which is actually just a shot in the dark. It was also found that 50 DART iterations were more than enough for most scenario's to find a good reconstruction.

The chapter concluded with a detailed inspection of SIRT. This algorithm performed the best throughout, mainly on data with high noise levels. One reason for this might be that SIRT computes a solution in a different norm than CGLS and LSQR do. This inherent scaling of SIRT was applied to CGLS, LSQR and LSMR to obtain scaled algorithms that supposedly performed better in high noise levels. It was found that these algorithms indeed produced somewhat better results that mimicked SIRT. The main advantage is that these scaled algorithms only need 20 iterations to arrive at those results while SIRT needs 100 iterations.

Finally, in Chapter 7 the classic DART was compared to the new DART. It seems that the new DART actually is an improvement over the classic DART for higher noise levels. Classic DART still performs somewhat better on data with little noise, but the reconstructions of new DART are by no means poor. The major drawback of new DART, mainly with SIRT as ARM, is the increased computation time. However, if scaled LSMR is used the computation times can be decreased dramatically at the expense of some accuracy. New DART also seemed to be able to work on experimental μCT data and produced superior reconstructions than classic DART when extremely few projection angles were used.

All in all it can be concluded that the DART algorithm can be improved in the sense of accuracy on noisy data by means of the new formulation. The classic DART most likely suffers from high noise levels because the reduced system only contains a fraction of the original variables. In the new DART all variables are available and only weak constraints are imposed. Moreover, the heuristic approaches are not part of the new algorithm anymore. SIRT is the best performing ARM, but if the computation time is a factor than scaled LSMR will provide more than reasonable reconstructions in about 1/4 the time SIRT needs.

8.2 Future Research

Although the new DART formulation performs very well there is always room for improvement. In this section some recommendations will be made for possible future research that could help improve DART even more.

This work investigated some choices of the d_i, the diagonal elements of the regularization matrix D in the new DART formulation. The magnitude of these values gave a measure for how much a certain pixel i was steered towards the grey value v_i. It was found that d_i dependent on the number of neighbouring pixels with a different segmented grey value performed quite nicely. A drawback of this construct is that it is more or less dependent on the number of segmentation values. If more densities are allowed in the object, more pixels will get a small d_i value. There may be better choices that lead to superior reconstructions. It might be possible to formulate an optimal choice for the d_i. One could also make the ARMs 'favor' solutions that contain the segmented grey values $\{\rho_1, \rho_2, \ldots, \rho_l\}$. In CGLS for example one might update the values d_i during the ARM iterations. This could even render the segmentation step redundant.

All the ARMs solve the tomography problem as if the problem is not discrete, consequently one can arrive at (non-segmented) solutions with negative grey values, this, of course, is not possible in practice and thus it might be sensible to adapt the ARMs such that negative values cannot be part of a solution. One then has to keep in mind that solutions with components in the null space $\mathcal{N}(W)$ might lead to poor results, and thus simply setting values to 0 as a grey value is negative is most likely not a good choice.

The reconstructions seemed to suffer from speckles in- and outside the object, mainly the problems containing more than two grey values. Better solutions might be obtained if the ARM used produces solutions with more homogeneous regions (although in the case of Bone this might be counter productive). Total variation minimization might do this trick. But it may also be obtained by nice regularization choices.

Segmentation was done by simply rounding of the continuous grey values to the nearest segmentation value. There are more elaborate segmentation schemes that might prove beneficial for the results of the algorithm.

The Reconstructed Residual Error was used to make an error quantification. This was done by summing up the absolute value of the RRE and the reconstruction with the minimal RRE was considered to be the best. A lot of information is disregarded by this approach. The RRE, thus the actual reconstruction of the segmentation error, might contain information that can be used to improve the subsequent reconstruction.

Noise was added to sinograms using a functionality from the ASTRA-toolbox. Although this does indeed provide one with a noisy sinogram, it has a couple of peculiarities. The level of noise needs to be given in by the number of counts, higher values yield less noise. This does not actually pose a real problem, but it is counter intuitive. What is a problem is that the level of noise is also dependent on the maximal value in the sinogram. Then noise levels become problem dependent and thus it becomes harder to compare problems. Analysis of experiments might benefit from a more proper simulation of the noise.

Lastly, it was found that the new DART had one major drawback: the computation time. With proper implementation on the GPU the computation time might drop considerably. Especially SIRT, which can be efficiently formulated as a matrix-vector product, is ideal for parallelization.

Bibliography

[1] K. J. Batenburg. A network flow algorithm for reconstructing binary images from continuous x-rays. *Journal of Mathematical Imaging and Vision*, 30(3):231–248, 2008.

[2] K. J. Batenburg, W. R. Fortes, R. Tijdeman, and L. Hajdu. Bounds On The Difference Between Reconstructions In Binary Tomography. In *Lecture Notes in Computer Science*, pages 369 – 380. Springer, 2011.

[3] K. J. Batenburg and J. Sijbers. *DART: A Fast Heuristic Algebraic Reconstruction Algorithm for Discrete Tomography*, pages IV–133–IV–136. IEEE, 2007.

[4] K. J. Batenburg and J. Sijbers. DART: A practical reconstruction algorithm for discrete tomography. *IEEE Transactions on Image Processing*, 20(9):2542–2553, 2011.

[5] A. Björk, T. Elfving, and Z. Strakos. Stability of conjugate gradient and lanczos methods for linear least squares problems. *SIAM Journal on Matrix Analysis and Applications*, 19(3):720–736, 1998.

[6] T. M. Buzug. *Computed Tomography: From Photon Statistics to Modern Cone-Beam CT*. Springer, Berlin, 2008.

[7] D. C. L. Fong and Saunders M. A. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011.

[8] D. Gale. A theorem on flows in networks. *Pacific Journal of Mathematics*, 7(2):1073–1082, 1957.

[9] R.J. Gardner, P. Gritzmann, and D. Prangenberg. On the computational complexity of reconstructing lattice sets from their x-rays. *Discrete Mathematics*, 202(1–3):45–71, 1999.

[10] P. Gritzmann, S. de Vries, and M. Wiegelmann. Approximating binary images from discrete x-rays. *SIAM Journal on Optimization*, 11(2):522–546, 2000.

[11] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, 1998.

[12] P. C. Hansen and M. Saxild-Hansen. AIR tools a MATLAB package of algebraic iterative reconstruction methods. *Journal of Computational and Applied Mathematics*, 236(8):2167–2178, 2012.

[13] K. M. Hanson. Noise and contrast discrimination in computed tomography. In T.H. Newton and D.G. Potts, editors, *Radiology of the Skull and Brain, Vol. 5: Technical aspects of*

computed tomography, volume 5 of *Radiology of the Skull and Brain*, pages 3941–3955. Mosby, Toronto, 1981.

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

[15] G. T. Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Springer, London, 2009.

[16] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. IEEE Press, New York, 1987.

[17] A. Kuba and G. T. Herman. Discrete tomography: A historical overview. In A. Kuba and G. T. Herman, editors, *Discrete Tomography: Foundations, Algorithms and Applications*, pages 3–34. Birkhäuser, Boston, 1999.

[18] F. Leroux, M. Gysemans, S. Bals, K.J. Batenburg, J. Snauwaert, T. Verbiest, C. Van Haesendonck, and G. Van Tendeloo. Three-dimensional characterization of helical silver nanochains mediated by protein assemblies. *Advanced Materials*, 22(19):2193–2197, 2010.

[19] H. Y. Liao and G. T. Herman. A coordinate ascent approach to tomographic reconstruction of label images from a few projection. *Discrete Applied Mathematics*, 151(1-3):184–197, 2005.

[20] L. Molina, H. Tan, E. Biermans, K. J. Batenburg, J. Verbeeck, S. Bals, and G. Van Tendeloo. Barrier efficiency of sponge-like la 2 zr 2 o 7 buffer layers for ybco-coated conductors. *Superconductor Science and Technology*, 24(6):065019, 2011.

[21] C. C. Paige and Saunders M. A. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[22] C. C. Paige and Saunders M. A. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transaction on Mathematical Software*, 8(1):43–71, 1982.

[23] Vision Lab: All Scale Tomographic Reconstruction Antwerp research group. ASTRA-toolbox. `http://www.visielab.ua.ac.be/software`, 2012.

[24] T. Roelandts, K. J. Batenburg, and J. Sijbers. Visualizing the segmentation error of a tomogram using the residual projection error. In *The Second International Conference on Image Formation in X-Ray Computed Tomography (CT Meeting)*, pages 293–296, 2012.

[25] H. J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canadian Journal of Mathematics*, 9:371–377, 1957.

[26] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, second edition, 2003.

[27] T. Schüle, C. Schnörr, S. Weber, and J. Hornegger. Discrete tomography by convex-concave regularization and d.c. programming. *Discrete Applied Mathematics*, 151(1-3):229–243, 2005.

[28] H. Stark, J. W. Woods, I. Paul, and R. Hingorani. Direct rourier reconstruction in computer tomography. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-29(2):237–245, 1981.

[29] F. Tabak. Robust Algorithms for Discrete Tomography, May 2012. Literature study for Master of Science Thesis.

[30] A. van der Sluis and H. A. van der Vorst. Numerical solution of large, sparse linear algebraic systems arising from tomographic problems. In G. Nolet, editor, *Seismic Tomography With Applications in Global Seismology and Exploration Geophysics*, pages 49–83. D. Reidel Publishing Company, Dordrecht, 1987.

[31] S. Webb. *From the Watching of Shadows: The Origins of Radiological Tomography.* Taylor & Francis, Bristol, 1990.

Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Appendix

Robust Algorithms for Discrete Tomography

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfilment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

FRANK TABAK

Delft, the Netherlands
October 2012

Appendix A

Figures

A.1 Figures: Best ARM

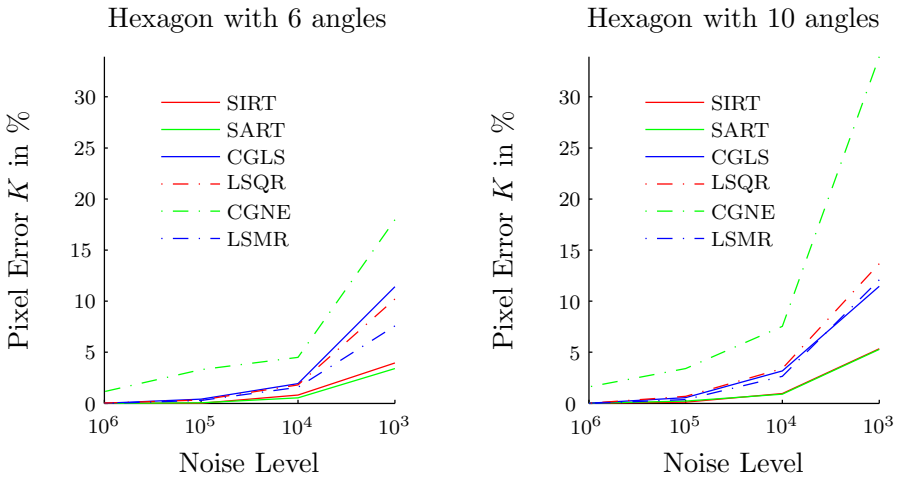


Figure A.1: *The pixel errors corresponding to the noise levels for Hexagon. Left: 6 angles, right: 10 angles.*

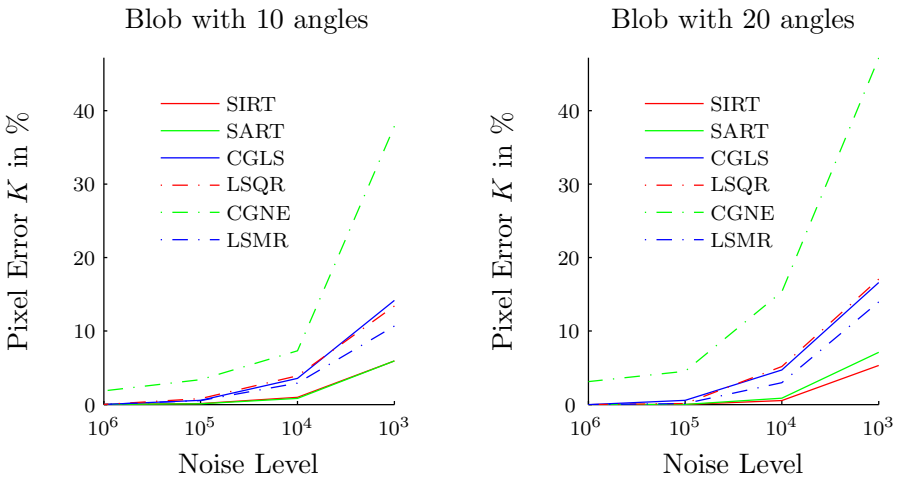


Figure A.2: *The pixel errors corresponding to the noise levels for Blob. Left: 10 angles, right: 20 angles.*

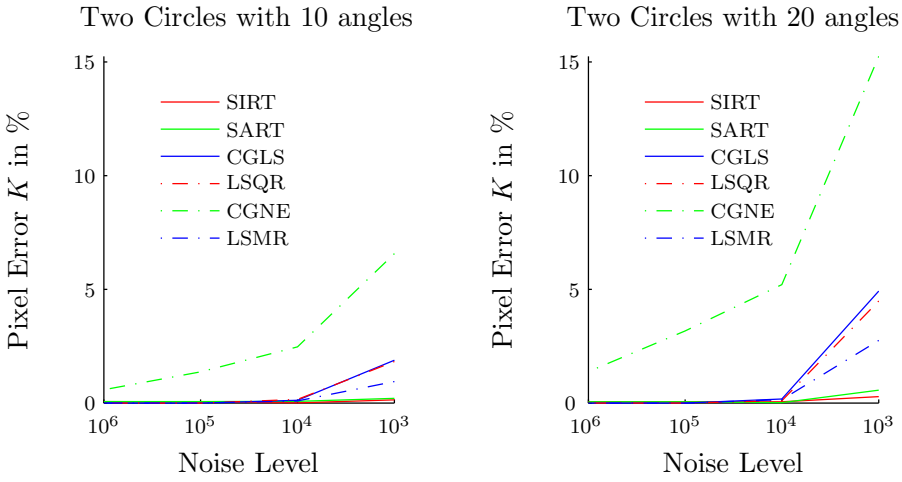


Figure A.3: *The pixel errors corresponding to the noise levels for Two Circles. Left: 10 angles, right: 20 angles.*

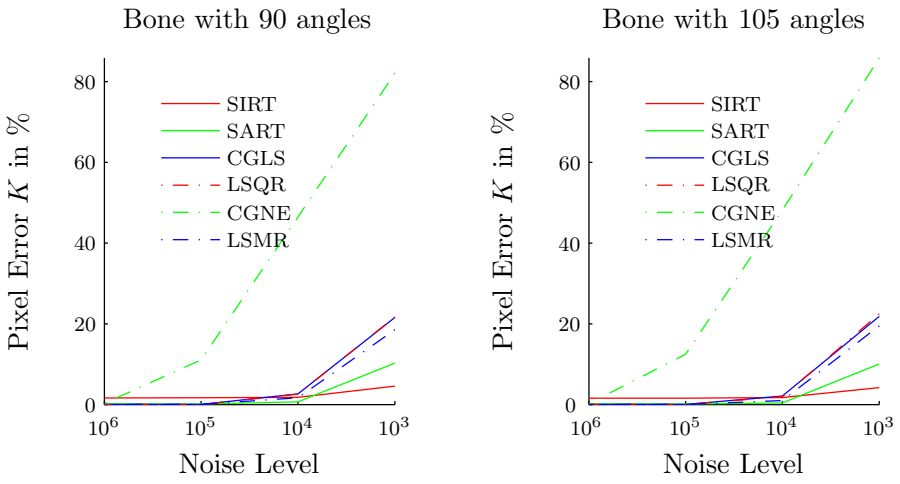


Figure A.4: *The pixel errors corresponding to the noise levels for Bone. Left: 90 angles, right: 105 angles.*

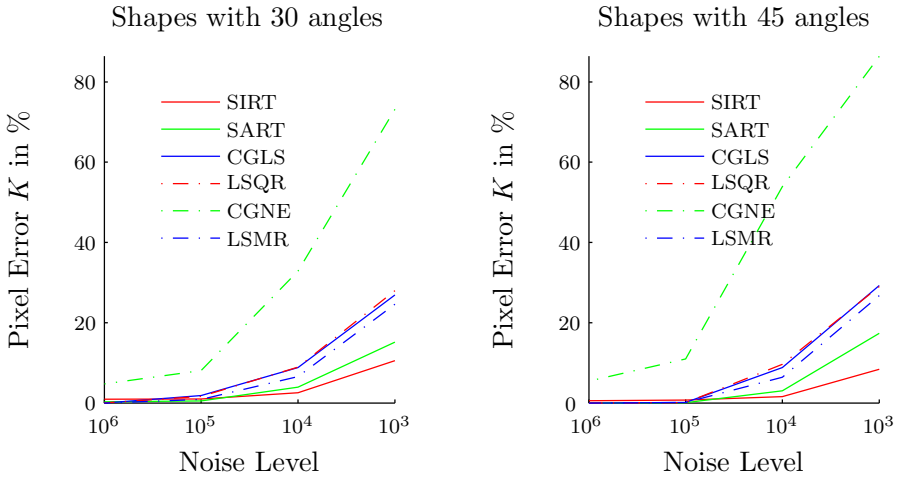


Figure A.5: *The pixel errors corresponding to the noise levels for Shapes. Left: 30 angles, right: 45 angles.*

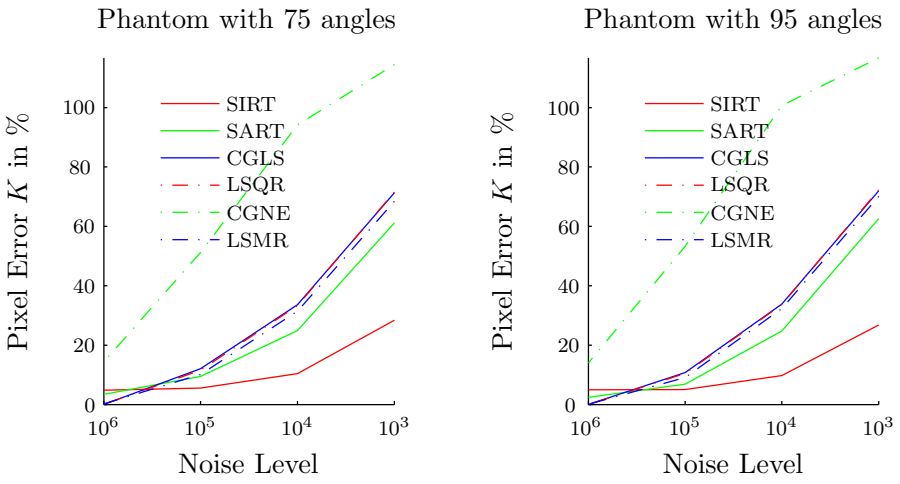


Figure A.6: *The pixel errors corresponding to the noise levels for Phantom. Left: 75 angles, right 95 angles.*

A.2 Figures: Number of ARM iterations

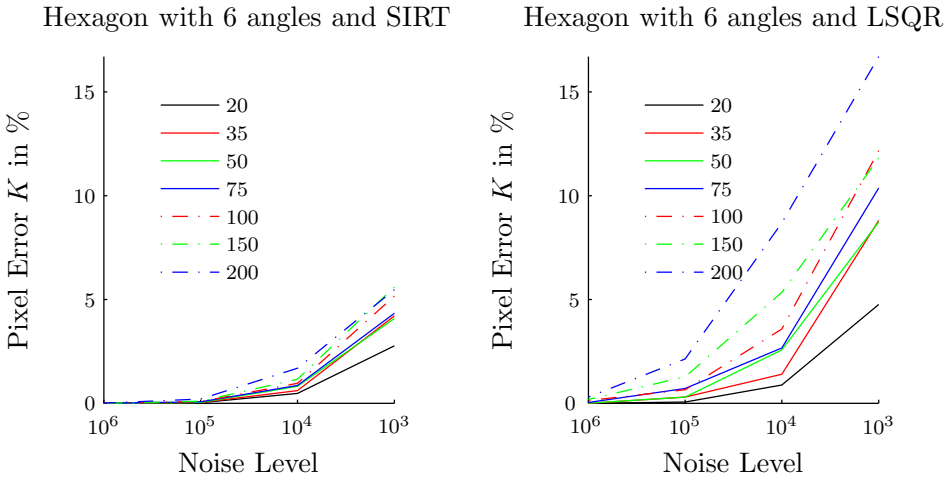


Figure A.7: *The pixel errors corresponding to the noise levels for Hexagon. Left: SIRT, right: LSQR.*

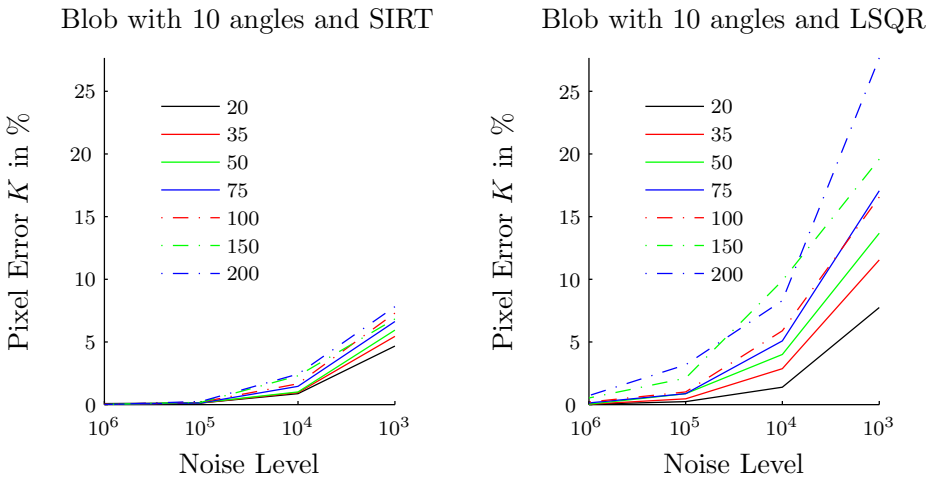


Figure A.8: *The pixel errors corresponding to the noise levels for Blob. Left: SIRT, right: LSQR.*

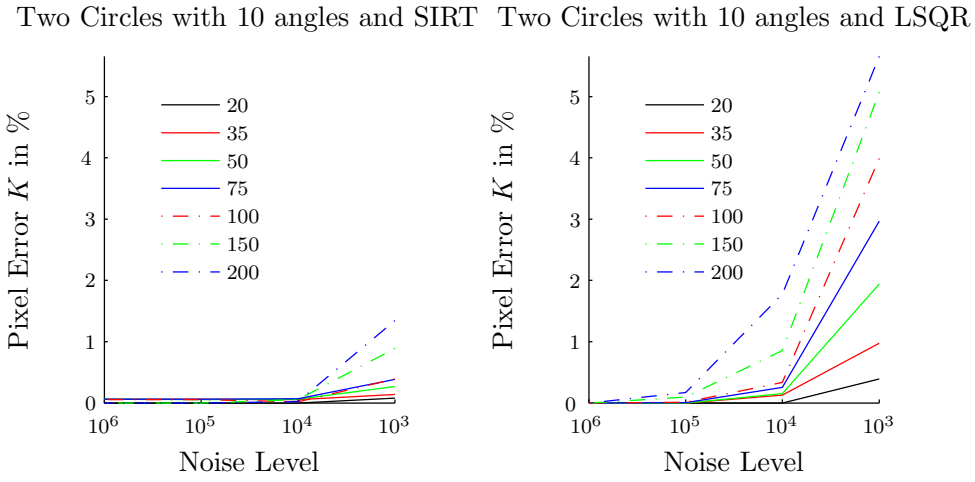


Figure A.9: *The pixel errors corresponding to the noise levels for Two Circles. Left: SIRT, right: LSQR.*

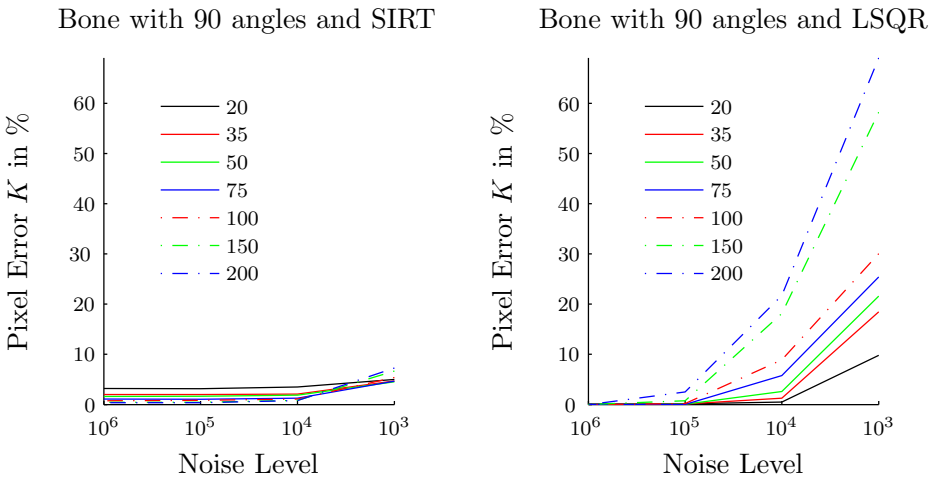


Figure A.10: *The pixel errors corresponding to the noise levels for Bone. Left: SIRT, right: LSQR.*

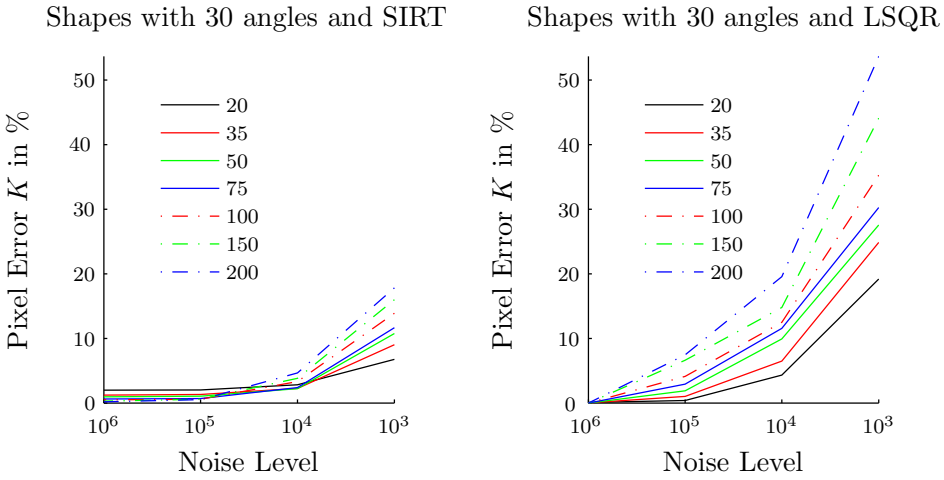


Figure A.11: *The pixel errors corresponding to the noise levels for Shapes. Left: SIRT, right: LSQR.*

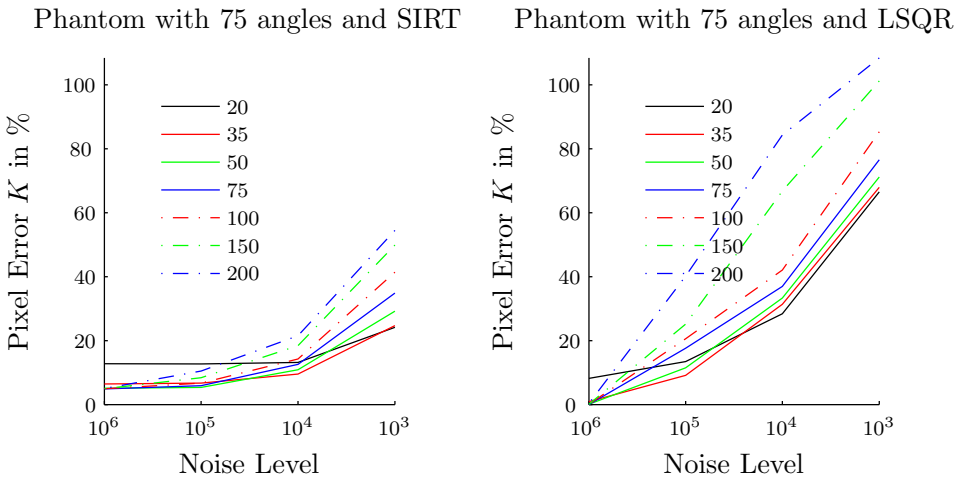


Figure A.12: *The pixel errors corresponding to the noise levels for Phantom. Left: SIRT, right: LSQR.*

A.3 Figures: Regularization on the Free Pixels

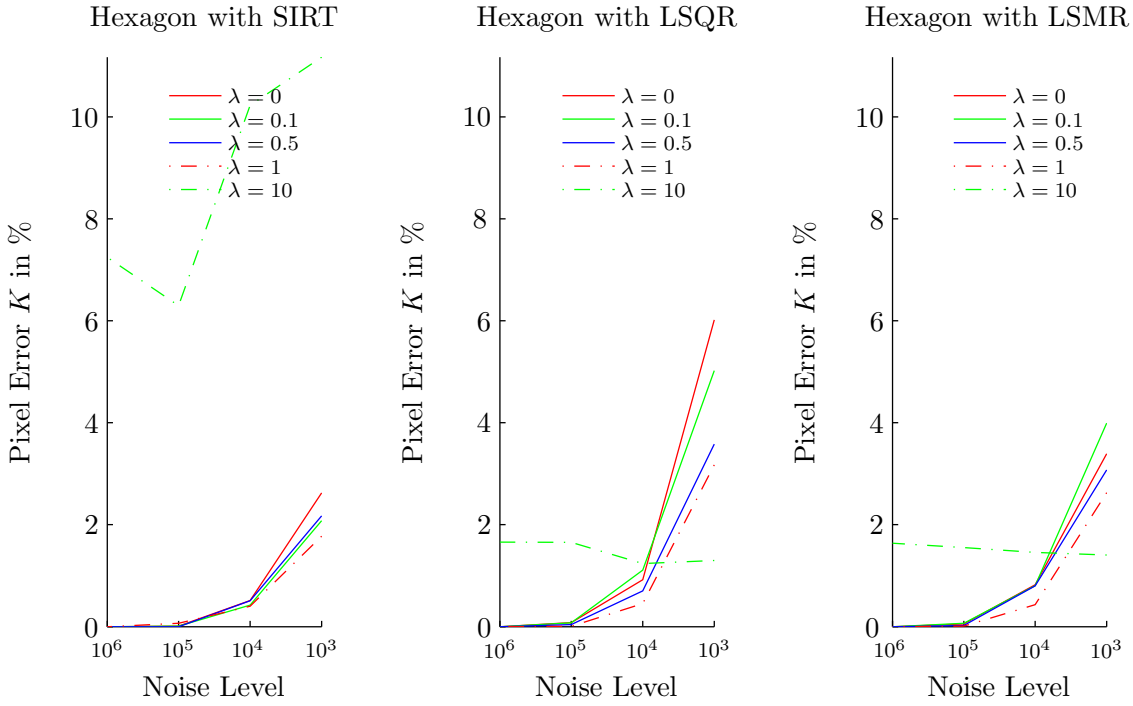


Figure A.13: *The performance of DART with Tikhonov regularization for Hexagon with 6 angles.*

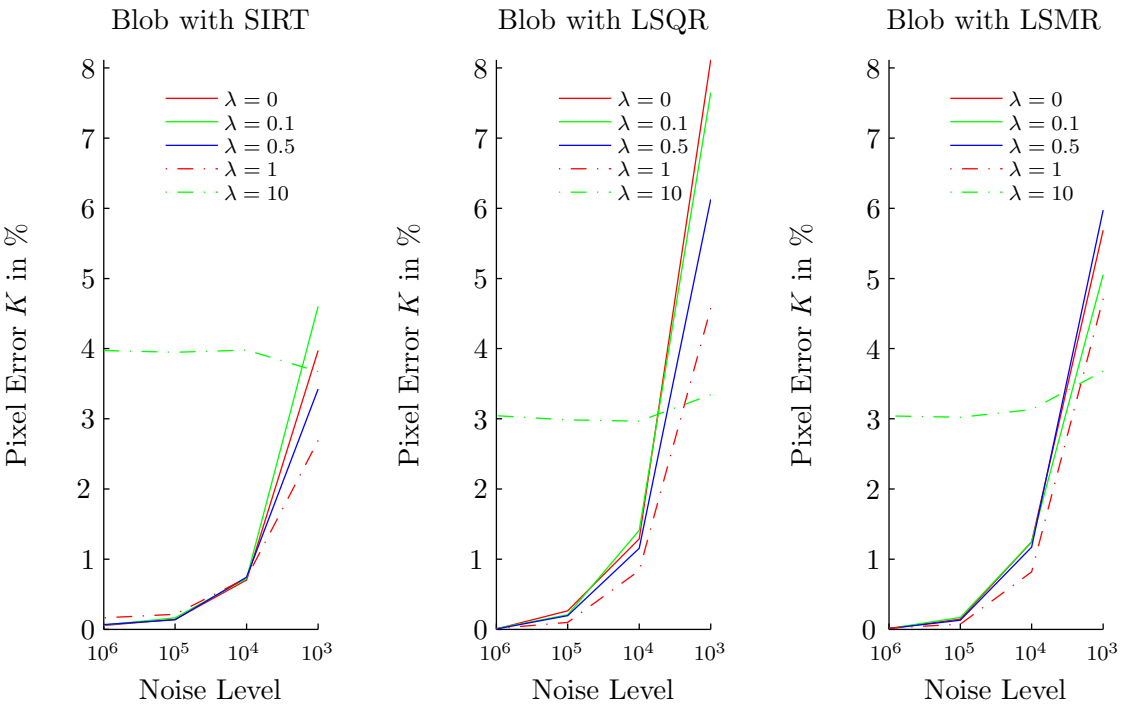


Figure A.14: *The performance of DART with Tikhonov regularization for Blob with 10 angles.*

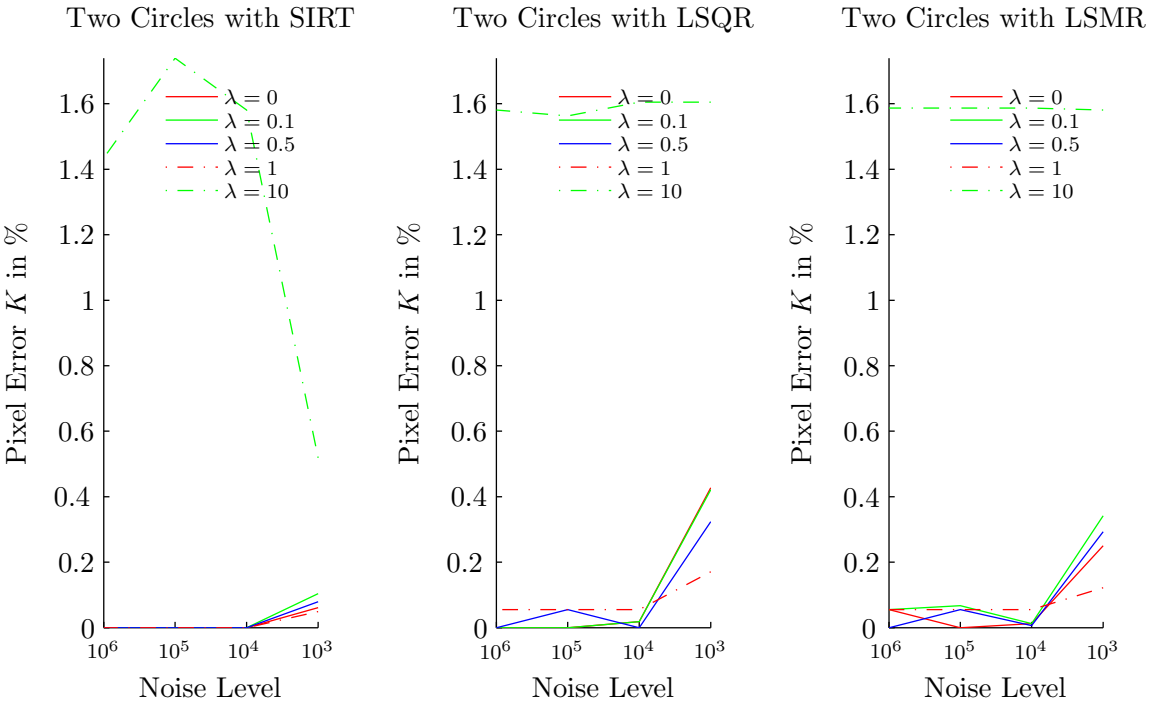


Figure A.15: *The performance of DART with Tikhonov regularization for Two Circles with 10 angles.*

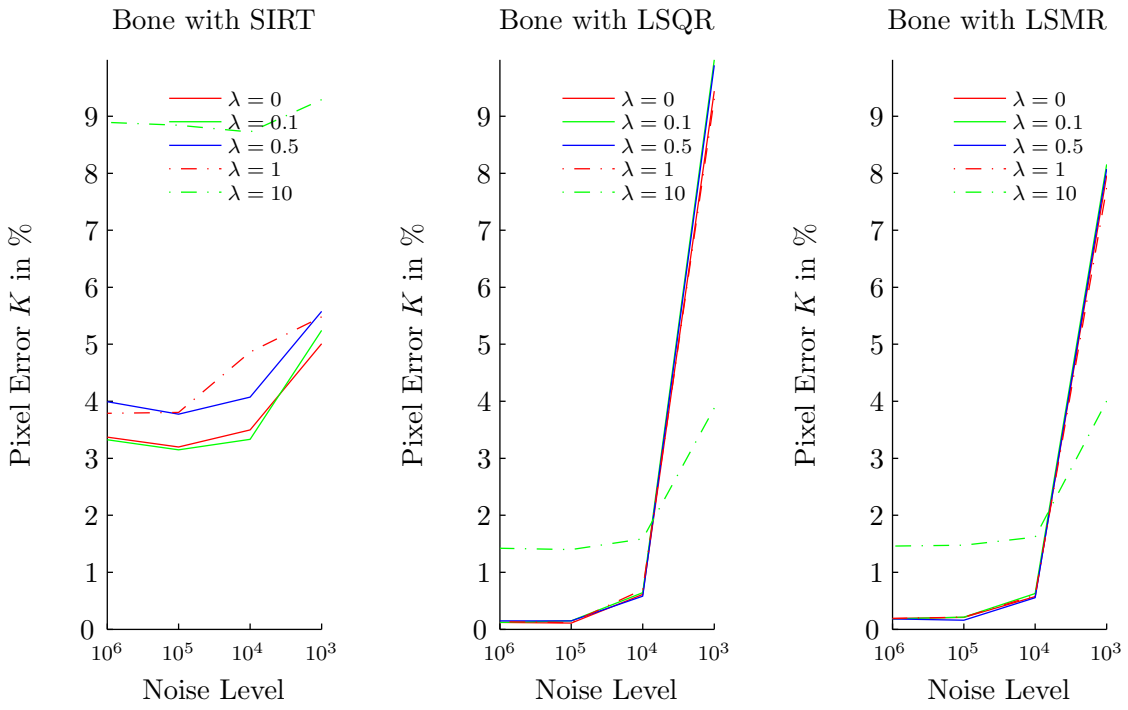


Figure A.16: *The performance of DART with Tikhonov regularization for Bone with 90 angles.*

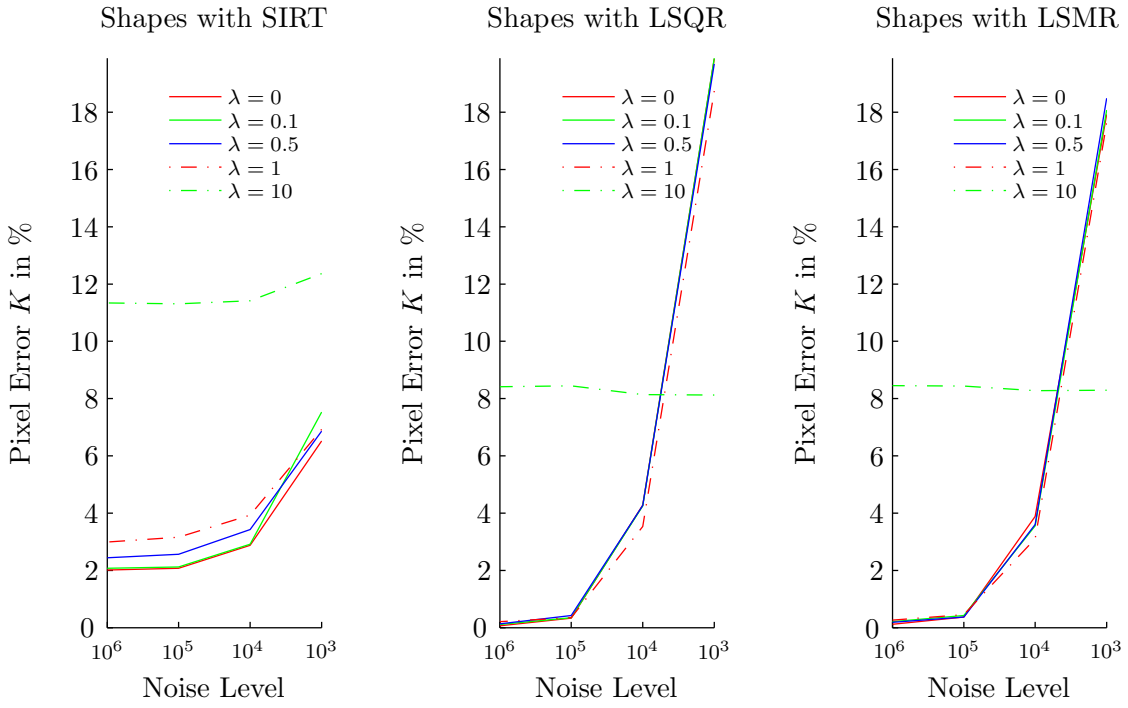


Figure A.17: *The performance of DART with Tikhonov regularization for Shapes with 30 angles.*

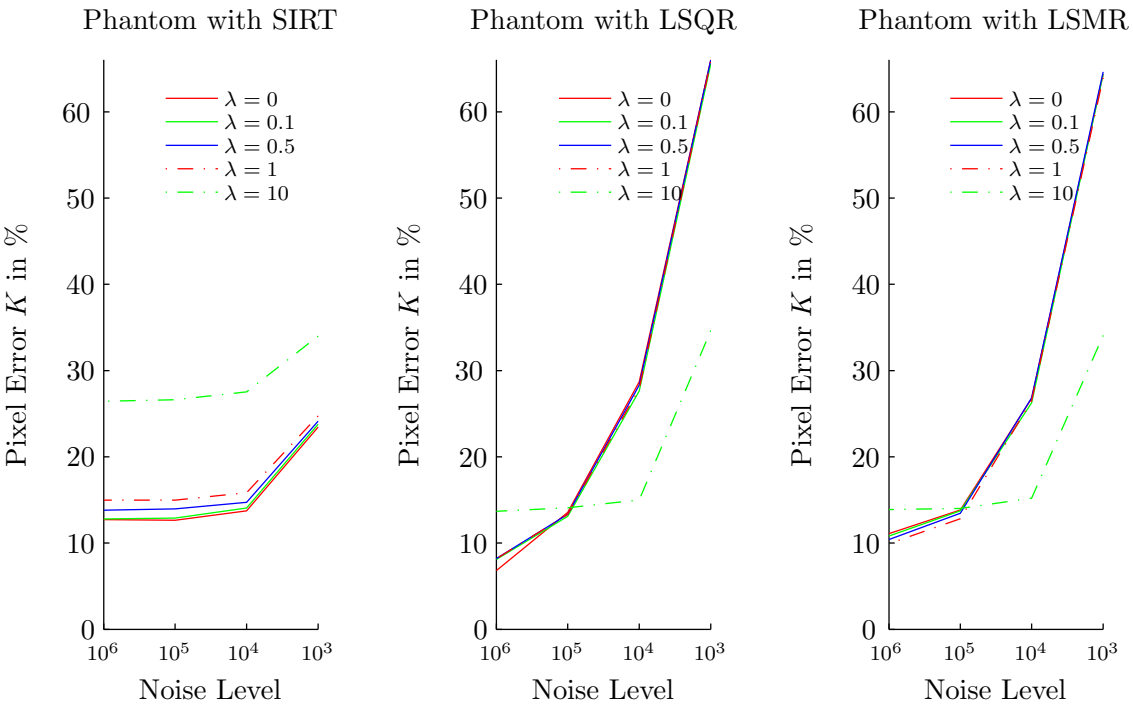


Figure A.18: *The performance of DART with Tikhonov regularization for Phantom with 75 angles.*

A.4 Figures: Alternative to the Fixed-Free Pixels Construct

A.4.1 Figures: Mimicking the Classical DART

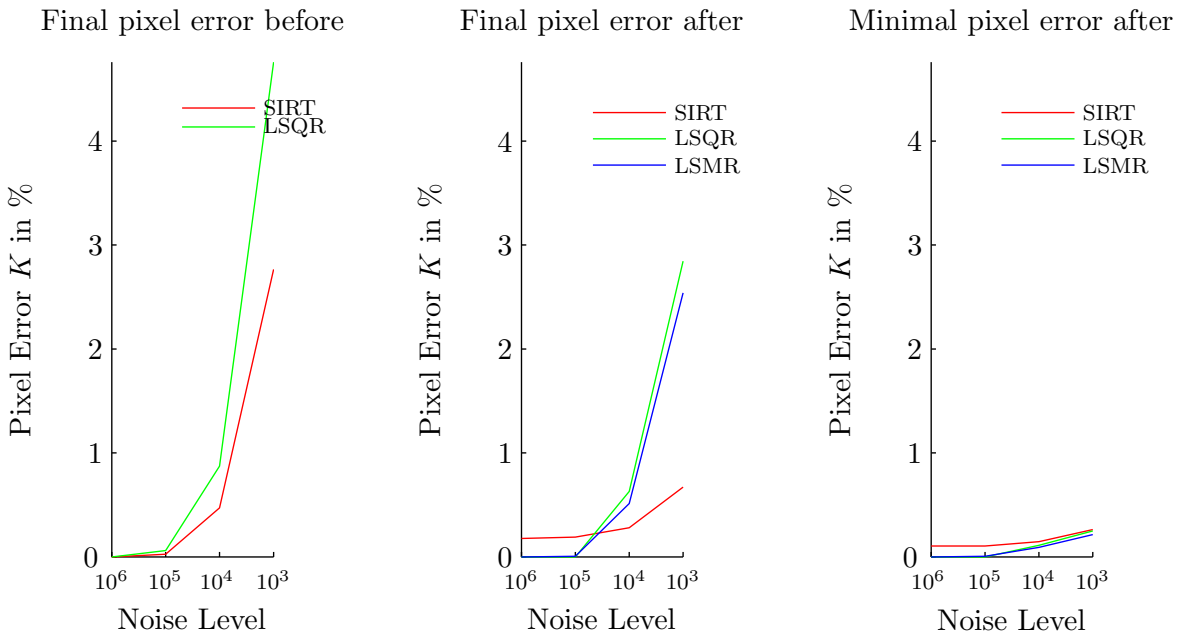


Figure A.19: *The performance of the new DART that aims to mimic the fixed-free pixels construct. Test problem Hexagon scanned using 6 projection angles.*

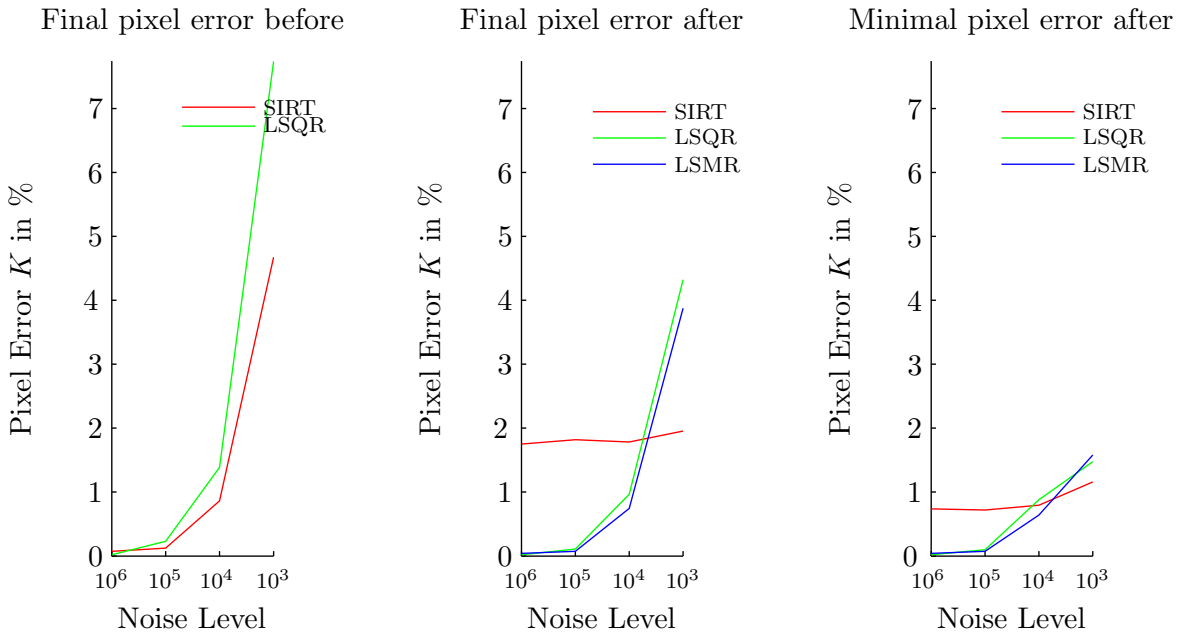


Figure A.20: *The performance of the after that aims to mimic the fixed-free pixels construct. Test problem Blob scanned using 10 projection angles.*

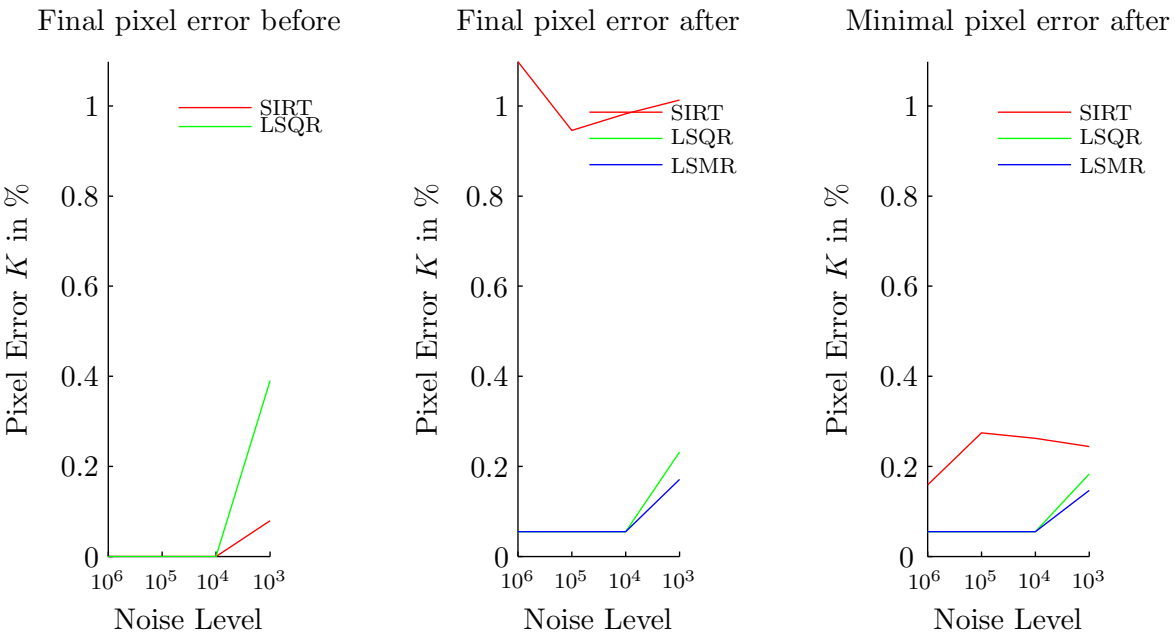


Figure A.21: *The performance of the new DART that aims to mimic the fixed-free pixels construct. Test problem Two Circles scanned using 10 projection angles.*

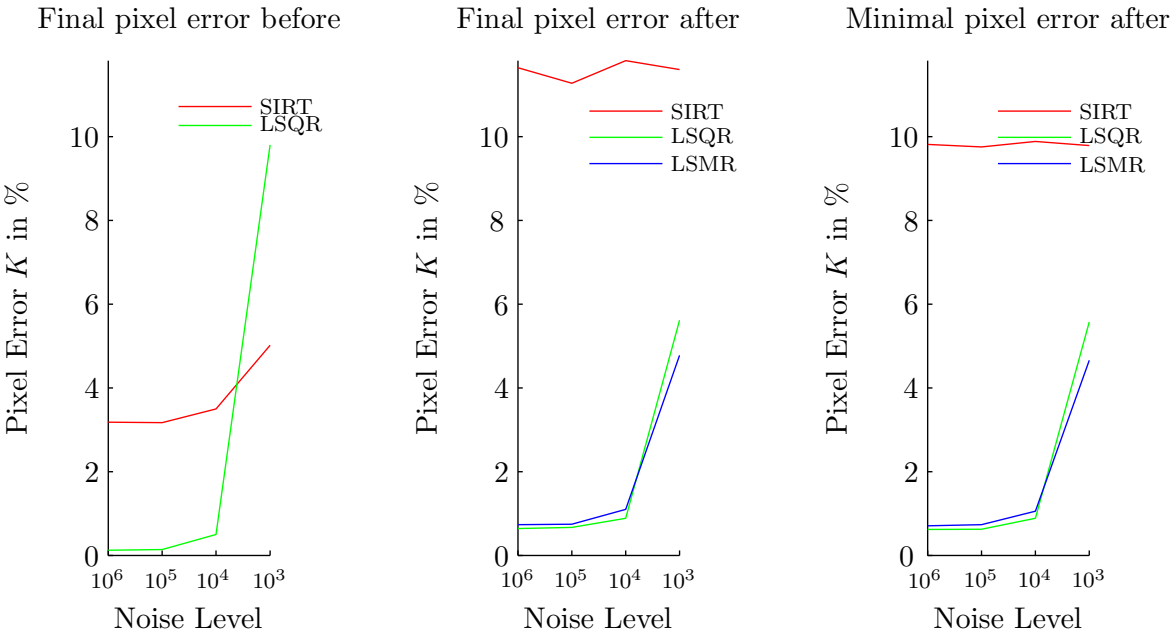


Figure A.22: *The performance of the new DART that aims to mimic the fixed-free pixels construct. Test problem Bone scanned using 90 projection angles.*

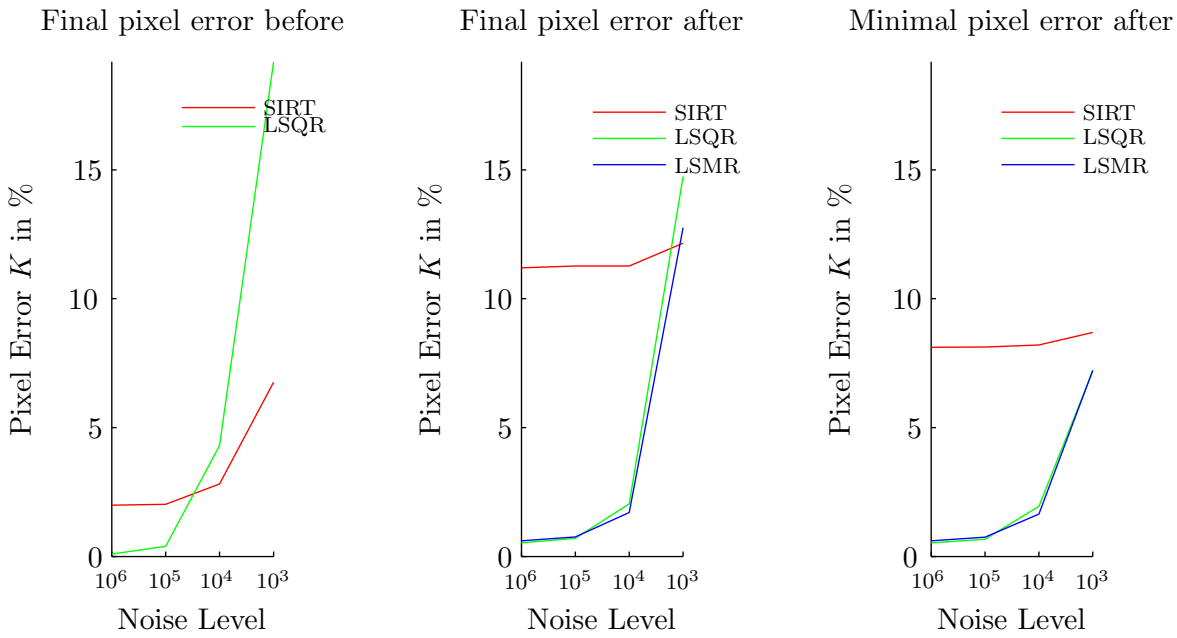


Figure A.23: *The performance of the after that aims to mimic the fixed-free pixels construct. Test problem Shapes scanned using 30 projection angles.*

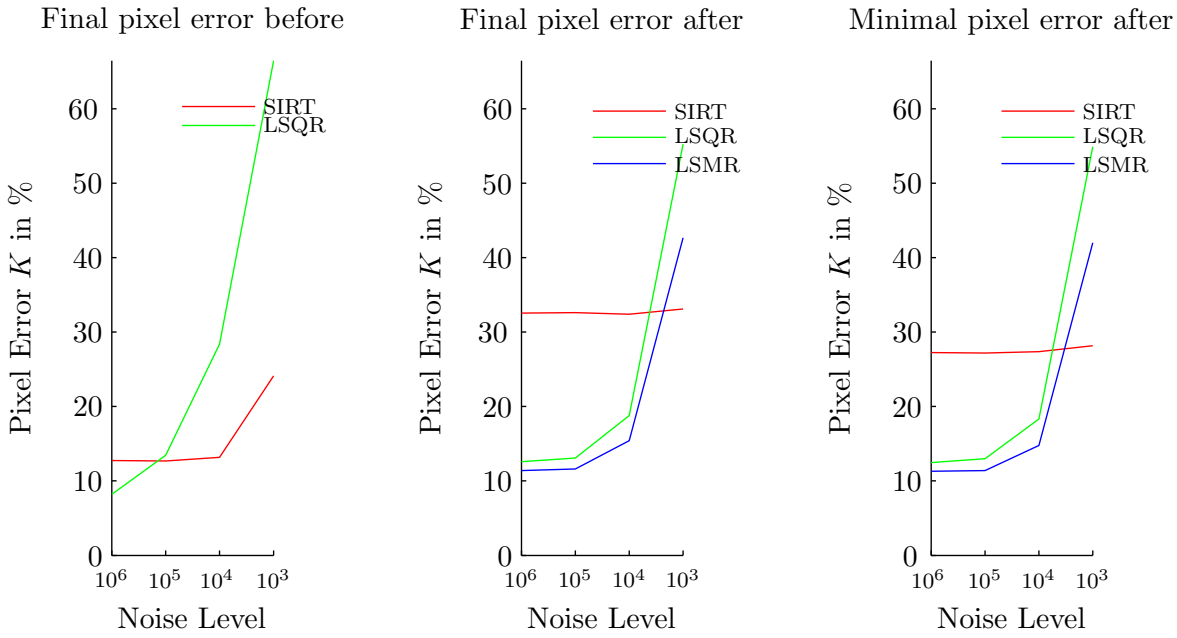


Figure A.24: *The performance of the after that aims to mimic the fixed-free pixels construct. Test problem Phantom scanned using 75 projection angles.*

A.4.2 Figures: More Degrees of Fixedness

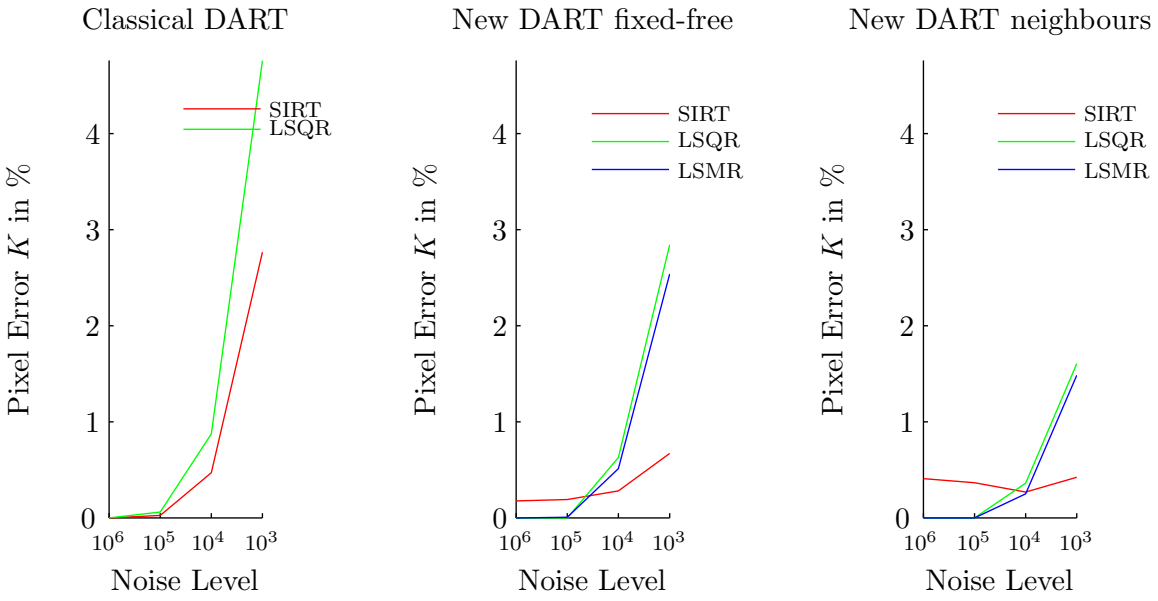


Figure A.25: *Performance of: left: the classical DART, middle: new DART with the fixed-free pixels construct, right: new DART with the neighbours construct. The test problem was Hexagon scanned using 6 angles.*

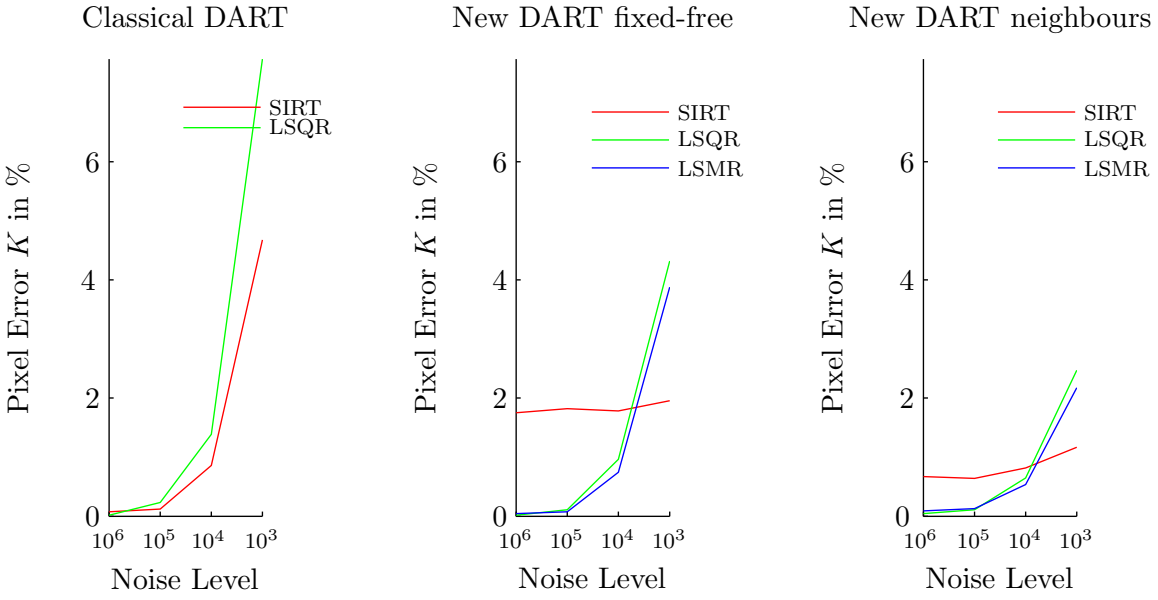


Figure A.26: *Performance of: left: the classical DART, middle: new DART with the fixed-free pixels construct, right: new DART with the neighbours construct. The test problem was Blob scanned using 10 angles.*

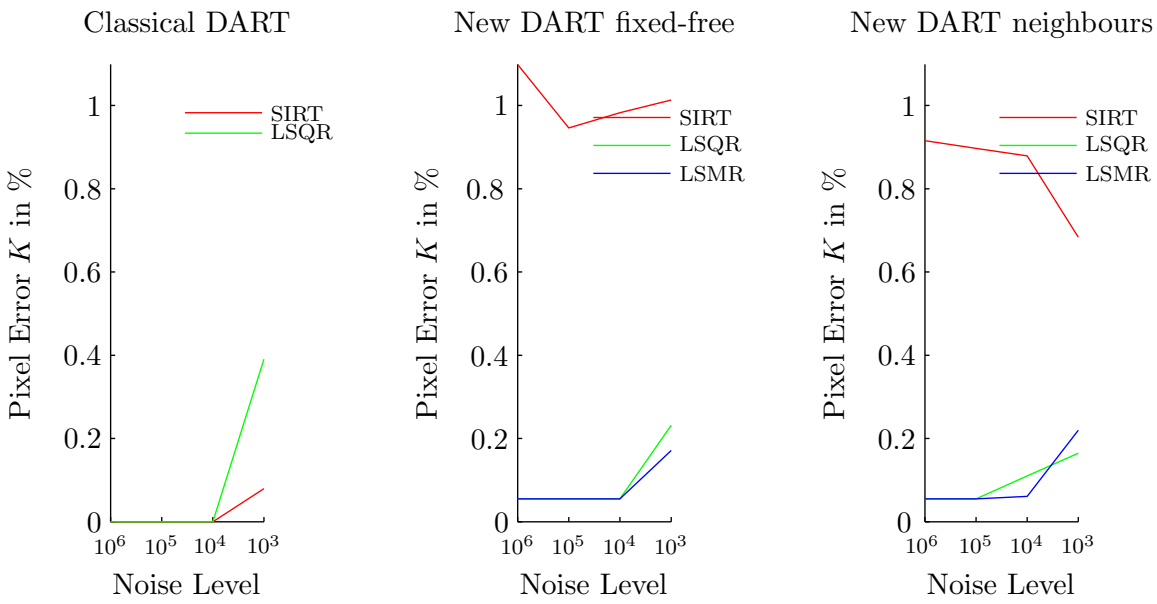


Figure A.27: *Performance of: left: the classical DART, middle: new DART with the fixed-free pixels construct, right: new DART with the neighbours construct. The test problem was Two Circles scanned using 10 angles.*

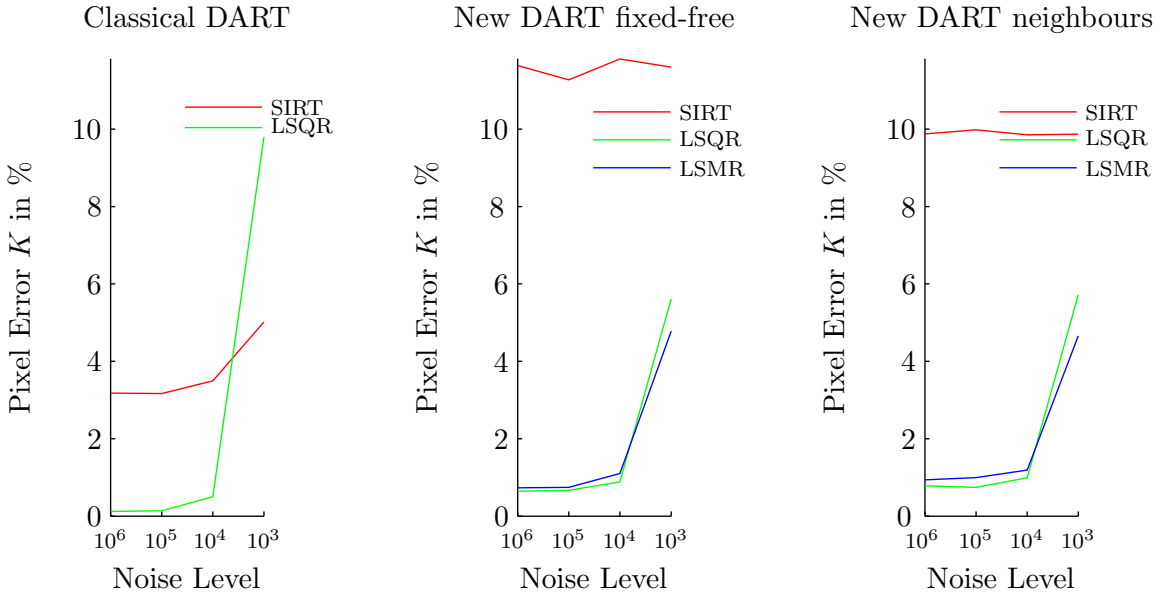


Figure A.28: *Performance of: left: the classical DART, middle: new DART with the fixed-free pixels construct, right: new DART with the neighbours construct. The test problem was Bone scanned using 90 angles.*

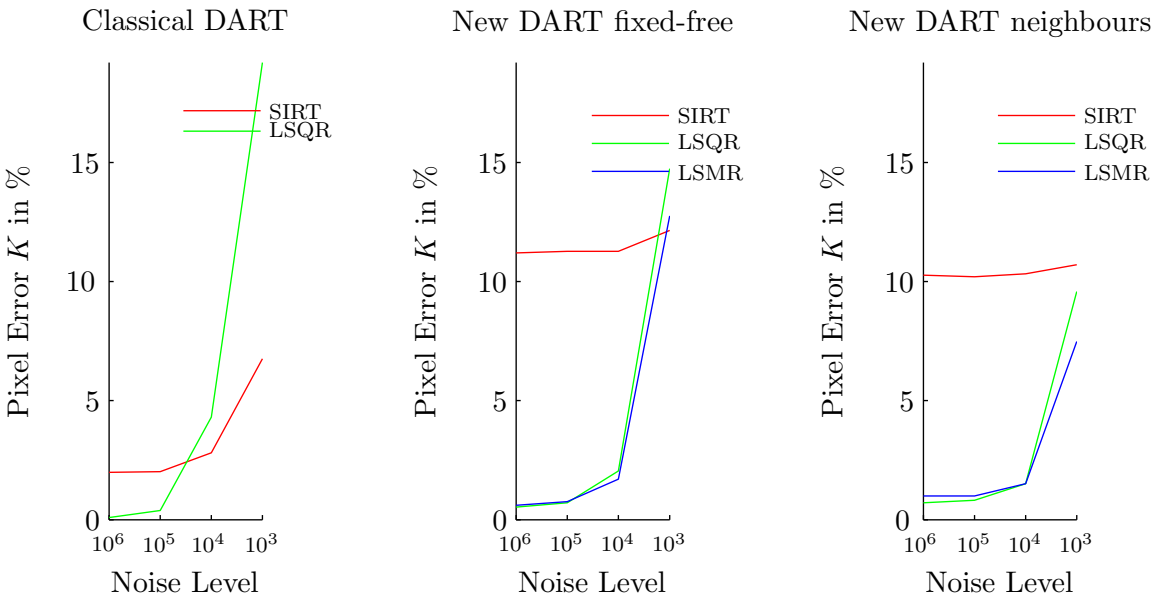


Figure A.29: *Performance of: left: the classical DART, middle: new DART with the fixed-free pixels construct, right: new DART with the neighbours construct. The test problem was Shapes scanned using 30 angles.*

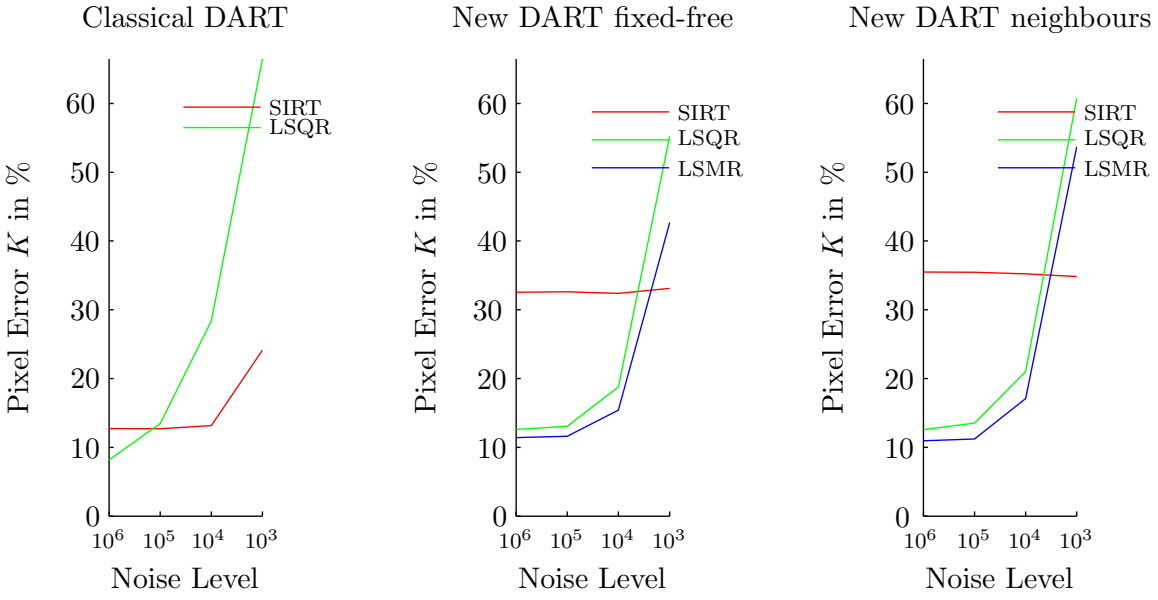


Figure A.30: *Performance of: left: the classical DART, middle: new DART with the fixed-free pixels construct, right: new DART with the neighbours construct. The test problem was Phantom scanned using 75 angles.*

A.4.3 Figures: Number of ARM iterations

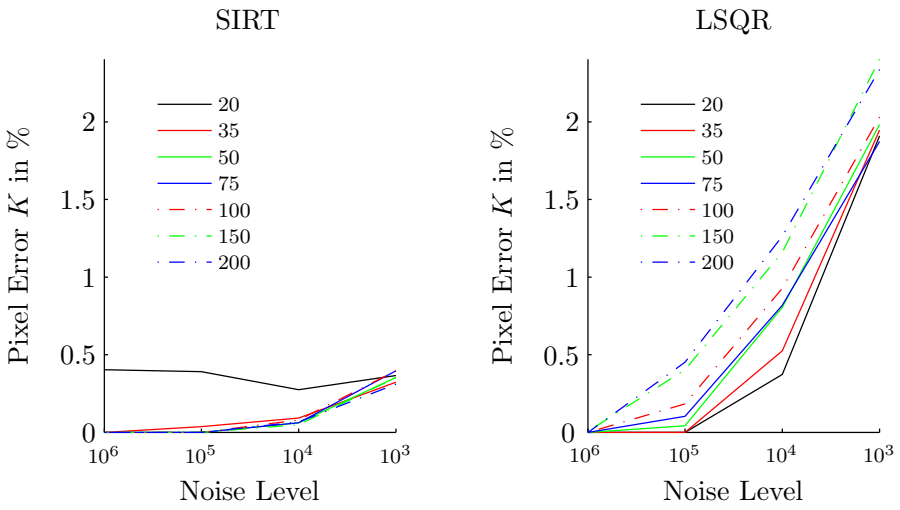


Figure A.31: *Performance of the new DART with the neighbours construct for a various number of ARM iterations. Test problem Hexagon scanned using 6 angles.*

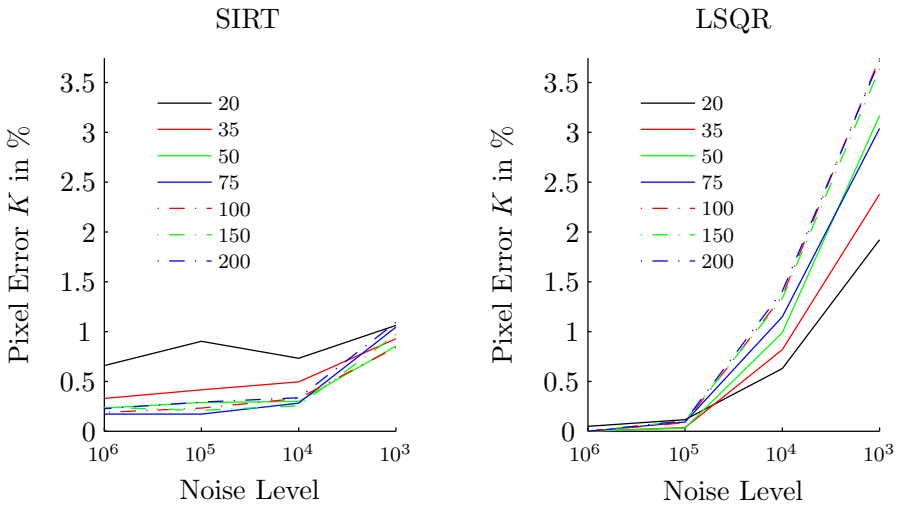


Figure A.32: *Performance of the new DART with the neighbours construct for a various number of ARM iterations. Test problem Blob scanned using 10 angles.*

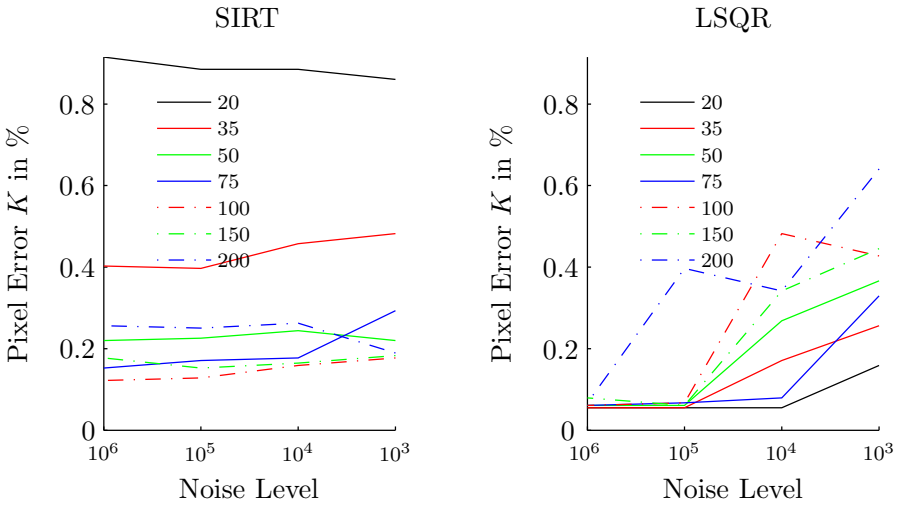


Figure A.33: *Performance of the new DART with the neighbours construct for a various number of ARM iterations. Test problem Two Circles scanned using 10 angles.*

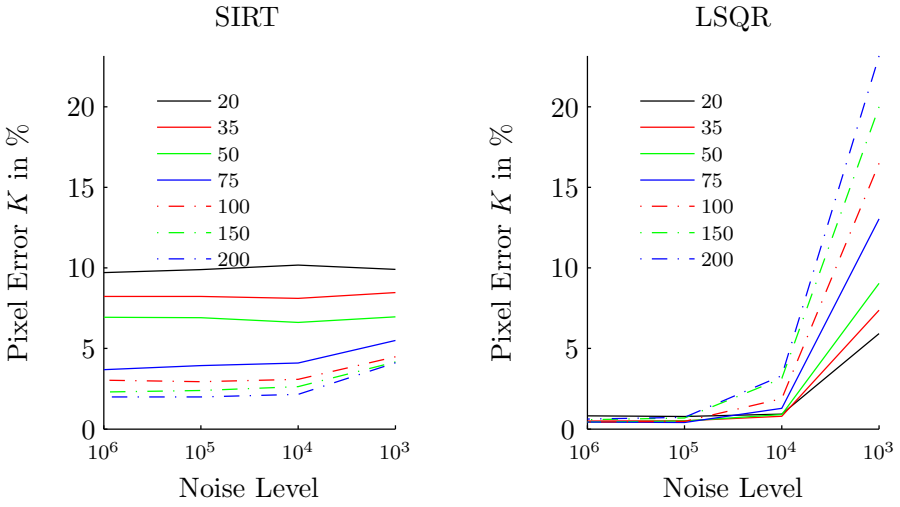


Figure A.34: *Performance of the new DART with the neighbours construct for a various number of ARM iterations. Test problem Bone scanned using 90 angles.*

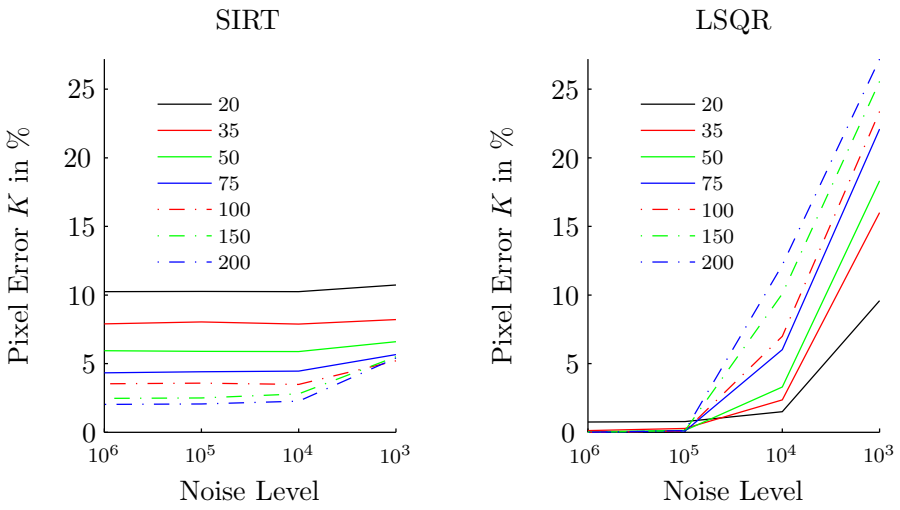


Figure A.35: *Performance of the new DART with the neighbours construct for a various number of ARM iterations. Test problem Shapes scanned using 30 angles.*

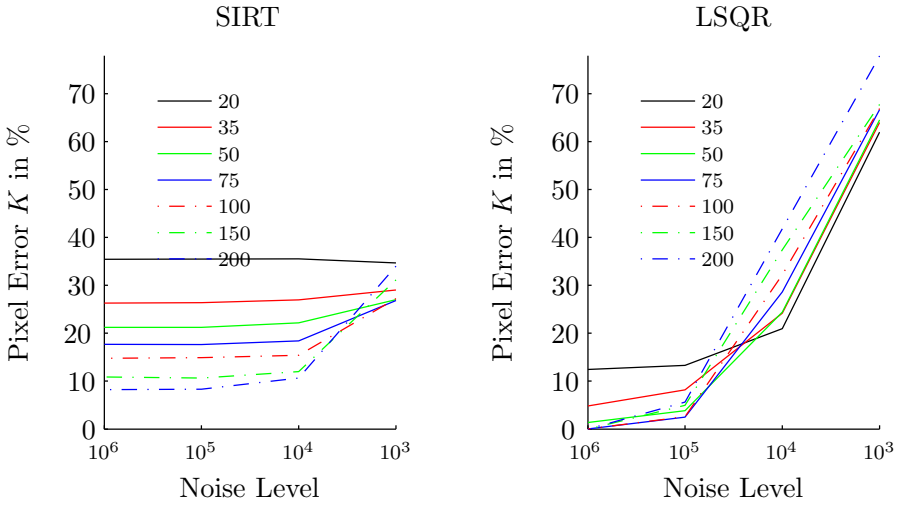


Figure A.36: *Performance of the new DART with the neighbours construct for a various number of ARM iterations. Test problem Phantom scanned using 75 angles.*

A.5 Figures: The Reconstructed Residual Eror

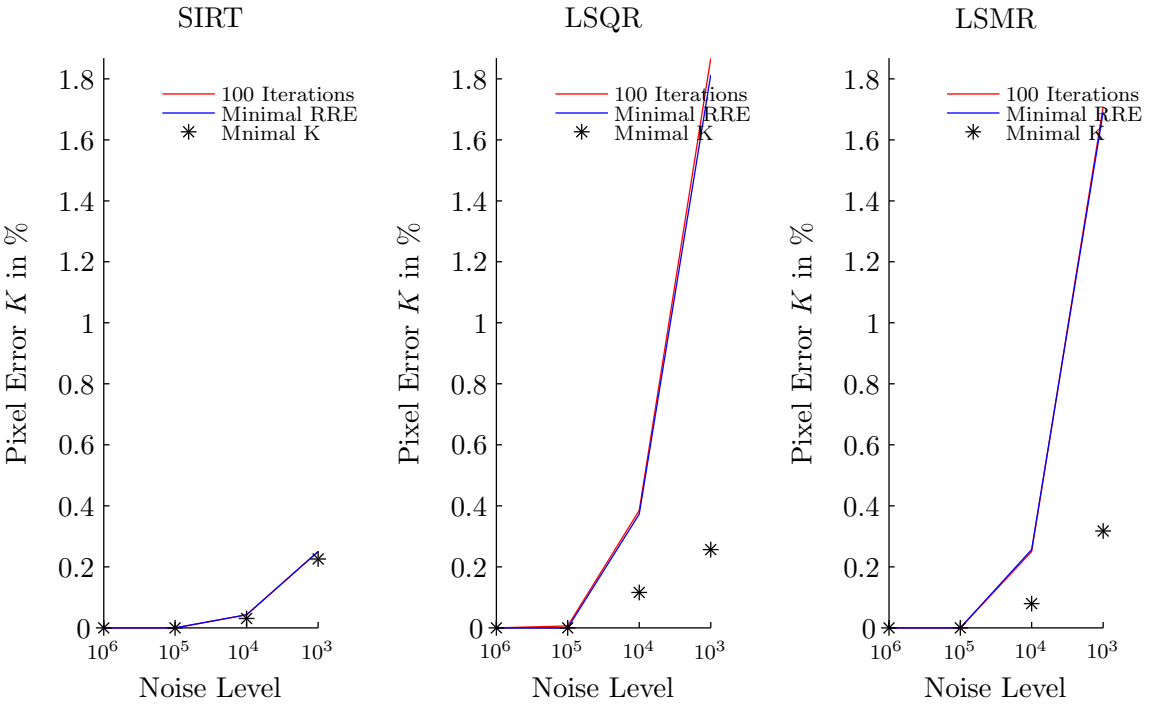


Figure A.37: *The pixel error after 100 DART iterations, when the reconstructed residual error is minimal and the minimal pixel error. Test problem Hexagon scanned using 6 angles.*

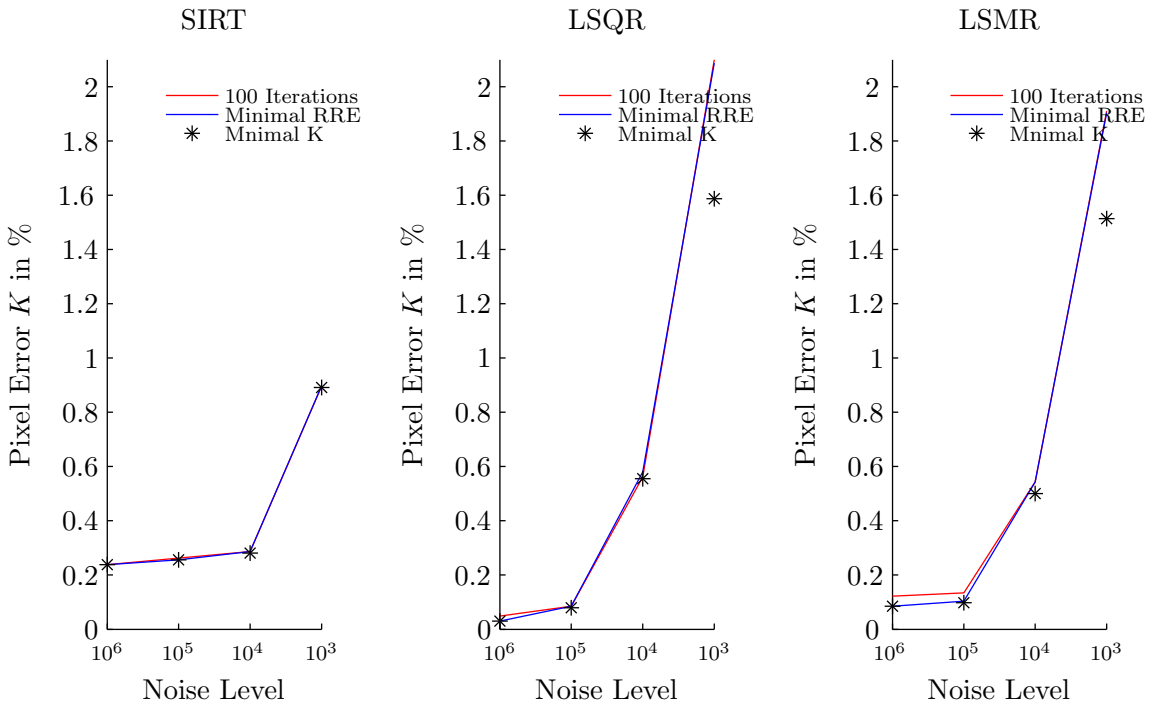


Figure A.38: *The pixel error after 100 DART iterations, when the reconstructed residual error is minimal and the minimal pixel error. Test problem Blob scanned using 10 angles.*

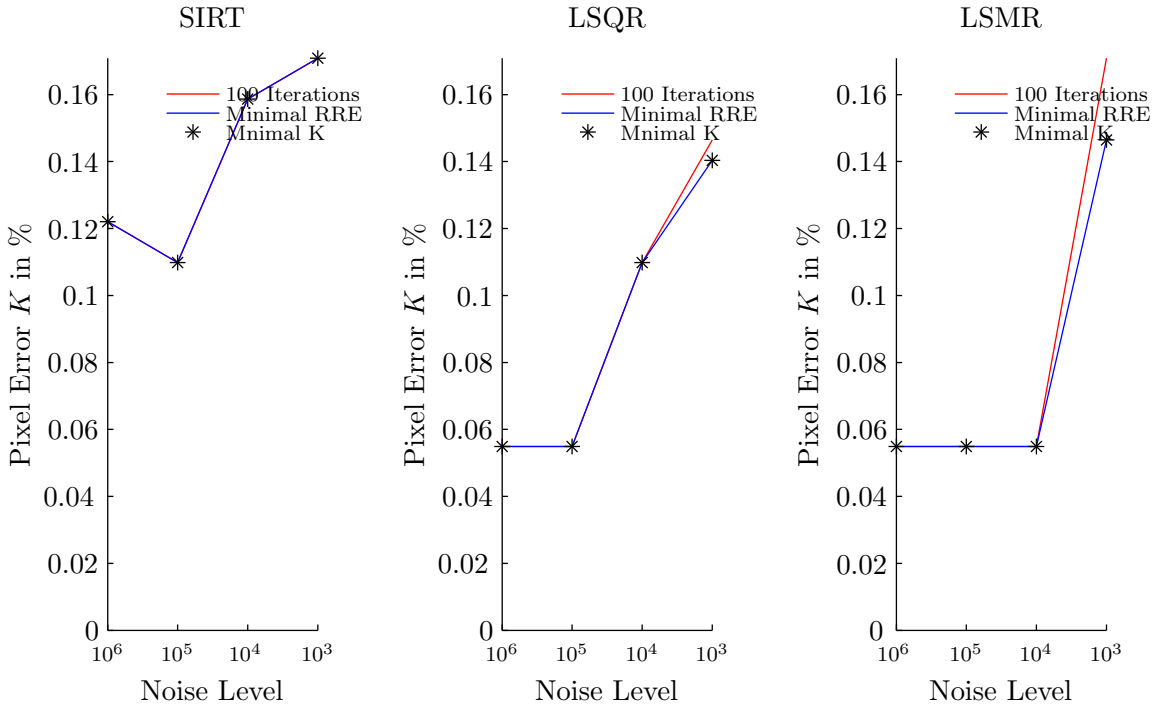


Figure A.39: *The pixel error after 100 DART iterations, when the reconstructed residual error is minimal and the minimal pixel error. Test problem Two Circles scanned using 10 angles.*

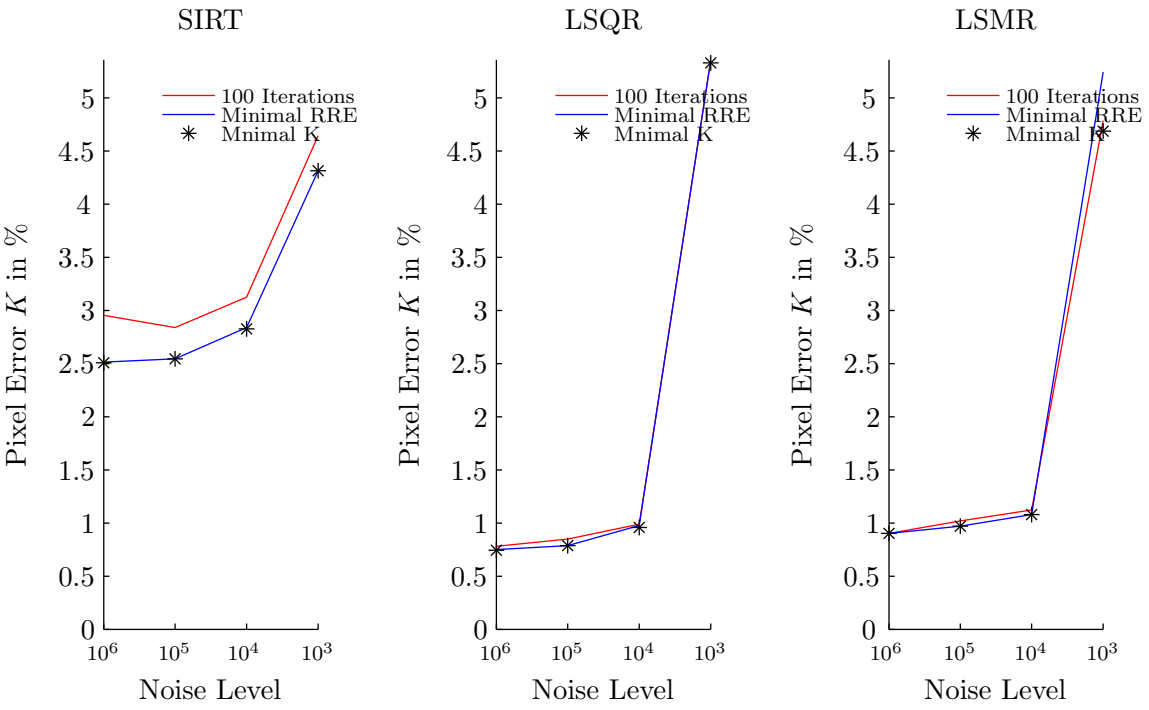


Figure A.40: *The pixel error after 100 DART iterations, when the reconstructed residual error is minimal and the minimal pixel error. Test problem Bone scanned using 90 angles.*

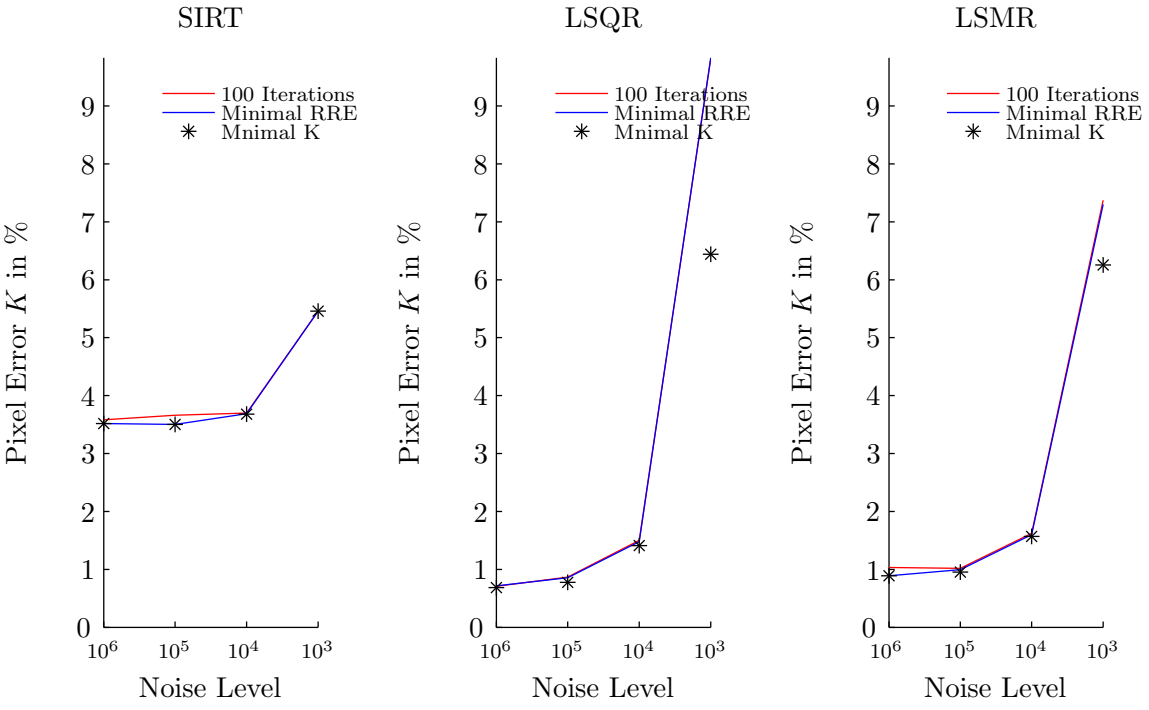


Figure A.41: *The pixel error after 100 DART iterations, when the reconstructed residual error is minimal and the minimal pixel error. Test problem Shapes scanned using 30 angles.*

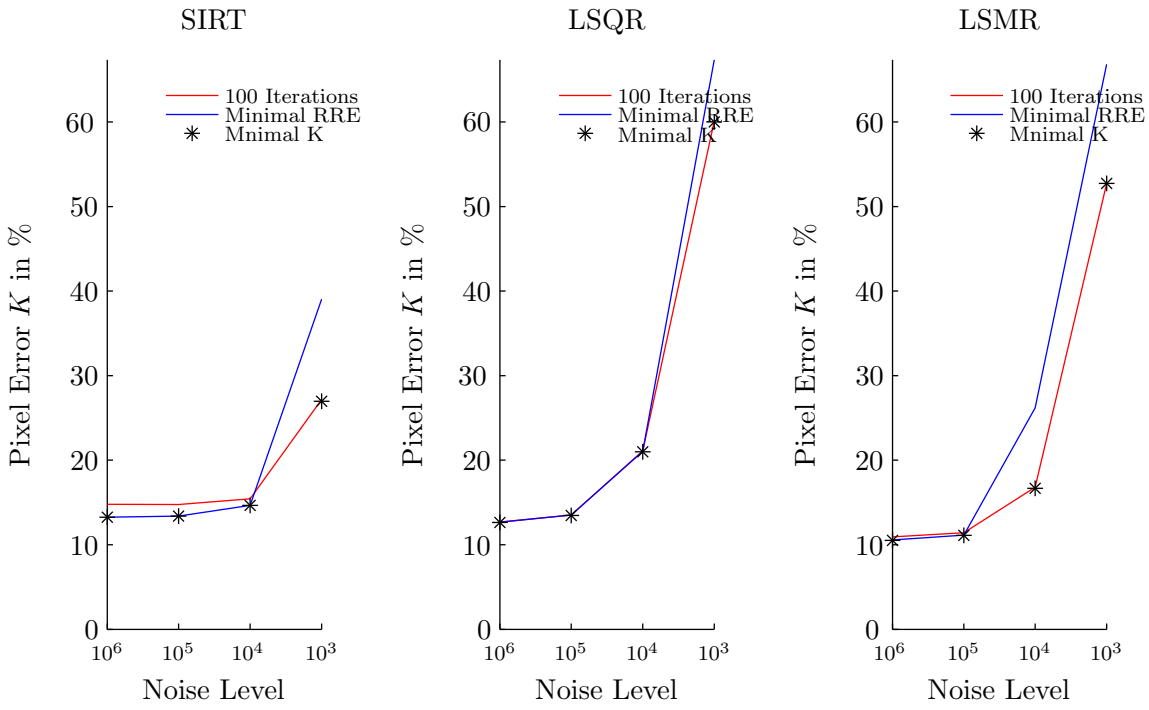


Figure A.42: *The pixel error after 100 DART iterations, when the reconstructed residual error is minimal and the minimal pixel error. Test problem Phantom scanned using 75 angles.*

A.6 Figures: A closer look at SIRT

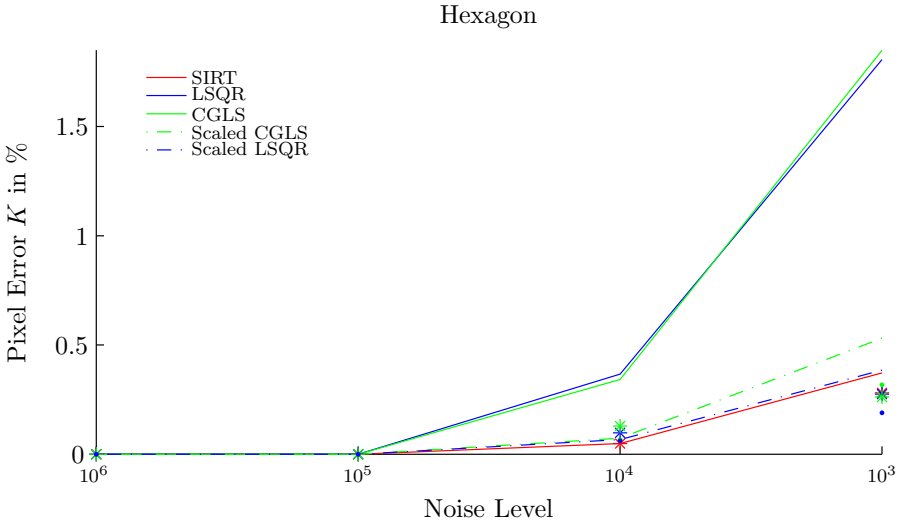


Figure A.43: *The performance of various ARMs, the minimal absolute RRE sum was used as error measure, the asterisks and diamonds correspond to the actual minimal pixel error of the traditional methods and the Scaled methods respectively. Test problem Hexagon scanned using 6 angles.*

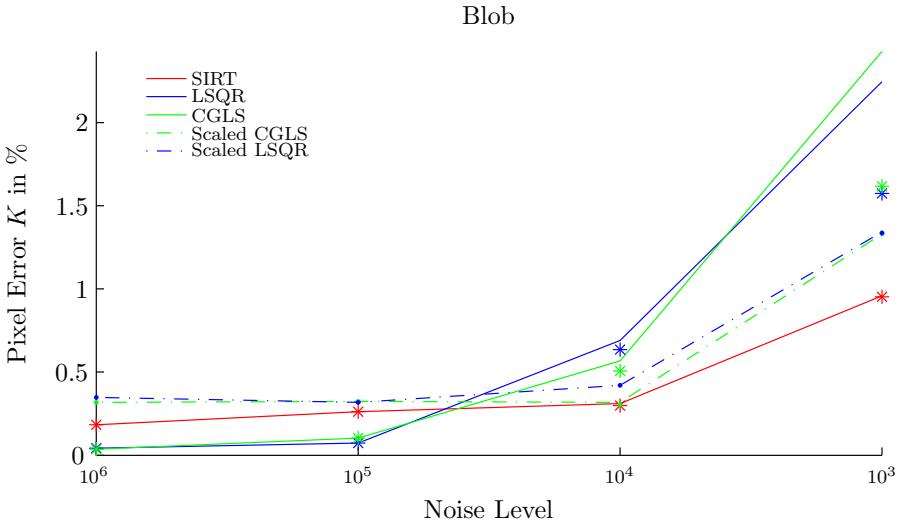


Figure A.44: *The performance of various ARMs, the minimal absolute RRE sum was used as error measure, the asterisks and diamonds correspond to the actual minimal pixel error of the traditional methods and the Scaled methods respectively. Test problem Blob scanned using 10 angles.*

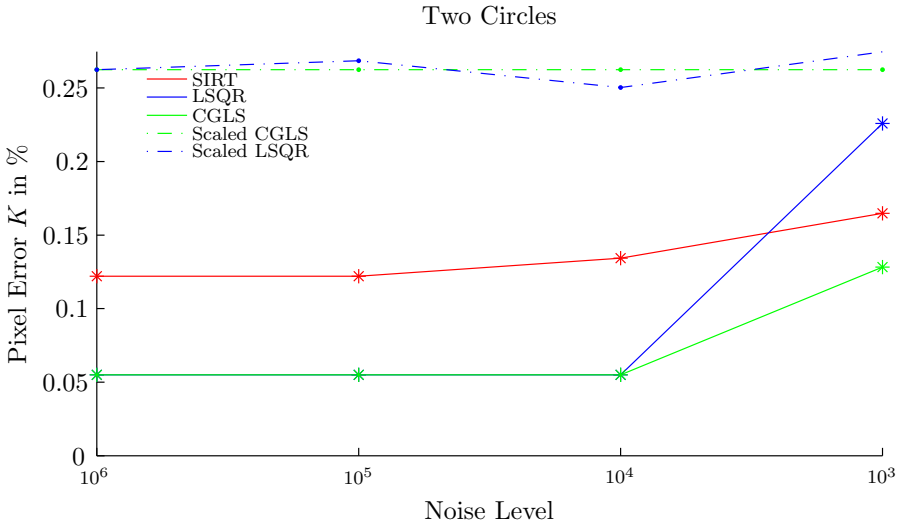


Figure A.45: *The performance of various ARMs, the minimal absolute RRE sum was used as error measure, the asterisks and diamonds correspond to the actual minimal pixel error of the traditional methods and the Scaled methods respectively. Test problem Two Circles scanned using 10 angles.*

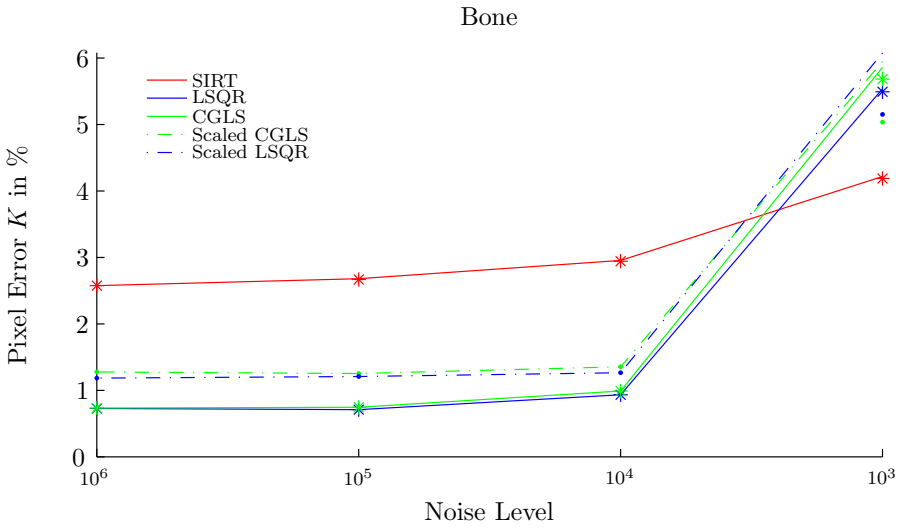


Figure A.46: *The performance of various ARMs, the minimal absolute RRE sum was used as error measure, the asterisks and diamonds correspond to the actual minimal pixel error of the traditional methods and the Scaled methods respectively. Test problem Bone scanned using 90 angles.*

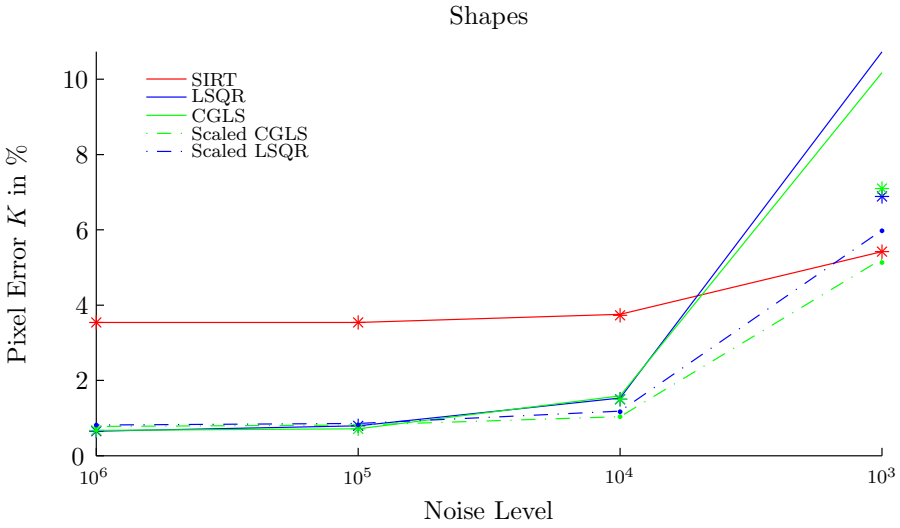


Figure A.47: *The performance of various ARMs, the minimal absolute RRE sum was used as error measure, the asterisks and diamonds correspond to the actual minimal pixel error of the traditional methods and the Scaled methods respectively. Test problem Shapes scanned using 30 angles.*

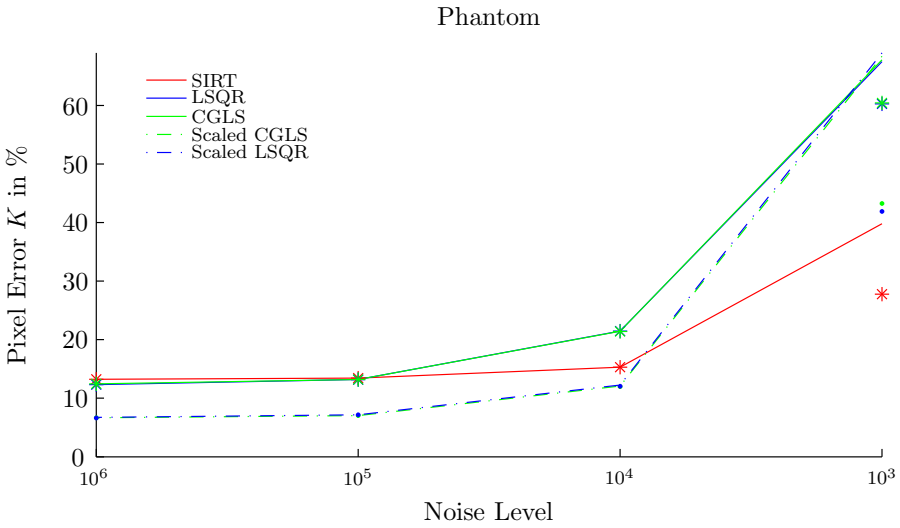


Figure A.48: *The performance of various ARMs, the minimal absolute RRE sum was used as error measure, the asterisks and diamonds correspond to the actual minimal pixel error of the traditional methods and the Scaled methods respectively. Test problem Phantom scanned using 75 angles.*

Appendix B

Tables

B.1 Tables: Minimal Angles

Method \ Angles	2	3
ART	36	2
SIRT	96	80
SART	140	135
CGLS	0	0
LSQR	0	0
CGNE	4	0
LSMR	0	0

Table B.1: *The pixel errors per angle for Hexagon.*

Method \ Angles	2	3	4	5	6	7
ART	1414	316	33	13	4	2
SIRT	1239	345	76	31	21	9
SART	1298	268	116	45	32	18
CGLS	1111	100	28	17	3	0
LSQR	1111	101	26	17	3	1
CGNE	1392	100	24	12	3	1
LSMR	1111	106	31	19	4	0

Table B.2: *The pixel errors per angle for Blob.*

Method \ Angles	2	3	4	5	6	7
ART	455	455	455	455	455	0
SIRT	455	455	455	455	455	0
SART	455	455	455	455	455	0
CGLS	455	455	455	455	455	0
LSQR	455	455	455	455	455	0
CGNE	455	455	455	455	455	0
LSMR	455	455	455	455	455	0

Table B.3: *The pixel errors per angle for Two Circles.*

Method \ Angles	40	45	50	55	60	65	70	75	80
ART	268	199	81	60	16	3	4	1	0
SIRT	506	457	338	363	322	320	291	278	284
SART	337	260	165	142	108	79	72	52	38
CGLS	316	143	78	84	22	18	12	11	9
LSQR	332	127	69	81	28	16	11	11	9
CGNE	296	163	93	48	26	32	5	1	0
LSMR	282	142	93	75	35	23	18	10	7

Table B.4: *The pixel errors per angle for Bone.*

Method \ Angles	2	3	4	5	6	7	8	9	10	15	20
ART	5357	4085	1953	2038	1536	391	206	211	164	62	0
SIRT	5250	4276	1962	1700	1271	883	345	371	370	224	131
SART	5367	4168	2086	1641	1369	961	375	391	373	192	67
CGLS	5125	5477	1946	1727	1706	287	202	228	154	78	0
LSQR	5125	5377	1976	1703	1736	284	209	233	162	78	0
CGNE	5575	5442	1987	1626	1684	280	210	226	157	59	0
LSMR	5125	5273	1988	1595	1711	276	212	224	165	81	2

Table B.5: *The pixel errors per angle for Shapes.*

Method \ Angles	25	30	35	40	45	50	55	60	65
ART	4383	3997	3710	3297	2447	2077	3	0	0
SIRT	3740	3526	3074	2548	1895	1393	993	865	829
SART	4163	3784	3608	3285	2928	2749	2227	1684	802
CGLS	4386	4023	3776	3324	3128	2486	42	11	2
LSQR	4384	4074	3761	3321	3083	2320	42	12	2
CGNE	4312	4003	3758	3278	2932	2222	4	2	0
LSMR	4342	4026	3777	3424	3146	2543	494	55	26

Table B.6: *The pixel errors per angle for Phantom.*

Method \ Angles	10	15	20	25
ART	467	124	26	3
SIRT	607	281	169	81
SART	595	283	96	39
CGLS	494	98	22	2
LSQR	475	98	26	2
CGNE	477	89	24	3
LSMR	484	100	28	2

Table B.7: *The pixel errors per angle for Cylinders.*

Method \ Angles	25	30	35	40
ART	142	191	5	0
SIRT	593	425	341	324
SART	481	425	280	200
CGLS	151	50	20	1
LSQR	173	51	19	2
CGNE	126	32	3	1
LSMR	237	120	63	14

Table B.8: *The pixel errors per angle for Speckled.*