Motion correction in X-ray tomography

submitted by

Ander Biguri

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Electrical and Electronic Engineering

September 2017

Abstract

bla

Contents

1 Introduction **3**

2 X-ray imaging in medicine **4**

3 The image reconstruction problem **5**

3.1 Geometry of CBCT 5

3.2 FDK 5

3.3 Iterative reconstruction algorithms 5

3.3.1 Algebraic Reconstruction Techniques 7

3.3.2 Conjugate Gradient Least Squares 7

3.3.3 Statistical inversion 7

3.3.4 Total variation minimization with POCS 7

3.3.5 Total variation regularization via Rudin-Osher-Fatemi model . . 15

3.4 Discussion 18

4 Applications of TIGRE **19**

5 Tolerance analysis of motion correction **20**

6 Conclusions and Future work **21**

Chapter 1

Introduction

Chapter 2

X-ray imaging in medicine

Chapter 3

The image reconstruction problem

This chapter tries to explain the mathematics behind CT reconstruction, the FDK algorithm and iterative reconstruction algorithms. After the formal proposition of the mathematical problem of integrating over straight lines,the FDK algorithm is introduced. Then, the alternative proposal of the iterative algebraic methods is shown, followed by a wide variety of different algorithms that can be used to solve the algebraic problem. These include gradient descend techniques, Krylov subspace methods, statistical approaches and compressed sensing techniques. Finally, a discussion of the challenges that arise from the use iterative algorithms are described.

3.1 Geometry of CBCT

3.2 FDK

3.3 Iterative reconstruction algorithms

Nowadays the FDK algorithm is the most widely used algorithm, and only until very recently [CITE] it has been the only algorithm available in any commercial medical or industrial CT device. While using FDK is advantageous in some cases, often the algorithm behaves poorly, specially when errors arise in the data, or the amount of data is limited. This is because FDK is based on an analytical approximation of straight path integrals in continuous spaces. Reality is far from straight path integrals, as due to X-ray physics photons do not behave linearly. Photons from CT machines are polychromatic, and human tissue is behaves non-linearly in respect to X-ray energy.

Additionally, Compton scattering is a common effect, where the photons get reflected in different angles dependent on their energy. Apart from photon physics related errors, electronic noise is always present in detector technology being the only feasible way of avoiding it longer exposition times, harmful to live tissue. Limited data can additionally harm the image reconstruction, as CT has generally less detector data than the amount of voxels are desired to reconstruct. All these effects make CT image reconstruction a challenging problem and have strong effect in the behaviour of FDK. As an alternative to FDK, iterative algebraic reconstruction algorithms try to minimize a functional by updating the image continuously and comparing it to the measured data. These algorithms have been shown to improve reconstruction quality when the data is noisy and/or limited. Additionally, as they are an algebraic tool, they allow careful tuning of the mathematics, enabling them to change their behaviour.

Iterative algorithms in CT generally reffer to those algorithms that, as the name says, iterate, but solve the linearized model

$$Ax = b + \tilde{e} \tag{3.1}$$

on where $x \in \mathbb{R}^{N_{voxels}}$ is a vector representing the lexicographically ordered voxels of the 3D image, $b \in \mathbb{R}^{N_{pixels}}$ a vector of the detector measured pixels. A is the linearized model matrix, a matrix that describes the behaviour of the CT system. Each row of the matrix A is describes the behaviour of the X-rays that affect each single pixel in the detector. The next chapter goes into a bit more detail on how to build matrix A. Errors from measurement are inevitable in any application, and there are linearization errors, as no model is perfect. In equation 3.1, $\tilde{e}$ represents all those errors.

As an exact solution for x can not be found, the problem in equation 3.1 is minimized as

$$\hat{x} = \arg\min_x \|Ax - b\|^2 + R(x), \tag{3.2}$$

on where $R(x)$ is an optional regularization function. This minimization function has been widely studied in mathematics and there are multiple algorithms that can solve it. However not all algorithms that solve the equation can be used in CT reconstruction, due to the nature of the A matrix, as it is very big (approximately $10^8 \times 10^8$ in a standard medical image) and very sparse (approximately 0.0017% of sparsity in a standard medical image). This makes the matrix severely ill-conditioned and impossible to store in memory. Additionally, often the CT problem can be underdetermined, making the problem ill-posed and further constraining the possible algorithms that can be used. That said, a wide variety of algorithm have been proposed to solve the CT algebraic problem and new ones are still being published. This section discusses a few of the

available and most common algorithms that have been studied in this work.

3.3.1 Algebraic Reconstruction Techniques

Arguably the most well known iterative algorithm is the method known as the algebraic reconstruction technique (ART)[CITE], known as the Kaczmarz method outside the CT imaging field. The ART algorithm, for matrix elements $a_{ij} \in R$ is defined as

$$x^{n+1} = x^n + \frac{b_i - \langle a_i, x^n \rangle}{\|a_i\|^2} a_i, \tag{3.3}$$

where a_i is the i-th row of matrix A and $\langle \, , \rangle$ denotes the inner product. The ART method projects the image into the hyperplane described by the equation in row i.

Relaxation parameter λ

3.3.2 Conjugate Gradient Least Squares

3.3.3 Statistical inversion

3.3.4 Total variation minimization with POCS

Sometimes solving a regularized problem may result in a better final image than just trying to solve the data constrain with the model. This is especially useful in more ill-conditioned problems, such as when the data is very noisy (thus the model does not fit the data with accuracy) or when few projections are available (the system is more under-determined). In these cases, regularisation can add a user constrain in the image domain that pushes the algorithm towards an specific solution among all the multiple possibilities. While a variety of regularization techniques and norms exist, the most suitable for CT imaging is the total variation (TV) norm.

The total variation norm is defined as the sum of the 2-norms of the directional gradients of the variable,

$$\|x\|_{\mathrm{TV}} = \sum_n \left\| \sum_\alpha \partial_\alpha x_n \right\|_2 . \tag{3.4}$$

Applied to CT imaging, the total variation norm is the sum of the total change occurred in the image. An image with less total variation would be an image that would have less change, or more flat, same valued regions. Regularizing with the TV norm as a minimization term will yield an image that is piecewise smooth and it happens that most of the objects imaged in CT scanners are piecewise smooth in linear attenuation, even more in medical CT imaging.

However, solving the minimization problem in equation XX is not trivial with the TV regularization. One of the first robust algorithm is the so called Adaptive Steepest Descend, Projection Onto Convex Subsets, or ASD-POCS algorithm [CITE]. This algorithm not only minimizes the data constrain with TV regularization but also adaptively controls the TV minimization update, in order to adapt its strength according to the data constrain update. Several adaptations and improvements of this algorithm have been proposed in the literature[CITES], all based in the same mathematical basis.

ASD-POCS

The previous algorithms discussed in this chapter where unconstrained minimization methods. While the TV minimization problem can be solved similarly (see section 3.3.5) formalizing the algorithm as a non-linear constrained minimization adds an advantage in the case where there system is under-determined. In an unconstrained problem such as in equation XX, the balance between the data constrain and the regularization constrain can be tuned via a hyperparameter, but in the case of an under-determined system, multiple solutions for the data fidelity term may exist. By reformulating it as it is shown in the rest of this section, the image with the same data fidelity 2-norm but the lowest TV norm can be chosen.

The minimization will yield an image $\vec{x}^*$ that minimizes

$$\vec{x}^* = \arg\min_x \|\vec{x}\|_{\mathrm{TV}} \tag{3.5}$$

subject to

$$\|A \cdot \vec{x} - \vec{b}\| \leq \epsilon, \tag{3.6}$$

$$\vec{x} \geq 0. \tag{3.7}$$

As previously described in this chapter, the data fidelity on equation 3.6 while desired to be zero, it will never reach to zero, due to inconsistencies in the data, model, noise, etcetera. Thus, in this algorithm it is introduced as a inequality constrain, instead of as the minimization problem itself. This introduces the parameter ϵ in the algorithm, the maximum 2-norm allowed for the data inconsistency. The problem in hand is now non-linear, due to the constrains, but convex.

The conditions for a constrained minimization to find the optimal solution can be obtained by satisfying the Karush Kuhn-Tucker conditions (a generalization of the Lagrange multiplies for inequality constrains). First, the Lagrangian for the current problems needs to be defined, as

$$\mathbf{L} = \|\vec{x}\|_{\mathrm{TV}} + \lambda_0 \cdot (\|A \cdot \vec{x} - \vec{b}\|^2 - \epsilon^2) - \vec{\lambda} \cdot \vec{x}, \tag{3.8}$$

where $\vec{\lambda}$ is a vector of the same size as the image, but λ_0 is a single value. Two inequality constrains are imposed to the Lagrange multipliers, namely non-negativity

$$\lambda_i \geq 0, \tag{3.9}$$

and complementarity

$$h_i(\vec{x})\lambda_i = 0, \tag{3.10}$$

where $i = 0, 1, ..., N_{pixels}$, and h_i is an alternative form of the inequality constrains as

$$h_0(\vec{x}) = \|A \cdot \vec{x} - \vec{b}\|^2 - \epsilon^2 \leq 0 \tag{3.11}$$

$$h_i(\vec{x}) = -x_i \leq 0 \qquad\qquad i \in [1, N_{pixels}] \tag{3.12}$$

Thus, only when the inequalities are violated does h_i turns non-zero, and with the complementarity condition, does the corresponding λ_i turns zero. A solution can be found for $\vec{x}$ when the gradient of the Lagrangian is zero, and if the differential operator is defined as

$$\nabla_{\vec{x}} Q(\vec{x}) = \sum_i \partial_{x_i} Q(\vec{x}) \vec{\delta}_i \tag{3.13}$$

where $\vec{\delta}_i$ is the Kronecker delta. The gradient of the Lagrangian can be then written as

$$\begin{aligned} \nabla_{\vec{x}} \mathbf{L} &= \nabla_{\vec{x}} \|\vec{x}\|_{\mathrm{TV}} + \lambda_0 \nabla_{\vec{x}} h_0(\vec{x}) + \sum_{i=1}^{N_{pixels}} \lambda_i \nabla_{\vec{x}} h_i(\vec{x}) = 0 \\ &= \nabla_{\vec{x}} \|\vec{x}\|_{\mathrm{TV}} + 2\lambda_0 A^T \cdot (A \cdot \vec{x} - \vec{b}) - \vec{\lambda} = 0 \end{aligned} \tag{3.14}$$

Further simplification can be applied to equation 3.14. As the non-negativity constrains are only active in zero valued voxels, the Lagrange multipliers are zero for strictly positive voxels. Thus, by adding an indicator function

$$\vec{x}_{indic} = \begin{cases} 1 & \vec{x} \neq 0 \\ 0 & \vec{x} = 0 \end{cases} \tag{3.15}$$

the Lagrangian gradient can be shortened to

$$\nabla_{\vec{x}}\mathbf{L} = \text{diag}(\vec{x}_{indic})\left(\nabla_{\vec{x}}\|\vec{x}\|_{\text{TV}} + \lambda_0 \nabla_{\vec{x}} h_0(\vec{x})\right) = 0. \tag{3.16}$$

Separating this new equation into two vectors,

$$\begin{aligned} \vec{d}_{\text{TV}} &= \text{diag}(\vec{x}_{indic})\left(\nabla_{\vec{x}}\|\vec{x}\|_{\text{TV}}\right) \\ \vec{d}_{\text{data}} &= \text{diag}(\vec{x}_{indic})\left(\nabla_{\vec{x}} h_0(\vec{x})\right) \end{aligned} \tag{3.17}$$

brings to the Karush Kuhn-Tucker conditions: $\vec{x}$ will be an optimal condition if $\vec{d}_{\text{TV}}$ and $\vec{d}_{\text{data}}$ are pointing in exactly the opposite direction. In practice the algorithm will only check if the vectors are pulling in opposite direction (by computing the dot product) and that the inequality constrains are satisfied. By checking the direction of the vectors the algorithm ensures that even if the data constrain is satisfied, only the optimal solution regarding both TV norm and data fidelity is chosen.

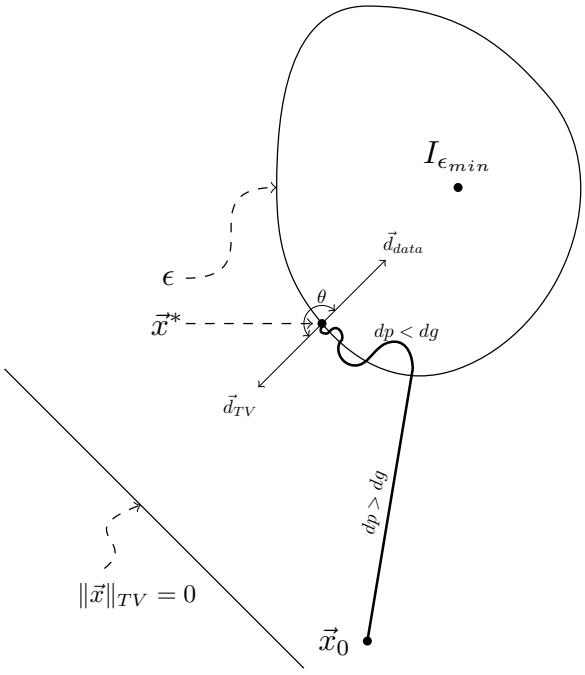

Figure 3-1: Conceptual diagram of the ASD-POCS algorithm path to the solution.

Figure 3-1 shows a conceptual diagram of the ASD-POCS algorithm. There is an area around the image with minimum data constrain, $I_{\epsilon_{min}}$. The solution $\vec{x}^*$ generally lies on the boundary of the area with the user specified ϵ. From an initial image $\vec{x}_0$, the algorithm walks towards the area of acceptable ϵ more strongly than towards the area

of minimum TV, as the step sizes of the vectors $\vec{d}_{\text{TV}}$ and $\vec{d}_{\text{data}}$, dp and dg respectively, are adaptively controlled to be $dp > dg$. Once the image is within acceptable 2 norm, then the step size is changed in order to have stronger $\vec{d}_{\text{TV}}$ ($dp < dg$). The optimal solution can be found when both vectors point in opposing direction, or in other words, when the angle between them is 180 degrees, or the cosine of it is -1,

$$\cos\theta = \frac{\vec{d}_{\text{TV}} \cdot \vec{d}_{\text{data}}}{\|\vec{d}_{\text{TV}}\|\|\vec{d}_{\text{data}}\|}. \tag{3.18}$$

The pseudocode for the algorithm can be seen in algorithm 1. The algorithm is essentially solving the two vector in equation 3.17, the data vector in lines [5-8] and the TV vector in lines [18-22]. Line [9] enforces the positivity constrain. In the algorithm, dtv is initialized according to α, an user specified TV hyperparameter for TV, together with dp, the step size performed by the data constrain. After the TV minimization is performed, the step size of the TV vector is rechecked. If the TV minimization step is too big (bigger than the data step size), and the desired ϵ is still not achieved, the step size is reduced further. This method of adaptively setting the step size of the TV iteration relating to the data step size is what ensures the optimal condition is achieved. Finally, the stopping criteria relies in either achieving the desired ϵ with with a desired $\cos\theta$, or stopping due to reaching a maximum amount of iterations (β decreases with iteration number). In the original proposition of the ASD-POCS algorithm (and shown here), the data constrain is solved using SART, however any other algorithm solving the same minimization problem can be used here (e.g. CGLS or OS-SART).

The algorithm has 7 parameters that need to be set up: β and β_{red} are the initial value and reduction ratio of the SART hyperparameter, similarly α and α_{red} serve as hyperparameter and reduction ratio for the TV minimization. r_{max} controls the maximum allowed ratio of change between the data minimization and the TV minimization, in order to adapt the step sizes. The number of iterations the TV minimization performs per iteration of the data minimization is defined as n_{TV}. Finally, the allowed data error is ϵ, as described before. The initial values of the variables in the algorithm are a key factor on its convergence. Empirical tests show that wrong parametrization of the algorithm can lead to severely noisy reconstructions. An study of the sensitivity of these parameters to changes has been performed by Lohvithee *et al*[CITE VEEs PAPER, at least as "future"]. The study shows that some parameters can be safely set up to a static value regardless of the data, such as the data hyperparameters, but that ϵ, n_{TV} and α are critical parameters to tune in order to get an usable reconstruction, and they are heavily data dependant. While some algorithms have successfully

replaced the initial set of α by some data based heuristics [CITE PCSD][1] to the best of the authors knowledge there is no mathematical proposal for setting these parameters. The biggest drawback of this method is that several reconstructions may be needed to find the best parameters for an specific application.

Note that this minimization approach, while used for TV minimization in the original article, can be used for a variety of different minimization functions. For example, the TV minimization step could be replaced by a prior image minimization[CITE PICS], or any other convex minimization function. Similarly, the data minimization step can be replaced by any other minimization algorithm, as long as it minimizes the problem in equation XX

[1]These algorithms, namely PCSD and Aw-PCSD, are also available in TIGRE, by Manasavee Lohvithee.

Algorithm 1 ASD-POCS

Set: $\beta, \beta_{red}, n_{TViter}, \alpha, \alpha_{red}, r_{max}$
$\vec{x} = 0$;
while Stopping criteria not met **do**
    $\vec{x}_{prev} = \vec{x}$
    for n_{angles} **do**
        $\vec{x} = \vec{x} + \beta V A^T W^{-1}(\vec{b} - A\vec{x})$ ▷ SART update
    end for
    $\beta = \beta * \beta_{red}$
    $\vec{x} = \max(0, \vec{x})$ ▷ Enforce positivity
    $\vec{x}_{out} = \vec{x}$
    $\epsilon_{now} = \|A\vec{x} - \vec{b}\|$ ▷ Current ϵ
    $dp = \|\vec{x} - \vec{x}_{prev}\|$ ▷ Change in $\vec{d}_{\text{data}}$
    if first iteration **then**
        $dtv = \alpha * dp$ ▷ Initialize TV hyperparameter
    end if
    $\vec{x}_{prev} = \vec{x}$

    for n_{TViter} **do** ▷ TV update
        $\vec{dx} = \nabla_{\vec{x}} \|\vec{x}\|_{TV}$
        $\hat{dx} = \vec{dx} / \|\vec{dx}\|$
        $\vec{x} = \vec{x} - dtv \cdot \hat{dx}$
    end for
    $dg = \|\vec{x} - \vec{x}_{prev}\|$ ▷ Change in $\vec{d}_{\text{TV}}$
    if $dg > r_{max} * dp$ **and** $\epsilon_{now} > \epsilon$ **then**
        $dtv = dtv * \alpha_{red}$
    end if
    ▷ Check stopping criteria
    $\cos\theta = \vec{dp} \cdot \vec{dg} / \|\vec{dp}\| \cdot \|\vec{dg}\|$
    if $(\cos\theta < -0.9$ **and** $\epsilon_{now} < \epsilon)$ **or** $\beta < 0.005$ **then**
        Stop
    end if
end while

The gradient of the TV norm

In order to minimize the TV norm via gradient descend, the gradient of the TV norm needs to be computed, $\nabla_{\vec{x}}||\vec{x}||_{TV}$, being $\vec{x}$ the vectorized form of a N-dimensional image.

The main challenge wit the $\nabla_{\vec{x}}||\vec{x}||_{TV}$ term is that $||\vec{x}||_{TV}$ is not differentiable in the general case. However, in the CT case, $\vec{x}$ can be described as x_{ijk}, a regularly discretized mesh of directional indices i, j, k of maximum value $i_{max}, j_{max}, k_{max}$. The gradient of x has an additional Cartesian index α:

$$g^{\alpha} = (\nabla x)^{\alpha} = \partial_{\alpha} x \tag{3.19}$$

$$g^{\alpha}_{ijk} = \partial_{\alpha} x_{ijk}. \tag{3.20}$$

The TV norm can me then defined as sum of the 2-norms of the gradient of x, g, over the Cartesian coordinate, resulting in a scalar.

$$\|x\|_{\mathrm{TV}} = \sum_{ijk} \sqrt{\sum_{\alpha} \left(g^{\alpha}_{ijk}\right)^2} = \sum_{ijk} \sqrt{\sum_{\alpha} \left(\partial_{\alpha} x_{ijk}\right)^2}, \tag{3.21}$$

This is the term that the total variation regularization algorithm minimizes with a gradient descend. In order to perform this, the gradient of this term with respect to x is needed, now defined in a scalar field

$$(\nabla_{\vec{x}}||x||_{TV})_{ijk}. \tag{3.22}$$

This derivative can be expanded to a 3 component value for each x_{ijk} as:

$$\begin{aligned}(\nabla_x \|x\|_{\mathrm{TV}})_{ijk} = \frac{\partial}{\partial x_{ijk}} \|x\|_{\mathrm{TV}} &= \partial_{x_{ijk}} \sum_{i'j'k'} \sqrt{\sum_{\alpha} \left(\partial_{\alpha} x_{i'j'k'}\right)^2} = \sum_{i'j'k'} \partial_{x_{ijk}} \sqrt{\sum_{\alpha} \left(\partial_{\alpha} x_{i'j'k'}\right)^2} \\ &= \sum_{i'j'k'} \frac{\sum_{\alpha} \left(\partial_{\alpha} x_{i'j'k'}\right) \partial_{x_{ijk}} \left(\partial_{\alpha} x_{i'j'k'}\right)}{\sqrt{\sum_{\alpha} \left(\partial_{\alpha} x_{i'j'k'}\right)^2}}.\end{aligned} \tag{3.23}$$

This term now contains ∂_{α} derivatives, i.e. derivatives in the Cartesian coordinate system $[x, y, z]$. These are defined as

$$
\begin{aligned}
\partial_x x_{i'j'k'} &= \lim_{h\to 0} \frac{x_{i'+h,j',k'} - x_{i',j',k'}}{h} \\
\partial_y x_{i'j'k'} &= \lim_{h\to 0} \frac{x_{i',j'+h,k'} - x_{i',j',k'}}{h} \\
\partial_z x_{i'j'k'} &= \lim_{h\to 0} \frac{x_{i',j',k'+h} - x_{i',j',k'}}{h}.
\end{aligned} \tag{3.24}
$$

However, x is discrete, thus the limit definition of the derivative can not be used to numerically compute it, but an approximation of it can. By setting $h = 1$, equation 3.24 becomes the backward finite differences of the first order approximation of a derivative, a very computationally cheap operation. The derivative w.r.t. the Cartesian coordinate can be rewritten as

$$
\partial_\alpha x_{i'j'k'} = \delta_{\alpha x}\left(x_{i',j',k'} - x_{i'-1,j',k'}\right) + \delta_{\alpha y}\left(x_{i',j',k'} - x_{i',j'-1,k'}\right) + \delta_{\alpha z}\left(x_{i',j',k'} - x_{i',j',k'-1}\right) \tag{3.25}
$$

where δ_α is a Kronecker delta for the Cartesian axis. The other partial derivative term that appears in equation 3.23 is $\partial_{x_{ijk}}\left(\partial_\alpha x_{i'j'k'}\right)$. As the derivative is w.r.t. x_{ijk}, each component of x is an independent variable, thus $\partial_{x_{ijk}}\left(\partial_\alpha x_{i'j'k'}\right)$ is zero everywhere but in indices $i = i' \wedge j = j' \wedge k = k'$, where the derivative is 1. The term then becomes

$$
\begin{aligned}
\partial_{x_{ijk}}\partial_x x_{i'j'k'} &= \partial_{x_{ijk}}\left(x_{i',j',k'} - x_{i'-1,j',k'}\right) = \delta_{i',i}\delta_{j',j}\delta_{k',k} - \delta_{i'-1,i}\delta_{j',j}\delta_{k',k} \\
&= \delta_{i',i}\delta_{j',j}\delta_{k',k} - \delta_{i',i+1}\delta_{j',j}\delta_{k',k} \\
\partial_{x_{ijk}}\partial_y x_{i'j'k'} &= \partial_{x_{ijk}}\left(x_{i',j',k'} - x_{i',j'-1,k'}\right) = \delta_{i',i}\delta_{j',j}\delta_{k',k} - \delta_{i',i}\delta_{j'-1,j}\delta_{k',k} \\
&= \delta_{i',i}\delta_{j',j}\delta_{k',k} - \delta_{i',i}\delta_{j',j+1}\delta_{k',k} \\
\partial_{x_{ijk}}\partial_z x_{i'j'k'} &= \partial_{x_{ijk}}\left(x_{i',j',k'} - f_{i',j',k'-1}\right) = \delta_{i',i}\delta_{j',j}\delta_{k',k} - \delta_{i',i}\delta_{j',j}\delta_{k'-1,k} \\
&= \delta_{i',i}\delta_{j',j}\delta_{k',k} - \delta_{i',i}\delta_{j',j}\delta_{k',k+1}.
\end{aligned} \tag{3.26}
$$

These terms are practically a selecting function for i', j', k' ,matching only in the indices $i, i+1, j, j+1, k, k+1$ in the sum of the right hand side of equation 3.23. However the indices are limited to $i' \in [1, i_{max}]$, $j' \in [1, j_{max}]$ and $k' \in [1, k_{max}]$. As boundary conditions, Neumann boundary conditions are set to zero. To enforce that, a Kronecker deltas can be introduced for each index, $(1 - \delta_{i,i_{max}})$, with the same approach with the other indices.

Finally, substituting in equation 3.23, the gradient of the TV norm can be described as

$$
\begin{aligned}
(\nabla_x \|x\|_{\mathrm{TV}})_{ijk} &= \sum_{i'j'k'} \frac{\sum_\alpha \left(\partial_\alpha x_{i'j'k'}\right) \partial_{x_{ijk}} \left(\partial_\alpha x_{i'j'k'}\right)}{\sqrt{\sum_\alpha \left(\partial_\alpha x_{i'j'k'}\right)^2}} \\
&= \sum_{i'j'k'} \frac{\partial_x x_{i'j'k'} \partial_{x_{ijk}} \left(\partial_x x_{i'j'k'}\right) + \partial_y x_{i'j'k'} \partial_{x_{ijk}} \left(\partial_y x_{i'j'k'}\right) + \partial_z x_{i'j'k'} \partial_{x_{ijk}} \left(\partial_z x_{i'j'k'}\right)}{\sqrt{\sum_\alpha \left(\partial_\alpha x_{i'j'k'}\right)^2}} \\
&= \frac{\partial_x x_{i,j,k} + \partial_y x_{i,j,k} + \partial_z x_{i,j,k}}{\sqrt{\sum_\alpha \left(\partial_\alpha x_{i,j,k}\right)^2}} \\
&- \frac{\left(1-\delta_{i,i_{max}}\right) \partial_z x_{i+1,j,k}}{\sqrt{\sum_\alpha \left(\partial_\alpha x_{i+1,j,k}\right)^2}} - \frac{\left(1-\delta_{j,j_{max}}\right) \partial_y x_{i,j+1,k}}{\sqrt{\sum_\alpha \left(\partial_\alpha x_{i,j+1,k}\right)^2}} - \frac{\left(1-\delta_{k,k_{max}}\right) \partial_z x_{i,j,k+1}}{\sqrt{\sum_\alpha \left(\partial_\alpha x_{i,j,k+1}\right)^2}}.
\end{aligned} \tag{3.27}
$$

Equation 3.27 is the numerical approximation of the gradient of the total variation norm, and describes scalar field of the same size of the image. The same approach can be used with central and forward differences to obtain a similar equation, however central differences may not correctly minimize the TV norm of the image. As central differences do not take into account the value of the current voxel ijk, a chequerboard pattern would have zero TV norm, and this is the opposite of the purpose of the algorithm, therefore only numerical approximations of derivatives that take immediately adjacent pixel values into account can be used (such as forward or backward finite differences).

3.3.5 Total variation regularization via Rudin-Osher-Fatemi model

A different minimization approach to POCS is the approach proposed by Jia *et al*[18], that uses the Rudin-Osher-Fatemi (ROF) model for total variation minimization, widely used in the denoising literature[31][12][43]. By starting from the same minimization problem, namely

$$
\hat{x} = \arg\min_x \|Ax - b\|^2 + \lambda \|x\|_{TV}, \tag{3.28}
$$

and a forward-backward splitting algorithm[10] is used to split the minimization into two alternating steps, the TV and the data steps. If the optimality condition is considered to be

$$
\frac{\partial}{\partial x_\alpha} \|Ax - b\|^2 + \lambda \frac{\partial}{\partial x_\alpha} \|x\|_{TV} = 0, \tag{3.29}
$$

being α the set of Cartesian coordinates, then the problem can be split into the following equations, where g is a auxiliary function and $\mu > 0$:

$$\lambda \frac{\partial}{\partial x_\alpha} \|x\|_{TV} = \mu(x - g) \tag{3.30}$$

$$\frac{\partial}{\partial x_\alpha} \|Ax - b\|^2 = -\mu(x - g). \tag{3.31}$$

By solving for g, the simplified version of the algorithm can be seen in 2.

Algorithm 2 TV minimization with ROF model

1: Solve: $g = x - \frac{\lambda}{\mu} \frac{\partial}{\partial x_\alpha} \|Ax - b\|^2$ ▷ SART
2: Minimize: $x = \arg\min_x \|x\|_{TV} + 0.5 * \mu \|x - g\|^2$
3: Enforce positivity: $x = \max(0, x)$

The first line of the algorithm its essentially a gradient descend iteration, which is essentially a SART iteration. Note that the SART iteration can be replaced by other data-minimization algorithms such as CGLS. The second line is the ROF model, widely researched in image denoising. The ROF model tries to find the image x with minimum total variation subject to having the minimal deviation from its original value g. By changing the value of he hyperparameter μ, the strength of this regularization is controlled. A high μ will ensure that the image is very similar to its original value, while a small μ will be more lax. The advantage of this approach compared to the ASD-POCS algorithm is that it requires no extra projection or backprojection operations. Additionally, minimizing the ROF model is a very well studied problem in the image processing field, and it has lead to finding computationally efficient methods.

In the article by Jia *et al*, they solve the ROF model via gradient descend and controlling its step size with Armijo's rule. In this work a different approach is taken, based on the image processing literature.

Primal Dual formulation of the ROF model

As previously shown in line 2 of algorithm 2, the ROF model can be formulated as

$$\hat{x}_{ROF} = \arg\min_x \|x\|_{TV} + \frac{\mu}{2} \|x - g\|^2. \tag{3.32}$$

A solution of this problem using a primal-dual (PDU) approach has been proposed in literature[49], by changing the minimization equation to a saddle point optimization problem. While a wide variety of methods have been proposed to minimize the ROF model[31][43][6], the PDU method has the advantage of being very parallelizable, thus

a perfect fit for GPU computing. The dual variable can be proposed by using the TV definition of $\|x\|_{TV} = \|\nabla x\|$ and observing the following consequence of the Cauchy-Schwartz inequality

$$\|\nabla x\| = \arg\max_{\|\mathbf{p}\|\leq 1} \|\mathbf{p}\nabla x\|, \tag{3.33}$$

where $\mathbf{p} = (p^1, p^2, p^3)^T$ (for the 3D case) is the said dual variable. Note that each p^i is the size of the image x. Equation 3.32 can be then rewritten as

$$\hat{x}_{ROF} = \arg\min_{x} \arg\max_{\|\mathbf{p}\|\leq 1} \|\mathbf{p}\nabla x\| + \frac{\mu}{2}\|x - g\|^2. \tag{3.34}$$

The primal and dual updates can be both obtained from this equation. For the primal update, differentiating the equation according to x results in

$$-\nabla \cdot \mathbf{p} + \mu(x - g) = 0, \tag{3.35}$$

and one can solve it for x by performing a gradient descend update as

$$x^{n+1} = x^n(1 + \tau_P^n) + \tau_P^n \left(g + \frac{1}{\mu}\nabla \cdot \mathbf{p} \right), \tag{3.36}$$

where τ_P is the primal step size. The dual update can be computed similarly, by differentiating equation 3.34 according to $\mathbf{p}$, the following equation si obtained:

$$\nabla x + \mathbf{p}\alpha = 0, \tag{3.37}$$

where α is a Lagrange multiplier for the inequality constrain $\|\mathbf{p}\| \leq 1$. This equation can be maximized with a gradient ascend method as

$$\mathbf{p}^{n+1} = \Pi_{B_0} \left(\mathbf{p}^n + \tau_D^n \nabla x \right), \tag{3.38}$$

where $\Pi_{B_0}(\mathbf{p}) = \frac{\mathbf{p}}{\max\{1, \|\mathbf{p}\|\}}$ is a projection onto the unit ball centred in the origin.

The PDU algorithm consists in updating $\mathbf{p}$ and x iteratively, by alternating the updates. In [49][19] a step size update is proposed for the primal and dual step sizes:

$$\begin{aligned} \tau_D^n &= 0.3 + 0.02n \\ \tau_P^n &= \frac{1}{\tau_D^n}\left(\frac{1}{6} - \frac{5}{15 + n} \right). \end{aligned} \tag{3.39}$$

The same update is used in this work, as the images in their work are structurally

similar to CT images and empirical test showed satisfactory results. The algorithm can be shown to converge as it is shown in [50] that the primal-dual gap decreases with each update of x^n, and the gap is suggested as a control variable for the stopping criteria. In this work the algorithm has been implemented without the stopping criteria check, and an user specified parameter for the number of iterations is passed as an input, with a default value of 50, as it empirically showed good results.

The discretization of the divergence and gradient operators are a key factor when numerically computing the PDU algorithm, as they need to be consequent with each other. Thus, the gradient can be approximated using forward differences, but as the divergence is the adjoint of the gradient, it must be approximated with backwards differences.

3.4 Discussion

Chapter 4

Applications of TIGRE

Chapter 5

Tolerance analysis of motion correction

Chapter 6

Conclusions and Future work

Bibliography

[1] Cern phase space tomography. `http://cern.ch/tomography`. Accessed: 2016-0912-19.

[2] Popi model webpage and data. `https://www.creatis.insa-lyon.fr/rio/popi-model_original_page`. Accessed: 2016-0912-19.

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016.

[4] Ander Biguri, Manjit Dosanjh, Steven Hancock, and Manuchehr Soleimani. TIGRE: a MATLAB-GPU toolbox for CBCT image reconstruction. *Biomedical Physics & Engineering Express*, 2(5):055010, 2016.

[5] J M Blackall, S Ahmad, M E Miquel, J R McClelland, D B Landau, and D J Hawkes. Mri-based measurements of respiratory motion variability and assessment of imaging strategies for radiotherapy planning. *Physics in Medicine and Biology*, 51(17):4147, 2006.

[6] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1):89–97, 2004.

[7] Cheng-Ying Chou, Yi-Yen Chuo, Yukai Hung, and Weichung Wang. A fast forward projection using multithreads for multirays on GPUs in medical image reconstruction. *Medical Physics*, 38(7):4052–4065, 2011.

[8] Cheng-Ying Chou, Yi-Yen Chuo, Yukai Hung, and Weichung Wang. A fast forward projection using multithreads for multirays on gpus in medical image reconstruction. *Medical Physics*, 38(7), 2011.

[9] Cheng-Ying Chou, Yi-Yen Chuo, Yukai Hung, and Weichung Wang. A fast forward projection using multithreads for multirays on gpus in medical image reconstruction. *Medical Physics*, 38(7), 2011.

[10] Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[11] Yi Du, Gongyi Yu, Xincheng Xiang, and Xiangang Wang. GPU accelerated voxel-driven forward projection for iterative reconstruction of cone-beam CT. *BioMedical Engineering OnLine*, 16(1):2, 2017.

[12] Joan Duran, Bartomeu Coll, and Catalina Sbert. Chambolle's projection algorithm for total variation denoising. *Image processing on Line*, 2013:311–331, 2013.

[13] Hao Gao. Fast parallel algorithms for the x-ray transform and its adjoint. *Medical physics*, 39(11):7110–7120, 2012.

[14] G. Han, Z. Liang, and J. You. A fast ray-tracing technique for TCT and ECT studies. In *Nuclear Science Symposium, 1999. Conference Record. 1999 IEEE*, volume 3, pages 1515–1518 vol.3, 1999.

[15] S Hancock, M Lindroos, and S Koscielniak. Longitudinal phase space tomography with space charge. *Physical Review Special Topics-Accelerators and Beams*, 3(12):124202, 2000.

[16] S Hancock, M Lindroos, E McIntosh, and Mike Metcalf. Tomographic measurements of longitudinal phase space density. *Computer Physics Communications*, 118(1):61–70, 1999.

[17] X. Jia, H. Yan, L. Cerviño, M. Folkerts, and S.B Jiang. A GPU tool for efficient, accurate, and realistic simulation of cone beam CT projections.

[18] Xun Jia, Yifei Lou, John Lewis, Ruijiang Li, Xuejun Gu, Chunhua Men, William Y Song, and Steve B Jiang. Gpu-based fast low-dose cone beam ct reconstruction via total variation. *Journal of X-ray science and technology*, 19(2):139–154, 2011.

[19] Florian Knoll, Markus Unger, Clemens Diwoky, Christian Clason, Thomas Pock, and Rudolf Stollberger. Fast reduction of undersampling artifacts in radial MR

angiography with 3D total variation on graphics hardware. *Magnetic resonance materials in physics, biology and medicine*, 23(2):103–114, 2010.

[20] Robert M Lewitt. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Physics in Medicine and Biology*, 37(3):705, 1992.

[21] H. Helen Liu, Peter Balter, Teresa Tutt, Bum Choi, Joy Zhang, Catherine Wang, Melinda Chi, Dershan Luo, Tinsu Pan, Sandeep Hunjan, George Starkschall, Isaac Rosen, Karl Prado, Zhongxing Liao, Joe Chang, Ritsuko Komaki, James D. Cox, Radhe Mohan, and Lei Dong. Assessing respiration-induced tumor motion and internal target volume using four-dimensional computed tomography for radiotherapy of lung cancer. *International Journal of Radiation Oncology*Biology*Physics*, 68(2):531 – 540, 2007.

[22] Yong Long, Jeffrey A Fessler, and James M Balter. 3D forward and back-projection for X-ray CT using separable footprints. *IEEE transactions on medical imaging*, 29(11):1839–1850, 2010.

[23] Bruno De Man and Samit Basu. Distance-driven projection and backprojection in three dimensions. *Physics in Medicine and Biology*, 49(11):2463, 2004.

[24] V. G. Nguyen, J. Jeong, and S. J. Lee. Gpu-accelerated iterative 3d ct reconstruction using exact ray-tracing method for both projection and backprojection. In *2013 IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC)*, pages 1–4, Oct 2013.

[25] Yusuke Okitsu, Fumihiko Ino, and Kenichi Hagihara. High-performance cone beam reconstruction using CUDA compatible GPUs. *Parallel Computing*, 36(2):129–141, 2010.

[26] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

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

[28] Eric Papenhausen, Ziyi Zheng, and Klaus Mueller. Gpu-accelerated back-projection revisited: squeezing performance by careful tuning.

[29] Hyeong-Gyu Park, Yeong-Gil Shin, and Ho Lee. A fully GPU-based ray-driven backprojector via a ray-culling scheme with voxel-level parallelization for cone-

beam ct reconstruction. *Technology in cancer research & treatment*, 14(6):709–720, 2015.

[30] James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with NAMD. *Journal of computational chemistry*, 26(16):1781–1802, 2005.

[31] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259 – 268, 1992.

[32] Holger Scherl, Benjamin Keck, Markus Kowarschik, and Joachim Hornegger. Fast gpu-based ct reconstruction using the common unified device architecture (cuda). In *Nuclear Science Symposium Conference Record, 2007. NSS'07. IEEE*, volume 6, pages 4464–4466. IEEE, 2007.

[33] Daniel Schlifske and Henry Medeiros. A fast GPU-based approach to branchless distance-driven projection and back-projection in cone beam CT. In *SPIE Medical Imaging*, pages 97832W–97832W. International Society for Optics and Photonics, 2016.

[34] Evan E. Schneider and Brant E. Robertson. CHOLLA: A new massively parallel hydrodynamics code for astrophysical simulation. *The Astrophysical Journal Supplement Series*, 217(2):24, 2015.

[35] WP Segars, G Sturgeon, S Mendonca, Jason Grimes, and Benjamin MW Tsui. 4D XCAT phantom for multimodality imaging research. *Medical physics*, 37(9):4902–4915, 2010.

[36] WP Segars, G Sturgeon, S Mendonca, Jason Grimes, and Benjamin MW Tsui. 4d xcat phantom for multimodality imaging research. *Medical physics*, 37(9):4902–4915, 2010.

[37] R.L. Siddon. Fast calculation of the exact radiological path for a three-dimensional ct array. *Medical physics*, 12(2):252–255, 1985.

[38] E.Y. Sidky and X. Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine and Biology*, 53(17):4777, 2008.

[39] Jan-Jakob Sonke, Joos Lebesque, and Marcel van Herk. Variability of four-dimensional computed tomography patient models. *International Journal of Radiation Oncology*Biology*Physics*, 70(2):590 – 598, 2008.

[40] William M Thompson and William RB Lionheart. GPU accelerated structure-exploiting matched forward and back projection for algebraic iterative cone beam ct reconstruction. 2014.

[41] W. van Aarle, W. Jan Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers. The {ASTRA} toolbox: A platform for advanced algorithm development in electron tomography. *Ultramicroscopy*, 157:35 – 47, 2015.

[42] Jef Vandemeulebroucke, David Sarrut, Patrick Clarysse, et al. The popi-model, a point-validated pixel-based breathing thorax model. 2007.

[43] Curtis R Vogel and Mary E Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.

[44] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal processing letters*, 9(3):81–84, 2002.

[45] Meng Wu and Jeffrey A Fessler. GPU acceleration of 3D forward and backward projection using separable footprints for x-ray CT image reconstruction. In *3rd Workshop on High Performance Image Reconstruction*, 2011.

[46] Fang Xu. Fast implementation of iterative reconstruction with exact ray-driven projector on gpus. *Tsinghua Science & Technology*, 15(1):30–35, 2010.

[47] Fang Xu and K. Mueller. A comparative study of popular interpolation and integration methods for use in computed tomography. In *3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006.*, pages 1252–1255, April 2006.

[48] Hao Yan, Xiaoyu Wang, Wotao Yin, Tinsu Pan, Moiz Ahmad, Xuanqin Mou, Laura Cervio, Xun Jia, and Steve B Jiang. Extracting respiratory signals from thoracic cone beam ct projections. *Physics in Medicine and Biology*, 58(5):1447, 2013.

[49] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration.

[50] Mingqiang Zhu, Stephen J Wright, and Tony F Chan. Duality-based algorithms for total-variation-regularized image restoration. *Computational Optimization and Applications*, 47(3):377–400, 2010.

[51] Andy Ziegler, Thomas Khler, Tim Nielsen, and Roland Proksa. Efficient projection and backprojection scheme for spherically symmetric basis functions in divergent beam geometry. *Medical Physics*, 33(12):4653–4663, 2006.

[52] Timo Zinsser and Benjamin Keck. Systematic performance optimization of cone-beam back-projection on the kepler architecture. *Proceedings of the 12th Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pages 225–228, 2013.