Fundamenta Informaticae 135 (2014) 21–42
DOI 10.3233/FI-2014-1110
IOS Press

Aligning Projection Images from Binary Volumes

Folkert Bleichrodt[*]
Centrum Wiskunde & Informatica
1098 XG Amsterdam, The Netherlands
F.Bleichrodt@cwi.nl

Jan De Beenhouwer
iMinds - Vision Lab, University of Antwerp
B-2610 Wilrijk, Belgium
jan.debeenhouwer@uantwerpen.be

Jan Sijbers
iMinds - Vision Lab, University of Antwerp
B-2610 Wilrijk, Belgium
jan.sijbers@uantwerpen.be

Kees Joost Batenburg
Centrum Wiskunde & Informatica
1098 XG Amsterdam, The Netherlands
K.J.Batenburg@cwi.nl

Abstract. In tomography, slight differences between the geometry of the scanner hardware and the geometric model used in the reconstruction lead to alignment artifacts. To exploit high-resolution detectors used in many applications of tomography, alignment of the projection data is essential.

Markerless alignment algorithms are the preferred choice over alignment with markers, in case a fully automatic tomography pipeline is required. Moreover, marker based alignment is often not feasible or even possible. At the same time, markerless alignment methods often fail in scenarios where only a small number of projections are available. In this case, the angular separation between projection images is large and therefore the correlation between them is low. This is a property that most markerless algorithms rely on. The intermediate reconstruction problem of alignment by projection matching is highly underdetermined in the limited data case. Therefore, we propose a projection matching method that incorporates prior knowledge of the ground truth. We focus on reconstructing binary volumes. A discrete tomography algorithm is employed to generate intermediate reconstructions. This type of reconstruction algorithm does not rely heavily on correlated projection images. Our numerical results suggest that alignment using discrete tomography projection matching produces much better results in the limited angle case, when compared to a projection matching method that employs an algebraic reconstruction method.

Keywords: alignment, automatic, binary tomography, prior knowledge

[*]Address for correspondence: P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

1. Introduction

Tomography deals with the reconstruction of an object based on projections; see Fig. 1. Projection images are acquired by scanning devices, such as X-ray based medical scanners or transmission electron microscopes [6, 15]. For high resolution microtomography or nanoscale imaging, the stability of the scanner hardware is a limiting factor in the reconstruction quality [16]. Motion of the object or limited accuracy of the mechanics leads to unaligned projection images that produce *alignment* artifacts in the reconstruction. Algorithms for aligning the projection images are essential to fully exploit modern detectors with high pixel density.

In electron tomography, the projection images are created using a beam of electrons. The instability of the sample holder in the electron microscope and technical limitations can lead to severe distortions in the geometry. Especially in the position of the object [12].

Another area in which alignment is important is in a synchrotron setup. A synchrotron produces monochromatic X-rays using a particle accelerator. This results in high resolution projection images that do not suffer from beam hardening. However, the instruments used in the projection acquisition are often not completely aligned before starting the experiment, resulting in the need for post-acquisition alignment. Also, during in-situ experiments it is challenging to keep the scanner setup in perfect alignment [25].

Current alignment methods are based on tracking of fixated markers, or are purely data-driven using a (markerless) projection dataset of the object [7, 9, 11]. The latter *markerless* methods are often used in applications at nanoscales, or other domains where using markers is not possible or not feasible. In many cases, a variant of cross-correlation techniques is employed. The cross-correlation between consecutive images with low angular separation can be exploited to estimate in-plane transformations of the projection images. In *projection matching* methods, an intermediate reconstruction is formed. By generating forward projections of the reconstructions and comparing them with the observed projections, it is possible to refine the alignment [22]. However, to obtain a reasonably accurate reconstruction, a relatively large number of projection images is required.

In recent years, a substantial number of publications have appeared about image reconstruction from highly limited data, in the fields of discrete tomography (DT) and compressive sensing (CS) [1, 3, 8, 23]. In discrete tomography, the unknown object is known to consist of just a few materials. Therefore, the number of grey values corresponding to these materials is small and possibly known in advance. This information can be used as prior knowledge in the reconstruction algorithm, to limit the solution space and yield more accurate reconstructions. In compressive sensing, sparseness of the total variation of the object (*i.e.* the ℓ_1 norm of the gradient image) can be exploited to reconstruct the object from a few projection images.

In this paper, we focus on DT, but the same concepts can be applied to CS techniques as well. To apply discrete tomography effectively, an aligned dataset is needed, as well as a good estimation of the grey levels. Methods have been proposed in the literature for grey level estimation [2], but for alignment, using markers is the only option. All of the markerless methods require a substantial number of projection images to perform well:

- for methods that exploit similarity between projections, the angular gap cannot be too large;
- for projection matching, to generate a decent reconstruction, enough projections should be available.

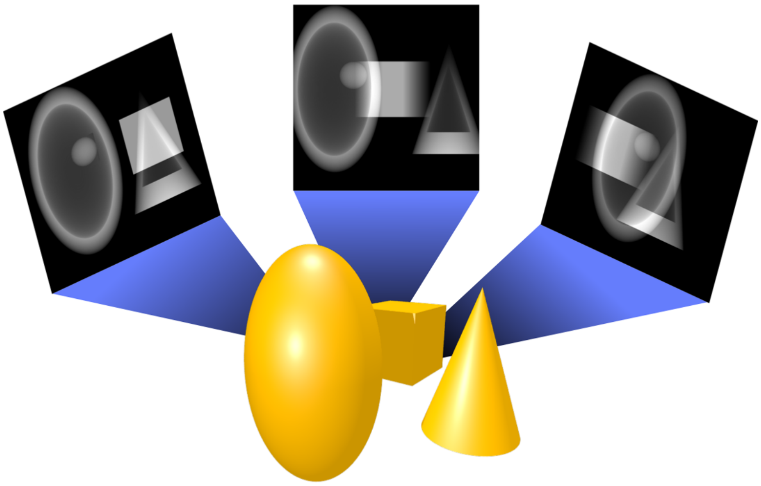


Figure 1: Acquisition of projection images from several angles. This figure shows the object with unknown interior and three of its projection images.

In this paper, we propose the Discrete Tomography Projection Matching (DTPM) method. DTPM incorporates a discrete tomography reconstruction algorithm in the projection matching alignment. We demonstrate with simulation experiments, that using this prior knowledge in the alignment phase, a substantially better alignment can be obtained. In comparison to standard projection matching methods, the DTPM alignment results in more accurate reconstructions.

The structure of this paper is as follows: in Section 2 the acquisition geometry for parallel beam tomography is described in detail. Here we introduce the geometrical parameters that are estimated by our alignment algorithm. We also give a formal, mathematical introduction to tomographic reconstruction. We show how the geometric parameters are included in the reconstruction and we define alignment by projection matching. A short description of discrete tomography is presented as well. In Section 3 we introduce our proposed DTPM alignment method. We show how discrete tomography is incorporated in the projection matching method. In Section 4 we describe a series of simulation experiments that were performed, also on noisy data, and the results are discussed. In Section 5 we discuss considerations that should be taken into account when applying our algorithm on data from real experiments. Finally, we conclude this paper in Section 6.

2. Method and Implementation

In this section we will explain the geometrical parameters that determine the projection geometry. We focus on the parallel beam geometry, illustrated in Fig. 2, but the method can be extended to other geometries as well. Furthermore, the mathematical background and algorithm details are given for the alignment method that estimates these parameters.

2.1. Geometrical Parameters

A typical setup for projection acquisition in tomography is illustrated in Fig. 2. We assume that the radiation source emits parallel beams that are perpendicular to the detector plane. The object is positioned at the origin of the x-y-z coordinate system and the z-axis indicates the rotation axis. The projection angle θ gives the rotation of the object around the z-axis. Note that rotation of the source and detector around the object is equivalent to rotation of the object itself (except for the sign). In our alignment algorithm, we do not align the projection angle θ, because we want to consider the same parameters that are typically used in image registration techniques.

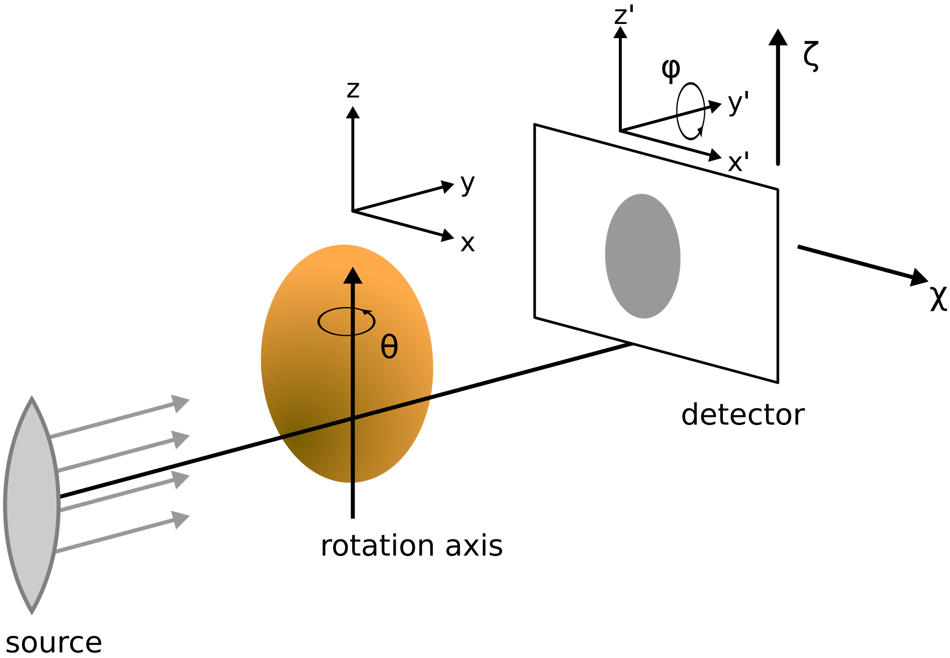


Figure 2: Overview of the geometrical parameters describing the geometry of the projection acquisition. The object in the middle is projected onto the detector on the right. The parallel rays are perpendicular to the detector.

To model shift and rotation in the projection domain (*i.e.* in the plane of the detector, both of the object as well as the detector itself) we define a second coordinate system x', y', z' that initially is aligned with the x, y, z coordinate system. The origin of the x', y', z' coordinates indicates the center of the detector. For the projection angle $\theta = 0$, the horizontal axis of the detector is aligned with the x'-axis and its vertical axis with the z'-axis. An in-plane shift of the detector (or of the object) can then be modeled by a detector shift of $(\chi, 0, \zeta)$ in the x', y', z' system. Note that the projection images are spanned in the x'-z'-plane. Therefore, the shift can be considered as a 2D vector in the x'-z'-plane. Since we consider a parallel beam geometry, object motion in the y'-direction does not change projections.

To model in-plane rotation of the detector or object, we introduce the parameter ϕ. The detector is rotated around the y'-axis over an angle ϕ. The coordinate system of the detector and that of the object are then related by the following coordinate transformations:

$$\begin{aligned} x' &= (x+\chi)\cos\phi + (z+\zeta)\sin\phi, \\ y' &= y, \\ z' &= -(x+\chi)\sin\phi + (z+\zeta)\cos\phi, \end{aligned} \tag{1}$$

So the coordinates are coupled by a translation and rotation operator.

Four parameters are sufficient to define the geometry of the acquisition of one projection image. In a tomography scanner, projection images are recorded at multiple angles,

$$\theta_1, \theta_2, \ldots, \theta_K \in V \subset [0, 2\pi). \tag{2}$$

In general, most datasets contain projection images with corresponding projection angles in the range $V = [0, \pi)$. For each projection image we have an in-plane shift,

$$(\chi_1, \zeta_1), (\chi_2, \zeta_2), \ldots, (\chi_K, \zeta_K), \tag{3}$$

and an in-plane rotation,

$$\phi_1, \phi_2, \ldots, \phi_K. \tag{4}$$

In markerless alignment methods based on image registration techniques, it is very common to estimate the parameters $\boldsymbol{\chi}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\phi}$, since these parameters act in the plane of the detector. For this reason, we also consider the parameters $\boldsymbol{\chi}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\phi}$ in our alignment method and do not estimate the other parameters. We note that the approach presented here can be extended to include other alignment parameters as well.

In the case of a perfectly aligned scanner setup that matches our theoretic model of the geometry, we would have $\chi_k = \zeta_k = \phi_k \equiv 0$ and the projection angles are known exactly. In reality, there is a shift in the order of 1-10 detector pixels and an in-plane rotation of several degrees. To some extent, the projection angles are known with limited accuracy.

These perturbations in the geometry are mainly caused by

- **Calibration errors** - If the rotation axis and the center of the detector are not aligned, a structural shift or in-plane rotation is introduced. This can be prevented by calibrating the hardware precisely. Calibrating, however, is a time consuming task and difficult if the pixel size of the detector is very small.

- **Mechanical inaccuracies** - The limited accuracy of the mechanics, such as the goniometer (the motor that selects the projection angle), is another source of misalignment.

- **Random motion** - Since the object is not completely fixed, some motion may occur when the goniometer rotates the object to the next projection angle. Also, the object might be moving while scanning, for example if a microscopy sample drifts within the sample holder.

For common reconstruction algorithms, it is assumed that no perturbations in the geometry are present and the projection angles $\tilde{\theta}_i$ are known exactly. The geometry that they impose is given by the set of parameters:

$$\left.\begin{aligned} \chi_i &= 0 \\ \zeta_i &= 0 \\ \phi_i &= 0 \\ \theta_i &= \tilde{\theta}_i \end{aligned}\right\} \quad i = 1, \ldots, K. \tag{5}$$

In reality, small perturbations are common even in the projection angles. Therefore, the geometry of the reconstruction method needs to be adjusted, to match the true geometry of the projection acquisition. For this purpose we need to introduce a mathematical framework that allows us to incorporate the geometry in the reconstruction. In the next section, we introduce this mathematical framework and discuss how we can incorporate the alignment in the reconstruction method.

2.2. Mathematical Formulation

The object can be related to its projections in a continuous manner by means of the Radon transform [18]. To solve the tomographic reconstruction problem, we define a discrete model of the Radon transform. The object is defined on a grid of unit cubes called *voxels*. Each voxel is assumed to have a constant grey value x_j. These are stored as a vector $\boldsymbol{x} \in \mathbb{R}^N$. The grey value is proportional to the attenuation coefficient of the corresponding material. The projection domain, consisting of a series of projection images, is discretized into a series of pixel grids. A detector value p_i is modeled as a weighted sum (a single line projection) of the object grey values

$$p_i := \sum_{j=1}^{N} w_{ij} x_j. \tag{6}$$

The weight w_{ij} models the attenuation of the source ray i caused by the material within voxel j. Different weight models exist that approximate the physical interaction of the radiation source with the object. In Figure 3, as an example the strip model is illustrated.

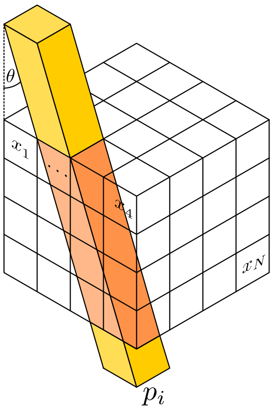


Figure 3: Discretization of the forward projection in three dimensions. The projection p_i is a weighted sum of the grey values. The weight w_{ij} of voxel j is determined by the intersection volume of ray i and the voxel. This is the so-called strip model. The ray is indicated in yellow.

We can now describe the relation between the measurements $\boldsymbol{p}$ and the unknown object $\boldsymbol{x}$ by a system of linear equations

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}. \tag{7}$$

The projection matrix $\boldsymbol{W}$ has dimensions $M \times N$, the line projections $\boldsymbol{p} \in \mathbb{R}^M$ and the object $\boldsymbol{x} \in \mathbb{R}^N$ are stored as column vectors. The number of projection images is K and the number of detector pixels is D, so the number of rows in $\boldsymbol{W}$ is given by $M = KD$.

The weights w_{ij} are not only determined by the ray width or voxel size, but they also depend on the geometry. This can be seen in Figure 3. Hence, by writing the *geometrical parameters* as vectors,

$$\begin{aligned}\boldsymbol{\theta} &= (\theta_1, \dots, \theta_K)^T,\\ \boldsymbol{\chi} &= (\chi_1, \dots, \chi_K)^T,\\ \boldsymbol{\zeta} &= (\zeta_1, \dots, \zeta_K)^T,\\ \boldsymbol{\phi} &= (\phi_1, \dots, \phi_K)^T,\end{aligned}$$

the dependency of the projection operator on the geometry is posed as a nonlinear equation system

$$\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} = \boldsymbol{p}. \tag{8}$$

In practice, it is not straightforward to use a closed-form expression for the weights w_{ij} as function of the geometrical parameters. They are generated on the fly based on the geometrical parameters by means of a ray tracing type algorithm. More details of this operation are discussed in Section 3.

If the unknown object can be represented exactly on a voxel grid, the system is consistent if the geometrical parameters are perfectly aligned and when the projection images do not contain noise. This enables us to formulate an alignment method as an algorithm for minimizing the inconsistency of Eq. (8):

$$\underset{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\chi},\boldsymbol{\zeta},\boldsymbol{\phi}}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{p}\|^2. \tag{9}$$

The factor $\frac{1}{2}$ is introduced to avoid a factor two in the gradient of this cost function and simplify the notation. There are many approaches to solve this optimization problem. Minimization with respect to $\boldsymbol{x}$ is a linear optimization problem, which is highly underdetermined. The dependency on the projection angles $\boldsymbol{\theta}$ makes Eq. (9) also a nonlinear problem. In addition to the large scale of the data, it is a very difficult optimization problem. However, by using the right approach, it is typically possible to solve this problem accurately. In the next section we will discuss in detail the approach of projection matching.

2.3. Projection Matching

In this section we will explain the projection matching method used for the alignment.

Minimizing the projection distance defined in Eq. (9) can be done in several ways. The most important goal is to find an accurate reconstruction $\boldsymbol{x}$. Without an accurate geometrical parameter set, this is not possible. Moreover, to estimate the geometrical parameters, a reasonable reconstruction should be available. It seems that this problem can only be solved effectively by considering the full optimization problem Eq. (9) and estimate $\boldsymbol{x}$ and the geometrical parameters simultaneously. This approach has been proposed in [5, 26]. Since the inverse problem in Eq. (8) is heavily underdetermined (*e.g.*, $N = 512 \times 512 \times 512$ and $M = 10 \times 512 \times 512$), ill-posed and nonlinear, this is a difficult task.

As an alternative approach, we can alternate between reconstruction and alignment. In this case, we treat both problems individually. During the reconstruction, we assume that the geometry is known, and during alignment, the reconstruction is fixed,

$$\text{Reconstruction}: \quad \underset{\boldsymbol{x}}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{p}\|^2, \tag{10a}$$

$$\text{Alignment}: \quad \underset{\boldsymbol{\theta},\boldsymbol{\chi},\boldsymbol{\zeta},\boldsymbol{\phi}}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{p}\|^2. \tag{10b}$$

Some algebraic reconstruction methods specifically solve Eq. (10a) in a certain norm. An example is the Simultaneous Iterative Reconstruction Technique (SIRT) [24]. Therefore, Eq. (10a) can be solved by employing a suitable reconstruction algorithm. In the second step, generated projections from the reconstruction using the forward model in Eq. (8), are matched to the observed projections. This subproblem only has a solution when the difference between the ground truth and reconstruction is in the null space of the projection operator $\boldsymbol{W}$, which is normally not the case. Nevertheless, given a reconstruction containing some details or crude outlines of the object, it may be possible to improve the alignment parameters. The resulting family of alignment methods is termed *projection matching*. Examples of these methods are discussed in [12, 22]. All of these algorithms use an intermediate reconstruction step and afterwards apply a form of alignment that is often based on image registration of projection images and forward projections of the reconstruction. The basic structure of these methods is illustrated in a flowchart in Fig. 4. In the experiment and results section, we compare our method with a standard projection matching algorithm that incorporates SIRT as the reconstruction method. We will refer to this method as PM-SIRT.

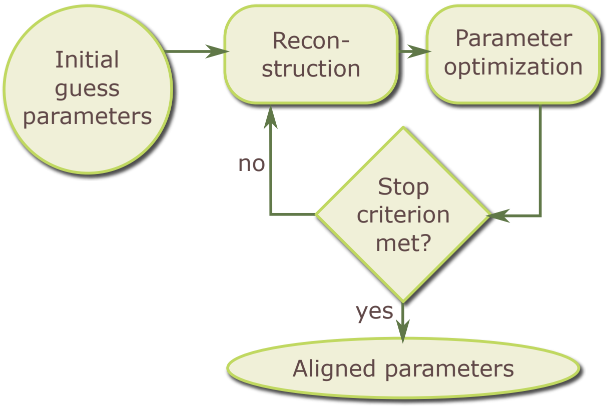


Figure 4: A flowchart of the projection matching algorithm that estimates the geometrical parameters.

2.4. Discrete Tomography

Many objects that are scanned in tomography consist of just a few materials. Each material has a corresponding, uniform grey value in the reconstruction. For example, when scanning bones (ex vivo), the reconstruction has two grey values, one for the bone and one for the background (assuming that the bone is approximately homogeneous). In this case, we would like to reconstruct a binary volume. The discrete nature of the grey values can be exploited in the reconstruction. Instead of solving Eq. (9) over a continuous domain, we can limit the domain of $\boldsymbol{x}$ to the set of grey values that are known in advance. Let $R = \{\rho_1, \ldots, \rho_l\}$ be the set of grey values in the ground truth image. The reconstruction problem that is considered in *discrete tomography* is posed as

$$\underset{\boldsymbol{x} \in \{\rho_1, \ldots, \rho_l\}^N}{\text{minimize}} \ \|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{p}\|^2 \tag{11}$$

In the experiment section, we consider projection datasets from an object with few grey values in the reconstruction (in this case two, these objects are called binary volumes). This allows us to use reconstruction algorithms for discrete tomography to approximately solve Eq. (11). Possible candidates for the reconstruction algorithm are described in [3, 4, 23].

3. Projection Matching with Discrete Tomography

To our knowledge, employing a discrete tomography prior in projection matching has never been proposed before. As discussed previously, all common markerless alignment methods are not as accurate when the number of projection images is small. By applying a discrete tomography reconstruction algorithm we can introduce prior knowledge in a projection matching method, to alleviate this limitation.

Essentially, the reconstruction phase Eq. (10a) in the projection matching algorithm is performed by a discrete tomography reconstruction algorithm that solves Eq. (11). We employ the binary reconstruction algorithm proposed by Schüle *et al.*, from now on referred to as DC [23]. This algorithm solves Eq. (10a) augmented with two priors using D.C. programming:

$$\underset{\boldsymbol{x}\in[0,1]^N}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{W}\boldsymbol{x}-\boldsymbol{p}\|^2+\alpha\boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x}+\frac{1}{2}\mu\langle\boldsymbol{x},\boldsymbol{e}-\boldsymbol{x}\rangle,\quad 0<\mu\in\mathbb{R}. \tag{12}$$

Here $\boldsymbol{e} := (1,\ldots,1)$. The middle term of Eq. (12) is a smoothness prior. The final term steers the solution to a binary volume. The parameters α and μ express weights to these terms.

From just a few projection images it is possible to make a perfect reconstruction, under some smoothness conditions on the ground truth [23]. It has not been investigated how well this algorithm performs in case of misaligned data. However, our numerical results suggest that the reconstructions in the early stages of the alignment algorithm are good enough for achieving convergence in the parameter estimation (10b).

The DT prior that is controlled by the parameter μ gradually forces the reconstruction towards a binary solution. That is, in the implementation μ is gradually increased while the change in the data fidelity term becomes smaller. The final DT reconstruction is approximately binary, but simply rounding by thresholding yields a true binary solution. This allows also the use of other DT reconstruction methods, such as the DART algorithm which also produces a discrete solution [3]. The reason we chose for the DC algorithm by Schulle et al. is that DART is not as robust in case the projections are very noisy.

The alignment of the projection images in the second sub-problem (10b) is solved by employing Levenberg-Marquardt. This method is a gradient-based trust region method [19]. The algorithm computes a sequence $\{\boldsymbol{y}_k\}$ of alignment parameters,

$$\boldsymbol{y}_k=(\boldsymbol{\chi}_k,\boldsymbol{\zeta}_k,\boldsymbol{\phi}_k)^T,\quad k=1,2,\ldots$$

that yield a decreasing cost function value (10b). The alignment parameters are updated by a descent direction $\boldsymbol{\eta}_k$,

$$\boldsymbol{y}_{k+1}=\boldsymbol{y}_k+\boldsymbol{\eta}_k \tag{13}$$

where the descent direction is found by solving the following equation:

$$\left(\boldsymbol{J}_k^T\boldsymbol{J}_k+\lambda_k\boldsymbol{D}^2\right)\boldsymbol{\eta}_k=-\boldsymbol{J}_k^T\boldsymbol{r}_k. \tag{14}$$

The residual $\boldsymbol{r}_k := \boldsymbol{W}(\boldsymbol{\theta},\boldsymbol{\chi}_k,\boldsymbol{\zeta}_k,\boldsymbol{\phi}_k)\boldsymbol{x}_k-\boldsymbol{p}$ has Jacobian $\boldsymbol{J}_k := \nabla_{\boldsymbol{y}_k}\boldsymbol{r}_k$. The descent direction is found by minimizing a second order Taylor approximation of the cost function represented by Eq. (14). The parameter λ_k controls the step size of the weighted descent direction, *i.e.*, it limits $\|\boldsymbol{D}\boldsymbol{\eta}_k\|$. The scaling (diagonal) matrix $\boldsymbol{D}$ is necessary to incorporate the different scales of the geometrical parameters. For example, shifts are in the order of voxels, while the in-plane rotation is in radians and is therefore about two orders of magnitude smaller. In our approach we selected the scalings manually. The scales of shifts $(\boldsymbol{\chi}, \boldsymbol{\zeta})$ are unchanged, *i.e.* $d_{ii} = 1$ for corresponding scales. The rotations $(\boldsymbol{\phi})$ are scaled in units of radians, *i.e.* $d_{ii} \approx \pi/180$. For further details of the Levenberg-Marquardt method we refer the reader to [14, 17].

As noted previously, the projection matrix $\boldsymbol{W}$ is generated on-the-fly. Therefore, the matrix is never fully formed in memory, which enables reconstruction of large datasets. Storing a full matrix is not feasible for many practical applications. The details of the ray tracing method are as follows: for each projection angle, the ray paths incident to each detector pixel are traced through the volume and the weights are measured based on an interpolation method by Joseph [13]. The direction of the rays, and therefore the voxels that are intersected are determined by the alignment parameters in Eq. (8). Since we take this approach for computing a projection, and because representing this matrix in closed form is not practical, we do not use an analytic expression for the Jacobian matrix. Instead, we employ a central finite difference approximation for gradients:

$$\nabla_{\chi_i}\boldsymbol{r} \approx \frac{\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi} + h\hat{\boldsymbol{e}}_i, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi} - h\hat{\boldsymbol{e}}_i, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x}}{2h}, \quad i = 1, \ldots, K \tag{15}$$

where $\hat{\boldsymbol{e}}_i$ is the ith basis vector. The gradients of the other parameters are computed likewise. The coefficients of $\boldsymbol{W}$ are only piecewise continuously differentiable with respect to the alignment parameters. This is due to discretization of the reconstruction volume. However, the experimental results in Section 4 suggest that the finite difference approach is effective and is not hampered by the fact that $\boldsymbol{W}$ is not differentiable in every point.

The step size of the finite difference method h is selected automatically, following the same procedure as proposed in [5]. Here we found that the step size is important for the accuracy of the gradient. If it is too large, the approximation in Eq. (15) is very poor. If the step size is too small, discretization effects and limited precision distort the measurement of the gradient. These effects are visible at fine scales in the cost function as erratic oscillations. The method we employed in [5] compares the cost function in Eq. (10b) to a spline fitted to sample points of the cost function. More sample points are compared to the spline and the relative fit yields a measure for the smoothness of the cost function. A step size h determines the distance between sample points. The smallest step size is selected for which the cost function behaves relatively smooth, to make the gradient more robust against these discretization effects. The selected step size behaves like a constant and does not change much in our pilot experiments. Therefore, it is computed once after the initial reconstruction and is fixed during the rest of the computations. This improves the performance of the method, since computing the step size is not cheap.

Since each parameter affects only detector values at a single projection angle, the Jacobian has the following sparse structure

$$\boldsymbol{J} := \begin{pmatrix} \boldsymbol{g}_1^{\chi} & & & & \boldsymbol{g}_1^{\zeta} & & & & \boldsymbol{g}_1^{\phi} & & & \\ & \boldsymbol{g}_2^{\chi} & & & & \boldsymbol{g}_2^{\zeta} & & & & \boldsymbol{g}_2^{\phi} & & \\ & & \ddots & & & & \ddots & & & & \ddots & \\ & & & \boldsymbol{g}_K^{\chi} & & & & \boldsymbol{g}_K^{\zeta} & & & & \boldsymbol{g}_K^{\phi} \end{pmatrix} \tag{16}$$

with $\boldsymbol{g}_1^{\chi}$ the numerical gradient of $\boldsymbol{r}$ with respect to χ_1. Due to the independence of the parameters between different projection images, these finite differences can be computed very efficiently. It costs only two forward projections for each parameter:

$$\begin{pmatrix} \boldsymbol{g}_1^{\chi} \\ \vdots \\ \boldsymbol{g}_K^{\chi} \end{pmatrix} := (\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi} + h\boldsymbol{e}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi} - h\boldsymbol{e}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x})/(2h)$$

with $\boldsymbol{e} := (1, \ldots, 1)^T$.

The full algorithm is listed in pseudo code in Algorithm 1. The Jacobian is needed to solve Eq. (14). We employ the linear least squares solver LSMR to solve this equation [10]. Note that the Jacobian is generated by computing Eq. (16). In the case we need to multiply the Jacobian with a vector, such as in LSMR, it can be efficiently computed by inner products. Since the Jacobian is constant in LSMR, the descent direction is found efficiently, even though we need to solve a system of equation. When the descent direction $\boldsymbol{\eta}_k$ is found, the value ρ is computed. The numerator represents the decrease in the cost function Eq. (10b). The denominator is the decrease in the quadratic model of the cost function based on a second order Taylor approximation:

$$\frac{1}{2}\|\boldsymbol{r}(\boldsymbol{y}+\boldsymbol{\eta})\|^2 \simeq L(\boldsymbol{\eta}) := \frac{1}{2}\boldsymbol{r}(\boldsymbol{y})^T\boldsymbol{r}(\boldsymbol{y}) + \boldsymbol{\eta}^T\boldsymbol{J}^T\boldsymbol{r} + \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{J}^T\boldsymbol{J}\boldsymbol{\eta}. \tag{17}$$

So, if ρ is strictly positive, we have found a descent direction. If, in addition, the fraction ρ is close to 1 the quadratic model is in good correspondence with the cost function. In that case, the damping parameter λ is decreased, such that a larger step is taken in the next iteration. Note that this parameter limits the ℓ^2 norm of the descent direction. Therefore, the step size is increased only if the quadratic model is accurate. Otherwise the step size is decreased.

4. Experiments and Results

In this section we discuss the results of a series of simulation experiments that have been performed to evaluate the capabilities of the DTPM algorithm, and compare it to projection matching without the use of prior knowledge. Based on a binary volume, we generated projection data and introduced misalignment in the geometry. From this data we can compare the accuracy of the standard projection matching using SIRT (PM-SIRT) as a reconstruction algorithm, with DTPM. Both algorithms are also tested for robustness against noise in the projection data.

The forward model in Eq. (7) can be used to generate projection data from a given volume $\boldsymbol{x} \in \mathbb{R}^N$. For creating forward and back projections as well as the SIRT and DC reconstruction algorithms the ASTRA toolbox was used [20, 21]. MATLAB was used as a scripting language that accesses the underlying GPU code through the C++ mex interface. The hardware used for the simulations was a workstation with Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz and a Geforce GTX 570 GPU.

The phantom we consider is depicted in Figure 5. It consists of the union and differences of several convex shapes. All objects are hollow, except for the cube, and the ellipsoid encloses a solid sphere. These simple shapes allow reconstructions from just a few projection images. In MATLAB, we generated the phantom on a $64 \times 64 \times 64$ voxel grid. The middle slice of the object is shown in Fig. 8a.

Algorithm 1 DTPM

Input: Projection data $\boldsymbol{p}$, initial geometry $\bar{\boldsymbol{y}}_0 = (\boldsymbol{\chi}_0, \boldsymbol{\zeta}_0, \boldsymbol{\phi}_0)^T$
Output: A binary reconstruction $\boldsymbol{x}_{i_{\max}}$ and aligned geometry $\bar{\boldsymbol{y}}_{i_{\max}}$

Let $\mathcal{DC}[\boldsymbol{y}](\boldsymbol{p})$ denote the operator that produces a binary DC reconstruction operating on $\boldsymbol{p}$, for a given geometry $\boldsymbol{y}$.
The parameter ϵ controls the convergence criterion, that monitors the relative change in the parameters.

```
for $i = 0, \ldots, i_{\max}$ do

    reconstruction phase
    $\boldsymbol{x}_i = \mathcal{DC}[\bar{\boldsymbol{y}}_i](\boldsymbol{p})$

    alignment phase, using Levenberg-Marquardt
    $\lambda = 5$
    $\nu = 2$
    $\boldsymbol{y}_0 = \bar{\boldsymbol{y}}_i$
    for $k = 0, \ldots, k_{\max}$ do
        $\boldsymbol{r}_k := \boldsymbol{W}(\boldsymbol{y}_k)\boldsymbol{x}_i - \boldsymbol{p}$
        compute $\boldsymbol{J}_k(\boldsymbol{x}_i)$
        Solve $\left(\boldsymbol{J}_k^T \boldsymbol{J}_k + \lambda \boldsymbol{D}^2\right) \boldsymbol{\eta}_k = -\boldsymbol{J}_k^T \boldsymbol{r}_k$
        if $\|\boldsymbol{\eta}_k\| \leq \epsilon(\|\boldsymbol{y}_k\| + \epsilon)$ then
            convergence
            $k = k_{\max}$
        else
            $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{\eta}_k$
            $\rho = (\|\boldsymbol{r}_k\|^2 - \|\boldsymbol{r}_{k+1}\|^2)/(L(\boldsymbol{0}) - L(\boldsymbol{\eta}_k))$
            if $\rho > 0$ then
                $\lambda \leftarrow \lambda \max\{\frac{1}{3}, 1 - (2\rho - 1)^3\}$
                $\nu = 2$
            else
                $\lambda \leftarrow \lambda\nu$
                $\nu \leftarrow 2\nu$
                $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k$
            end if
        end if
    end for
    $\bar{\boldsymbol{y}}_{i+1} = \boldsymbol{y}_{k_{\max}}$
end for
```

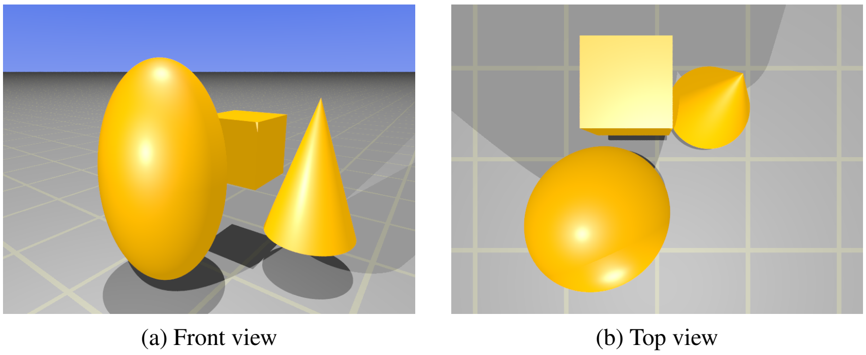

(a) Front view (b) Top view

Figure 5: Binary phantom consisting of an ellipsoid, a cone and a cube. A plane is shown for more perspective and improved visibility. The phantom can be enclosed in a cube consisting of $2 \times 2 \times 2$ grid cells (as seen from the top view).

4.1. Experiment I - the Effect of Discrete Tomography

In the first experiment we focus on the effects of DT reconstruction. Therefore we look at a perfectly aligned dataset and do not consider alignment.

The strength of the DC algorithm is in its ability to accurately reconstruct a dataset from just a few projections. It is not clear for this phantom, how many projection angles are necessary to create an accurate reconstruction. Therefore, we compare SIRT and DC reconstructions for a varying number of projection angles. We generated projection data for 50 angles. A subset of these projection images were used to create a reconstruction.

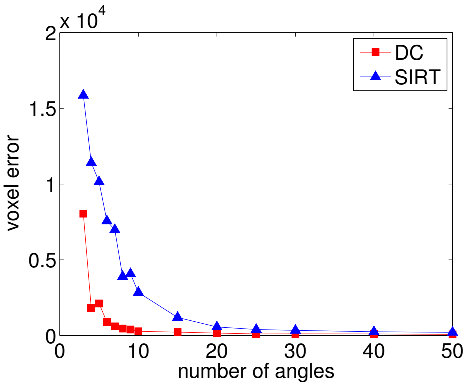


Figure 6: Number of incorrect voxels, in comparison with the ground truth, for a varying number of projection angles.

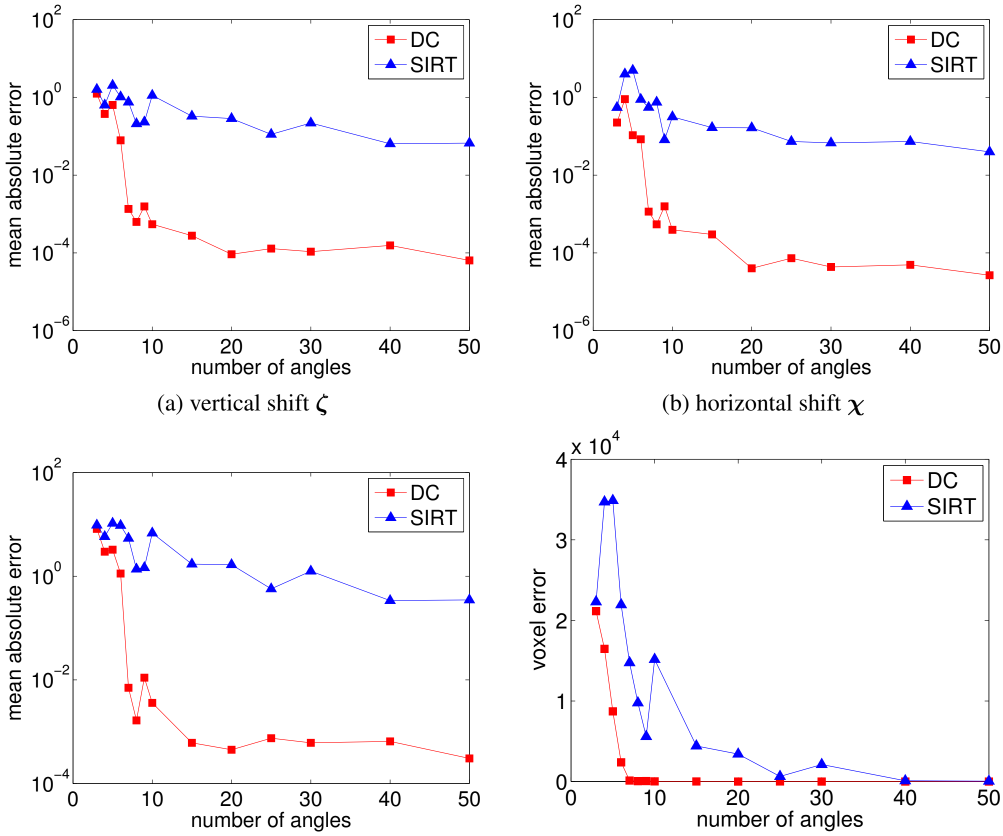
Figure 6 shows a plot of the number of incorrect voxels, called the *voxel error*, as a function of the number of projection angles. The number of projection angles that were considered are: 3 to 10, 20, 25, 30, 40 and 50. The smoothness parameter $\alpha = 0.08$ for DC was chosen empirically.

As expected, DC outperforms SIRT substantially in the limited angle case. However, this difference disappears if many projection images are used. Therefore we focus our simulations on the limited angle case (less than 10 projection angles).

4.2. Experiment II - Aligning Projection Data

In the second experiment, perturbations were applied to the projection geometry used to simulate the projection images. From the parameters in section 2, we have included the in-plane shift (χ, ζ) and in-plane rotation ϕ. Because these parameters operate in the plane of the detector, the alignment phase (10b) can be seen as an image registration problem, in which the observed projection image acts as the reference image.

An in-plane detector shift of maximum ± 10 detector pixels, both in y- and z-directions was simulated. The in-plane rotation is at most ± 15 degrees. These perturbations were generated randomly with uniform distribution. The misaligned projection data was used as input for the alignment software.



(a) vertical shift ζ

(b) horizontal shift χ

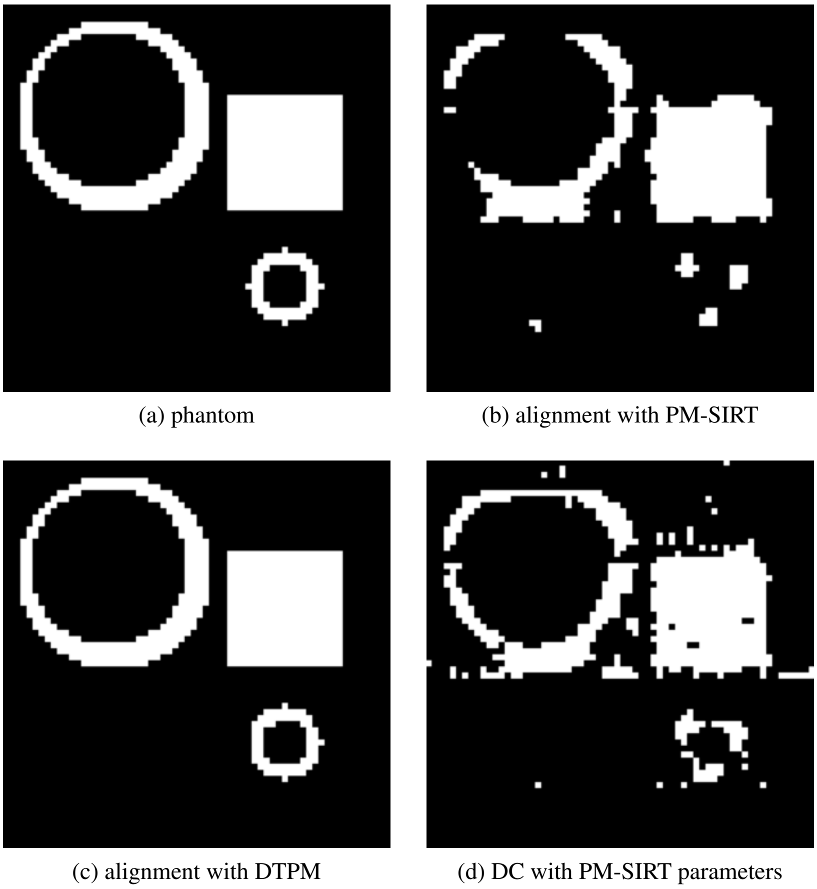
(c) in-plane rotation ϕ

(d) pixel error

Figure 7: Alignment results for varying numbers of projection angles. The error measure is the absolute error averaged over the number of projection angles: $\|\chi - \chi^*\|/K$ where χ^* is the true shift. Similar measures are used for the other parameters. (d) A plot of the voxel error reveals that the DTPM alignment produces perfect reconstructions, if the number of angles is large enough.

In Fig. 7a-7c, the final alignment error for the three parameters is shown for a varying number of projection angles. The errors are averaged over all projection angles, so they represent the mean error per projection image. With a maximum misalignment of 10 voxels (in the object position), the initial average error should be close to 5. While the initial average error for the in-plane rotation will be around 7.5 degrees. The performance of PM-SIRT is unsatisfying and improves only moderately when the number of angles is increased. In case of shifts, the parameters are found with an accuracy of 1-2 detector pixels. Such errors still produce smearing in the reconstruction. The DTPM algorithm using DC also fails with $3-5$ projection angles. However, for 8 projection images, the average error in the shift is in the order of 10^{-4}, so sub-pixel accuracy is achieved. The voxel error in Fig. 7d indicates that this kind of accuracy in the alignment is sufficient, since the corresponding voxel error is close to zero. Hence, the DC approach seems suitable in the limited angle case and achieves a higher accuracy overall.

For a qualitative comparison, the middle slices of the phantom and reconstructions are shown in Fig. 8. The reconstructions are computed from eight projection images. From these results it is clear that the alignment error for PM-SIRT is not small enough to eliminate all alignment artifacts. In contrast, the DTPM alignment produces an almost perfect reconstruction. It is not clear, however, if the difference in reconstruction quality is due to the alignment. It might be the case that DC creates a more accurate



(a) phantom

(b) alignment with PM-SIRT

(c) alignment with DTPM

(d) DC with PM-SIRT parameters

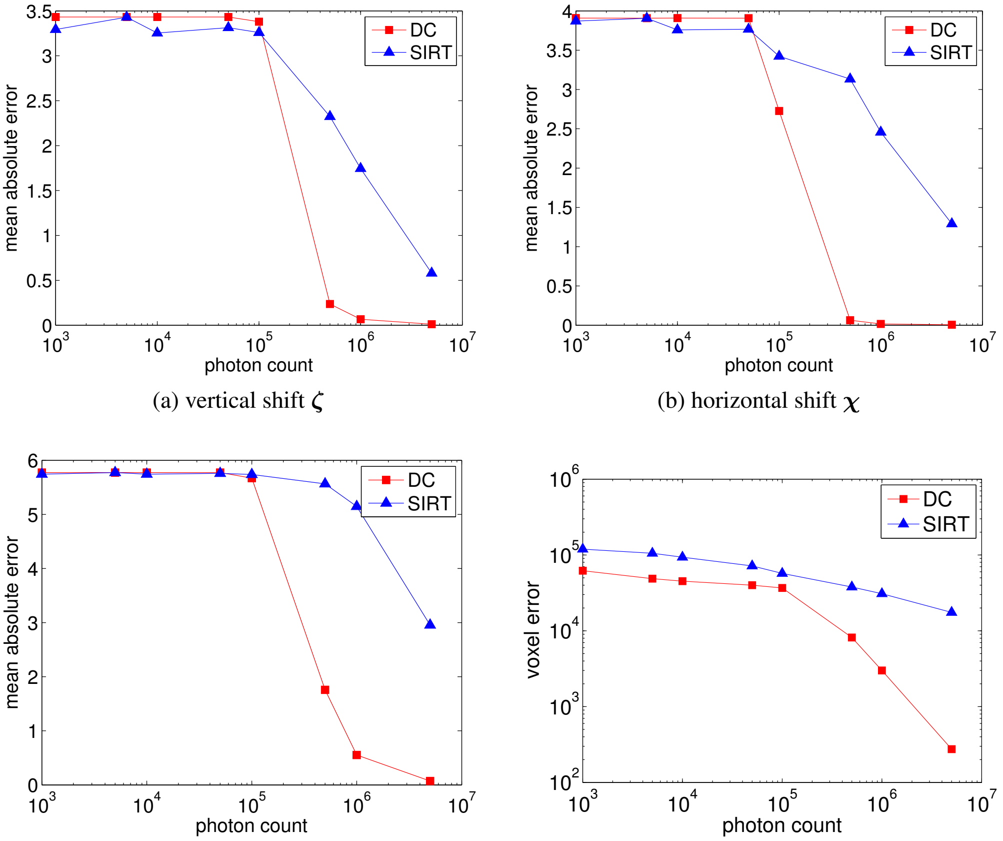
Figure 8: Comparison of a reconstructed slice (the middle one) resulting from PM-SIRT and DTPM. Reconstruction is based on 8 projection images. (d) Shows a DC reconstruction using the geometrical parameters found by PM-SIRT. This shows that the quality is not due to the final reconstruction using DC, but due to the alignment.

final reconstruction, while the alignment parameters are comparable. To exclude this possibility we have also created a DC reconstruction using the alignment parameters found by PM-SIRT, as shown in Figure 8d. This DC reconstruction still contains artifacts in the surface of the objects. This indicates that the intermediate DC reconstructions yield a better alignment, and it is not just the final discrete tomography reconstruction that accounts for the differences in quality.

4.3. Experiment III - Aligning Noisy Projection Data

In this experiment we test the robustness against noise of the DTPM algorithm, to determine if the discrete tomography prior still works well if the projection data contains noise.

We focus on reconstructing from eight projection images. From the previous experiment we found that this number of angles should be enough for accurate alignment. We applied Poisson noise to the projection data. The amount of noise is indicated by the simulated photon count for the projection data. A lower photon count corresponds to a higher noise level. In this particular case, we vary the noise level to match a simulated photon count from 10^3 to $5 \cdot 10^6$. These can be considered moderate to limited noise levels.



(a) vertical shift ζ

(b) horizontal shift χ

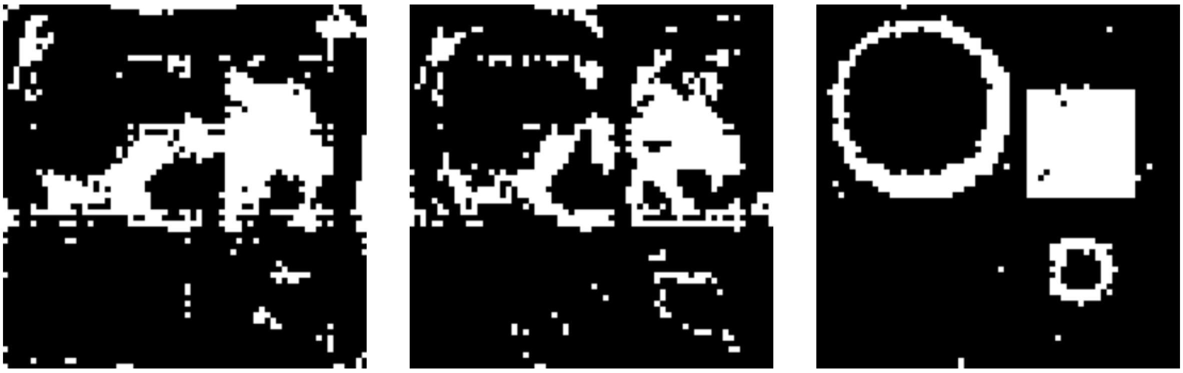
(c) in-plane rotation ϕ

(d) voxel error

Figure 9: Alignment results for eight projection images perturbed by Poisson noise.

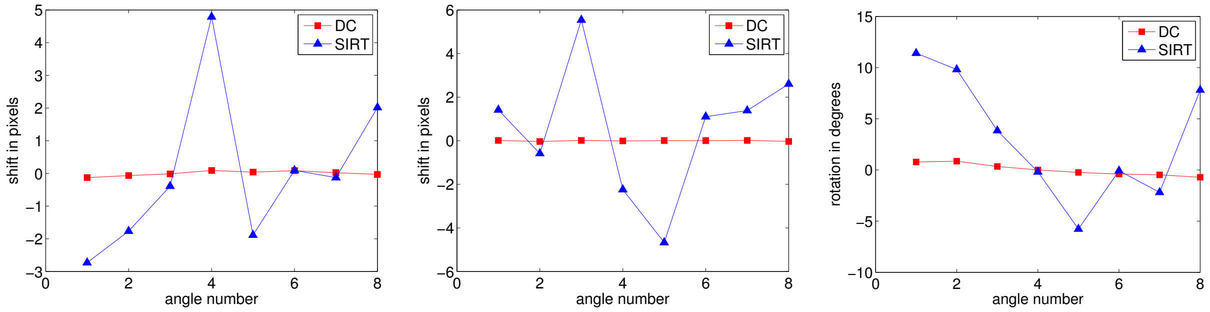
In Fig. 9, the absolute, averaged alignment errors are shown for the full range of noise levels. These plots show that alignment is very difficult when the noise levels are high. Clearly, it is not possible to accurately align a dataset if the noise level crosses a certain threshold. The DTPM alignment seems decent if the photon count is at least 10^6 (small amount of noise). In this case, sub-pixel accuracy is achieved, while in-plane rotation is found with an accuracy of a few degrees. The voxel error in Fig. 9d reveals that the DTPM reconstructions are reasonable for photon counts above 10^5. A voxel error of 10^4 corresponds to approximately 4% incorrect voxels. The PM-SIRT alignment results are much worse. While the vertical alignment improves with increased photon counts, the horizontal shift and in-plane rotation do not improve much. Presumably, due to the noise, details are missing in the SIRT reconstruction, such that the alignment step does not improve the parameters in a direction that will create a better reconstruction in the next iteration.

To visually assess the quality of the reconstructions, we show the middle slices in Fig. 10a and Fig. 10.c for PM-SIRT and DTPM respectively, corresponding to a dataset with simulated photon count of 10^6. Although the DTPM reconstruction is not as good as in the noiseless case, the contours of the objects are clearly visible. Moreover, the surfaces of the ellipsoid and cone in the DTPM reconstruction are fully closed. The cube is showing very minor salt and pepper noise, but its cross-section is clearly a square. The PM-SIRT reconstruction, in contrast, fails to produce the cone and the surface of the ellipsoid is far from closed. Again we reconstructed using DC with the alignment parameters from PM-SIRT. This reconstruction, shown in Fig. 10b, is slightly better, because the cross-section of the cone is visible. The result again indicates that the alignment parameters found by DTPM are more accurate and that not the final DC reconstruction causes the qualitative differences. This is confirmed by the final errors in the alignment parameters shown in Fig. 11. The horizontal and vertical shifts are found with sub-pixel accuracy by DTPM. The in-plane rotation is accurate up to 1 or 2 degrees. In comparison, the alignment by PM-SIRT is very poor.



(a) PM-SIRT (b) DC with PM-SIRT parameters (c) DTPM

Figure 10: Alignment result for eight projection images. The Poisson noise has an intensity corresponding to a photon count of 10^6. The figures show the middle slice of the reconstruction resulting from the alignment algorithm. In (b) a DC reconstruction is shown, using the aligned parameters found by PM-SIRT.



(a) residual vertical shift ζ (b) residual horizontal shift χ (c) residual in-plane rotation ϕ

Figure 11: The errors in the alignment parameters after alignment. The dataset used 8 projection images perturbed by noise, corresponding to a photon count of 10^6.

4.4. Performance Considerations

In this subsection we compare and discuss the computation times of PM-SIRT and DTPM. As computation times depend on the implementation and the particular dataset that is considered, we provide a rather general discussion of the performance. The alignment phase is the same for both methods and therefore has the same complexity. Differences in computation times for the alignment are caused by differences in the number of iterations that are required for convergence. However, the largest difference is due to the reconstruction algorithm that is employed. The method SIRT requires a forward projection and a back projection in each iteration. This is computed as two matrix vector products, using Eq. (7) (one by $\boldsymbol{W}$ and one by $\boldsymbol{W}^T$). The method DC has an inner and outer loop. The outer loop controls the parameter μ, but does not perform actual computations. The inner loop consists of a forward and back projection and is therefore comparable to a SIRT iteration. In the experiments, the number of SIRT iterations was kept constant at 300 iterations. Typically, the number of outer loop iterations of DC is half that number. Each iteration of the outer loop, the inner loop is run on average 10 times, so in total the DC method performs about 5 times as many matrix vector products, compared to SIRT.

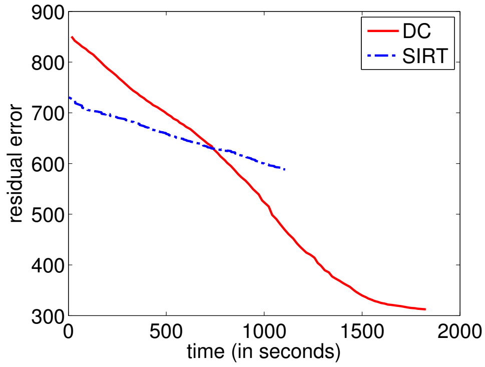


Figure 12: Computation times

For any given dataset it will be difficult to estimate the performance between DC and SIRT, because we do not know the number of inner or outer loop iterations that will be performed by DC. The number of iterations of both loops is determined by their convergence criteria. Therefore, we compare performance results of a single run of the algorithm based on our phantom dataset.

In order to compare computation times, the residual error of both methods are plotted against the computation time, shown in Fig. 12. The same noisy dataset was used as in experiment III. This result shows that DTPM is about two times slower compared to PM-SIRT. However, the final residual error of PM-SIRT is achieved much faster using DTPM. In that sense, DTPM converges faster and is a more efficient method. Notice that the initial residual error of DTPM is higher than that of PM-SIRT. This shows that the DC reconstruction method is not as accurate compared to SIRT when the alignment error is large. Most likely, the discrete tomography prior does not improve the reconstruction if the data consistency is small, due to misalignment.

5. Discussion

The results from the experiments on simulated projection data show that the DTPM method achieves a lower error in the reconstruction as well as the alignment parameters, when compared to PM-SIRT. In the experiments we applied both methods to a dataset using simulated projection data from a phantom object. In this section we will discuss our expectations and considerations for applying our method to data from real experiments.

First of all, the results of experiment I show that 8 projections are enough to accurately reconstruct the binary phantom object. If more than 20 projection are available, the difference between SIRT and DC reconstructions is only minor. So in order to benefit from using the DTPM alignment method on real data, the number of projections should be within a certain interval.

When reconstructing from experimental data we cannot determine the voxel error of a reconstruction, since we do not have a ground truth. Neither can we determine the optimal number of projection angles using the voxel error. However, it is still possible to estimate a reasonable number of projections. If a manually aligned dataset could be obtained, we can compare residual errors between SIRT and DC. The residual error compares simulated projections from the final reconstruction with the measured projection data. Using this norm, we can create a similar plot as shown in Fig. 6. Although the residual error has no one-to-one correspondence with the voxel error, it still can give an approximate insight to the real error.

Secondly, the amount of noise should be limited in order to benefit from using the DTPM alignment method. From Fig. 9 we see that the alignment fails if the signal-to-noise ratio is too low. Acceptable noise levels can be determined by qualitatively observing the aligned reconstruction. As an alternative, the total variance of the projection images could be considered as a measure for the amount of noise.

Moreover, it is important to note that alignment errors can be present in the projection data that cannot be modeled by our set of alignment parameters. Although the alignment parameters we consider have typically the largest perturbations in an experimental dataset, other errors can be present as well. For example, we do not consider rotation of the object around the x-axis. A small rotation would result in vertical "shrinking" of the projection image and certain features would overlap. Such an alignment error can only be corrected for if the rotation around the x-axis is included in the parameters of DTPM. However, this parameter is highly correlated with the shift parameters used in DTPM. Likewise, errors in the rotation angle are difficult to find, due to the high non-linear nature of the parameter. Nevertheless, we think that estimating in-plane shifts and rotations will improve the alignment of an experimental dataset even if other alignment errors exist.

Finally, the choice of the smoothness prior parameter α is important in the DC reconstruction. This parameter reflects the spatial coherency of solutions. The phantom in Fig. 5 is very smooth indeed. In reality, this prior might not be as accurate, depending on the nature of the object. Selecting the parameter α can be achieved in a similar fashion as choosing the number of projections. By minimizing a measurable error norm, such as the residual error, we can select the value for α that results in the lowest error.

6. Conclusions

A new method was proposed for alignment of binary tomography datasets from limited data. Prior knowledge of the binary grey values was included as a regularization method. It was found that the use of discrete tomography in a projection matching method, results in more accurate intermediate reconstructions. As a result, the subsequent alignment step by matching projection images is better defined.

For aligning the projection images, a variant of the Levenberg-Marquardt algorithm was used. By using finite differences for computing derivatives, the method does not require analytic gradients.

From our numerical results for simulated data, we see that the binary tomography algorithm DC yields much more accurate reconstructions in the case of few projection images (less than 15) [23]. The projection matching method DTPM is able to effectively align datasets from 6 projection images or more. The alignment combined with SIRT fails to find accurate alignment parameters with few projection angles. Only for a large number of angles the results are improving.

Qualitatively, the difference between PM-SIRT and DTPM was clear. While DTPM results in almost perfect reconstructions, for datasets with limited noise, the PM-SIRT reconstructions have many artifacts. By creating a DC reconstruction combined with geometrical parameters found by PM-SIRT, we showed that differences in quality in the reconstructions are mainly due to the alignment accuracy.

The results show that employing binary tomography as regularization is an essential step in projection matching alignment, when only a few projection images are available.

Acknowledgements

This research was supported by the Netherlands Organisation for Scientific Research (NWO), programme 639.072.005 and by iMinds (Interdisciplinary Institute for Technology, a research institute founded by the Flemish Government) and the IWT SBO Tomfood project. Networking support was provided by the EXTREMA COST Action MP1207.

References

[1] Batenburg, K. J.: An evolutionary algorithm for discrete tomography, *Discrete Applied Mathematics*, **151**(1), 2005, 36–54.

[2] Batenburg, K. J., van Aarle, W., Sijbers, J.: A semi-automatic algorithm for grey level estimation in tomography, *Pattern Recognition Letters*, **32**(9), 2011, 1395–1405.

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

[4] Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences*, **2**(1), 2009, 183–202.

[5] Bleichrodt, F., Batenburg, K. J.: Automatic Optimization of Alignment Parameters for Tomography Datasets, in: *Image Analysis*, Springer, 2013, 489–500.

[6] Boas, F. E., Fleischmann, D.: Evaluation of two iterative techniques for reducing metal artifacts in computed tomography, *Radiology*, **259**(3), 2011, 894–902.

[7] Brandt, S., Heikkonen, J., Engelhardt, P.: Automatic alignment of transmission electron microscope tilt series without fiducial markers, *Journal of structural biology*, **136**(3), 2001, 201–213.

[8] Candès, E. J., Romberg, J., Tao, T.: Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE Transactions on Information Theory*, **52**(2), 2006, 489–509.

[9] Fitchard, E., Aldridge, J., Reckwerdt, P., Mackie, T.: Registration of synthetic tomographic projection data sets using cross-correlation, *Physics in medicine and biology*, **43**(6), 1999, 1645.

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

[11] Frank, J.: *Electron tomography: Three-dimensional imaging with the transmission electron microscope*, Plenum Pub Corp, 1992.

[12] Houben, L., Bar Sadan, M.: Refinement procedure for the image alignment in high-resolution electron tomography, *Ultramicroscopy*, **111**, 2011, 1512–1520.

[13] Joseph, P.: An improved algorithm for reprojecting rays through pixel images, *Medical Imaging, IEEE Transactions on*, **1**(3), 1982, 192–196.

[14] Madsen, K., Nielsen, H. B., Tingleff, O.: Methods for Non-Linear Least Squares Problems (2nd ed.), 2004.

[15] Midgley, P. A., Dunin-Borkowski, R. E.: Electron tomography and holography in materials science, *Nature materials*, **8**(4), 2009, 271–280.

[16] Midgley, P. A., Weyland, M.: 3D electron microscopy in the physical sciences: the development of Z-contrast and EFTEM tomography, *Ultramicroscopy*, **96**(3), 2003, 413–431.

[17] Moré, J. J.: The Levenberg-Marquardt algorithm: implementation and theory, in: *Numerical analysis*, Springer, 1978, 105–116.

[18] Natterer, F., Wübbeling, F.: *Mathematical methods in image reconstruction*, SIAM, 2001.

[19] Nocedal, J., Wright, S.: *Numerical optimization*, 2nd edition, Springer Verlag, 2006.

[20] Palenstijn, W. J., Batenburg, K. J., Sijbers, J.: Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs), *Journal of Structural Biology*, **176**(2), 2011, 250–253.

[21] Palenstijn, W. J., Batenburg, K. J., Sijbers, J.: The ASTRA Tomography Toolbox, *CMMSE 2013 : Proceedings of the 13th International Conference on Computational and Mathematical Methods - Almería , Spain*, 2013.

[22] Parkinson, D. Y., Knoechel, C., Yang, C., Larabell, C. A., Le Gros, M. A.: Automatic alignment and reconstruction of images for soft X-ray tomography, *Journal of Structural Biology*, **177**(2), 2012, 259–266.

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

[24] Slaney, M., Kak, A.: *Principles of computerized tomographic imaging*, SIAM, 1988.

[25] Williams, J. J., Yazzie, K. E., Phillips, N. C., Chawla, N., Xiao, X., De Carlo, F., Iyyer, N., Kittur, M.: On the Correlation Between Fatigue Striation Spacing and Crack Growth Rate: A Three-Dimensional (3-D) X-ray Synchrotron Tomography Study, *Metallurgical and Materials Transactions A*, **42**(13), 2011, 3845–3848.

[26] Yang, C., Ng, E., Penczek, P.: Unified 3-D structure and projection orientation refinement using quasi-Newton algorithm, *Journal of structural biology*, **149**(1), 2005, 53–64.