DELFT UNIVERSITY OF TECHNOLOGY

THESIS PROJECT
WI5005

LITERATURE REVIEW

Author:
GIULIA IPPOLITO
(4949617)

Supervisors:
DR. IR. M. B. VAN GIJZEN,
IR. M. L. DE LEEUW DEN BOUTER

April 6, 2020

Contents

Introduction

Magnetic Resonance Imaging (MRI) is an imaging technique which produces three dimensional detailed anatomical images. It is one of the most powerful non-invasive technologies available at the current moment, but unfortunately it is also extremely expensive. Superconducting magnets are used in conventional MRI scanner which produce very high magnetic fields and guarantee high quality resulting images. These magnets have a nominal cost of $ 1M per Tesla (T) of magnetic field, and scanners in common use require 1.5 or 3 T [16]. Furthermore, the generation of such a strong magnetic field produces a tremendous amount of heat. For this reason, cooling is required, which is done using liquid helium, and requires a lot of energy [11].

Furthermore, a high level of expertise is required for operation and repair, making the scanners completely out of reach for many communities, especially in low-development countries. A significant reduction in cost is possible by eliminating the superconducting magnet from the MRI system, but this comes with a large reduction in available magnetic field strength [13].

For the past four years teams at the TU Delft, Leiden University Medical Center (LUMC), Pennsylvania State University (PSU) and Mbarara University of Science and Technology (MUST) have been working on developing a low-cost, portable MRI scan with a magnetic field in the milliTesla (mT) range ($\sim$ 50 mT). In Figure 1 the scanner currently at LUMC. The main motivation behind this scanner lies in the will to be able to export these scanner to African countries, such as Uganda, in order to make it possible to detect hydrocephalus in children and newborns.

In particular, the CURE children's hospital in Uganda, specialized in the detection and treatment of infant hydrocephalus, uses Computed Tomography (CT) brain imaging, the use of which is potentially dangerous for developing children due to X-ray radiation.

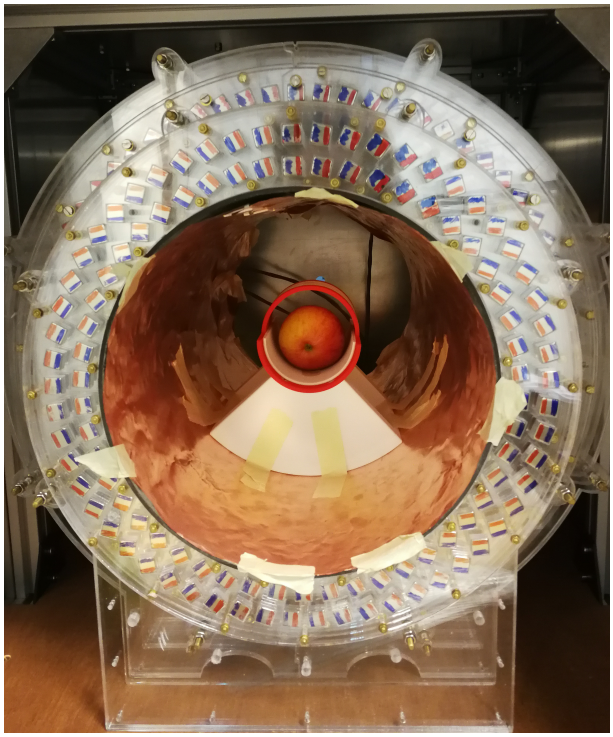


Figure 1: LUMC MRI scanner during a measurement of a phantom (an apple, in this case).

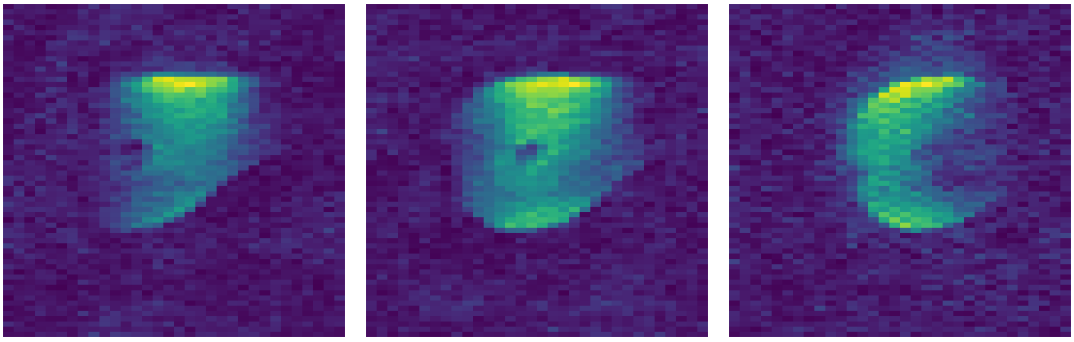
Hydrocephalus (from the Greek 'hydro', meaning water, and 'cephalus', head) is a condition of which the primary characteristic is excessive accumulation of fluid in the brain. The 'water' is actually cerebrospinal fluid (CSF), a clear fluid that surrounds the brain and spinal cord [1]. The build-up of this liquid causes a widening of the brain ventricles, which in turn creates potentially harmful pressure on the tissues of the brain. Without treatment, death or permanent disability may occur [1].

It is estimated that hundreds of thousands of children in sub-Saharan Africa develop hydrocephalus every year, with 1,000 to 2,000 cases in Uganda alone [11].

Motivation

The motivation for this literature review comes from the will to explore possibilities to improve both the resolution and the scanning times for the MRI scanner at LUMC. Indeed, the major problem with such low-field MRI systems is simply signal-to-noise ratio (SNR), since reducing the magnetic field strength from a typical clinical strength of 1.5 T to $\sim$ 50 mT comes with a several hundred fold reduction in SNR [13]. Furthermore, it is also crucial to be able to reduce the scanning time as much as possible, in order to account for possible (likely) movements from the subject scanned, especially because the goal is to be able to effectively scan children and even newborns. Simultaneously, the image quality must be high enough to allow to detect signs of hydrocephalus.

In [6] Super Resolution Reconstruction (SRR) techniques were already presented as a possibility to reach these goals. Unfortunately, by the time this research was carried out, it was not possible yet to generate proper images using the scanner at LUMC or PSU, hence it was not possible to test SRR techniques on actually working low-field machines. Since the start of the low-field MRI project major progress has been made, and it is now possible to do tests on real data (see Figure 2), which is the first innovation with respect to the previous work.



(a) (b) (c)

Figure 2: MRI scans of a pear, shifted of 1.7 cm one from the other. yz plain. Data acquired on 28-01-20.

Furthermore, as appears clear from Figure 2, we only have enough intensity in the central part of the scan. This is due to the small Field of View (FoV), but also to inhomogeneities in the magnetic field and gradient non-linearities along the bore of the scanner. This implies that, in order to have a complete observation of a subject (if the subject is bigger than the FoV), more than one scan is needed and those scans need to be combined.

In order to acquire multiple scans within a shorter time, it is possible to coarsen the resolution in the through-plane direction. The slice thickness can be substantially higher than the in-plane resolution in order to increase the Signal to Noise Ratio (SNR) [15]. However, even if the scanned subject fits in the FoV, there is a great advantage of multi slice images, compared to full 3D acquisitions. That is that it is possible to interleave

the acquisition of slices, which makes it possible to record multi-slice images significantly faster than full 3D images with the same resolution [15].

It has also been shown by several authors how, for a given acquisition time, it is possible to obtain super resolution reconstructed images which have a higher SNR than images directly acquired at the same resolution [17].

Hence, the idea is to acquire multiple scans within a short time by coarsening the resolution in the through-plane direction, and then combine those scans into a high resolution image with isotropic voxels. This would result in a significant step forward on the path of reducing the scanning time and improve the quality of the images. That is why the focus of this research will be on SRR techniques, which will be tested on real data produced by LUMC low-field MRI scanner.

Sections Outline

After the Introduction, Section 1 covers the basics of the working of a conventional MRI scanner, together with some information on the low-field version at LUMC. Next, in Section 2 and 3 respectively, the standard Super Resolution Reconstruction and some solution methods are presented. Section 4 contains a few details about the data used for the initial experiments, which are described together with the current results in Section 5. Finally, Section 6 focuses on the research questions for this thesis project and ideas for its development.

1 MRI Background

In this section we will briefly and concisely describe the process which allows us to obtain 2D and 3D data from a conventional MRI scanner. Finally, we will make some observations to put in relation what was said with the low-field MRI scanner at LUMC. The section is mainly based on [12].

1.1 Form of the Signal

MRI scanners use the body's natural magnetic properties to produce detailed images from any part of the body [4]. The strong magnetic field inside the scanner has an effect on the body by forcing the protons in the hydrogen molecules to align with that field. This uniform alignment creates a magnetic vector oriented along the axis of the MRI scanner [4]. When a radiofrequency current is added to the magnetic field, the magnetic vector is deflected. When the radiofrequency source is switched off the magnetic vector returns to its resting state (the protons realign with the magnetic field). This process is called *free precession*, and it causes a receiving signal to be emitted, and from this signal images are obtained.

An MRI scanner is made of three components: a main magnet, a magnetic field gradient system and a Radio Frequency (RF) system.
The main magnet generates the strong, uniform static magnetic field B_0. The gradient system is made of three orthogonal coils, which serve the purpose of generating time-varying magnetic fields which vary linearly in each spatial dimension. Finally, the RF system is made of a *transmitter* coil and a *receiver* coil. The first generates the B_1 field (also known as RF pulse), which is responsible for exciting the electron spins in the object (or person) that is to be imaged [6], while the latter transforms the energy released as the protons realign with the magnetic field into a signal.

The form of the receiving signal is

$$S(t) = \int_{\text{object}} M_{xy}(\boldsymbol{r}, 0) e^{-i\gamma \Delta B(\boldsymbol{r}) t} d\boldsymbol{r}, \tag{1.1}$$

where $\boldsymbol{M}$ is the net magnetization. The net magnetization vector in MRI is the summation of all the magnetic moments of the individual hydrogen nuclei. A parallel and antiparallel hydrogen nuclei have equal but opposite magnetic moments and cancel each other out [3]. However, a parallel alignment constitutes a lower energy state than an antiparallel alignment. Therefore, spins will be more likely to assume the parallel alignment [6], which means there are always slightly more hydrogen nuclei parallel to B_0. This slight difference constitutes the net magnetization vector $\boldsymbol{M}$. For an extensive background on the derivation of the signal model, the reader is referred to [12].

1.2 Spatial Information Encoding

For 2D imaging, MRI scanners turn the 3D imaged object into a 2D image. In order to do so, a slice has to be selected and only the hydrogen protons in that slice will be excited. This selective excitation is done through two components: a gradient field and a shaped RF pulse. A RF pulse can only be frequency selective, and protons at different spatial locations will be excited in the same way if they resonate at the same frequency. This means that the RF pulse has to be made spatially selected by making the protons resonance frequency position-dependent. A way to do so is to augment the B_0-field with a linear gradient field during the excitation period.

After the RF pulse, spatial information can be encoded into the signal during the free precession period, which we mentioned earlier. Since the signal is in the form of a complex exponential, there are essentially two ways to encode this information: *frequency* encoding and *phase* encoding [12].

Frequency encoding makes the oscillation frequency of an activated MR signal linearly dependent on its spatial origin, and it is used to determine one axis in the xy-plane of the selected slice. This is done by applying

a linear gradient field to the magnetic field after the RF pulse, for instance a gradient G_x in the x-direction. As thoroughly described in [12], this leads to the signal

$$S(t) = \iint_{\text{object}} \rho(x,y) e^{-i\gamma G_x x t} dx dy. \tag{1.2}$$

Because the magnetic field strength only varies in one direction, Equation 1.2 enables us to distinguish between different locations along the x-direction only. For the two-dimensional spatial localisation, phase encoding is needed.

The basic idea behind phase encoding is to pre-frequency encode the signal for a short time interval. After an RF pulse, a phase-encoding gradient G_y is turned on for a short interval T_{pe}, and then turned off. During the interval $0 \leq t \leq T_{\text{pe}}$, the local signal is frequency encoded. As a result of this frequency encoding, signals from different y-positions accumulate different phase angles after a time interval T_{pe}. At time T_{pe}, the signal will have an initial phase angle

$$\varphi(y) = -\gamma G_y y T_{\text{pe}} \tag{1.3}$$

$\varphi(y)$ is linearly dependent on the position y, so the signal is phase-encoded.
After demodulation (i.e. removal of the carrier signal $e^{-i\omega_0 t}$), we have

$$S(t) = \iint_{\text{object}} \rho(x,y) e^{-i\gamma G_y y T_{pe}} dx dy. \tag{1.4}$$

Combined with the previous signal model, we can now determine both the x- and y-coordinates of each signal component. So if a frequency-encoding gradient is turned on in the x-direction and a phase-encoding gradient in the y-direction, we obtain

$$S(t) = \iint_{\text{object}} \rho(x,y) e^{-i\left(\gamma G_x x t + \gamma G_y y T_{pe}\right)} dx dy. \tag{1.5}$$

It is possible to rewrite the signal in the following form

$$S(k_x, k_y) = \iint_{\text{object}} \rho(x,y) e^{-i(k_x x + k_y y)} dx dy \tag{1.6}$$

by performing the transformation

$$\begin{cases} k_x = \gamma G_x t \\ k_y = \gamma G_y T_{pe} \end{cases} \tag{1.7}$$

Hence, we expressed the signal in the so-called *k-space* notation. k-space is an abstract concept and refers to a data matrix containing the raw MRI data [2].

1.3 2D Image Reconstruction

Since we are imaging a 3D object, of which we have selected one slice, ρ will actually also depend on z, i.e $\rho(x,y,z)$. Hence, 1.6 needs to be replaced by

$$S(k_x, k_y) = \iint_{\text{object}} \int_{\text{slice}} \rho(x,y,z) e^{-i(k_x x + k_y y)} dx dy dz. \tag{1.8}$$

Denoting the desired image function by $I(x,y)$, in the slice-selective imaging case from [12] we have that

$$I(x,y) = \int_{z_0 - \Delta z/2}^{z_0 + \Delta z/2} \rho(x,y,z) dz, \tag{1.9}$$

Substituting 1.9 to $\rho(x,y)$ in 1.8, the basic imaging equation is the 2D Fourier transform

$$S(k_x,k_y) = \iint_{\text{object}} I(x,y)e^{-i2\pi(k_x x+k_y y)}dxdy, \tag{1.10}$$

where now

$$\begin{cases} k_x = \dfrac{\gamma}{2\pi}G_x t \\ k_y = \dfrac{\gamma}{2\pi}G_y T_{pe} \end{cases} \tag{1.11}$$

A few more observations lead to the final Fourier reconstruction formula

$$\begin{aligned} I(x,y) = \Delta k_x \Delta k_y \sum_{n=-N/2}^{N/2-1} \sum_{m=-M/2}^{M/2-1} S\left(n\Delta k_x, m\Delta k_y\right) e^{i2\pi\left(n\Delta k_x x + m\Delta k_y y\right)} \\ |x| < \frac{1}{\Delta k_x}, |y| < \frac{1}{\Delta k_y}. \end{aligned} \tag{1.12}$$

$I(x,y)$ can then be obtained by performing an inverse FFT, i.e. using an inverse Fast Fourier Transform algorithm.

1.4 3D Case

A future goal for this thesis project is to apply SRR to 3D data. The challenge in three-dimensional imaging lies in sorting out all the signal components from different spatial locations. This could be done by using slice-selective excitations for localization in the third dimension, leaving the other two dimensions to be done with encoding methods. Alternatively, it is possible to encode information along all three dimensions into the activated signals. Because of the nature of MRI signals, one dimension would be frequency encoded, while the other two could be either phase or frequency encoded. The general imaging equation for 3D would be in any case of the following form

$$S\left(k_x,k_y,k_z\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x,y,z)e^{-i2\pi\left(k_x x+k_y y+k_z z\right)}dxdydz. \tag{1.13}$$

It is important to note that the description of MRI given in this section also applies to low-field MRI, with some variations described in [13]. In particular, the image is obtained from the signal by performing an inverse FFT for both the conventional and the low-field MRI. Unfortunately, as mentioned in the Introduction, towards the edge of the scanner our gradients deviate from a linear pattern. Especially the gradient in the z-direction is very nonlinear towards the edges, which is consistent with the images in Figure 2. Moreover, the background field is more inhomogeneous and there is much more noise in the signal of the low-field scanner. All these factors combined result in the relationship between signal and image not to be perfectly governed by an FFT. As a consequence, we can notice irregularities and *artifacts* in the images.

2 Super Resolution Reconstruction

In this section the idea behind Super Resolution Reconstruction (SRR) is explained and the standard technique is presented. This section is mainly based on [6, 14], as well on some other sources which will be cited later in the text.

SRR techniques are devoted to the restoration of high resolution (HR) images from a set of several degraded and aliased low resolution (LR) samples. The idea is to have multiple LR observations of the same "scene", and to use the non-redundant information contained in those images in order to generate a new one. This ideally results in a HR image in which high frequency components are enhanced and lost ones are retrieved.

In order for each of these LR images to contain new information, we need them to be shifted with subpixel precision one from the other. Indeed, if we would have integer shifts, every image would contain the same information and therefore it would not be possible to retrieve the desired HR image.

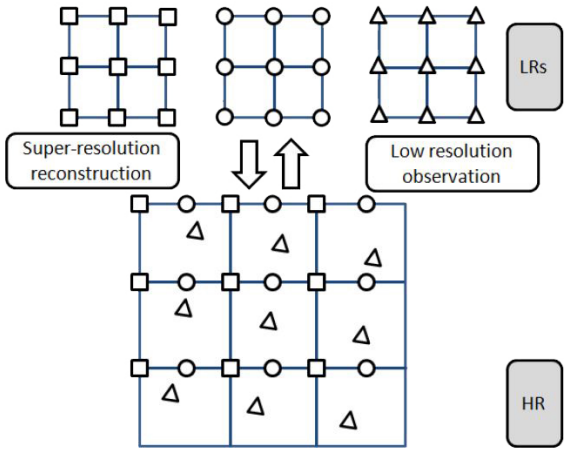


Figure 3: Sub-pixel shifts provide new information and lead to HR images. Source [20].

Depending on the kind of scene which is observed and by the mechanism of the observation (e.g a camera, an MRI scan, etc...), those sub-pixel shifts may be due to *uncontrolled motions*, like movements of the objects, or *controlled motions*, like images acquired from orbiting satellites. If those motions, whether controlled or not, can be estimated with sub-pixel accuracy, it is possible to combine the down-sampled LR images into an HR image.

2.1 Observation Model

In order to talk about SRR techniques, the first step to be taken is the formulation of an *observation model* which links the original HR images to the LR images. We first introduce the following notation. X and Y_k, $k = 1,\dots,p$ denote the $L_1N_1 \times L_2N_2$ HR image and the $N_1 \times N_2$ LR images respectively in matrix form. L_1 and L_2 represent the down-sampling factors for the horizontal and vertical directions respectively. $\boldsymbol{x}$ and $\boldsymbol{y}_k$, $k = 1,\dots,P$ denote instead the same images, but in lexicographical vector form. Hence, for instance we have

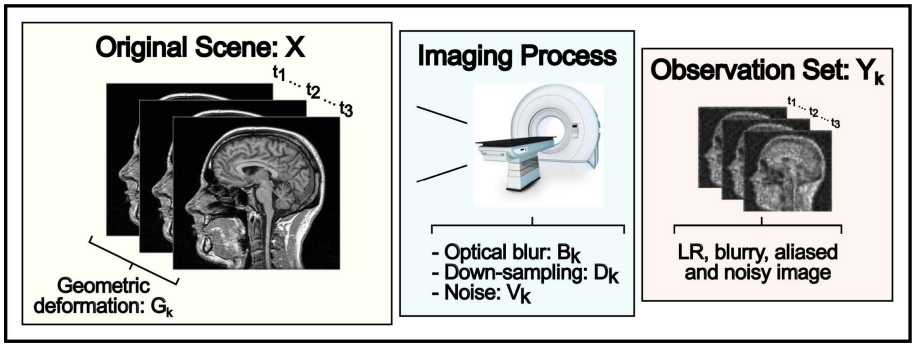


Figure 4: Scheme of the acquisition process. Source [17].

$$\boldsymbol{x} = \text{vec}(X) = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix},$$

with $N = L_1N_1 \times L_2N_2$ and where $\boldsymbol{x}_i$ is the ith column of X. In the following observation model it is assumed that $\boldsymbol{x}$ remains constant during the acquisition of the samples, except for motions and degradations allowed by the model. In Figure 4 a scheme of the acquisition process is shown. From a mathematical point of view, the model can be represented as

$$\boldsymbol{y}_k = D_k B_k G_k \boldsymbol{x} + \boldsymbol{v}_k, \qquad k = 1, \dots, P. \tag{2.1}$$

In 2.1, $\boldsymbol{v}_k$ is the noise, which is usually assumed to be independent, identically distributed Gaussian noise. D_k, B_k and G_k are three matrices that represent the transformations that lead from the original scene $\boldsymbol{x}$ to the LR images $\boldsymbol{y}_k$, with dimensions $N_1N_2 \times L_1N_1L_2N_2$, $L_1N_1L_2N_2 \times L_1N_1L_2N_2$ and $L_1N_1L_2N_2 \times L_1N_1L_2N_2$ respectively. As shown in 4, first a *geometric deformation* G_k is applied, which can be a sub-pixel translation, rotation, or both. B_k is the *blurring matrix*, which models the blurring occurring during the imaging process. This is done through the Point Spread Function (PSF). The PSF is a function that describes the blurring of one pixel over its surrounding pixels [6]. Finally, D_k models the *down-sampling operator*. It is usually assumed that every LR image has the same down-sampling factor and that the blurring operator is spatially invariant. So D_k and B_k do not actually depend on k. Hence, the system in 2.1 can be replaced by

$$\boldsymbol{y}_k = DBG_k \boldsymbol{x} + \boldsymbol{v}_k, \qquad k = 1, \dots, P. \tag{2.2}$$

Expressing the matrix product in the RHS of 2.2 as a single matrix yields

$$\boldsymbol{y}_k = A_k \boldsymbol{x} + \boldsymbol{v}_k, \qquad k = 1, \dots, P. \tag{2.3}$$

If we now vertically concatenate the vectors $\boldsymbol{y}_k$, the matrices A_k and the noise vectors $\boldsymbol{v}_k$ as follows

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_P \end{bmatrix}, \quad A = \begin{bmatrix} A_1 \\ \vdots \\ A_P \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_P \end{bmatrix},$$

we obtain the final system

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{v}, \tag{2.4}$$

where the matrix A has dimensions $N_1N_2P \times L_1N_1L_2N_2$, $\boldsymbol{y}$ and $\boldsymbol{v}$ are $N_1N_2P \times 1$ vectors, and $\boldsymbol{x}$ is a $L_1N_1L_2N_2 \times 1$ vector.

The details concerning the implementation of the matrices themselves will be explored in Section 5.

2.2 Least-Squares Formulation

The aim of SRR is to solve system 2.4 for $\boldsymbol{x}$, therefore "inverting" the process which leads to the blurred, downsampled LR images and retrieving the HR image. We can immediately notice that the matrix A in 2.4 is not necessarily square, unless $P = L_1L_2$.

We recall the following definitions for a system

$$A\boldsymbol{x} = \boldsymbol{y}, \quad \text{with } A \in \mathbb{C}^{m\times n}, \quad \boldsymbol{x} \in \mathbb{R}^n, \quad \boldsymbol{b} \in \mathbb{C}^m. \tag{2.5}$$

Definition 2.1. A linear system is called *underdetermined* if the number of unknown variables exceeds the number of equations ($m < n$).

Definition 2.2. A system is called formally *determined* if the number of unknowns is the same as the number of equations. ($m = n$)

Definition 2.3. A system is called *overdetermined* if there are more equations than unknowns ($m > n$) [19].

From these we can tell that system 2.4 might have zero, one, or infinitely many solutions. In particular, if A is invertible, then a determined system has a unique solution.
When A is not square, a system equivalent to 2.5 can be solved:

$$A^*A\boldsymbol{x} = A^*\boldsymbol{y}. \tag{2.6}$$

This system is Hermitian semi-positive definite (semi HPD) and it is called the system of *normal equations* associated with the *least-squares problem*

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 \tag{2.7}$$

Equation 2.6 is generally used when dealing with overdetermined systems. In this case, if the system is consistent (i.e. it has at least one solution), $\text{rank}(A) = n$ with $m \geq n$ and A^*A is invertible, then the least squares minimum norm (LSMN) solution is given by

$$\boldsymbol{x}_{\text{ls}} = (A^*A)^{-1}A^*\boldsymbol{b}. \tag{2.8}$$

Equation 2.8 is called LSMN because it is one particular solution of 2.7 which minimizes both $\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2$ and the 2-norm of the solution itself.

Most of the times when dealing with SRR for MRI data, our system will be underdetermined. Similarly as we had before, if the system is consistent and $\text{rank}(A) = m$ with $m \leq n$, the LSMN solution is

$$\boldsymbol{x}_{\text{ls}} = A^*(AA^*)^{-1}\boldsymbol{y}. \tag{2.9}$$

Indeed, if $\boldsymbol{y} = A\boldsymbol{x}$, we can set $\boldsymbol{x} = A^*\boldsymbol{u}$. So we will have

$$\boldsymbol{y} = A\boldsymbol{x} \Leftrightarrow \boldsymbol{y} = AA^*\boldsymbol{u}.$$

Since A is full rank, AA^* is invertible. Thus, we have $\boldsymbol{u} = (AA^*)^{-1}\boldsymbol{y} \Rightarrow \boldsymbol{x} = A^*(AA^*)^{-1}\boldsymbol{y}$.

A first attempt of determining 2.9 could then be done by computing the *pseudoinverse* of A. In particular, we will compute the *Moore-Penrose inverse*. We recall that the SVD of A is

$$A = U\Sigma W^*, \tag{2.10}$$

with U, Σ and V respectively the left singular vectors, the singular values σ_i and the right singular vectors of A.
Given 2.10, the Moore-Penrose inverse of A is $A^+ = W\Sigma^+U^*$, and $\boldsymbol{x} = A^+\boldsymbol{y}$ is our $\boldsymbol{x}_{\text{ls}}$. Note that W^+ is the pseudoinverse of W, defined as

$$W^+ = \begin{bmatrix} W_m^+ & 0 \\ 0 & 0 \end{bmatrix}, \quad W_m^+ = \text{diag}(1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_r), \quad \text{where} \quad r = \text{rank}(A). \tag{2.11}$$

However, this naive method becomes unreliable when having large, noisy systems with very small singular values. $\boldsymbol{x}_{\text{ls}}$ is equal to

$$\boldsymbol{x}_{\text{ls}} = A^+\boldsymbol{y} = W\Sigma^+U^*\boldsymbol{y} = \boldsymbol{x} + W\Sigma^+U^*\boldsymbol{v}, \tag{2.12}$$

where we used that $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{v}$. The last term in the equation is called *inverted noise*, and it can be explicitly computed as

$$W\Sigma^+U^*\boldsymbol{v} = \sum_{i=1}^{r} \frac{\boldsymbol{u}_i^*\boldsymbol{v}}{\sigma_i}\boldsymbol{w}_i \tag{2.13}$$

It appears clear that if the singular values are too close to 0, the inverted noise will grow and contaminate the reconstructed image.

2.3 Regularization

As we have seen, naive solutions like the pseudoinverse method lead to images potentially heavily corrupted by noise. Furthermore, we introduce the following definition.

According to Jacques Hadamard, a problem is *well-posed* if:

1. A solution exists
2. The solution is unique
3. The solution's behaviour changes continuously with the initial conditions.

A problem is said to be *ill-posed* if one or more of these properties do not hold. System 2.4 is ill-posed, as inverse problems typically are. Indeed, the condition number of matrix A becomes very large, and A is said to be ill-conditioned. As a consequence, solutions become very sensitive to small perturbations in the right-hand side $\boldsymbol{y}$. This is why we introduce the idea of *regularization*. Through the regularization term, we add additional information about $\boldsymbol{x}$, which results in more feasible solutions.

2.3.1 Tikhonov Regularization

Adding the Tikhonov regularization term to 2.7 yields the following new minimization problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 + \frac{1}{2}\lambda\|F\boldsymbol{x}\|_2^2, \tag{2.14}$$

where λ is the regularization parameter and F is the regularization matrix, typically a discrete finite difference approximation of a derivative operator. In the context of MRI scans, it can in particular be chosen to be a first-order difference matrix. In MRI images it is very likely for neighbouring pixels to have the same value, and such a choice of a regularization matrix penalizes jumps between pixels.
The parameter λ gives the balance between minimizing $\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2$ and minimizing $\|F\boldsymbol{x}\|_2^2$, hence the balance between a solution which fits the data well and enforcing prior information on the solution.

There exist several ways to determine the value of λ. One of the most popular choices is to employ the $L-$curve method [8].
The L-curve is a log-log-plot of $\|\boldsymbol{y} - A\boldsymbol{x}\|_2$ versus $\|F\boldsymbol{x}\|_2$, with λ as the parameter. The curve basically consists of two parts: a "flat" part where the regularization errors dominates, and a "steep" part where the perturbation error dominates. The optimal regularization parameter is supposed to lie somewhere near the L-curve's corner. The scale is chosen as such because it emphasizes the two parts and the corner of the curve.

Observe now that the objective function of minimization problem 2.14 is a convex function (linear combination of composition of l_2-norms), hence by posing its gradient equal to 0 we can find the global optimal solution to our problem. Therefore we have

$$(A^*A + \lambda F^*F)\boldsymbol{x} = A^*\boldsymbol{y} \tag{2.15}$$

which leads to

$$\boldsymbol{x} = (A^*A + \lambda F^*F)^{-1}A^*\boldsymbol{y}. \tag{2.16}$$

Unfortunately it is extremely computationally demanding to use the previous equation to find $\boldsymbol{x}$ by inverting $A^*A + \lambda F^*F$. In fact, if X is a $n \times m$ image, for what we have observed in Section 2.1, $A^*A + \lambda F^*F$ will be a $nm \times nm$ matrix. So if X is a HR image, then the previous matrix becomes very large.

2.3.2 Total Variation Regularization

Another possible regularization is the Total Variation (TV) regularization, which leads to the following minimization problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 + \frac{1}{2}\lambda\|F\boldsymbol{x}\|_1. \tag{2.17}$$

Here F is a first-order difference matrix defined such that

$$\|F\boldsymbol{x}\|_1 = \sum_{k=1}^{n}\sum_{l=2}^{m}\left|X_{l,k} - X_{l-1,k}\right| + \sum_{l=1}^{m}\sum_{k=2}^{n}\left|X_{l,k} - X_{l,k-1}\right|. \tag{2.18}$$

Similarly to the Tikhonov case, this choice of F penalizes jumps between neighbouring pixels, even though not as harshly. Indeed, in the minimization problem 2.14, the l_2-norm makes the regularisation term grow quadratically with the difference in value between neighbouring pixels, while in 2.17 the penalization grows linearly, due to the l_1-norm.

3 Solution Methods

In the following section we will present some solution methods for the standard SRR problem. The main sources for this section are [6, 18].

3.1 Krylov-Subspace Iterative Methods for Tikhonov Regularization

As we have seen in the previous section, it is not feasible to directly compute $\boldsymbol{x}$ as in 2.16 when dealing with high resolution images. Nevertheless, it is still possible to solve 2.15 by using iterative solvers, and in particular Krylov-subspace iterative solvers. From [18] we have the following definition

Definition 3.1. Given a matrix $A \in \mathbb{R}^{n\times n}$ and a vector $\boldsymbol{r}_0 \in \mathbb{R}^n$, the space $K^k(A;\boldsymbol{r}_0)$ defined as

$$K^k(A;\boldsymbol{r}_0) = \operatorname{span}\left\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \dots, A^{i-1}\boldsymbol{r}_0\right\}. \tag{3.1}$$

is called the Krylov-space of dimension k corresponding to matrix A and initial residual $\boldsymbol{r}_0$.

If we now consider a basic iterative method for a system $A\boldsymbol{u} = \boldsymbol{f}$ which follows the recursion

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + (\boldsymbol{f} - A\boldsymbol{u}^k) = \boldsymbol{u}^k + \boldsymbol{r}^k$$

we can observe how

$$\boldsymbol{u}^k \in \boldsymbol{u}^0 + \operatorname{span}\left\{\boldsymbol{r}^0, A\boldsymbol{r}^0, \dots, (A)^{k-1}\boldsymbol{r}^0\right\}.$$

Therefore a $\boldsymbol{u}^k$ calculated by a basic iterative method is an element of $\boldsymbol{u}^0 + K^k\left(A;\boldsymbol{r}^0\right)$.

The *Conjugate Gradient* (CG) method is an iterative method developed by Hestenes and Stiefel [10]. It is suitable for solving linear systems in the form $A\boldsymbol{u} = \boldsymbol{f}$, where A is a square HPD matrix.

In the following sections we will describe the CG and some of its variations. We will follow [18] and make the assumptions that $M = I$, and $\boldsymbol{u}_0 = 0$ so $\boldsymbol{r}_0 = \boldsymbol{f}$. These assumptions are not necessary for the CG method itself, but are useful to facilitate the formulas.

3.1.1 Generalization of the Problem

In Section 2.3.1 we have considered minimization problem 2.14

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 + \frac{1}{2}\lambda\|F\boldsymbol{x}\|_2^2.$$

As carefully explained in [6], this is a special form of the following problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{x}\|_{C^{-1}}^2 + \frac{1}{2}\lambda\|\boldsymbol{x}\|_R^2, \tag{3.2}$$

where $R = F^*F$ and C is the covariance matrix of the noise. Similarly to what we have seen before, the optimality condition for 3.2 is

$$(A^*C^{-1}A + \lambda R)\boldsymbol{x} = A^*C^{-1}\boldsymbol{y} \tag{3.3}$$

We can now observe that 3.2 can be formulated as a constrained optimisation problem

$$\begin{aligned} &\min_{\boldsymbol{r},\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{r}\|_C^2 + \frac{1}{2}\lambda\|\boldsymbol{x}\|_R^2 \\ &\text{subject to } \boldsymbol{r} = C^{-1}(\boldsymbol{y} - A\boldsymbol{x}) \end{aligned} \tag{3.4}$$

Indeed,

$$\|\boldsymbol{r}\|_C^2 = \boldsymbol{r} * C\boldsymbol{r} = (\boldsymbol{y} - A\boldsymbol{x})^* C^{-1}(\boldsymbol{y} - A\boldsymbol{x}) = \|\boldsymbol{y} - A\boldsymbol{x}\|_{C^{-1}}^2$$

If we now use the technique of lagrangian multipliers on 3.4, we find

$$\boldsymbol{r} = C^{-1}(\boldsymbol{y} - A\boldsymbol{x}), \quad \lambda R\boldsymbol{x} = A^*\boldsymbol{r}, \tag{3.5}$$

which if λR is invertible yields

$$\boldsymbol{x} = \frac{1}{\lambda} R^{-1} A^* \boldsymbol{r}, \quad \left(\frac{1}{\lambda} A R^{-1} A^* + C\right)\boldsymbol{r} = \boldsymbol{y} \tag{3.6}$$

If we now define $\boldsymbol{z} := \dfrac{1}{\lambda}\boldsymbol{r}$, we can extend the result to the case $\lambda = 0$ and have

$$(AR^{-1}A^* + \lambda C)\boldsymbol{z} = \boldsymbol{y}, \quad \boldsymbol{x} = R^{-1}A^*\boldsymbol{z} \tag{3.7}$$

From the equivalence between 3.2 and 3.4 follows the one between 3.3 and 3.7.

3.1.2 Standard CG

CG is an iterative solver which finds a solution to the following minimization problem

$$\left\|\boldsymbol{u} - \boldsymbol{u}^k\right\|_A = \min_{\boldsymbol{y} \in K^k(A;\boldsymbol{r}^0)} \|\boldsymbol{u} - \boldsymbol{y}\|_A \tag{3.8}$$

Thus the method builds a sequence of improving approximate solution vectors $\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots$ and eventually it finds $\boldsymbol{u}^k$ in the Krylov subspace $K^k(A; \boldsymbol{r}^0)$ such that $(\boldsymbol{u} - \boldsymbol{u}^k)^* A(\boldsymbol{u} - \boldsymbol{u}^k)$ is minimized (i.e. in the A-norm).
At each step, the residual is computed as

$$\boldsymbol{s}_k = \boldsymbol{f} - A\boldsymbol{u}_k. \tag{3.9}$$

Concerning the rate of convergence of the CG method, from [18] we have the following theorem

Theorem 3.1. *The iterates $\boldsymbol{u}_k$ obtained from the CG algorithm satisfy the following inequality:*

$$\|\boldsymbol{u} - \boldsymbol{u}_k\|_A \leq 2\left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right)^k \|\boldsymbol{u} - \boldsymbol{u}_0\|_A, \tag{3.10}$$

where $\kappa_2(A)$ is the condition number of the matrix A in the 2-norm.

Hence, a smaller condition number implies a faster convergence. In Algorithm 1 the complete standard CG is shown.

Algorithm 1 Standard CG

Require: $A \in \mathbb{C}^{n\times n}$, $\boldsymbol{u}_0 \in \mathbb{C}^n$, $\boldsymbol{f} \in \mathbb{C}^n$
Ensure: Approximate solution $\boldsymbol{u}_k$ such that $\|\boldsymbol{f} - A\boldsymbol{u}_k\| \leq TOL$
procedure CG($A, \boldsymbol{f}$) ▷ Solve $A\boldsymbol{u} = \boldsymbol{f}$
Initialize: $\boldsymbol{s}_0 = \boldsymbol{f} - A\boldsymbol{u}_0$; $\boldsymbol{p}_0 = \boldsymbol{s}_0$; $\gamma_0 = \boldsymbol{s}_0^* \boldsymbol{s}_0$; $k = 0$;
  while $\sqrt{\gamma_k} > TOL$ **and** $k < k_{\max}$ **do**
    $\xi_k = \boldsymbol{p}_k^* A \boldsymbol{p}_k$
    $\alpha_k = \dfrac{\gamma_k}{\xi_k}$
    $\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \alpha_k \boldsymbol{p}_k$
    $\boldsymbol{s}_{k+1} = \boldsymbol{s}_k - \alpha_k A \boldsymbol{p}_k$
    $\gamma_{k+1} = \boldsymbol{s}_{k+1}^* \boldsymbol{s}_{k+1}$
    $\beta_k = \dfrac{\gamma_{k+1}}{\gamma_k}$
    $\boldsymbol{p}_{k+1} = \boldsymbol{s}_{k+1} + \beta_k \boldsymbol{p}_k$
    $k = k + 1$
  end while
end procedure

3.1.3 CGLS

When dealing with a system $A\boldsymbol{u} = \boldsymbol{f}$ where A is not hermitian positive definite, or not even square, it is possible to use a variant of the CG method to solve the system, namely the CGLS. This variant of the CG is obtained by applying the standard CG to the system of normal equations $A^*A\boldsymbol{u} = A^*\boldsymbol{f}$. Indeed, even if A is not, A^*A is HPD and the CG can be applied. As for the standard CG, the CGLS minimizes the 2−norm of the residual at every iteration. In fact

$$\|\boldsymbol{e}_k\|_{A^*A} = \|\boldsymbol{u} - \boldsymbol{u}_k\|_{A^*A} = (\boldsymbol{u} - \boldsymbol{u}_k)^* A^* A (\boldsymbol{u} - \boldsymbol{u}_k) = (\boldsymbol{f} - A\boldsymbol{u}_k)^* (\boldsymbol{f} - A\boldsymbol{u}_k) = \|\boldsymbol{s}_k\|_2^2 \tag{3.11}$$

So the error is minimized in the A^*A-norm, while the residual in the 2−norm.
In order to enhance the stability of the algorithm, it is possible to make some modifications [6]. Indeed, if we want to include Tikhonov regularization and solve minimization problem 3.2, we want to apply the CG to 3.3 instead of the standard normal equations. So in the general form $A\boldsymbol{u} = \boldsymbol{f}$ we substitute A with $A^*C^{-1}A + \lambda R$, $\boldsymbol{u}$ by x and $\boldsymbol{f}$ by $A^*C^{-1}\boldsymbol{y}$. From 3.4 we define

$$\boldsymbol{r}_k := C^{-1}(\boldsymbol{y} - A\boldsymbol{x}_k)$$

and we can introduce a recursion for the residual

$$\begin{aligned} \boldsymbol{s}_{k+1} &= A^*C^{-1}\boldsymbol{y} - (A^*C^{-1}A + \lambda R)\boldsymbol{x}_{k+1} = A^*C^{-1}(\boldsymbol{y} - A\boldsymbol{x}_{k+1}) - \lambda R\boldsymbol{x}_{k+1} \\ &= A^*\boldsymbol{r}_{k+1} - \lambda R\boldsymbol{x}_{k+1} \end{aligned} \tag{3.12}$$

The full algorithm for solving 3.3 is shown in Algorithm 2.

Algorithm 2 CGLS

Require: $A \in \mathbb{C}^{m\times n}$, $C \in \mathbb{C}^{m\times m}$, $R \in \mathbb{C}^{n\times n}$, $\boldsymbol{x}_0 \in \mathbb{C}^n$, $\boldsymbol{y} \in \mathbb{C}^m$, $\lambda \in \mathbb{R}_{\geq 0}$
Ensure: Approximate solution $\boldsymbol{x}_k$ such that $\|A^*\boldsymbol{r}_k - \lambda R\boldsymbol{x}_k\| \leq TOL$
procedure CGLS($A, \boldsymbol{y}$) ▷ Solve $A\boldsymbol{x} = \boldsymbol{y}$
Initialize: $\boldsymbol{r}_0 = C^{-1}(\boldsymbol{y} - A\boldsymbol{x}_0)$; $\boldsymbol{s}_0 = A^*\boldsymbol{r}_0 - \lambda R\boldsymbol{x}_0$; $\boldsymbol{p}_0 = \boldsymbol{s}_0$; $\boldsymbol{q}_0 = A\boldsymbol{p}_0$;$\gamma_0 = \boldsymbol{s}_0^*\boldsymbol{s}_0$; $k = 0$;
  while $\sqrt{\gamma_k} > TOL$ **and** $k < k_{\max}$ **do**
    $\xi_k = \boldsymbol{q}_k^* C^{-1}\boldsymbol{q}_k + \lambda \boldsymbol{p}_k^* R\boldsymbol{p}_k$
    $\alpha_k = \dfrac{\gamma_k}{\xi_k}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
    $R\boldsymbol{x}_{k+1} = R\boldsymbol{x}_k + \alpha_k R\boldsymbol{p}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k C^{-1}\boldsymbol{q}_k$
    $\boldsymbol{s}_{k+1} = A^*\boldsymbol{r}_{k+1} - \lambda R\boldsymbol{x}_{k+1}$
    $\gamma_{k+1} = \boldsymbol{s}_{k+1}^*\boldsymbol{s}_{k+1}$
    $\beta_k = \dfrac{\gamma_{k+1}}{\gamma_k}$
    $\boldsymbol{p}_{k+1} = \boldsymbol{s}_{k+1} + \beta_k \boldsymbol{p}_k$
    $\boldsymbol{q}_{k+1} = A\boldsymbol{p}_{k+1}$
    $k = k + 1$
  end while
end procedure

3.1.4 CGNE

CGLS is not the only variant of the conjugate gradient. The Conjugate Gradient Normal Error (CGNE) denotes the CG applied to the system $AA^*\boldsymbol{z} = \boldsymbol{y}$, where we set $\boldsymbol{x} = A^*\boldsymbol{z}$.
An interesting fact to notice about CGNE is that, unlike CGLS, the error $\boldsymbol{e}_k$ is minimized in the 2-norm. In fact,

$$\|\boldsymbol{x} - \boldsymbol{x}_k\|_{AA^*}^2 = (A(\boldsymbol{z} - \boldsymbol{z}_k))^*(A(\boldsymbol{z} - \boldsymbol{z}_k)) = (\boldsymbol{x} - \boldsymbol{x}_k)^*(\boldsymbol{x} - \boldsymbol{x}_k) = \|\boldsymbol{e}_k\|_2^2 \tag{3.13}$$

This method is a natural choice when dealing with underdetermined problems, which is quite often the case when talking about SRR, since it is not always possible to have access to enough data. The reason lies in the fact that for an underdeterminate problem AA^* has a lower dimension than A^*A. Algorithm shows the complete pseudo-code for CGNE.

Algorithm 3 CGNE

Require: $A \in \mathbb{C}^{m\times n}$, $C \in \mathbb{C}^{m\times m}$, $R \in \mathbb{C}^{n\times n}$, $\boldsymbol{x}_0 \in \mathbb{C}^n$, $\boldsymbol{y} \in \mathbb{C}^m$, $\lambda \in \mathbb{R}_{\geq 0}$
Ensure: Approximate solution $\boldsymbol{x}_k$ such that $\left\|\boldsymbol{y} - A\boldsymbol{x}_k - C\boldsymbol{r}_k\right\| \leq TOL$

procedure CGNE($A, \boldsymbol{y}$) ▷ Solve $A\boldsymbol{x} = \boldsymbol{y}$
  if $\lambda > 0$ **then**
    $\boldsymbol{x}_0 = \frac{1}{\lambda} R^{-1} A^* \boldsymbol{r}_0$
  else
    $\boldsymbol{x}_0 = 0$; $\boldsymbol{r}_0 = 0$
  end if
  Initialize: $\boldsymbol{s}_0 = \boldsymbol{y} - A\boldsymbol{x}_0 - C\boldsymbol{r}_0$; $\boldsymbol{p}_0 = \boldsymbol{s}_0$; $\boldsymbol{q}_0 = A^*\boldsymbol{p}_0$;$\gamma_0 = \boldsymbol{s}_0^*\boldsymbol{s}_0$; $k = 0$;
  while $\sqrt{\gamma_k} > TOL$ **and** $k < k_{\max}$ **do**
    $\xi_k = \boldsymbol{q}_k^* R^{-1} \boldsymbol{q}_k + \lambda \boldsymbol{p}_k^* C \boldsymbol{p}_k$
    $\alpha_k = \frac{\gamma_k}{\xi_k}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k R^{-1} \boldsymbol{q}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \lambda \alpha_k \boldsymbol{p}_k$
    $\boldsymbol{s}_{k+1} = \boldsymbol{s}_k - \alpha_k (AR^{-1}\boldsymbol{q}_k + \lambda C \boldsymbol{p}_k)$
    $\gamma_{k+1} = \boldsymbol{s}_{k+1}^* \boldsymbol{s}_{k+1}$
    $\beta_k = \frac{\gamma_{k+1}}{\gamma_k}$
    $\boldsymbol{p}_{k+1} = \boldsymbol{s}_{k+1} + \beta_k \boldsymbol{p}_k$
    $\boldsymbol{q}_{k+1} = A^* \boldsymbol{p}_{k+1}$
    $k = k + 1$
  end while
end procedure

3.2 Methods of Total Variation

In the previous sections we presented methods to solve 3.3, thus finding the optimal solution for minimization problem 3.2 with Tikhonov regularization. Nevertheless, in section 2.3.2 we talked about another kind of regularization, Total Variation.
When dealing with this kind of regularization, it's not possible to directly apply CG or its variants due to the presence of the l_1-penalty. ADMM (Alternating Directions Method of Multipliers) is a possible method to solve minimization problem 2.17, which we recall being

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 + \frac{1}{2}\lambda \|F\boldsymbol{x}\|_1.$$

In [5] ADMM is explained. First, the objective function is split into two functions

$$h(\boldsymbol{x}) := \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2, \quad g(F\boldsymbol{x}) := \frac{\lambda}{2}\|F\boldsymbol{x}\|_1,$$

so that the minimization problem can be written as

$$\begin{aligned} &\min_{\boldsymbol{x},\boldsymbol{v}} h(\boldsymbol{x}) + g(\boldsymbol{v}) \\ &\text{subject to } F\boldsymbol{x} = \boldsymbol{v}. \end{aligned} \tag{3.14}$$

The idea behind this method is to approximately perform the optimization of $(\boldsymbol{x}, \boldsymbol{v})$ in two steps using alternating minimization of $\boldsymbol{x}$ and $\boldsymbol{v}$. Despite not giving exactly the same result as 2.17 would give, it should converge

to the same solution and much easier to obtain. Indeed, the functions $h(\boldsymbol{x})$ and $g(\boldsymbol{v})$ are straightforward to minimize individually.

4 Dataset

On the 28th of January 2020 some measurements were performed at LUMC using a pear (and some other fruit and vegetables) as phantom. In Figure 2 of section 1.1 a few samples were already shown, where we can see how the pear is shifted in the z direction of 1.7 cm.

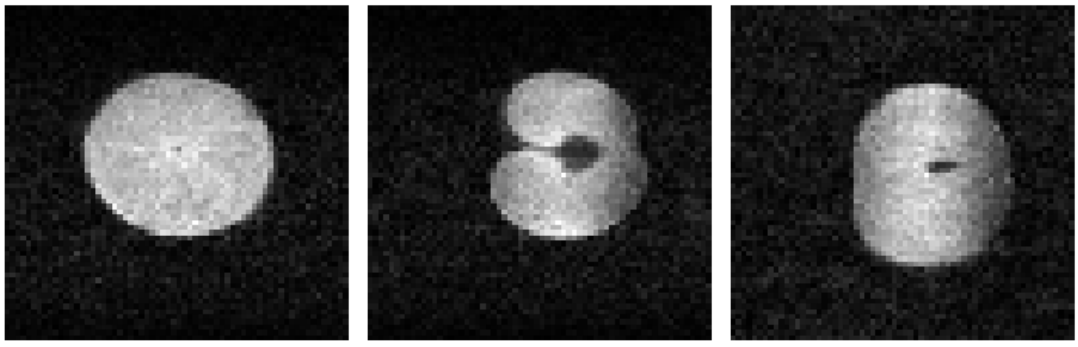
The data acquired have a resolution of $64 \times 64 \times 32$ in the x, y and z directions respectively. We have isotropic voxels in the xy plane (2×2 mm), and anisotropic voxels in the xz and yz planes (2×4 mm). The coarser resolution in the z direction helped reducing the scanning times, which is 2 times less than if we had already acquired the data with voxels of $2 \times 2 \times 2$ mm.

Unfortunately, the shift between the different samples was too big to be able to apply SRR (which we remind, requires sub-pixel shifts), which led to new measurements carried out on the 27th of February 2020. The first results in section 5 were obtained on these data.

Again, a pear and an apple were imaged with isotropic voxels in the xy plane (2×2 mm), and anisotropic voxels in the xz and yz planes (2×8 mm). The resolution is $64 \times 64 \times 16$ pixels. Shifts of 2 mm were made, thus we obtained 4 measurements shifted of 1/4 of a pixel one from the other (sub-pixel shifts). Furthermore, a $2 \times 2 \times 2$ mm^3 sample was acquired, in order to compare the SRR result with it. In this literature review we will only consider the scans of the apple, leaving the other data for future experiments. Table 1 groups some data concerning the various measurements. In Appendix A further details about the acquisition parameters for one of the LR samples.

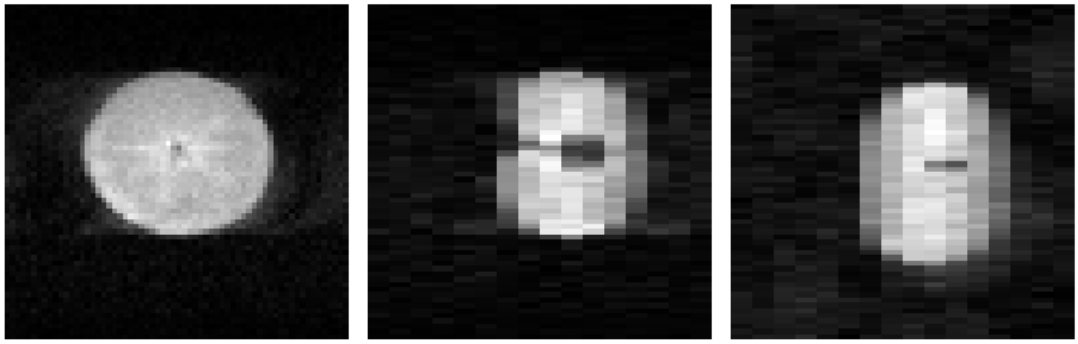
Table 1: MRI measurements performed at LUMC on 28-01 and 27-02-2020.

Date	Object	Voxel size	Resolution	FOV read	Shifts	N. of scans
28-01-20	Pear	$2 \times 2 \times 4$ mm^3	$64 \times 64 \times 32$	128	0,17,34,51,68,85 mm	6
28-01-20	Pointy pepper	$2 \times 2 \times 4$ mm^3	$64 \times 64 \times 32$	128	0,17,34,51,68,85 mm	6
27-02-20	Pear	$2 \times 2 \times 2$ mm^3	$64 \times 64 \times 64$	128	0 mm	1
27-02-20	Pear	$2 \times 2 \times 8$ mm^3	$64 \times 64 \times 16$	128	$0, 2, 4, 6$ mm	4
27-02-20	Apple	$2 \times 2 \times 2$ mm^3	$64 \times 64 \times 64$	128	0 mm	1
27-02-20	Apple	$2 \times 2 \times 8$ mm^3	$64 \times 64 \times 16$	128	$0, 2, 4, 6$ mm	4



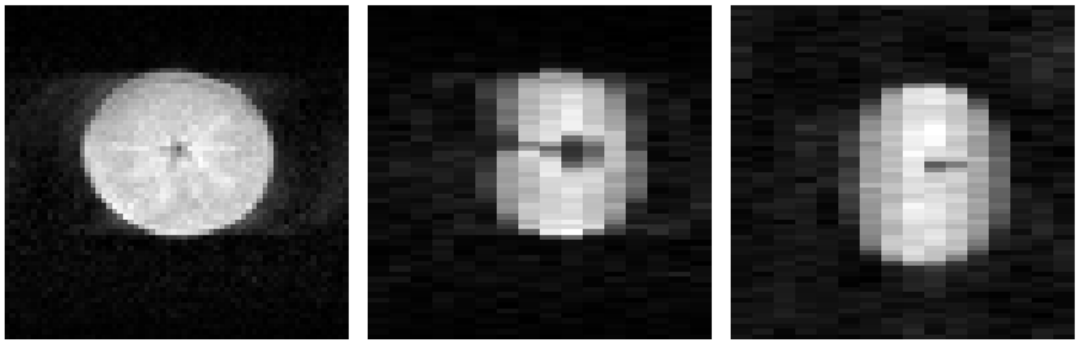
(a) xy plane (b) yz plane (c) xz plane

Figure 5: HR apple scan. Position 0.



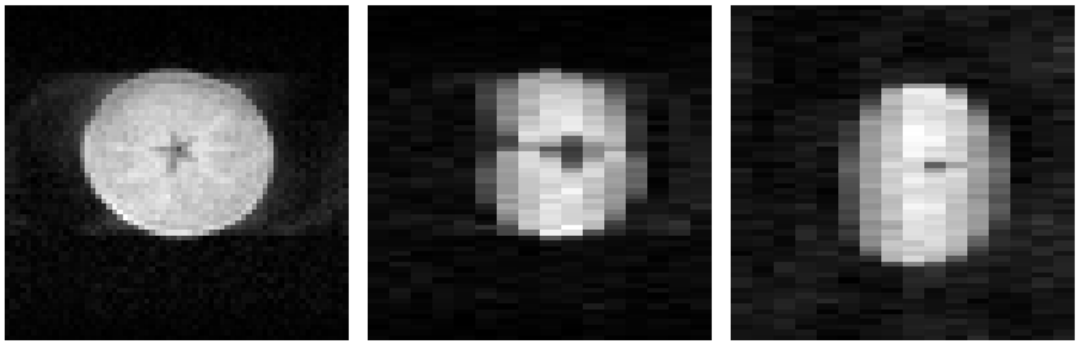
(a) xy plane (b) yz plane (c) xz plane

Figure 6: LR apple scan. Position 0 (0 mm shift).



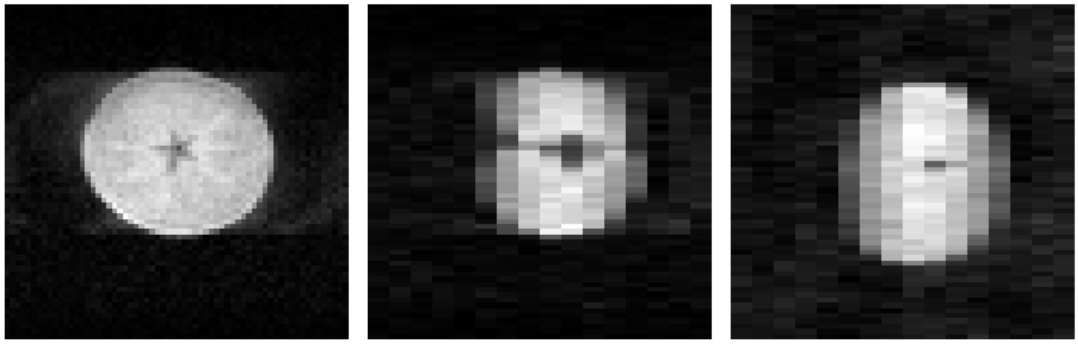
(a) xy plane (b) yz plane (c) xz plane

Figure 7: LR apple scan. Position 1 (2 mm shift).



(a) xy plane (b) yz plane (c) xz plane

Figure 8: LR apple scan. Position 2 (4 mm shift).



(a) xy plane (b) yz plane (c) xz plane

Figure 9: LR apple scan. Position 3 (6 mm shift).

5 Initial Results

In this section some preliminary results on both the Shepp-Logan phantom and real data are shown.

5.1 Observation Model

Firstly, we implemented the observation model. The three matrices D, B and G_k in system 2.2 where built.

A shift to the right in the x-direction was chosen as geometric deformation in both cases. Since we are dealing with gray-scale images, every pixel is identified by a unique value, which normalized is between 0 and 1. When shifting, it is clear that part of the image will go beyond the borders, and on the left we will have pixels missing. Therefore the images were padded with zeros. That is coherent with the type of data that we are considering, since in MRI scans the background is black (i.e. pixel value 0) and the central part is the one generally containing the information.

Having a shift of a in the right direction, we can decompose it into integer and decimal part:

$$a = \lfloor a \rfloor + a_{\text{dec}}. \tag{5.1}$$

When the image is shifted of a fraction of the pixel length, there will be an overlap between the pixel themselves. Given a pixel (i, j) of image X, its new value will be a weighted average of the value of the neighboring pixels on their left and right. Calling $\tilde{X}$ the shifted image, we have

$$\tilde{X}_{i,j} = (1 - a_{\text{dec}})\, X_{i,j-\lfloor a \rfloor} + a_{\text{dec}}\, X_{i,j-\lfloor a \rfloor - 1} \tag{5.2}$$

The down-sampling was implemented by doing an averaging of neighboring pixels. Having a HR image of size $L_1 N_1 \times L_2 N_2$, where L_1 and L_2 are the down-sampling factors in the y and x direction respectively, then the down-sampled image is obtained by averaging over $L_1 L_2$ pixels. In our case, since we only had a shift in the x-direction, down-sampling was implemented in 1D only, so $L_2 = 1$, $L_1 = L$.

Finally, the blurring was modeled through the PSF. As already mentioned in Section 2.1, the PSF describes the blurring of one pixel over its surrounding pixels. The function was modeled as a zero-mean Gaussian with standard deviation σ, as suggested by most articles including [17]. Similarly to the down-sampling operator, also the blurring was modeled in one direction only. This led to a PSF array $\boldsymbol{p}$, where p_i is given by

$$p_i = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{i^2}{2\sigma^2}} \tag{5.3}$$

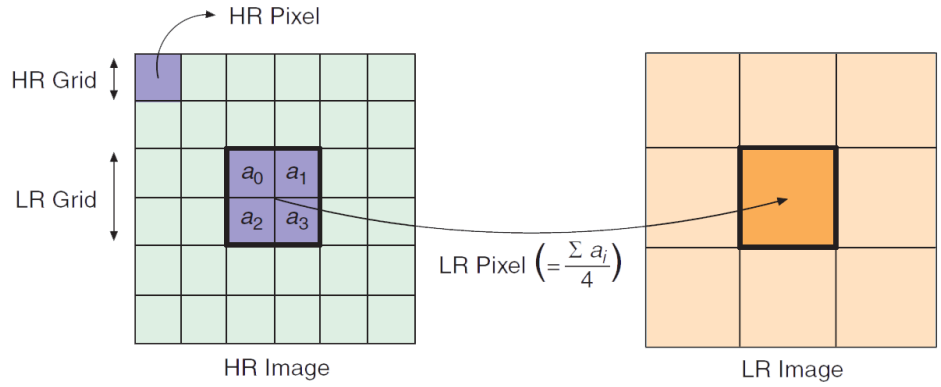


Figure 10: Down-sampling by averaging when $L_1 = L_2 = 2$. Source [14].

In order to obtain the blurred image $\overline{X}$, $\boldsymbol{p}$ is convoluted with the image X as described in [9]. The convolution is performed by rotating $\boldsymbol{p}$ of 180°, then placing it on top of the image horizontally so that the central element in P overlaps with element $X_{i,j}$. Finally, the corresponding components are multiplied and the results are summed, resulting in $\overline{X}_{i,j}$.

An important thing to notice is that when placing $\boldsymbol{p}$ over X, some of the values of the PSF array could not match with any value in X and go beyond the borders of the image. That is why it is necessary to include *boundary conditions* in our discussion. We chose the so-called "Zero Boundary Conditions", hence the image was padded with zeros. Putting the image X into lexicographic form $\boldsymbol{x}$ as in 2.2, the blurring matrix B was built so that $B\boldsymbol{x}$ would result in the blurred image. Below an example of how to obtain the matrix B when X is a 2×3 matrix and $\boldsymbol{p}$ has length 5.

$$\boldsymbol{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix}, \qquad X = \begin{bmatrix} x_1 & x_3 & x_5 \\ x_2 & x_4 & x_6 \end{bmatrix}, \qquad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}.$$

$$\begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \overline{x}_3 \\ \overline{x}_4 \\ \overline{x}_5 \\ \overline{x}_6 \end{bmatrix} = \underbrace{\begin{bmatrix} p_3 & p_2 & p_1 & & & \\ & p_3 & p_2 & p_1 & & \\ p_4 & & p_3 & p_2 & & \\ & p_4 & & p_3 & p_2 & \\ p_5 & & p_4 & & p_3 & \\ & p_5 & & p_4 & & p_3 \end{bmatrix}}_{B} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \tag{5.4}$$

5.2 Phantom

For testing purposes, a 256×256 Shepp-Logan phantom was used as HR image (see Figure 11). Four LR images were created, by shifting, blurring and down-sampling the HR version. The first was created by blurring and down-sampling, but no shifting. The other three images had shifts in the x-direction of $0.25, 0.5$ and 0.75

pixels with respect to the first. Note that in order to create a LR image which is shifted of a from the original image, the HR image needs to be shifted of $\bar{a} = aL$, where L is the down-sampling factor. This way, the LR image resulting from the down-sampling will be shifted of exactly a.

Figure 11: HR Shepp-Logan phantom. 256×256.

The down-sampling factor was chosen as $L = 4$, and the standard deviation for the blurring $\sigma = 2$. The final LR images are then 256×64. The process is illustrated in Figure 12.
The four LR images were then transformed into vectors by lexicographical ordering and stacked, leading to the vector $\boldsymbol{y}$ of 2.4. The matrices D, B and G_k where multiplied and stacked, resulting in A. $\boldsymbol{y}$ and A have dimensions respectively 65536×1 and 65536×65536. In fact $L*256*64 = 65536$, just like $256*256 = 65536$.

Firstly, we decided to compare the result of the Python built-in solver for least-squares with sparse matrices with the result of the CGLS without regularization. The default tolerance for the `scipy.sparse.linalg.lsqr` function is 10^{-8}, and it took ~ 30 sec. to reach that tolerance. Due to the availability of the original HR image, we were able to compare the reconstructed images not only visually, but also through the relative error between the reconstructed image $\boldsymbol{x}_r$ and the original image $\boldsymbol{x}$. For the built-in solver, the relative error in the l_1-norm was 0.025. It took the CGLS a tolerance of 10^{-10}, 3702 iterations and ~ 59 sec. to reach the same relative error. The resulting images are shown in Figure 13). It is clear that they are qualitatively comparable from a visual point of view. We notice no significant difference and the results are highly satisfactory.

Table 2: l_1 and l_2-norm of the relative error between $\boldsymbol{x}$ and $\boldsymbol{x}_r$ for CGLS.

λ	10^{-20}	10^{-15}	10^{-7}	10^{-5}
l_1	0.031	0.036	0.159	0.244
l_2	0.023	0.025	0.065	0.117

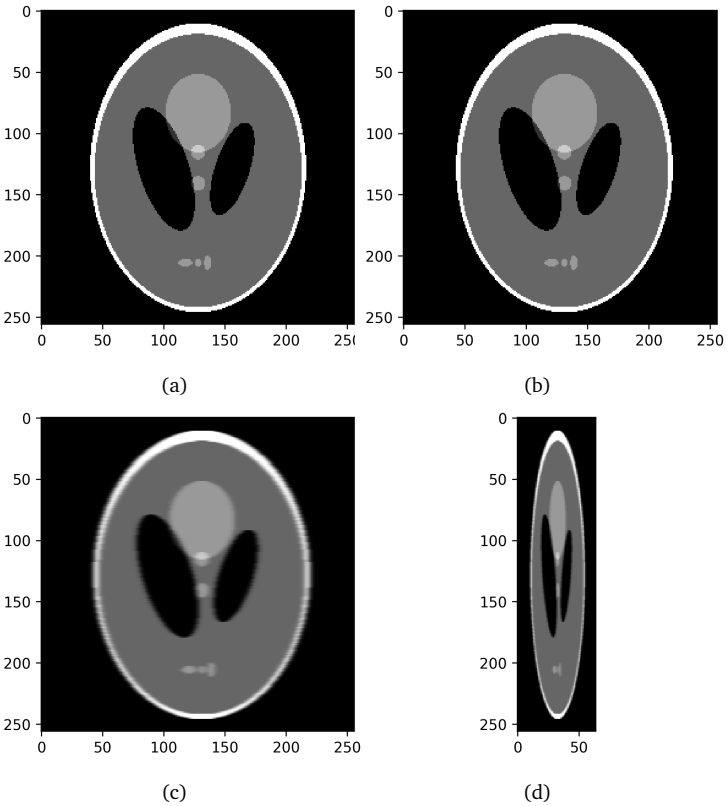


Figure 12: From (a) to (d), image is shifted of $3 = 4 * 0.75$ pixels, blurred and down-sampled.

Afterwards, we wanted to analyze the results for the CGLS with a second-order difference matrix with Neumann boundary conditions as he regularization matrix R. The same number of maximum iterations was used, and the same tolerance of 10^{-10}.
Figure 14 shows four reconstructions for different values of λ, and Table 2 contains the relative errors.
The CGLS performs well for $\lambda = 10^{-20}, 10^{-15}$, giving visually good results, as well as quite low relative errors. The features of the HR image are clearly recognizable and the images are very similar to the ones in 13. In Figure 14c we see how the image starts getting blurrier and in 14d we see the smoothing due to the higher regularization parameter.

In this particular case we obtain better results by not regularizing or by using a very low λ (which actually does not yield any regularization). The edges in the resulting images are much sharper in these cases. This was to be expected, since the original image is not smooth, and regularizing means imposing a certain level of smoothness on the image.

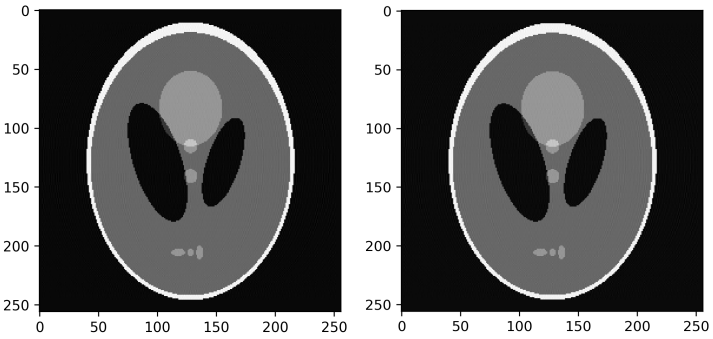


Figure 13: (a) `scipy.sparse.linalg.lsqr` vs (b) non-regularized CGLS. Relative error in l_1-norm: 0.025.

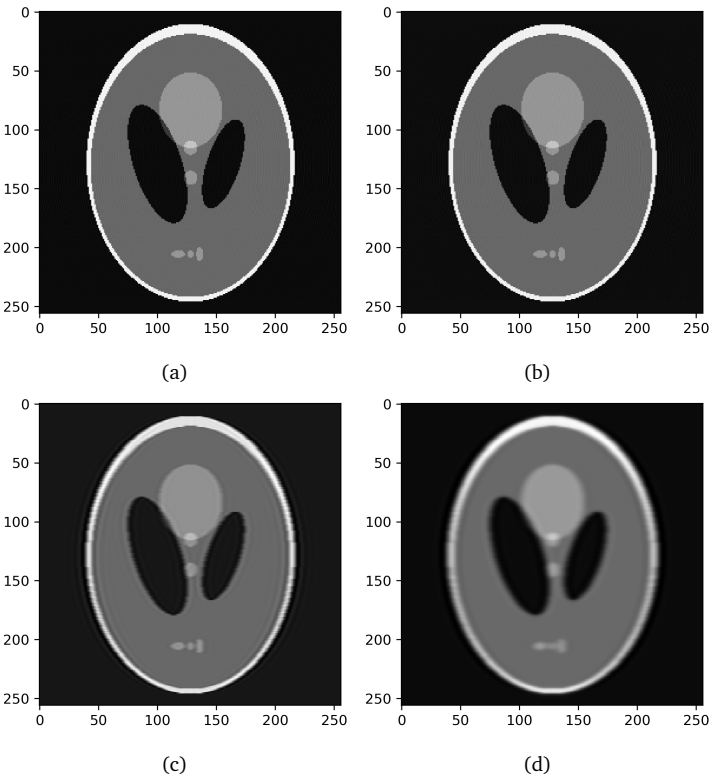
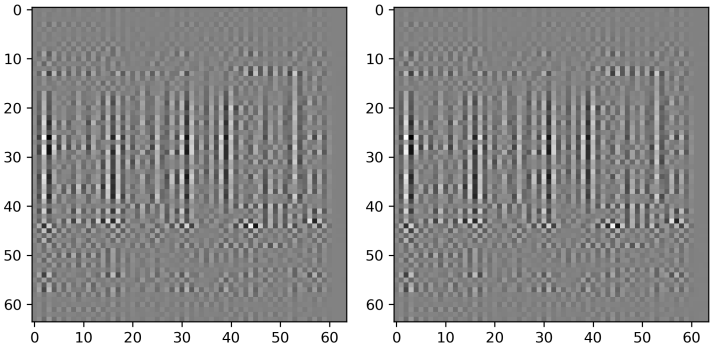


Figure 14: From (a) to (d), phantom reconstructions for different values of λ: $10^{-20}, 10^{-15} 10^{-7}, 10^{-5}$.

5.3 Real Data

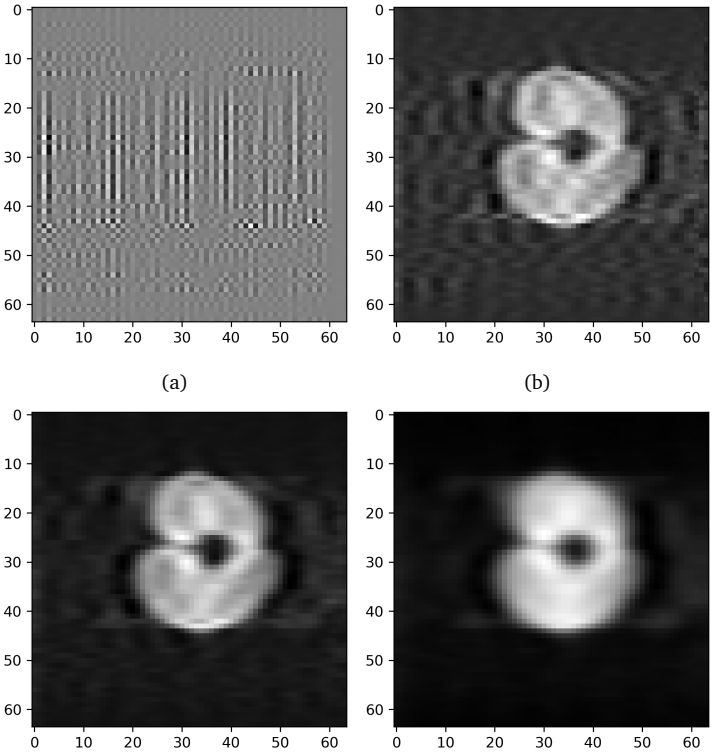
After experimenting with the Shepp-Logan phantom, the next step was using real data. The set of apple scans described in Section 4 was used. SRR was applied in the yz plane. We have four LR images of size 64×16. Hence, we have again a down-sampling factor $L = 4$ in the x-direction. The final matrix A is 4096×4096, $\boldsymbol{y}$ is a 4096×1 vector, just like the unknown $\boldsymbol{x}$.
We took the same steps as in the previous section, by first comparing the non-regularized CGLS with the built-

in Python solver, and then analyzing the results for the regularized CGLS. Because of the availability of a HR scan ($2 \times 2 \times 2$ voxels), we could compare it with the reconstructed images and again, set our parameter λ accordingly.



(a) (b)

Figure 15: (a) `scipy.sparse.linalg.lsqr` vs (b) non-regularized CGLS. Relative error in l_1-norm: ~ 12191.



(a) (b)

(c) (d)

Figure 16: From (a) to (d), apple reconstructions for different values of λ: $10^{-20}, 10^{-6}, 10^{-5}, 10^{-4}$.

We can clearly see from Figure 15 that, unlike the results we had for the phantom, in this situation the absence of a regularization results in both the algorithms not being able to reconstruct the image. This is also confirmed by the relative error, being ~ 12191 in both cases. In Table 3 the relative errors are shown, and we can notice a steep decrease in the values when going from 16a to 16b. Furthermore, we notice how the values keep decreasing as the image gets smoother. Image 16d though results too smooth and blurred after a visual assessment, and the shape of the yz plane of the apple in more recognizable in 16c. However, the colors of 16d (i.e. the pixel values) resembles the ones of Figure 5 slightly more, having less noise in the central row and a subject with softer color transitions. We deem then that this could be the reason for a lower score for the last image. We conclude that a value of λ between 10^{-6} and 10^{-5} could be a good trade-off.

Overall, we have shown that regularization is certainly needed, hence CGLS outperforms the built-in Python solver. Image 16c represents a huge improvement with respect to the LR images of Section 4.

There is, of course, room for improvement, as for instance using a more rigorous criterion when choosing λ and work on the *registration*. Indeed, the shifts between samples are assumed to be of a certain length, but we are aware that it is an approximation. The estimation of motion information is referred to as registration [14]. This is also certainly a reason for the relative error not being as low as it would b expected. Indeed, the HR image that we are taking into consideration is probably not exactly aligned with the reconstructed image, resulting in a higher error.

Table 3: l_1 and l_2-norm of the relative error between $\boldsymbol{x}$ and $\boldsymbol{x}_\text{r}$ for CGLS.

λ	10^{-20}	10^{-6}	10^{-5}	10^{-4}
l_1	12239.943	2.876	2.790	2.718
l_2	10222.370	3.090	3.075	3.023

6 Research Questions

This literature review is part of a thesis research project which aims to give an answer to the following questions:

1. **Main question:** Can SRR yield comparable results to images directly acquired in high resolution?
2. Is SRR in the frequency domain a valid approach, and how does it compare to the spatial domain techniques?
3. Is it possible to obtain comparable or better results than standard techniques when using a Deep Learning approach?
4. Which of these approaches yields the best results on real data?

Along the way of answering those questions, several steps have to be made. We will now list a few ideas for each of the research questions, numerated accordingly.

1. Initial results have already been given in Section 5, but there are still several aspects that can be improved and more methods to implement.

   A standard Gaussian kernel has been implemented, but a proper modeling of the Point Spread Function (PSF) is missing at the moment. Furthermore, it is unclear at the moment how to determine the parameters for the kernel, such as the length and the value of the standard deviation, when dealing with real data.

   The L-curve method or other rigorous ways to determine the value of the regularization parameter λ are yet to be implemented.

   Registration should be implemented, and would probably result in a great improvement in the quality of the images.

   The only solver implemented at the moment is CGLS, which solves the minimization problem for Tikhonov regularization. The next step is to implement CGNE, as well as a solver for Total Variation regularization.

2. Frequency domain approach is described in [14, 20]. Although is it theoretically intuitive, it has been discarded in favor of spatial domain methods due to some relevant limitations. The observation model is indeed restricted to only global translational motion and Linear Space Invariant (LSI) blur.

   Nevertheless, since we are conducting this research under these very hypotheses, we are interested in exploring the possibilities of the frequency domain approach in comparison to spatial domain techniques.

3. In 2014 a deep learning method for SRR was first presented [7]. Since then, these techniques have been greatly explored. We are interested in implementing selected deep learning techniques and draw a comparison with the previously mentioned standard ones.

4. In order to evaluate the methods a visual analysis was made. Furthermore, when possible the l_1 and l_2-norms of the error between HR image and reconstructed image where compared and analyzed as well. Other Image Quality Assessment (IQA) tools might be used.

Note that the previous side research questions and the steps to pursue them might be subject to modifications along the way, in order to serve the main question in the best possible way.

References

[1] *Hydrocephalus Fact Sheet*. https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Fact-Sheets/Hydrocephalus-Fact-Sheet. Accessed: 24-03-2020.

[2] D. J. Bell, U. Bashir, et al., *k-space*. https://radiopaedia.org/articles/k-space-1. Accessed: 01-04-2020.

[3] D. J. Bell, J. Jones, et al., *Net magnetization vector*. https://radiopaedia.org/articles/net-magnetisation-vector. Accessed: 31-03-2020.

[4] A. Berger, *How does it work?: Magnetic resonance imaging*, BMJ: British Medical Journal, 324 (2002), p. 35.

[5] C. A. Bouman, *Model based image processing*, Purdue University, (2013).

[6] M. L. de Leeuw den Bouter, *Image reconstruction in low-field MRI: a super-resolution approach*, Master's thesis, Delft University of Technology, (2017).

[7] C. Dong, C. C. Loy, K. He, and X. Tang, *Learning a Deep Convolutional Network for Image Super-Resolution*, in ECCV (4), D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., vol. 8692 of Lecture Notes in Computer Science, Springer, (2014), pp. 184–199.

[8] P. C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM review, 34 (1992), pp. 561–580.

[9] P. C. Hansen, J. G. Nagy, and D. P. O'leary, *Deblurring images: matrices, spectra, and filtering*, vol. 3, Siam, 2006.

[10] M. R. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409–436.

[11] D. Hoving, *MRI for Africa*, Delta, Journalistic platform TU Delft, (2017). Accessed: 24-03-2020.

[12] Z.-P. Liang and P. C. Lauterbur, *Principles of magnetic resonance imaging: a signal processing perspective*, SPIE Optical Engineering Press, (2000).

[13] T. O'Reilly, W. Teeuwisse, and A. Webb, *Three-dimensional MRI in a homogenous 27 cm diameter bore Halbach array magnet*, Journal of Magnetic Resonance, 307 (2019), p. 106578.

[14] S. C. Park, M. K. Park, and M. G. Kang, *Super-resolution image reconstruction: a technical overview*, Signal Processing Magazine, IEEE, 20 (2003), pp. 21–36.

[15] D. H. J. Poot, V. V. Meir, and J. Sijbers, *General and Efficient Super-Resolution Method for Multi-slice MRI.*, in MICCAI (1), T. Jiang, N. Navab, J. P. W. Pluim, and M. A. Viergever, eds., vol. 6361 of Lecture Notes in Computer Science, Springer, (2010), pp. 615–622.

[16] M. Sarracanie, C. D. LaPierre, N. Salameh, D. E. Waddington, T. Witzel, and M. S. Rosen, *Low-cost high-performance MRI*, Scientific reports, 5 (2015), pp. 1–9.

[17] E. Van Reeth, I. W. K. Tham, C. H. Tan, and C. L. Poh, *Super-resolution in magnetic resonance imaging: A review*, Concepts in Magnetic Resonance Part A, 40A (2012), pp. 306–325.

[18] C. Vuik and D. Lahaye, *Scientific computing*, Lecture notes for WI4201, (2012).

[19] M. WIJCHERS, *Image reconstuction in MRI: The possibilities of portable low-cost MRI scanners*, Master's thesis, Delft University of Technology, (2016).

[20] J. YANG AND T. HUANG, *Image super-resolution: Historical overview and future challenges*, CRC Press, 1 2017, pp. 1–33.

A Measurements 27-02

Acquisition parameter for one sample (LR apple).

```
90Amplitude = -28
180Amplitude = -22
accumulate = "yes"
acqTime = 3.2
autoPhase = "no"
b1Freq = 2.14882d
bandwidth = 20
dataDirectory = "C:\Users\Low Field\Desktop\20200227 - Super resolution"
dwellTime = 50.0
echoTime = 20e3
etLength = 32
experiment = "TSE3D_V2_1"
expName = "TSE3D_v2.1"
expNr = 16
filter = "no"
filterType = "none"
flatFilter = "yes"
FOVphase1 = 128
FOVphase2 = 128
FOVread = 128
gradCorr = 480
incExpNr = "yes"
kTraject = "In-out"
nPhase1 = 64
nPhase2 = 16
nrPnts = 64
nrScans = 1
plane = "yz"
position = [48,143]
pulseLength = 100
repTime = 2e3
rxGain = 31
rxPhase = 0
saveData = "true"
timeMag = "yes"
usePhaseCycle = "yes"
windowSize = "large"
```