Reconstruction of binary matrices
from fan-beam projections

Antal Nagy* and Attila Kuba*

Abstract

The problem of the reconstruction of binary matrices from their fan-beam projections is investigated here. A fan-beam projection model is implemented and afterwards employed in systematic experiments to determine the optimal parameter values for a data acquisition and reconstruction algorithm. The fan-beam model, the reconstruction algorithm which uses the optimization method of Simulated Annealing, the simulation experiments, and the results are then discussed in turn.

1   Introduction

Tomography is an imaging procedure where the cross-sections of the 3D object being studied are determined from its projection images. The projection images can be created by some rays that are emitted from a source (like X-rays from an X-ray tube), transmitted through and partially absorbed by the object, and finally detected by some array (plane or line) of detectors. The pixels of the projection image represent the total absorption of the rays along the lines between the source and the corresponding detector elements. Usually several projections of the object are acquired from different directions. Then the task is to compute the cross-sections of the object via some mathematical procedure [1] called *reconstruction from projections*. This imaging technique is routinely used in *computerized tomography* (CT) for example, where the section images of the human body are computed from a huge number of measurements using transmitted X-rays. The general method for the reconstruction of 3D objects is that the 2D cross-sections of the objects are reconstructed from the projections measured in the plane of the selected section, effectively reducing the 3D reconstruction problem to a series of *2D reconstruction problems*. In the case of so-called *truly 3D reconstruction* the whole 3D object is reconstructed using rays in the whole 3D space.

Discrete tomography (DT) is a special kind of tomography that can be applied if the object to be reconstructed consists of only a few known homogeneous materials

---

*Department of Image Processing and Computer Graphics, University of Szeged, Szeged, Hungary, Email: `nagya@inf.u-szeged.hu`

359

(e.g. metal and wood). This information can be incorporated into the reconstruction process, giving one the opportunity of reconstructing simple objects from a much smaller number of projection values than is necessary for more complex objects. For this reason discrete tomography seems to be important in applications where the object is so simple and there is no opportunity or it is too costly to acquire lots of projections, like those in non-destructive testing, electron microscopy and medicine. For a summary of the theory and applications of DT, see [2].

(a) Parallel projections
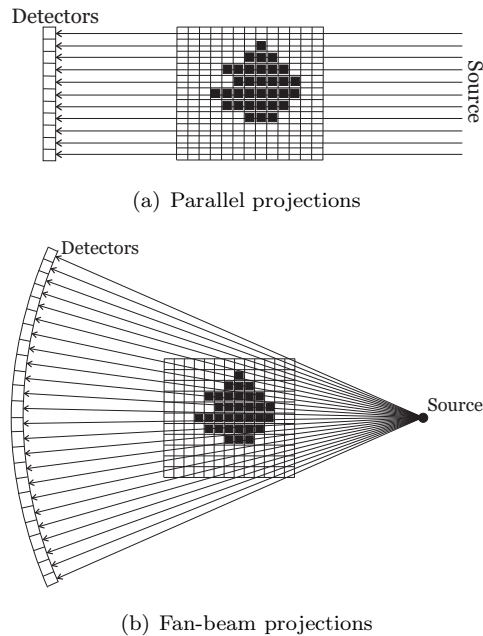

(b) Fan-beam projections

Figure 1: Parallel and fan-beam projections.

There are basically two ways of acquiring the necessary projections (see Fig. 1). In the case of *parallel projections*, the rays parallel to a given direction are transmitted and measured in one phase of the acquisition process. By rotating the system other rays parallel to other directions can be created. In the case of *fan-beam projections* the rays coming from the actual source position (like a fan) are measured at each step. By rotating the source and the detectors around the object new fan-beam projections can be created. Although the two kinds of projection are really equivalent from the viewpoint of information available for the reconstruction (by a suitable rearrangement of the rays one can be transformed to the other), there are different reconstruction methods for parallel and fan-beam projections. This type of classification is common in applications where, owing to the technical possibilities, one of these arrangements is always used.

In this paper we discuss a special discrete tomography problem, namely the reconstruction of binary matrices from their fan-beam projections. The reconstruc-

tion of binary matrices from parallel projections is a classical problem and it has been intensively investigated (see [3, 4], for example). Even the reconstruction from fan-beam (in 2D) or from cone-beam (in 3D) projections is well understood (see [5, 6]). It is interesting that at the same time there are very few papers about DT using fan-beam/cone-beam projections. To our knowledge, the only paper to date that deals with this problem is the article by Peyrin *et al.* [7]. There they discuss results related to truly 3D reconstructions of objects from cone-beam projections. The main reason for this might be that from a mathematical viewpoint some reconstruction results of parallel projections can be applied directly in the case of fan-beam projections, but this is not the case when we desire, for example, the optimal number of X-ray sources, the necessary minimum number of measurements, and so on.

We believe that 2D reconstruction using fan-beam rays is an important and interesting problem especially from an application point of view.

There are several applications of tomography that make use of fan-beam projections and could be of interest in DT e.g. non-destructive testing using X-rays [8] or neutron beams [9].

The aim of this paper is to investigate the quality of reconstruction as a function of the parameters of the fan-beam projection model and the reconstruction algorithm applied. The method employed is the simulation of the whole process from acquiring projections to a comparison of the reconstructed images. The whole simulation procedure is realized in the following way. Binary matrices are created which represent the 2D objects to be reconstructed. Then a model for computing the fan-beam projections is set up and implemented. The fan-beam model contains several parameters like the number of sources and detector elements. With suitable parameter settings different fan-beam data acquisition systems can be simulated. The projections are afterwards computed analytically based on the parameter values of the fan-beam model. The measurement errors can be simulated in our system by some additive random noise. A random-search optimization method was implemented here to reconstruct binary images from the input data. In order to compare the reconstructed images with the original image in an objective way, several measures are implemented; here we present the results of this comparison expressed as a relative mean error. The effects of each parameter are studied in such a way that a sequence of reconstructions is performed by varying only one parameter value and keeping the others fixed. In this way we can produce a curve of the values for the relative mean square and show the effect of the given parameter on the reconstruction process.

The structure of the paper is as follows. In Section 2 the reconstruction problem is introduced with the necessary definitions and our notation for fan-beam projections. Afterwards, we describe the details of the fan-beam model used in our simulation experiments. In Section 4 our DT reconstruction problem is reformulated as an optimization problem, which is then solved using the method of Simulated Annealing (SA). The results of our experiments together with related discussions are given in Section 5. Finally, in the last section, we present conclusions from the studies carried out so far and make suggestions for future work.

2 The reconstruction problem for fan-beam pro- jections

Let f be an integrable real function in the $\mathbb{R}^2$ plane. Let S be a point called the *source point*, and v_θ be a unit vector in the direction $\theta \in [0, 2\pi)$ in the plane. Consider the integrals of f along the half-lines starting from S in direction v_θ

$$[\mathcal{R}f](S, \theta) = \int\limits_0^\infty f(S + u \cdot v_\theta) du \ . \tag{1}$$

The transformation defined by (1) is called the *projection of f taken from the point S in the direction θ*, or the *fan-beam projection of f taken from the point S*. (Another way of acquiring projections is when the integrals of f are taken along parallel straight lines in given directions. This kind of projection is called *parallel*.) Such a projection model can be applied to computerized tomography (see [1], say) where the projections are taken from several hundred source points around the object to be reconstructed.

Given a set of the source points $\mathcal{S}$, the *reconstruction problem using fan-beam projections* can be stated as follows:

FB($\mathcal{S}$) RECONSTRUCTION PROBLEM

Given: *A function $g : \mathcal{S} \times [0, 2\pi) \longrightarrow \mathbb{R}$.*

Task: *Construct a function f such that*

$$[\mathcal{R}f](S, \theta) \ = \ g(S, \theta)$$

for all $S \in \mathcal{S}$ for almost every $\theta \in [0, 2\pi)$.

There are several methods for solving the FB reconstruction problem. For a summary the interested reader may read [1], for example.

In this paper we are interested in the reconstruction of special types of functions from fan-beam projections. Henceforth, let us suppose that the support of f can be covered by an $n \times n$ regular lattice W such that f is constant on each 1×1 square of the lattice, so f can take a value 0 or 1. That is, f can be represented by a binary-valued matrix or, equivalently, by a vector $\mathbf{x} \in \{0, 1\}^J$ where x_j denotes the jth element of the matrix, say, in successive order, where $j = 0, 1, \ldots, J$ and $J = n^2$.

In the majority of applications the projections are acquired from only a finite number of points, S_k, $k = 1, 2, \ldots, K$, along a finite (L) number of half-lines from each point. In this case the ith projection, b_i, from the point S_k in direction v_l ($i = (k-1) \cdot K + l$) can be described by the linear equation

$$\sum_{j=0}^J a_{ij} x_j \ = \ b_i \ , \qquad i = 1, 2, \ldots, I \ , \tag{2}$$

where a_{ij} denotes the length of the intersection of the ith half-line with the jth unit square of W and $I = K \cdot L$. In the linear equation system (2) the projections are obtained (within a certain error) by measurements. The elements of matrix $A = (a_{ij})_{I \times J}$ can be computed knowing the positions of the squares in W and the half-lines starting from the source points. The special feature of (2) is that the unknown vector $\mathbf{x}$ is binary here, i.e. $x_j \in \{0, 1\}$ for all $j = 1, 2, \ldots, J$.

3 The fan-beam model

Based on the method of data acquisition in most of the DT applications, as in our fan-beam model (see Fig. 2), the source points S_k, $k = 1, 2, \ldots, K$, lie on a circle $C_r = \{(x, y) \mid x^2 + y^2 = r^2\}$ around the origin O, where $r > 0$ is large enough for W to be in C_r. Furthermore, it is also usual that the source points are uniformly distributed on C_r, that is $S_k = (r \cdot \cos \theta_k, r \cdot \sin \theta_k)$, where $\theta_k = \theta_0 + (k - 1) \cdot 2\pi/K$ for all $k = 1, 2, \ldots, K$. The *start angle* $\theta_0 \in [0, 2\pi)$ determines not only the position of the first point, but all source points. (The reason for the inclusion of the start angle θ_0 into the model is that since we usually have only a few source points (like 2–4), their number as well as their positions can have a strong influence on the reconstruction, as we shall show later.) For example, a start angle of $0°$ means that the first source lies on the intersection of circle C_r and also on the positive part of the x axis.
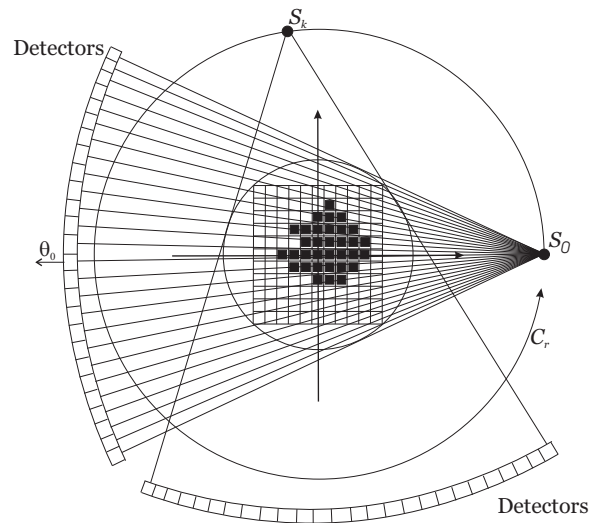

Figure 2: The geometry of our fan-beam model.

In our model the integrals along the half-lines starting from the source point S_k $(k = 1, 2, \ldots, K)$ are measured by L detectors, uniformly placed on the other

side of the object on an arc having its center point in S_k (see Fig. 2). The arc of detectors is big enough for the whole image to lie between the half-lines drawn from the source to the endpoints of the detector arc. Each detector measures one projection value b_i.

For simplicity, we suppose that the center of the rectangle W is at the origin of the coordinate system (see Fig. 2).

We are going to study the effect of noise as well. That is why Gaussian noise was generated and added to the exact projections to create noisy projection data.

In our fan-beam model the following parameters can be varied:

n: the size of the binary matrix to be reconstructed ($n \times n$);

r: the radius of circle C_r, i.e. the distance of the source points from the origin;

θ_0: the start angle which determines the position of the first source point;

K: the number of source points;

L: the number of detector elements or, equivalently, the number of measurements from one source point;

η: the percentage of Gaussian noise added in the projections.

4 Reconstruction as an optimization problem

As we saw earlier the solution of the FB (see Section 2) reconstruction problem in our fan-beam model is equivalent to finding a solution of the linear equation system

$$\mathbf{Ax} \;=\; \mathbf{b}\,, \qquad \text{where } \mathbf{x} \text{ is a binary-valued vector}\,. \tag{3}$$

Since any half-line in our model at most intersects $O(n)$ squares of W, the matrix $\mathbf{A} = (a_{ij})_{I \times J}$ is sparse (it contains only a few non-zero elements). Another important property of this matrix equation in DT applications is that the number of equations (i.e. the number of projections) is usually much less than the number of unknowns, hence $I \ll J$. It means that it can have several solutions, even binary-valued ones. Furthermore, due to measurement errors it is also possible that (3) has no exact solution, so it is better to try to find a binary-valued $\mathbf{x}$ which satisfies (3), at least approximately.

The reconstruction methods like the Algebraic Reconstruction Techniques (ART) (see [1], say) do not necessarily provide a binary-valued $\mathbf{x}$ that satisfies (2). It cannot be applied here because a non-binary solution might be quite different from a binary one.

Actually, a possible way of solving (3) at least approximately is to reformulate it as an optimization problem. Formally, we should find the minimum of the following objective function

$$C(\mathbf{x}) = |\mathbf{Ax} - \mathbf{b}| + \gamma \cdot \Phi(\mathbf{x})\,, \qquad \text{where } \mathbf{x} \text{ is a binary-valued vector}\,. \tag{4}$$

The first term on the rhs ensures that we have an $\mathbf{x}$ satisfying (3) at least approximately. The second term allows us to include *a priori* knowledge about $\mathbf{x}$ into the optimization if there are several good binary vector candidates that keep $|\mathbf{Ax} - \mathbf{b}|$ low. For example, $\Phi(\mathbf{x}) = |\mathbf{x} - \mathbf{x}^{(0)}|$ might be such a case, where $\mathbf{x}^{(0)}$ is a given (probably binary-valued) vector, called a *prototype*. With such a Φ we can look for an $\mathbf{x}$ that is similar to the prototype $\mathbf{x}^{(0)}$. The regularization coefficient γ is needed to weight the two terms in C. For example, if the measurements are very noisy then we can choose a bigger γ than that for the noiseless case to give the second term in (4) a higher weight, so it will play a more important role in the optimization. γ can also be used to control the effects of noise.

Since we are looking for a binary-valued $\mathbf{x}$ in the optimization of (4), the usual numerical optimization methods seem unsuitable here. The combinatorial optimization methods looked more promising and turned out to be useful. Among them we selected the *simulated annealing* (SA) optimization procedure [10]. The reason for this selection was that it was easy to implement; it can be easily adapted to the objective function (i.e. with very small modifications the program is suitable for optimizing other objective functions). Besides, since our main aim is to study the effects of the fan-beam model parameters, the selection of the optimization method actually plays only a secondary role here. We believe that we would probably get similar results with other optimization methods because they have to optimize the objective function as well.

4.1   Simulated annealing

SA is a random-search technique that is based on the physical phenomenon of metal cooling. The system of metal particles gradually reaches the minimum energy level where the metal freezes into a crystalline structure. Based on previous works [11], we implemented the SA algorithm like so (see Fig. 3):

The algorithm starts from an arbitrary initial binary image $\mathbf{x}^{(0)}$, an initial (high) temperature $T^{(0)}$ and calculates the objective function value $C(\mathbf{x})$. Then a position j is randomly chosen in the image $\mathbf{x}$. Let $\mathbf{x}'$ be the image that differs from $\mathbf{x}$ only by changing the value of $\mathbf{x}$ in position j to the other binary value, i.e. $x'_j = 1 - x_j$. This change is accepted by the algorithm, i.e. $\mathbf{x}$ is replaced by $\mathbf{x}'$ if $C(\mathbf{x}') < C(\mathbf{x})$. Even if the objective function does not get smaller, the change is accepted with a probability depending on the difference $\Delta C = C(\mathbf{x}') - C(\mathbf{x})$. Formally, the change is accepted even in that case when

$$\exp(-\Delta C/\kappa T) > z\,, \tag{5}$$

where κ, T and z are, respectively, the Boltzmann constant ($11.3805 \times 10^{-23} \times m^2 kg s^{-2} K^{-1}$), current temperature and a randomly generated number from a uniform distribution in the interval $[0, 1]$. Otherwise, the change is rejected, i.e. $\mathbf{x}$ does not change in this iteration step.

If a change is rejected then we test the level of *efficiency* of changes in the image in the last iterations. It means that we count the number of rejections in
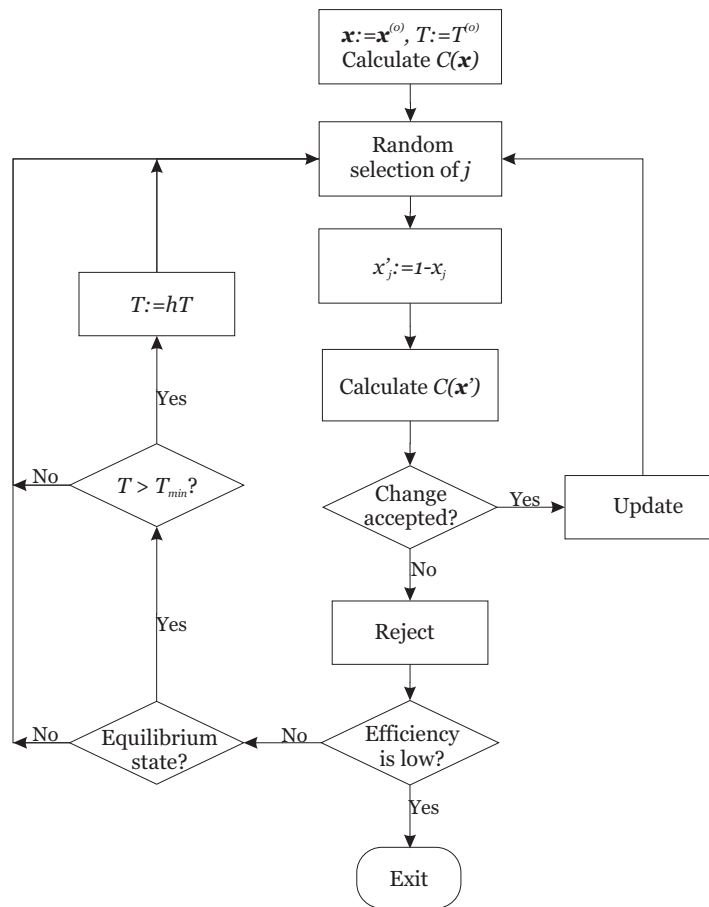
Figure 3: Flow-chart of the implemented SA algorithm.

the last N_{iter} iterations. If this number is greater than a given threshold value R_{thr} then the efficiency of changes is too low and the SA optimization algorithm will be terminated.

We calculate the variance of the cost function in the last N_{var} iterations. A so-called *equilibrium state* is said to be attained if the present estimate of the current ΔC variance is greater than the previous variance estimate. If the equilibrium state is achieved, we reduce the current temperature (allowing changes with smaller probabilities when the value of the objective function is greater) and let the algorithm run with a lower temperature value (T is replaced by $h \cdot T$, where h is the so-called *cooling factor*). In our experiments we chose the same value for the parameter as in [7], namely $h = 0.9$.

Our SA algorithm then has the following parameters:

$\mathbf{x}^{(0)}$: initial image;

$T^{(0)}$: initial temperature;

N_{iter}: number of iterations used in the computation of efficiency;

R_{thr}: threshold value for the number of rejected changes in the last N_{iter} iterations;

N_{var}: number of accepted iterations used in the computation of the variance of the cost function.

5   Results and discussion

In this section the experimental results using our fan-beam model and the implemented SA algorithm are presented together with a discussion of each.

The simulation experiments were performed with phantom images each having a size 200×200 (i.e. $n = 200$). The projections of the phantom images were computed based on (2) for each parameter setting. The images were then reconstructed from the projections using the SA algorithm outlined above. In order to get quantitative results, the original phantom images were compared pixel by pixel according to the relative mean error

$$M_e = \frac{\sum\limits_{j=1}^{J} |x_j - \hat{x}_j|}{\sum\limits_{j=1}^{J} \hat{x}_j} \, , \tag{6}$$

where $\hat{\mathbf{x}} = \{\hat{x}_j\}_{j=1}^{J}$ denotes the vector of the original image. Clearly, $M_e \geq 0$ and the smaller value indicates a better comparison result. Furthermore, $M_e = 0$ if and only if $\mathbf{x} = \hat{\mathbf{x}}$.

Since we had an optimization process based on a random-search, we repeated each test 100 times with the same parameter setting. The mean of the 100 M_e values was computed and presented later as the result for each test with the given parameter setting. The average image of the 100 binary images was given as the result of the reconstruction for one parameter setting.

Naturally, several parameter settings were tested. One of them, the so-called *baseline parameter setting*, played a special role. Here only one of the parameters was allowed to change at a time, the others having the same values as in the baseline parameter setting case. In order to see the effect of the parameters on the quality of the reconstruction, a sequence of tests was performed for each parameter. During a test sequence only the value of the selected parameter was changed and the other parameters always had the same values as in the baseline parameter setting case. For instance, to see the effect of increasing the number of sources, we changed the value of K in the model (lying in the range 2–32), computed the projections for the same phantom image, ran the reconstruction algorithm with the

Table 1: Baseline parameter setting

Parameter	Baseline value	Range
Distance from sources to origin (r)	250	$[250, 1750]$
First source position angle (θ_0)	$0°$	$[0°, 360°/K]$
Number of sources (K)	$K \in \{22, 32\}$	$[2, 32]$
Number of detector elements (L)	401	$[101, 401]$
Additive noise	$\eta \in \{0\%, 5\%\}$	$[0\%, 45\%]$
Initial temperature (T)	$4°$	$[4°, 104°]$
Number of iterations (N_{iter})	10000	$[1000, 10000]$
Rejected iterations (R_{thr})	9990	$[990, 9990]$
Variance iterations (N_{var})	5000	$[500, 5000]$
Regularization parameter (γ)	145	$[0, 145]$

same parameter settings 100 times, took the reconstructed 100 images, computed the 100 M_e values for the reconstructed and original images, calculated the average images, then finally drew a curve showing the changes of M_e as a function of the number of projections. The curve drawn from the mean values of a sequence like this is presented as the final result of the observations associated with the chosen parameter. The baseline parameter setting together with the range of the parameter values are given in Table 1.

Figure 4: Baseline software phantom used in the tests.

The experiments were executed with the Φ_{poz} penalty term and $\gamma = 145$ regularization parameter (Section 5.4).

As can be seen from Table 1, the noise contribution was only 0% or 5%. In order to assess the effect of noise we repeated the tests not just with 0% noise but also with 5% noise. From the test results, we obtained two curves, one without noise and one with 5% noise.

We repeated the test for $K = 32$ and $K = 22$ in most of the experiments. These produced two more curves for the experiments that were performed.

The parameters and the corresponding results can be divided into four groups. This is why we have chosen to discuss the results separately in the subsequent

four subsections. They show the effect of changing the parameters of the fan-beam model, the SA optimization algorithm, the complexity of the phantom image to be reconstructed, the regularization parameter γ given in (4), and the effect of adding noise.

5.1 The parameters of the fan-beam model

In this group of tests we studied the effects of changing those parameters related to the fan-beam model listed in Section 3.

5.1.1 Distance between sources and origin

This parameter was varied between 250 and 1750 while the detector angular aperture covered the W lattice to be reconstructed and the number of detectors L were kept constant (see Fig. 5). Of course, as the distance between the source points and the origin increases, the fan-beam model approaches the model of parallel projections when the same detector parameters are used.
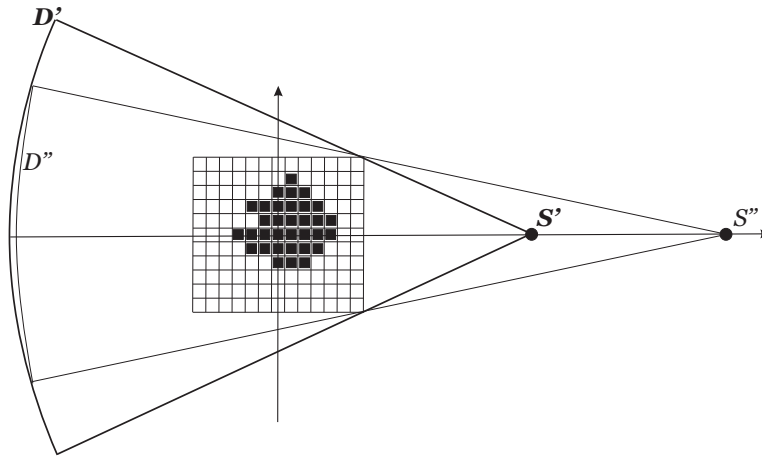
Figure 5: Changing the distance between source and origin. Detector arc D' is for the source point S' and detector arc D'' is for the source point S''.

The curves in Fig. 6 both show that there is no big difference between the results when the source is close to or far from the origin. More generally, there is no real difference between the fan-beam and parallel-beam projections if we change this distance. This is primarily because the further we go from the origin, the more parallel the beams become. It seems that the equations belonging to the near-parallel rays determine the image in both cases.
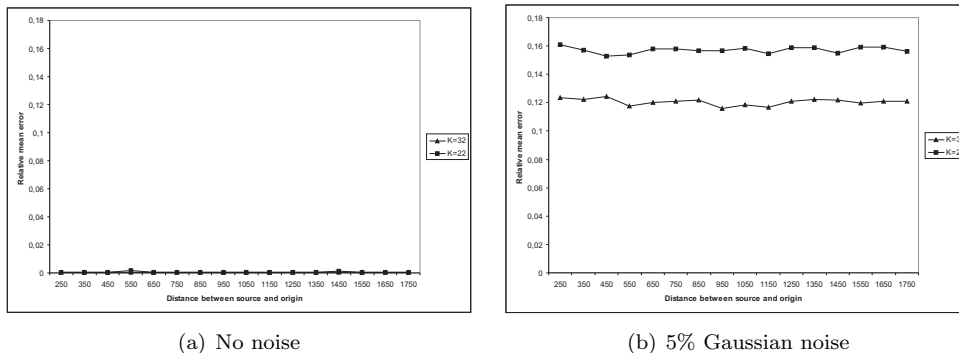
(a) No noise

(b) 5% Gaussian noise

Figure 6: Relative mean error as a function of the distance between source and origin.
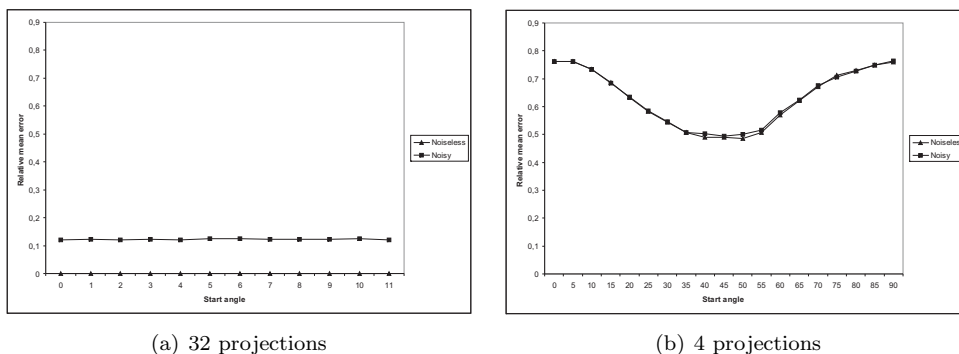
(a) 32 projections

(b) 4 projections

Figure 7: Relative mean error as a function of the start angle.

5.1.2   Start angle

We varied the value of the start angle parameter from $0°$ to $360°/K$ degrees. It is clear that by determining the position of the first source we also determine the positions of all sources around the circle C_r (if the number of sources is fixed). The relative mean square curves describing the effects of these changes are given in Fig. 7(a) for the case when the number of sources is just the base setting (i.e. $K = 32$). The curves indicate that there is little real difference in the relative mean square if we have a relatively large number of source points.

But we obtain quite different curves when the number of projections is small. For example, for $K = 4$ source points we get curves which show that the quality of the reconstruction changes depends on the source positions (see Fig. 7(b)).

The reason for the shape of the curves in Figs. 7(a) and 7(b) is that there are certain projection values which provide more information for the reconstruction

than others. For example, a projection value of 0° means that along the corresponding half-line all pixels intersecting the half-line have a value of 0° (which may be called an *empty line of pixels*). Another is when a projection value is the same as the length of the intersection of the square in W with the corresponding half-line; in that case all pixels along the half-line have a value 1 (let us call them a *full line of pixels*). If the source points have positions such that there are many pixels lying in almost empty or almost full lines then a large part of the image can be reconstructed from a few projections. Looking at Fig. 4 we notice that this is indeed the case when one of the source points is in a position near 40°.

The reconstructed images can be seen in Fig. 8 when the number of sources is 4. These figures confirm the earlier belief that the quality will be better when the first source point lies almost on the same line as the centers of 3 circles.
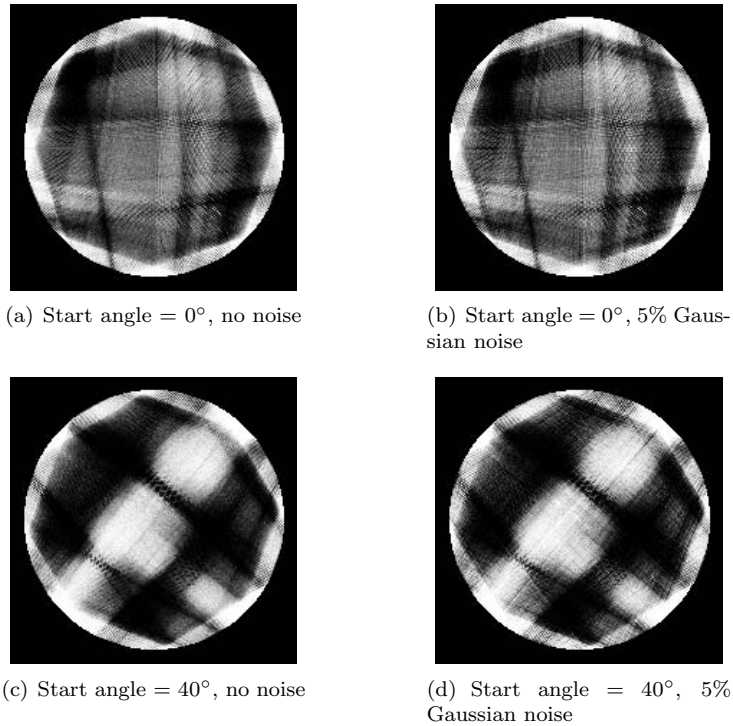

(a) Start angle = 0°, no noise

(b) Start angle = 0°, 5% Gaussian noise

(c) Start angle = 40°, no noise

(d) Start angle = 40°, 5% Gaussian noise

Figure 8: Images reconstructed from 4 projections with different start angles.

5.1.3 Number of sources

The number of sources in the simualtions was varied from 2 to 32. It came as no surprise that increasing the number of sources and number of projection values in

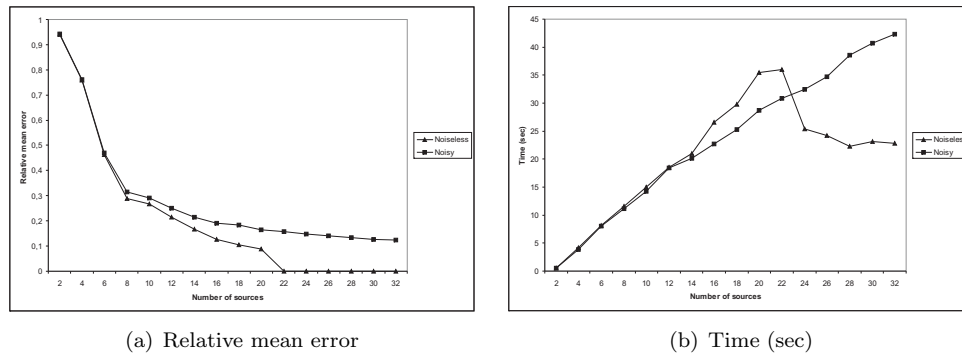(a) Relative mean error   (b) Time (sec)

Figure 9: The effect of varying the number of sources.

turn increased the reconstruction quality. But the real question here was how to find the minimum number needed to produce a good reconstructed image for a particular case. Figure 9 provides the information needed to answer this question. It is also tells us what this number depends on.

The graphs in Fig. 9(a) show that taking more than 12 sources improves the mean square error by a very small amount. When there were noiseless projections the data from 22 sources was sufficient to achieve high quality reconstructions.

The view, based on the graphs in Fig. 9(a), that 22 or more source points hardly affects the quality of the reconstruction is indeed borne out by inspecting the reconstructed images (Fig. 10). At the same time reconstructing the phantom from 22 projections takes more time than reconstructing it from 32 projections when no Gaussian noise was added (see Fig. 9(b)).

5.1.4   Number of detector elements

The number of detector elements also means that the number of projection values measured will belong to a source point. If we have more detector elements we will also have more equations in (2), hence more information about the image. But, of course, for the detector elements beyond a certain number we cannot obtain a better mean square error (Fig. 11).

That increasing the number of detector elements will bring an improvement only up to some limit can be readily explained. Here it is about 281 when $K = 32$ (see Figs. 11 and 12) and the binary image seems to be determined by the exact projection data.

This experiment was repeated with $K = 22$. We got approximately the same relative mean error level for $L = 401$ when no noise was added (see Fig. 11(a)). The number of equations to be solved is almost the same in both cases (i.e. $32 \cdot 281 \approx 22 \cdot 401$), which helps to explain why the results were so similar.
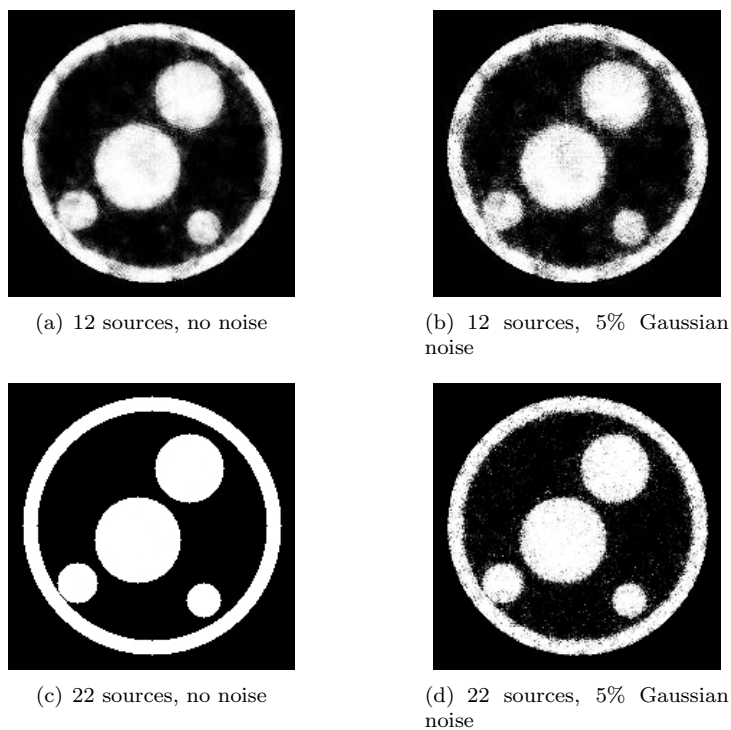
(a) 12 sources, no noise

(b) 12 sources, 5% Gaussian noise

(c) 22 sources, no noise

(d) 22 sources, 5% Gaussian noise

Figure 10: Images reconstructed from projections with a different number of sources.

(a) No noise

(b) 5% Gaussian noise
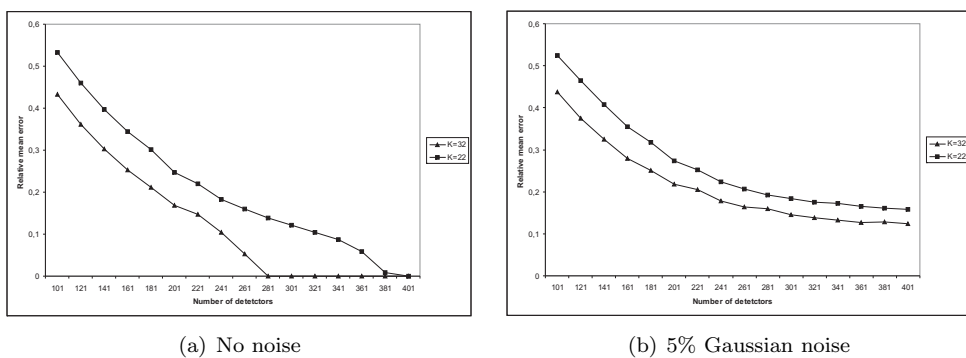
Figure 11: Relative mean error obtained from varying the number of detector elements.

(a) 101 detector elements, no noise

(b) 101 detector elements, 5% Gaussian noise

(c) 281 detector elements, no noise

(d) 281 detector elements, 5% Gaussian noise
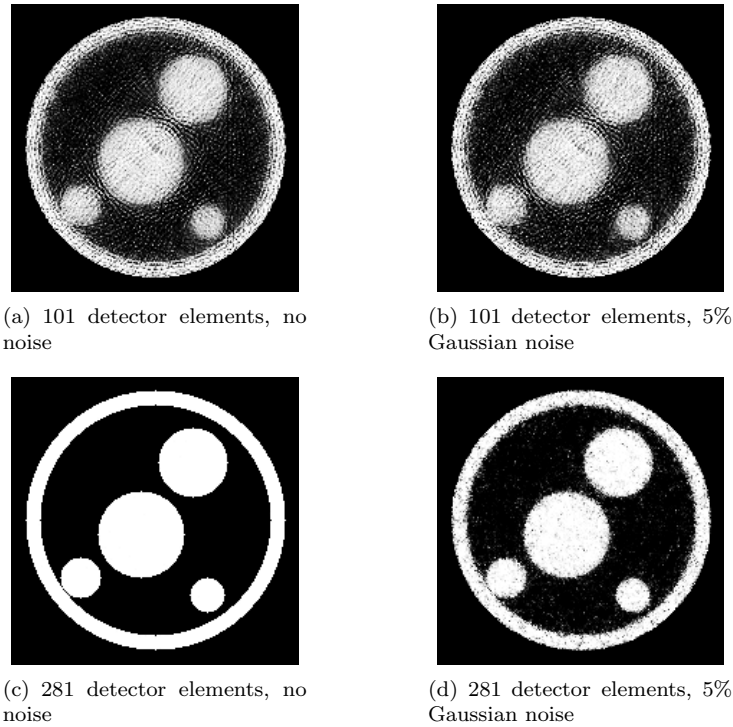
Figure 12: Images reconstructed from projections using a different number of detector elements ($K = 32$).

5.2   The parameters of the SA optimization algorithm

The SA algorithm we implemented has several parameters. We will investigate here what happens when we vary the initial temperature and stopping criteria.

5.2.1   Initial temperature

As is well known, a higher temperature in the SA algorithm may make the optimisation process go in the 'wrong' direction (thus we accept changes with a higher probability when the objective function C increases, so $C(\mathbf{x}') > C(\mathbf{x})$) . If the temperature is very low then changes in the 'wrong' direction have small probabilities. When the initial temperature is high the algorithm in the first iterations can make 'wrong' changes with a higher probability. There is then a chance of making "big jumps", thus it can get near the global minimum. However, an excessively high initial temperature can cause unnecessary iterations during the execution of the algorithm. Thus the parameter setting must be chosen with care.

In our program the initial temperature was varied from 4 to 104 (see Fig. 13).
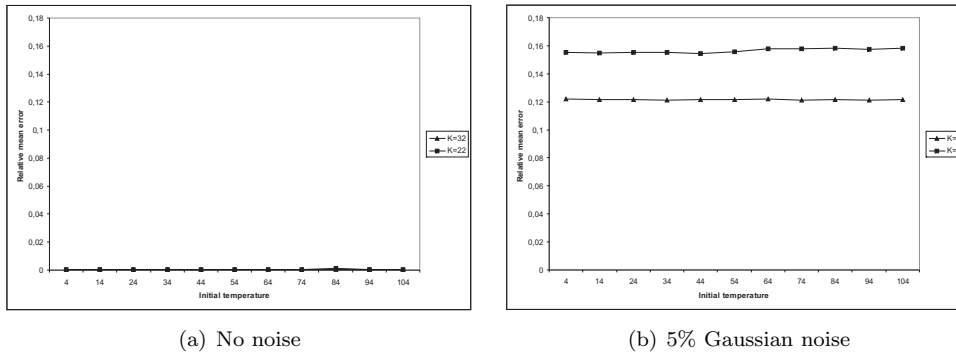
(a) No noise

(b) 5% Gaussian noise

Figure 13: Relative mean error as a function of the initial temperature.

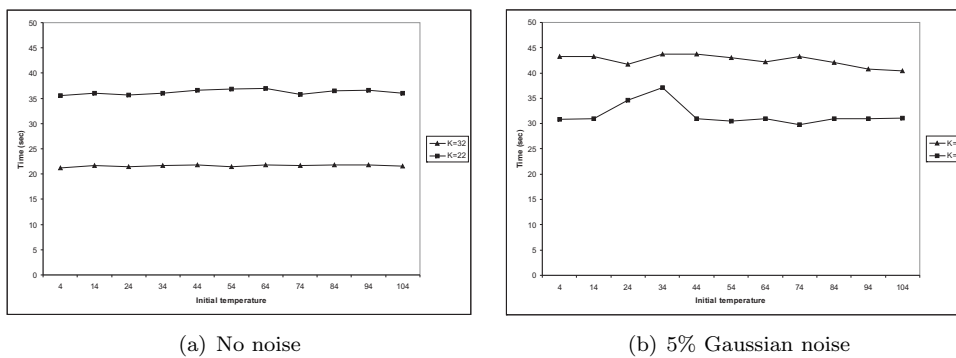(a) No noise

(b) 5% Gaussian noise

Figure 14: Execution time as a function of the initial temperature.

At the base of the curves we can see (see Fig. 13) that the influence of the initial temperature on the relative mean error is very limited. This is because these temperatures provide enough freedom for the SA method to find the global minimum. The curves of the execution times (see Fig. 14) do not show any major differences with a higher initial temperature. It is noticeable, however, that when there was no noise added, the algorithm when $K = 22$ case took more time to reconstruct the phantom image than when $K = 32$ because the fewer number of equation meant a less determined system of equations with, perhaps, more local minima. The results show the opposite case when 5% Gaussian noise was added to the projections. The explanation might be that when noise was added, the $K = 32$ case had more equations and more local minima than the $K = 22$ case.
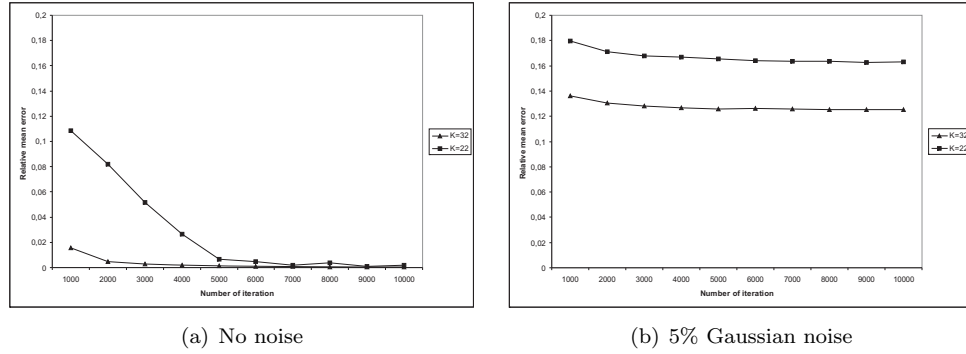
(a) No noise   (b) 5% Gaussian noise

Figure 15: Relative mean error as a function of the number of iterations in the stopping criteria ($R_{\mathrm{thr}} = N_{\mathrm{iter}} - 10$ and $N_{\mathrm{var}} = N_{\mathrm{iter}}/2$).
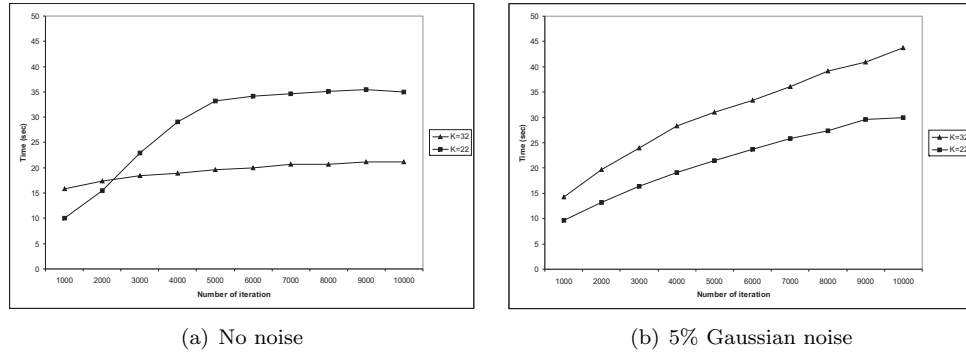

(a) No noise   (b) 5% Gaussian noise

Figure 16: Execution time as a function of number of iterations in the stopping criteria ($R_{\mathrm{thr}} = N_{\mathrm{iter}} - 10$ and $N_{\mathrm{var}} = N_{\mathrm{iter}}/2$).

5.2.2 Stopping criteria

The stopping criteria in our SA algorithm is determined by counting the rejected iterations within the last N_{iter} number of iterations. If this number is greater than a given threshold (R_{thr}) then we say that the *efficiency* is low, because too few changes were accepted.

We investigated the effects of varying these two parameters separately. First, we changed N_{iter} and kept the threshold number R_{thr} high (see Figs. 15 and 16). The estimate for the variance of the objective function was calculated from the last $N_{\mathrm{var}} = N_{\mathrm{iter}}/2$ number of accepted changes. Second, we fixed the number of iterations and increased the efficiency (i.e. R_{thr}, the number of rejected iterations). The results are displayed in Figs. 17 and 18.

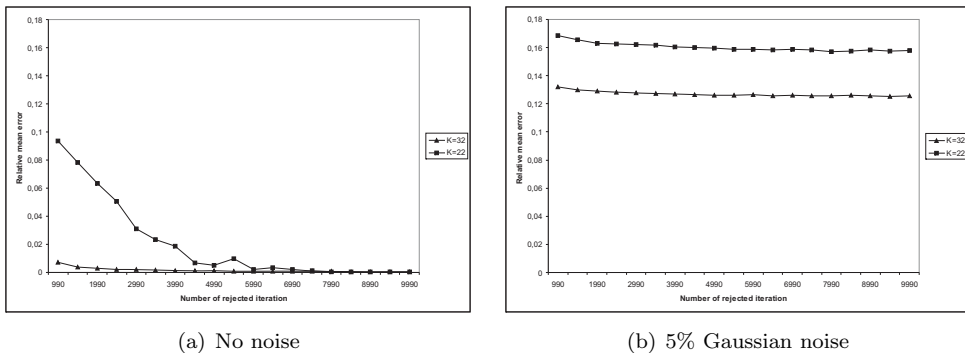The relative mean error showed a downward trend in both experiments owing

(a) No noise

(b) 5% Gaussian noise

Figure 17: Relative mean error as a function of the number of rejected iterations in the stopping criteria ($N_{\text{iter}} = 10000$ and $N_{\text{var}} = 5000$).
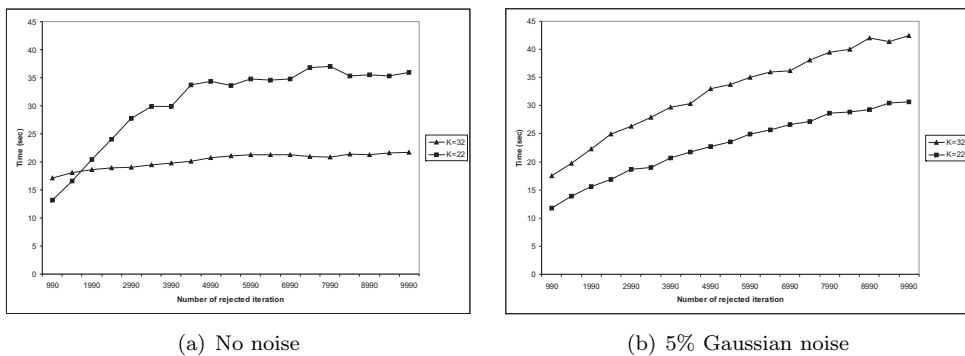

(a) No noise

(b) 5% Gaussian noise

Figure 18: Execution time as a function of the number of rejected iterations in the stopping criteria ($N_{\text{iter}} = 10000$ and $N_{\text{var}} = 5000$).

to the behaviour of the algorithm used. The execution time, in contrast, showed a rising trend. From these findings we may conclude that a greater efficiency produces better results, but at a price.

5.3 Complexity

Here we generated 10 different software phantoms. These phantoms had 1, 2,..., 10 small circles inside a big ring of a given size (see Fig. 19). The experiment was repeated 100 times for each software phantom. The results of these tests are displayed in Fig. 20. The curves here clearly show that the situation is quite different for the noise-free and noisy projections. If the projections are noiseless and $K = 22$, more complex images can be reconstructed but these will have a higher error. It seems that when $K = 32$ this many sources gives a sufficient

number of equations to enable us to reconstruct complex images like the ones shown. When Gaussian noise was afterwards added to the projections the quality of the reconstruction did not change markedly. The latter was anticipated because the objective function then has more local minima than it does for the noiseless case.
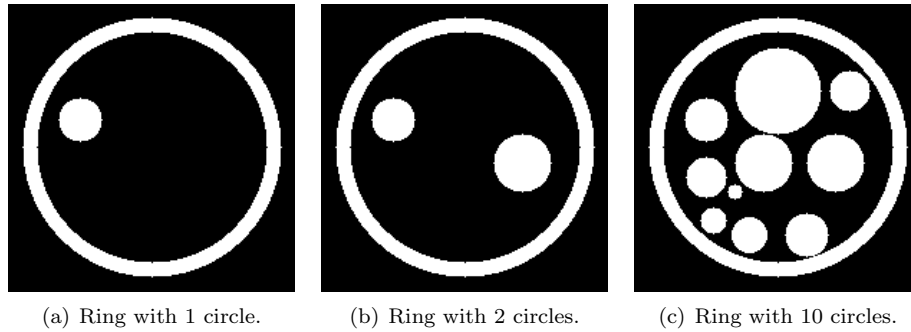

(a) Ring with 1 circle.   (b) Ring with 2 circles.   (c) Ring with 10 circles.

Figure 19: Software phantom images made for testing the complexity.

(a) No noise   (b) 5% Gaussian noise
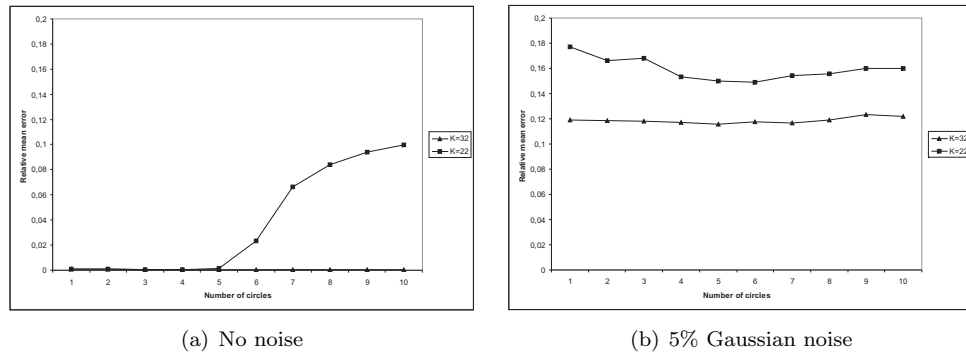
Figure 20: Relative mean error as a function of the number of circles (complexity of the image).

5.4   The regularization parameter

In the previous cases the regularization parameter γ in (4) was assigned a value of 145. Varying the value of γ from 0 to 145 the second term becomes more important. In the experiments we used a special kind of function for $\Phi(x)$, namely

$$\Phi(x) = \Phi_{\mathrm{poz}}(x) = \sum_{j=0}^{nm-1} poz(f_j - f_j^{(0)}), \qquad (7)$$

where *poz* denotes the positive part of y. Formally,

$$poz(y) = \begin{cases} y, \text{ if } y > 0 \\ 0, \text{ otherwise,} \end{cases} \tag{8}$$

and $f_j^{(0)}$ is a so-called *prototype function*. For the phantom shown in Fig. 21(a), the prototype function $f^{(0)}$ was the mask in Fig. 21(b). The regularization parameter and penalty term bias the algorithm according to the mask function. Optimizing the objective function we get a result like this, which will lie inside the given mask.

We carried out experiments where we varied the regularization parameter γ in Φ_{poz} (see Figs. 22 and 23). We got a qualitative improvement in the noisy case using the Φ_{poz} penalty term and increasing the value of the regularization parameter (see Fig. 22(b)). We also obtained good results when we had $K = 22$ and no noise was added to the projections (see Fig. 22(a)). This is because a pixel did not change outside the $f^{(0)}$ mask, and the $\Phi_{\text{poz}}(x)$ regularization tag penalised the objective function when it did so.
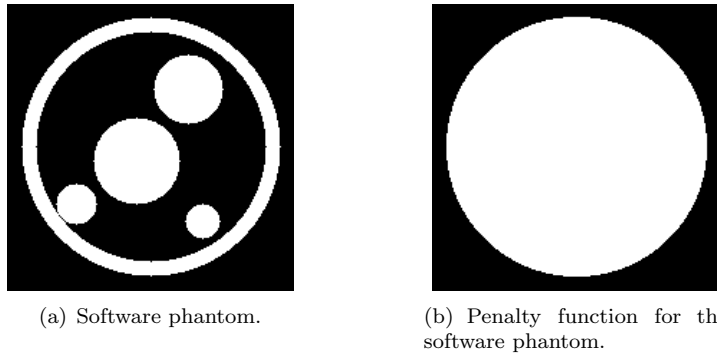

(a) Software phantom.

(b) Penalty function for the software phantom.

Figure 21: Images of the software phantom and the penalty function.

(a) No noise

(b) 5% Gaussian noise
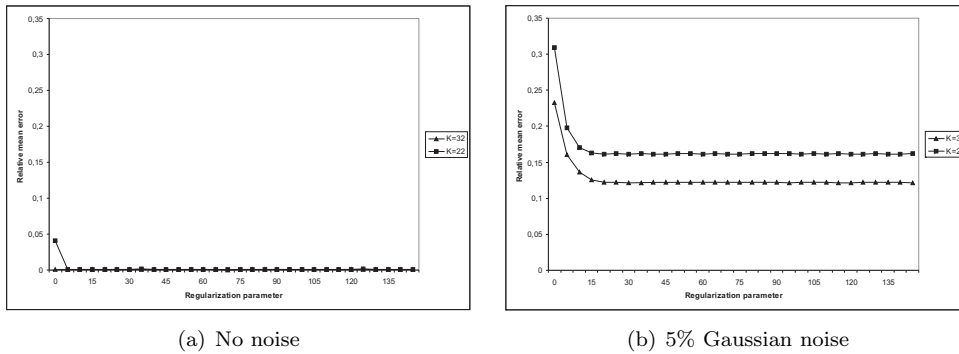
Figure 22: Relative mean error as a function of the regularization parameter (Φ_{poz}).

(a) No noise

(b) 5% Gaussian noise
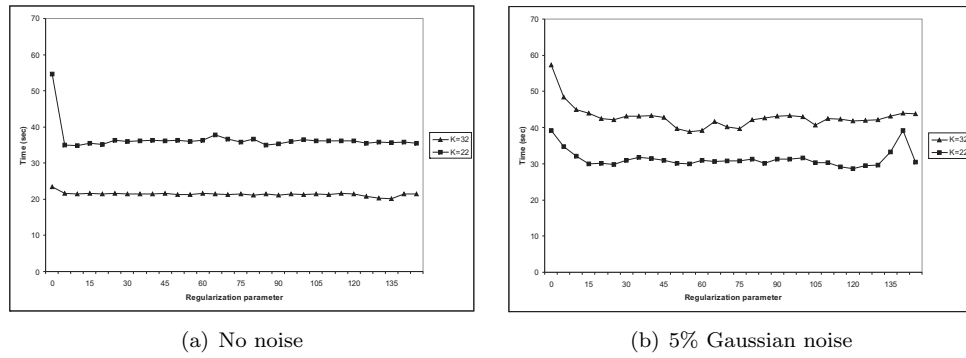
Figure 23: Execution time as a function of the regularization parameter (Φ_{poz}).

(a) Gamma = 0, no noise.   (b) Gamma = 20, no noise.   (c) Gamma = 145, no noise.

(d) Gamma = 0, 5% Gaussian noise.   (e) Gamma = 20, 5% Gaussian noise.   (f) Gamma = 145, 5% Gaussian noise.
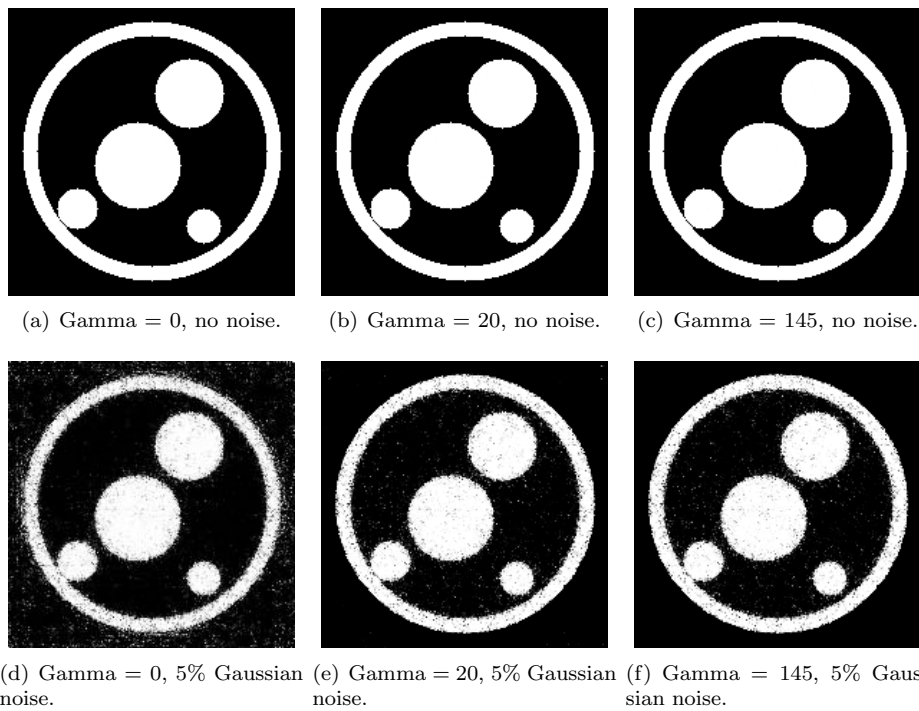
Figure 24: Images obtained when varying the Φ_{poz} regularization parameter ($K = 32$).

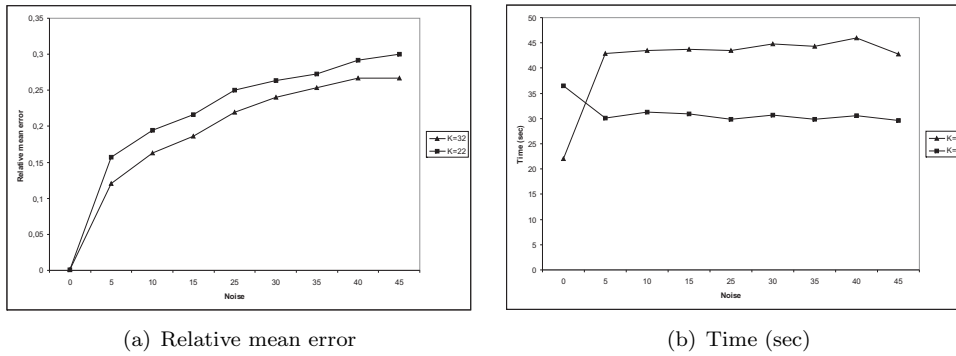(a) Relative mean error   (b) Time (sec)

Figure 25: The effect of changing the noise ratio.

5.5 Noise

The noise ratio η in our experiments was varied from 0% to 45%. Once again we focused on the relative mean error (see Fig. 25(a)) and the time (see Fig. 25(b)). Increasing the amount of noise in the projections we got worse results, as expected (see Fig. 25(a)). The algorithm was found to halt in the $K = 22$ case when we had fewer equations and more noise. The reason for this is the number of equations needed. If we have fewer equations the system is less determined in the noiseless case. In the $K = 32$ case when we add noise the number of local minima seems to be more than that for the $K = 22$ case.

6 Discussion and Conclusions

Now we will summarise the results of the previous sections. First, we studied the effects of varying the parameters of the fan-beam projections and the SA reconstruction algorithm with binary-valued matrices. With binary-valued phantoms we carried out experiments to study the effects of varying parameters on the quality of the reconstructed objects. In each test only one parameter was varied. The tests were repeated 100 times to get a better approximation for the relative mean error and for the execution time.

From our findings it is apparent that, when using the SA method, there is no major difference between reconstructing an image from a fan-beam and reconstructing one from parallel-beam projections. We saw this when we varied the distance between the sources and the origin in the experiments (see Fig. 6).

Changing the start angle when the source number was 4 yielded better results only in certain special source positions (see Fig. 7). When the number of sources (see Fig. 9) and number of detector elements (see Fig. 11) were varied in the experiments, we observed big variations in the quality of the reconstruction.

Upon increasing the initial temperature we did not, as expected, notice any

big difference between the results (see Fig. 13). Varying the stopping criteria we found that good reconstruction results could be obtained but they required longer execution times (see Figs. 15, 17, 16 and 18). The main task here, however, was to determine the stopping criteria needed to produce good reconstruction images with an acceptable execution time. We found such criteria for our particular software phantom and we expect that there should be similar criteria with other phantoms.

The results of the complexity test in Section 5.3 show that, when K was changed from 22 to 32, there was no great change in the relative mean error when noise was present (see Fig. 20(b)). In this test we found we needed a feasible number of equations (number of sources or number of detectors) when no noise was added to the projections (see Fig. 20(a)). This number was found to depend on the geometrical complexity of the phantom.

In Section 5.4 we noticed that, beyond a certain point, varying the regularization parameter did not yield better results (see Fig 22). This value was lower for the noiseless case than for the noisy case and it was found to depend on the number of equations used in the reconstruction process.

The simulated experiment in Section 5.5 with noise shows that when we have more information and noisy projections, it usually requires more time to reconstruct the object from these projections (see Fig. 25). This is because the equation system is less determined in the noisy case.

Overall, the study revealed that the number-of-equations parameter is strongly related to other parameters as well like the stopping criteria, regularization parameter and complexity of the phantom. We obtained similar results with different parameter settings and found the execution time was also an important factor in the reconstruction process. A rehearsal of the experiments will, of course, be necessary for reconstructing real objects with well-defined, measurable parameters.

Acknowledgments

References

[1] G.T. Herman, Image Reconstruction from Projections, *Academic Press*, New York, (1980).

[2] G.T. Herman and A. Kuba, Discrete Tomography. Foundations, Algorithms, and Applications, *Birkhauser*, Boston, (1999).

[3] R.A. Brualdi, *Matrices of zeros and ones with fixed row and column sum vectors*, Linear Algebra AppI. 33 (1980) 159-231.

[4] A. Kuba, G.T. Herman, *A historical introduction*, in [2], (1999).

[5] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, New York, NY: IEEE Press, Inc., (1988).

[6] P. Grangeat, *Mathematical framework of cone beam 3D reconstruction via the first derivative of the Radon transform*, Mathematical Methods in Tomography, Lecture notes in Mathematics, eds. G.T. Herman, A.K. Louis, and F. Natterer, 1497 (1991) 66 – 97.

[7] N. Robert, F. Peyrin, M. J. Yaffe, *Binary vascular reconstruction from a limited number of cone beam projections*, Med. Phys. 21, (1994) 1839 – 1851.

[8] B. Chalmond, F. Coldefy, B. Lavayssiere, *Tomographic reconstruction from non-calibrated noisy projections in non-destructive evaluation*, Inverse Problems 15, (1999) 399 – 411.

[9] A. Kuba, L. Ruskó, L. Rodek, Z. Kiss: *Preliminary results of Discrete Tomography in Neutron Imaging*, in press *IEEE Trans. Nucl. Sci.* (2005).

[10] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, *Equation of state calculation by fast computing machines*, J. Chem. Phys. 21 (1953) 1087 – 1092.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by simulated annealing*, Science 220 (1983) 671 – 680.