Image Restoration by blind deconvolution

Wolfgang Stefan

2003/06/27

Diploma Thesis in Techno - Mathematics

Technische Universität München (TUM)
Arizona State University (ASU)

Advisers:
ASU: Prof. Dr. Rosemary Renaut
TUM: Prof. Dr. Rupert Lasser

Abstract

In this Diploma Thesis we will present some methods to improve the quality of a given pictures. In particular blind deconvolution will be applied to deblur the images. The deconvolution tries to invert the blurring of an image that is modeled by the convolution \( g = f \ast h \). Blind deconvolution tries to do this without knowledge of the point spread function that blurred the image. In the Thesis first the degradation model is described as well as problems that have to be addressed when applying deconvolution i.e. the extreme ill posedness of the deconvolution and the singularity of the blind deconvolution. A method of inverse filtering is presented, the NASRIF by Deepa Kundur and a more promising approach using total variation (TV) and Tikhonov (TK) regularization is shown. Numerical experiments presented in this thesis show that it is beneficial to use TV on \( f \) but TK regularization on the point spread function \( h \) rather then using TV on both or TK on both as done in previous publications. The reader can easily reproduce the experiments that lead to this conclusion by using the Matlab-GUI attached to this document. At the end it is shown how to carry out the deblurring in physical space as well as in Fourier space. The best results can be achieved by restoring the image in Fourier space but the point spread function in physical space.
# Contents

## 1 Introduction
- 1.1 Image restoration in medical imaging .................................................. 7
- 1.2 Existing image restoration approaches .................................................. 7

## 2 Degradation Models
- 2.1 Discrete model ......................................................................................... 9
  - 2.1.1 A priori constraints ........................................................................... 10
- 2.2 Point spread functions ............................................................................ 10
  - 2.2.1 Gauss blur ....................................................................................... 11
- 2.3 Out-of-focus blur ..................................................................................... 12
- 2.4 Motion blur ............................................................................................. 13
- 2.5 Noise ....................................................................................................... 13

## 3 Problem properties
- 3.1 Singular value decomposition [13, 12] ...................................................... 15

## 4 Theoretical Solution Techniques
- 4.1 Matrix inversion ....................................................................................... 19
- 4.2 Inverse filtering ......................................................................................... 19
- 4.3 A Forward Model ..................................................................................... 22
  - 4.3.1 Fit-to-data functional ......................................................................... 22
  - 4.3.2 Regularization .................................................................................... 23
  - 4.3.3 Constrained minimization ................................................................. 23

## 5 Tikhonov regularization
- 5.1 Non-blind deconvolution ......................................................................... 25
  - 5.1.1 Choosing the parameter (L-curve) ....................................................... 25
- 5.2 Blind deconvolution ................................................................................. 26
  - 5.2.1 Alternating Minimization (AM) .......................................................... 26
  - 5.2.2 Global behavior ................................................................................ 29

## 6 Total variation regularization
- 6.1 Non-blind deconvolution ......................................................................... 33
- 6.2 Blind deconvolution ................................................................................. 36

## 7 Numerical implementation and examples
- 7.1 1 dimensional signals .............................................................................. 38
  - 7.1.1 1-d slice of a MRI scan .................................................................... 40
- 7.2 Restoration of images .............................................................................. 42
  - 7.2.1 Restoration in Fourier space .............................................................. 44
  - 7.2.2 2d examples ..................................................................................... 45

## 8 Conclusions ................................................................................................. 49
List of Tables

1  Condition numbers of some convolution matrices . . . . . . . . . . . . . . . . . 15
2  Noise amplification after matrix inversion . . . . . . . . . . . . . . . . . . . 19
3  Alternating minimization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
List of Figures

1. The Shepp-Logan Phantom ........................................... 11
2. Example of a Gaussian point spread function ......................... 11
3. Example of a Gaussian blurred image .................................. 12
4. Example of an out-of-focus psf ....................................... 12
5. An out-of-focus example ................................................ 13
6. Example of a motion psf ................................................. 13
7. A motion blur example ................................................... 14
8. Example of a MRI picture with a noise level of 9% ..................... 14
9. 1d slice of a MRI scan with 9% noise ................................ 15
10. Singular values of a Gauss psf .......................................... 16
11. 1d Matrix inversion example. The Figure shows the test problem as dashed line, the blurred version as dotted line and the reconstruction using matrix inversion as solid line. ......................................................... 17
12. Plot of the singular values of a Gauss psf (o) and the coefficients $u_i^T g$ of the test problem seen in Figure 11 (□) with no noise on the blurred function. It can be seen that none of the larger coefficients are larger than the singular values. ............................... 18
13. Plot of the singular values of a Gauss psf (o) and the coefficients $u_i^T g$ for a noise level of $10^{-7}$ (x) and $10^{-2}$ (□). It can be seen that in both cases some of the coefficients are larger than the singular values. .............................................. 18
14. Effect of blurring to the support. The process of bluring the image increases the support of the image. ................................................. 20
15. Example of a restoration using the NASRIF algorithm .................. 21
16. Point spread function and the approximation of its inverse ............. 21
17. Impact of noise on the NASRIF-reconstruction .......................... 22
18. Tikhonov non blind deconvolution. 10% noise was added to the blurred function. ............................ 25
19. Tikhonov non blind deconvolution. No noise was added to the blurred function. .............................. 26
20. The generic form of the l-curve, taken from [12] ............................ 27
22. Example of a blind deconvolution using Tikhonov regularization on $f$ and $h$. In this case the AM was carried out with only a few steps in the direction of $h$ in order to preserve some information of the initial condition after the minimization w.r.t. $h$. The algorithm was started at the exact solution i.e. with the true $f$ and the true psf $h$. ................................................................. 30
23. Example of a blind deconvolution using Tikhonov Regularization on $f$ and $h$. The algorithm was started with the observed signal for $f$ and a delta-function for $h$. ............................. 31
24. Example of a blind deconvolution using Tikhonov Regularization on $f$ and $h$. The algorithm was started with the observed signal for $f$ and a large out-of-focus psf for $h$. ......................................................... 32
25. Total variation nonblind deconvolution. 10% noise was added to the blurred function. ................................. 34
26. Total variation nonblind deconvolution. 10% noise was added to the blurred function. ................................. 35
27. TV based denoising, taken from [22] ....................................... 36
28. TV regularized blind deconvolution started at the given data and a rough estimation of the psf. ................................. 37
29. TV regularized blind deconvolution started at the given data and a delta function. ................................. 38
Example of a practical problem. The given data is taken from a simulation of a MRI scan. The true psf in this case is not available. The true signal is also not available and in the figure only an approximation of the true signal.

Shepp-Logan Phantom

Shepp-Logan Phantom blurred by a gauss psf with noise of a level of about 20%

Shepp-Logan Phantom deblurred by non blind deconvolution with TV regularization. $$\lambda_f = 100$$

Shepp-Logan Phantom deblurred by non blind deconvolution with TV regularization. $$\lambda_f = 1000$$

Shepp-Logan Phantom deblurred by blind deconvolution with TV regularization.

Point spread function of the restoration shown in Figure 35

Unprocessed image of Jupiter

Restored image of Jupiter using blind deconvolution

Point spread function belonging to the Jupiter restoration
1 Introduction

The goal of image restoration is to find uncorrupted images from noisy, blurred ones. There are many applications in different fields such as astronomy and medical imaging. Mostly a blurred image is given, for example blurred by the atmosphere or by a defocused photo camera, and the image as it would be without blurring is to be found. In the examples above the process of blurring is known and therefore can be used to reconstruct the image. In other applications, like in some medical imaging problems, the blurring process is unknown and has to be reconstructed together with the image.

The degradation process is usually modelled as a convolution:

\[ g = f * h + n \]  

where \( g \) is the blurred, noisy image i.e. the data available, \( f \) is the original image, \( h \) is the point spread function modelling the blurring and \( n \) is some kind of noise added to the blurred image. In the next Section there is a detailed description of the image degradation model.

The process of image restoration falls into two classes, dependent on whether information about the degradation process is provided. If this process is known, i.e. an estimation of the point spread function \( h \) is known, then the restoration of \( f \) from \( g \) is known as deconvolution. If, on the other hand, there is very little or no information about the point spread function the deconvolution is regarded as blind deconvolution.

The deconvolution of \( g \) is a typical inverse problem [13, 12]. As often with inverse problems this problem is ill-posed[13, 2] and often even singular[2]. There are many interpretations of the expression ill-posed: one way to look at this is, that little change in the input variable, here \( g \), or both \( g \) and \( h \), generates a large change in the output or a completely different output, here \( f \). In other words, for ill-posed problems the output does not depend continuously on the input. This means in particular that a large uncontrolled amplification of the added noise can be expected and as a result the actual solution of the inverse problem is useless.

The problem is regarded as singular if in addition to being ill-posed no unique solution exists to the problem. In both cases regularization is required in order to find the correct solution. Regularization techniques in general provide additional information for solving the ill-posed problem. Here, for example, additional information about the image to be restored is utilized. Possible additional information would be for example non negativity or a finite support of the image\(^1\), statistical properties of the image, or perhaps information about the degree of smoothness of the image\(^2\). Mostly this results in a tradeoff between the smoothness of the reconstruction (and with that less noise) and the degree of the deblurring. In Section 4.3 there is a detailed discussion of some of these regularization techniques.

The solution of a blind deconvolution problem is not unique. Consider the solution pair \((f_0, h_0)\) for (1). If this is a valid solution then so is the pair \((af_0, \frac{1}{a}h_0)\) for any \(a \neq 0 \in \mathbb{R}\). Additional information has to be provided to solve this problem. Thus blind deconvolution is singular and a good regularization is required.

The field of standard deconvolution is already widely used and well covered in books like [2, 15, 13]. Whereas the field of blind deconvolution seems to be still in development.

\(^1\) e.g. [16]  
\(^2\) e.g. [6]
1.1 Image restoration in medical imaging

PET (Positron Emission Tomography) and SPECT (Single Photon Emission Computed Tomography) are two methods to obtain images non-invasively from the interior of a patient[21]. In both cases a radioactive tracer is injected into the patients body emitting gamma ray photons which are collected by a detector array. The actual image is the output of an inverse reconstruction algorithm. The underlying process is ill conditioned and therefore yields a very noisy and blurred image. Other medical imaging techniques, like MRI or ultrasonic imaging also result in general in very noisy and blurred images. In this paper we will see how the deconvolution idea can be used to improve the quality of the pictures. It is important to keep in mind, that these kind of images have a very high noise level (10% and more). This makes the interpretation of these images very difficult, especially if the image is used as input of a pattern classification algorithm. In medical imaging such an algorithm will normally try to find the type of tissue at each pixel or voxel. In such a case it is particularly important to have a picture that does not contain much noise and has sharp edges. A high noise level or a blurred image leads in such a case usually to a miss classification of the pixel or voxel. Later in this thesis there are examples of output of a regularized deconvolution as presented in this paper which shows a significant reduction in the noise level.

Unfortunately the commonly used classification algorithms like SPM (Statistical Parametric Mapping) [1] need 3 dimensional data sets for the classification. The deconvolution algorithms, on the other hand, are computationally extremely expensive. Therefore it was not possible to produce a 3 dimensional data set to test with the standard classification packages.

However there are several ways to produce 3d data despite the high costs of algorithm. One way would be to produce the 3d data by stacking deblurred 2d slices. However this would result in high frequency noise in cross sections of this stacked 3d image. Also some of the properties of the total variation regularization indicate that it is possible to split up the problem in several smaller sub problems as we will point out later in the thesis. Moreover all calculations for this work were done in Matlab. Switching to a different implementation in C++ or FORTRAN might also make the algorithm more efficient. Clearly more work in this direction has to be done.

1.2 Existing image restoration approaches

Recently J. Meunier, M. Mignotte, C. Janicki and J. Meunier compared in [20] several restoration techniques on 3D SPECT imaging. Two of the algorithms are blind deconvolution: IBD$^3$ and the NASRIF Algorithm, the subject of Section 4.2. Their observation was, that the best results can be accomplished using the NASRIF Algorithm. The method requires the knowledge of the support. In their paper the authors used a Marcovian segmentation to estimate the support from a given SPECT image.

The relatively good results they gained in the enhancement of the given image give rise to some more investigations of the potential of such restoration techniques. Especially the “forward approach” described in Section 4.3.

Also worth looking at is the combination of the reconstruction and restoration of the image in one single algorithm. This means that the blurring of the image is taken into account during the reconstruction of the image.

The advantage of such a combined approach would be, better control over the expected properties of the reconstructed image, especially with respect to the noise. For example if

---

$^3$iterative blind deconvolution, not covered in this paper. Refer to [4] for more details.
we know the image has to have a certain smoothness\textsuperscript{4} we can take that information into the reconstruction algorithm.

A first attempt to integrate a total variation regularization into a PET reconstruction algorithm was made by E. Jonsson, Sung-Cheng Huang and T. Chan in [14]. Although they used it only to reduce the noise in the reconstruction.

\textsuperscript{4}for example with respect to the total variation norm. We will look into this in detail in Section 6
2 Degradation Models

The process by which the original image is blurred is usually very complex and often unknown. To simplify the calculations the degradation often is modelled as a linear functional. A more detailed discussion of linear and nonlinear degradation models can be found in [2].

For the rest of this paper we will assume a linear degradation model:

\[
g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \nu) h(x, y, \xi, \nu) d\xi d\nu + n(x, y) \tag{2}
\]

where \(g(x, y)\) is the blurred noisy image, \(f(x, y)\) is the original image, \(n(x, y)\) is some noise and \(h(x, y, \xi, \nu)\) is the point spread function referred to as psf. In general the point spread function depends on the spacial location i.e. at each point in the image domain there exists a different point spread function. Such a point spread function is called a \textit{spatially variant} point spread function (SVPSF). Otherwise, the point spread function can be written as \(h(x, y, \xi, \nu) = h(x - \xi, y - \nu)\), and is called \textit{spatially invariant}, (SIPSF). In this case the integral term in (2) simplifies and the degradation process is modelled as an ordinary two dimensional convolution,

\[
g(x, y) = (f * h)(x, y) + n(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \nu) h(x - \xi, y - \nu) d\xi d\nu + n(x, y) \tag{3}
\]

the convolution \(\ast\) is, as in the one dimensional case, related to the Fourier transform:

\[
f * h = \mathcal{F}\{f\}\mathcal{F}\{h\}\tag{4}
\]

where \(\mathcal{F}\{f\}\) is used to denote the Fourier transform of function \(f\).

2.1 Discrete model

In the discrete - discrete model i.e. both \(g\) and \(h\) are discrete (2) becomes

\[
g_{i,j} = \sum_{k} \sum_{l} f_{k,l} h_{i,j,k,l} + n_{i,j} \tag{5}
\]

here \(f_{i,j}, g_{i,j}\) and \(n_{i,j}\) can be regarded as matrix elements of matrices \(F, G\) and \(N\), respectively, and \(h_{i,j,k,l}\) is an element of a four dimensional array. The summations are over all “valid” indices, such that every term that is not zero is taken into account. Usually both the original image and the point spread function have finite support and thus the sum can be expected to be finite. In the spatially invariant case, \(h_{i,j,k,l} = h_{i-k,j-l}\) we obtain the discrete equivalent of (3),

\[
g_{i,j} = \sum_{k} \sum_{l} f_{k,l} h_{i-k,j-l} + n_{i,j}. \tag{6}
\]

As in the continuous case there is an equivalent formulation using the discrete Fourier transform:

\[
f * h = \sum_{k} \sum_{l} f_{k,l} h_{i-k,j-l} = dft\{f\}dft\{h\}. \tag{7}
\]

There are different ways to deal with the sum in equation (6). If we assume that the value of \(f, g\) and \(h\) is 0 outside some supporting region (zero padding) the convolution in (6) will yield a
larger image $g$ than $f$. Zero padding can produce problems when we try to process only a part of the image where we know that the values outside of the processing region is not 0. On the other hand if we only sum up all valid indices i.e. where both $f$ and $h$ actually have values we will end up with a smaller image $g$. Note that in this case we do not use zero padding. A third approach is to regard the functions $f$ and $h$ as periodic resulting in an infinite and also periodic $g$. Note that the relation (7) is only valid for periodic $f$ and $g$. It is possible to make the periodic case equivalent to the other two cases by using zero padding and using only a subregion of the output of the convolution, at least in a subregion of the picture. This is particularly of interest if the expensive sums are to be replaced by FFTs.

The convolution in (6) can also be expressed by a matrix - vector multiplication by rearranging the matrices $F, G$ and $N$ to vectors $f, g$ and $n[2]$. (6) becomes:

$$g = [H]f + n,$$

where the matrix $[H]$ can be constructed out of the discrete point spread function $h$ and it has the following Toeplitz structure:

$$[H] = \begin{pmatrix}
h_0 & & \\
h_1 & h_0 & \\
h_2 & h_1 & h_0 \\
& \vdots & \vdots & \ddots
\end{pmatrix}.$$  

(9)

This matrix-vector form can be helpful in the analysis of this problem. Yet, if it is to be used for computations it is most important to make use of the special structure of the matrix, otherwise the computation is very expensive.

### 2.1.1 A priori constraints

Since most image data represents some kind of a physical object, for example the body of a patient, it can be expected that both the object $f$ and the image $g$ obey a non negativity constraint. Note that this does not necessarily mean that the psf $h$ is positive. As an example think about the approximation of a positive function using the Dirichlet kernel. This kernel is non positive, yet if the values of the function $f$ is sufficiently far away from 0 then $f * h$ will show the usual under and overshoots, known as Gibbs phenomenon but will stay always positive. Never the less sometimes it makes sense to impose non negativity to the point spread function $h$.

$$f(\xi, \nu) \geq 0 \quad \forall (\xi, \nu)$$

$$g(x, y) \geq 0 \quad \forall (x, y)$$

(10)

Equivalently, in the discrete - discrete model:

$$f_{k,l} \geq 0 \quad \forall k, l$$

$$g_{i,j} \geq 0 \quad \forall i, j$$

(11)

In particular, if $g$ is obtained through an image reconstruction algorithm, the algorithm will usually have been designed to enforce the non-negativity of $g$ through a constraint condition.

### 2.2 Point spread functions

In a linear model the point spread function $h$ models the blurring of the image. In general this process is not reversible. Theoretically it can be seen from (4) or (7) that the convolutions with
the point spread function can be reversed if the spectrum of $h$ has no zeros in it. Note this is only true for periodic problems. Yet from a practical point of view this is of little usefulness. Especially since usually round off-noise and other noise is involved. To illustrate the problem let’s look at the convolution as a filtering operation. A filter can be described as a convolution as in (6). In this case the Fourier coefficients are the filter coefficients. If some of the filter coefficients are very small the resulting spectral coefficients of $g$ will be very small and possibly lost to the noise. We have an information loss and the reconstruction of the original function is not possible.

2.2.1 Gauss blur

The Gauss blur is defined by the following psf

$$h(x, y) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

(12)
where $\sigma$ is a parameter of the Gaussian, in statistics it is usually called variance. This kind of blur occurs for example due to long time atmosphere exposure. Figure 2 shows a Gaussian point spread function and Figure 3 shows a image blurred with a Gauss psf.

### 2.3 Out-of-focus blur

This blurring is produced for example by a defocused optical system. It distributes a single point uniformly over a disk surrounding the point. Figure 4 illustrates the psf and Figure 5 shows an example of a blurred image. The psf is given by

$$h(x, y) = c \begin{cases} 1, & \sqrt{(x - c_x)^2 + (y - c_y)^2} \leq r \\ 0, & \text{otherwise} \end{cases}$$

where $r$ is the radius and $(c_x, c_y)$ is the center of the out of focus psf. The scaling factor $c$ has to be chosen such that $\int \int h(x, y)dxdy = 1$ i.e. in the continous case $c = \frac{1}{\pi r^2}$. 

Figure 3: Example of a Gaussian blurred image

Figure 4: Example of an out-of-focus psf
2.4 Motion blur

When an object or the camera is moved during light exposure a motion blurred image is produced. Figure 6 shows an example of such a psf, figure 7 shows a image that was blurred with this psf.

2.5 Noise

As mentioned before the noise level of an actual problem can be expected to be very high (10% for MRI and even more for PET). The random distribution of the noise is usually also unknown. Figure 8 shows an example of a MRI picture with noise, and figure 9 shows a slice of this image.
Figure 7: A motion blur example

Figure 8: Example of a MRI picture with a noise level of 9%
Figure 9: 1d slice of a MRI scan with 9% noise

<table>
<thead>
<tr>
<th>N</th>
<th>Out of Focus</th>
<th>Gauss</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>20.3</td>
<td>9.5 · 10^9</td>
</tr>
<tr>
<td>100</td>
<td>81.4</td>
<td>4.0 · 10^7</td>
</tr>
<tr>
<td>1000</td>
<td>650</td>
<td>3.7 · 10^8</td>
</tr>
</tbody>
</table>

Table 1: Condition numbers of some convolution matrices

3 Problem properties

To be able to better understand the problem and its behavior lets look again at the matrix-vector formulation of the problem:

\[ g = [H]f + n, \]  

Suppose that the matrix \([H]\) is non-singular. If the matrix is non-singular we can theoretically obtain an estimation of \(f\) by inverting \([H]\):

\[ \hat{f} = [H]^{-1}g \]  

which inserting in equation (8) yields

\[ \hat{f} = [H]^{-1}([H]f + n) = f + [H]^{-1}n. \]  

Thus the estimate for the restored image is composed of two parts: the original image and a term involving the noise. Due to the poor condition of the problem (8) the matrix \([H]\) will have a large condition number i.e. \([H]\) is nearly-singular, see some examples in Table 1. Thus the matrix \([H]^{-1}\) will have large entries, and consequently the term \([H]^{-1}n\) can dominate the term containing the solution \(f\). We illustrate this effect in Section 4.1.

3.1 Singular value decomposition [13, 12]

The singular value decomposition (svd) can provide much insight into the problem. The svd can be computed for any \(A \in \mathbb{R}^{m \times n}, \ m \geq n\) as

\[ A = U \Sigma V^T = \sum_{i=1}^{n} u_i \sigma_i v_i^T \]  

15
where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices i.e. $U^T U = V^T V = I$ and $\Sigma$ is a diagonal matrix $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$. The numbers $\sigma_i$ are called singular values and are nonnegative [13]:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$$

The svd is closely related to the eigen value decomposition of the matrices $A^T A$ and $AA^T$.

$$A^T A = V \Sigma^2 V^T$$
$$AA^T = U \Sigma^2 U^T$$

If $A$ is invertible its inverse is given by

$$A^{-1} = \sum_{i=1}^{n} v_i \sigma_i^{-1} u_i^T$$

(17)

From this and the very large condition number shown in Table 1 we can expect that there are very small singular values for the matrix $[H]$. Figure 10 shows the singular values of a Gauss point spread function.

In general there are two important properties

- The singular values decay rapidly to zero and the number of small singular values increases with the size of $[H]$.

- with increasing $i$ (i.e. decreasing $\sigma_i$) $u_i$ and $v_i$ have more sign changes in them. This means that the small singular values correspond to high frequency components i.e. in the inverse problem we have to expect a high amplification of high frequency components.

These properties can be observed in most discrete ill posed problems. Unfortunately it is very difficult or even impossible to prove in general. From equation 17 we can see how important the decay of singular values is with respect to the solution of the inverse problem. In other words the faster the singular values decay the more singular is the problem. This observation gives rise to the definition of the degree of ill-posedness $\alpha$:

$$\sigma_i = O(i^{-\alpha})$$

(18)

Figure 10: Singular values of a Gauss psf
Problems of this kind are called mildly or moderately ill posed. If on the other hand

$$\sigma_i = O(e^{-\alpha i})$$  \hspace{1cm} (19)$$

then the problem is called severely ill posed. Figure 10 shows with $\alpha \sim .17$ such a problem.

Using the singular value decomposition we can obtain a solution of the kind of (14) even for non-invertible $[H]$ in form of the least square solution.

$$f_{LSQ} = \sum_{i=1}^{n} \frac{u_i^T g}{\sigma_i} v_i$$  \hspace{1cm} (20)$$

This also reveals the problem with a solution of this kind: If the coefficient $u_i^T g$ decays slower to zero than the singular value $\sigma_i$ then we will have large coefficients in front of $v_i$ which we stated above has many sign changes in it. So, we expect to see large amplitudes of the high frequencies.

Figure 12 shows an example of a blurred function with no noise in it. The function is the same as the one in Figure 11 except that no noise was added after blurring. The fact that none of the coefficients with high index are larger than the singular values indicates that this problem is invertible. In fact it is possible to compute a solution of the kind of (14) and it fits the original data perfectly.

On the other hand Figure 13 shows the same test problem with noise. In both cases (a noise level of $10^{-7}$ and 1% noise) it can be seen that at least some of the coefficients $u_i^T g$ are bigger than the singular values thus resulting in a bad reconstruction. Figure 11 shows the reconstruction of the example with a noise level of $10^{-7}$.

The condition that the coefficients $u_i^T g$ have to be smaller than $\sigma_i$ or that $\frac{u_i^T g}{\sigma_i} < 1$ is also called the discrete Picard condition \cite{12}.
Figure 12: Plot of the singular values of a Gauss psf (o) and the coefficients $u_i^T g$ of the test problem seen in Figure 11 (□) with no noise on the blurred function. It can be seen that none of the larger coefficients are larger than the singular values.

Figure 13: Plot of the singular values of a Gauss psf (o) and the coefficients $u_i^T g$ for a noise level of $10^{-7}$ (x) and $10^{-2}$ (□). It can be seen that in both cases some of the coefficients are larger than the singular values.
Table 2: Noise amplification after matrix inversion

<table>
<thead>
<tr>
<th>noise level in $g$</th>
<th>$|g - f|_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>$10^1$</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>$10^5$</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>$10^3$</td>
</tr>
</tbody>
</table>

4 Theoretical Solution Techniques

We overview techniques for solving the ill-posed singular deconvolution problem. Equivalently we need techniques for finding the solution $f$ in (2) or (3), when $g$ and $h$ are given, or in case of blind deconvolution, finding $f$ and $h$ when $g$ is given.

4.1 Matrix inversion

A first approach of solving would be to try to invert the matrix in (8) and apply it to the blurred signal.

$$\hat{f} = [H]^{-1}g.$$  

As shown in previous section this is not a very good idea because of the ill-posedness of the problem. To illustrate this let’s look at the one dimensional test function shown in Figure 11. The figure also shows the function blurred with a Gauss point spread function. After blurring the function noise with a level of $10^{-7}$ was added. It is no surprise that the direct deconvolution using the inverse of $[H]$ amplifies the noise to a huge extent considering the very low noise level. Table 2 shows the error of the reconstruction with various noise levels in $g$. We can clearly see that we cannot seek the “exact” solution of the problem. This is of course because of the ill-posedness of the problem.

This also applies to other techniques of retrieving the “exact” solution like using the Fourier transformation of the function and the point-spread function\(^5\). We rather need to find a solution that “behaves nicely” and is “near” to the “exact” solution. The solution that obeys certain constraints is called a regularized solution.

It should also be mentioned that the “exact” solution referred to in the previous section isn’t exact at all since it ignores the fact that there is noise in $g$ and that the real solution convolved with $f$ produces $g$ without noise.

4.2 Inverse filtering

Let’s stay for a moment with the idea of trying to invert the blurring. We can try to approximate a filter $u$ such that $g * u \approx f$, which we call the inverse filter to $h$. We must keep in mind that we face the same problems as with the matrix inversion i.e. we really don’t want to find the “exact” inverse filter because of ill posedness of the problem and the resulting amplification of the noise.

We need to incorporate some additional information into the solution. For example we may consider the support of the original function $g$ i.e. we know that the signal $f$ is zero\(^6\) outside

---

\(^5\) In Fourier space the convolution is a simple multiplication

\(^6\) or some background gray level $LB$
If we look at the process of blurring we can realize that the points of the original function, i.e. $g$, are spread over a larger area than occupied by $g$ (therefore the name point-spread-function). Figure 14 illustrates this process. This also means that an inverse filter has to map the signal $g$ to a signal $\hat{f}$ that is inside of the support.

We can tell if an approximation of $u$ is “good” by looking at the “deblurred” signal i.e. $\hat{f} = g * u$. If $\hat{f}$ has many and/or large values outside of the support then it is not a good approximation of the inverse of the filter $h$. As stated in Section 2.1.1 we can also assume that the desired function $f$ has no negative values.

So, let’s construct a penalty-function that penalizes values outside of the support and values that are smaller than zero. The NASRIF algorithm, as first utilized in this case by D. Kundur [16] does exactly this. It seeks the minimum of the following functional:

$$J(u) = \sum_{x \in D_{sup}} \|\hat{f}(x)\|^2 \frac{1 - \text{sign}(\hat{f}(x))}{2} + \sum_{x \in D_{sup}} (f(x) - LB) + \gamma \left( \sum u(x) - 1 \right)$$

(22)

Here $\hat{f}(x)$ is obtained as the filtered version of $g(x)$:

$$\hat{f}(x) = g(x) * u(x),$$

(23)

where $u$ represents the approximate inverse for the point spread function $h$. In addition, $LB$ is the background gray level and the parameter $\gamma$ is only nonzero if $LB = 0$ and forces the solution away from the trivial solution. The set $D_{sup}$ is the set of points inside the area of support.

Note that in this algorithm no a priori knowledge about the point spread function is imposed. In fact, the inverse of the point spread function is used as the variable over which the cost functional is minimized. The NASRIF algorithm thus belongs to the class of blind deconvolution methods.

The only constraint on the smoothness of the function is that the signal should be zero or the background level outside the support. This means that the application of the noise is suppered because it would lead to high function values outside of the support. This suppresses to some extent the wild oscillations seen in the matrix inversion case. This method therefore applies...
some kind of regularization, yet as we will see the imposed regularization reaches its limits very quickly.

Using the Hessian-matrix of (22) it can be shown, that (22) is convex and therefore that there exists a global minimum [16]. To minimize (22) D. Kundur suggested a conjugate gradient method (CG) with line search.

Figure 15 shows an 1d example of a deconvolution using the NASRIF algorithm. The dotted line shows the original function. The dashed line the convolution of the original function with the psf shown in figure 16 on the left side plus noise of a level of 1%. This data is given to the NASRIF algorithm. The solid line in figure 15 shows the result of the deblurring i.e. $\hat{f} = g \ast u$.

We can clearly see that we have a large amplification of the noise but still much smaller then the amplification we have seen in the matrix inversion case in figure 11.

Figure 16 also shows the approximation of the inverse psf u reconstructed by the NASRIF algorithm in the middle. The picture on the right in figure 16 shows the convolution of h and u. If u would really be the inverse psf then $h \ast u$ would be a delta function. The fact that $h \ast u$ is not a delta function is not so much because the algorithm got the “wrong” solution but more because we are not looking for the “exact” solution, rather one that fits our needs i.e. has only small values outside the support. The example above shows large amplification of the noise. The practically invisible noise of a level of 1% is amplified to a considerable level in the restoration. Figure 17 shows the impact of different noise levels on the solution. We can see that even with noise levels under 10% NASRIF produces errors in the restoration that have almost 5 times the
energy of the signal itself. We can also see that the noise amplification is very random and thus uncontrollable.

4.3 A Forward Model

The magnitude of the amplification of the noise seems to be a general problem when trying to invert the blurring directly even with regularization. Thus, we need a different approach to address our inverse problem.

It seems quite reasonable to consider the image itself, and possibly the psf, as the unknown variables over which a minimization is required, rather than the inverse psf. In this case, it suggests that we attempt to find the image that fits best to the given data and at the same time behaves nicely i.e. obeys certain constraints. For example we may need again some smoothness conditions, or support conditions on the image and the point spread function. The basic regularization techniques are no different than if we try to solve the problem in the inverse direction.

4.3.1 Fit-to-data functional

First, we have to define the fit-to-data functional such that the restored image will fit the provided data. The fit-to-data functional is needed to make sure that in the restoration $f$ (and $h$) we do obtain the appropriate fit to the data $g$ i.e. given the restored image $\hat{f}$ blurred with the psf $h$ (or $\hat{h}$ if $h$ is subject to the restoration), we should restore the original data measured in some appropriate norm:

$$\|f \ast h - g\|_2.$$  

Here we suppose the usual $L^2$-norm, but other choices are possible. Numerical experiments show, however that the choice of norm has only little impact on the solution. So we choose a norm that is nice to handle. We can also look at the square of the fit to data functional which makes in some cases the analysis of the problem easier but also changes the nature of the problem which will have an impact on the minimization process especially if we have nonlinear constraints as for the total variation norm introduced later.
4.3.2 Regularization

Due to the ill-posed nature of the problem we also need to put some constraints on the restoration, due to the ill-posed nature of the problem. Usually this is done by some smoothness constraint on \( \hat{f} \) (and \( \hat{h} \)). In general we can say that we want that some norm of \( \hat{f} \) and \( \hat{h} \) is small i.e.

\[
\| \hat{f} \|_f "small" \\
\| \hat{h} \|_h "small"
\]

The choice of the norms for \( \hat{f} \) and \( \hat{h} \) is of central meaning in this process. Different choices of the norm result in very different meanings of the statement that \( \hat{f} \) or \( \hat{h} \) is smooth. We will look at different choices of the norm and its properties and the different results in the restoration in the sections 5 and 6.

4.3.3 Constrained minimization

Now we are able to formulate the deblurring problem as a constrained minimization i.e.

\[
\min_{f,h} \| f \ast h - g \|_2
\]

with respect to

\[
\| \hat{f} \|_f "small" \\
\| \hat{h} \|_h "small"
\]

(25)

To solve this constrained minimization we introduce Lagrange multipliers and gain the following objective function

\[
J(f,h) = \| f \ast h - g \|_2 + \lambda_1 \| \hat{f} \|_f + \lambda_2 \| \hat{h} \|_h
\]

(26)

with the Lagrange multipliers \( \lambda_1 \) and \( \lambda_2 \). Equation (26) represents the central equation of regularized deconvolution. We are unable to evaluate \( J(f,h) \) for given \( f \) and \( h \) since we haven’t said what the norms \( \| \cdot \|_f \) and \( \| \cdot \|_h \) will be. Also note that often the squares of (24) and (25) are used.

As stated before we want to have a “smooth” solutions without high frequency oscillations. It seems reasonable to demand that some norm of the derivative of \( f \) (or \( h \)) is small i.e. (25) becomes

\[
\| L_d \hat{f} \|_p "small" \\
\| L_d \hat{h} \|_q "small"
\]

(27)

with \( L \) an approximation of the \( d \)-th order derivative operator. \( p \) and \( q \) are usually chosen to be 1 or 2.

If we look at a discrete representation of (27) then \( L_d \) is represented by a matrix \( L \). The matrix \( L \) has to be chosen such that its null space is the discrete version of the nullspace of the continuous derivative operator, i.e.

- for the 1st order derivative: The derivative of a constant function has to be zero i.e.

\[
L_1 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}
\]
• for the 2nd order derivative: The derivative of a constant function and a linear function has to be zero i.e.

\[
L_2 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}
\]

and

\[
L_2 \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}
\]

• …

This leaves us in case of a finite difference approximation with a \((n - d) \times n\) matrix e.g. for the 1st order derivative\(^7\)

\[
L_1 = \begin{pmatrix}
-1 & 1 & 0 & 0 & \cdots & 0 \\
0 & -1 & 1 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \cdots & \vdots \\
0 & \cdots & \cdots & -1 & 1
\end{pmatrix}
\]

(28)

Note that it is not necessary to include the scaling factor \(1/dx\) even if \(dx \neq 1\) because it can be absorbed into the the Lagrange parameter \(\lambda_1\) or \(\lambda_2\) in (26). However this must be kept in mind when choosing \(\lambda_1\) or \(\lambda_2\) if \(dx\) is different from 1.

\(^7\) e.g. produced by the function `getL` in [12]
Figure 18: Tikhonov non blind deconvolution. 10% noise was added to the blurred function.

5 Tikhonov regularization

If the norm for \( f \) or \( h \) in (27) is chosen to be the the 2-norm then the regularization is called Tikhonov regularization.

5.1 Non-blind deconvolution

The field of non-blind deconvolution with Tikhonov regularization is very well covered\(^8\). The deconvolution is carried out by minimizing the functional

\[
J(f, h) = \|f * h - g\|^2_2 + \lambda \|\nabla f\|^2_2
\]  
(29)

This functional can be shown to be convex and therefore there exists a unique minimum that is only dependent on the parameter \( \lambda \) in some parameter space \( \Lambda \) usually \( \mathbb{R} \).

The Tikhonov regularization does not allow sharp edges due to the use of the 2-norm. This results in smooth edges and oscillations as shown in figure 18. However as illustrated in figure 19 if no noise is added to the blurred signal and a small enough \( \lambda \) is chosen the Tikhonov regularized deblurring works very well.

5.1.1 Choosing the parameter (L-curve)

Different choices of the parameter \( \lambda \) result in a trade-off between the smoothness of the signal i.e. \( \|\nabla f\|_2 \) is small and good fit-to-data i.e. \( \|f * h - g\|_2 \) is small. One way of choosing this

\(^8\)see e.g. [11], [23] or [19]
Figure 19: Tikhonov non blind deconvolution. No noise was added to the blurred function.

parameter is to plot the L-curve [13]. If we sweep through the parameter space $\lambda \in \Lambda$, computing $\|f * h - g\|_2$ and $\|\nabla f\|_2$ for each parameter and plot them as a point in the $\|\nabla f\|_2 - \|f * h - g\|_2$ plane, we will see the L-curve, illustrated in figure 20. More filtering leads to a bigger norm of the residual $\|f * h - g\|_2$ and less filtering results in a non-smooth signal $f$ indicated by a big $\|\nabla f\|_2$.

The goal of the L-curve analysis is to find the $\lambda$ for which both norms are as small as possible i.e. to pick the $\lambda$ that corresponds to the corner of the L-curve. Figure 21 shows a “real” L-curve generated using the 1-d image reconstruction test problem in [12].

5.2 Blind deconvolution

In case of blind-Tikhonov regularization (26) is represented by

$$J(f, h) = \|f * h - g\|_2^2 + \lambda_1 \|\nabla f\|_2^2 + \lambda_2 \|\nabla h\|_2^2$$

(30)

Note that in this case the norms are squared. In [24] it is shown that this particular functional is convex w.r.t. to $f$ if $h$ is fixed i.e. $J(f) := J(f, h)$ is convex. This guarantees a unique solution for the problem if $h$ is fixed. The same is true for fixed $f$ w.r.t. $h$. Unfortunately this does not mean that $J(f, h)$ is convex w.r.t. both $f$ and $h$.

5.2.1 Alternating Minimization (AM)

You and Kaveh suggested in [24] an alternating minimization (AM) for this problem. There are two ways to carry out the AM algorithm. One is to first minimize w.r.t. $h$ and then w.r.t. $f$ (AMHF) and the other is to first minimize w.r.t. $f$ and then w.r.t. $h$ (AMFH) [7], see Table
Figure 20: The generic form of the l-curve, taken from [12]

Figure 21: l-curve of the 1d-image restoration test problem in [12]
AM Algorithm - min h then min f (AMHF):
Given \( f^0 \): iterate \( k = 1, 2, ... \)

- solve \( h^k = \arg \min_h J(f^{k-1}, h) \)
- solve \( f^k = \arg \min_f J(f, h^k) \)

AM Algorithm - min f then min h (AMFH):
Given \( h^0 \): iterate \( k = 1, 2, ... \)

- solve \( f^{k-1} = \arg \min_f J(f, h^{k-1}) \)
- solve \( h^k = \arg \min_h J(f^{k-1}, h) \)

Table 3: Alternating minimization

3. Chan and Wong give a detailed analysis of the (AM) algorithm in [7]. They found that the algorithm converges globally for each choice of a initial condition. But the solution depends on the initial condition. In other words there exists a unique solution for each choice of the initial condition. They also found out that the two versions of that AM algorithm i.e. (AMHF) and (AMFH) produce the same sequence of iteratives for a appropriate choice of the initial condition and that the solution satisfies the following properties

- Total Intensity Preserving

\[
\sum_{p=0}^{m-1} \sum_{q=0}^{n-1} f(p, q) = \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} f^{true}(p, q)
\]

and

\[
\sum_{p=0}^{m-1} \sum_{q=0}^{n-1} h(p, q) = 1
\]

- Phase Invariant

\( \theta_F = \theta_G \)

- Relation of Magnitude

\[
|F(\xi_x, \xi_y)| = \sqrt{\frac{\lambda_2}{\lambda_1}} |H(\xi_x, \xi_y)|
\]

Where \( F \) represents the Fourier transform of \( f \), \( G \) and \( H \) respectively. \( \theta_F \) denotes the phase of the Fourier transform of \( f \). Especially interesting is the third property, it shows that the solution will have the same magnitudes in the Fourier domain as the point spread function. In general this is of course not the case for the real solution, thus showing that the solution using Tikhonov Regularization in general gives a poor reconstruction of the true signal.
5.2.2 Global behavior

The analysis in the previous paragraph assumes that the minimization in each step is actually carried out until a minimum is found. A different minimum of (30) will be found if the minimization in each step is only carried out a few steps.

Consider for example the AMHF algorithm described in table 3. This algorithm does not depend on the initial guess of $h$ since $J(f, h)$ is convex w.r.t. $h$

\[
\begin{pmatrix} \dot{f} \\ \dot{h} \end{pmatrix} = \nabla J(f, h)
\]

where $\nabla J(f, h)$ is the gradient of $J(f, h)$. In a discrete representation this is a gradient system in a $(N_f + N_h)$ dimensional space. Such systems are extremely difficult if not impossible to analyse.

We illustrate this in the following examples. The first example (Figure 22) shows the result of the blind deconvolution when starting at the correct solution. No noise was added to the blurred function. This represents the best case scenario. Only a few steps of the minimization in the direction of $h$ were carried out each step of the AM in order to preserve the shape of the psf after the first minimization. Since the Tikhonov regularization on $h$ does not allow sharp edges we see a blurring of the initial $h$. The oscillations in the reconstruction of $f$ are both a result of the blurring of the psf and the Tikhonov regularization on $f$.

The second example (Figure 23) was started with the observed signal as an initial guess for $f$ and with a delta function as an initial guess for $h$. Usually such a deconvolution is started with these initial values because they are always available. The reconstruction is much more inaccurate that the solution shown in figure 22, yet it is a local minimum of $J(f, h)$.

The third example, shown in Figure 24, shows the result if the minimization is started with a large out of focus psf i.e. $h_i = 1/n$ where $n$ is the number of points of $h$. The initial guess for $f$ was again the observed signal $g$. This solution also is a local minimum of $J(f, h)$ although it has almost nothing to do with the correct solution.

These examples show that it is extremely important to start the minimization near the correct solution, especially with the initial estimate of the psf.
Figure 22: Example of a blind deconvolution using Tikhonov regularization on \( f \) and \( h \). In this case the AM was carried out with only a few steps in the direction of \( h \) in order to preserve some information of the initial condition after the minimization w.r.t. \( h \). The algorithm was started at the exact solution i.e. with the true \( f \) and the true psf \( h \).
Figure 23: Example of a blind deconvolution using Tikhonov Regularization on $f$ and $h$. The algorithm was started with the observed signal for $f$ and a delta-function for $h$. 
Figure 24: Example of a blind deconvolution using Tikhonov Regularization on $f$ and $h$. The algorithm was started with the observed signal for $f$ and a large out-of-focus psf for $h$. 
6 Total variation regularization

[6, 23] Another regularization functional seeks to minimize the total variation:

\[
\|\nabla f\|_{L^1}
\]

(31)

In case of a discrete representation the \(L^1\)-norm becomes the standard 1-norm i.e.

\[
\|x\|_1 = \sum_{i=1}^{n} |x_i|.
\]

The total variation norm has some interesting properties, perhaps the most important one is, that in contrast to the Tikhonov regularization the TV norm allows jumps in the reconstructed signal making it particularly interesting for piece wise smooth functions.

6.1 Non-blind deconvolution

The non-blind deconvolution is as in the Tikhonov case (eq (29)) also carried out by minimizing a functional

\[
J(f, h) = \|f \ast h - g\|_2^2 + \lambda \|\nabla f\|_1
\]

(32)

In order to get a feeling how the TV regularization works let’s take a look at the example in Figure 25. The example shows a non blind deconvolution using total variation restoration with 10% noise in the blurred signal. The example shows that the restored signal has basically the same edges as the original signal in contrast to the Tikhonov solution seen in figure 18. The example also illustrated the typical loss of contrast in small structures of TV restoration like the peak on the left side of the picture. We can almost completely remove the noise by increasing the parameter \(\lambda\) but at the price of loosing more contrast in the small structures as shown in figure 26.

In [22] Strong and Chan look at the class of piece wise constant functions in 1-d and piece wise constant and rotational symmetric functions in 2-d and 3-d and the effect of the TV denoising\(^9\). They were able to quantitatively describe the effect of loosing the contrast in small structures. Figure 27 illustrates the result of the denoising. They show that the reconstructed signal has the following properties under certain conditions\(^10\):

- the edges of the original signal are preserved
- the contrast of the original signal i.e. the change in intensity levels is reduced. Indicated by \(\delta_i\) in figure 27.
- the decrease of the contrast is directly proportional to the regularization parameter \(\lambda\) and indirectly proportional to the scale of the constant region.

In particular they found that the numbers \(\delta_i\) can be found explicitly by

\[
\begin{array}{c|c|c|c|c}
\mathbb{R}^1 & \mathbb{R}^2 & \mathbb{R}^3 & \text{Type of Region} \\
\hline
\frac{2}{x_i-x_{i-1}} \lambda & -\frac{2}{r_i-r_{i-1}} \lambda & \frac{3(r_i^2 + r_{i-1}^2)}{r_i^2 - r_{i-1}^2} \lambda & \text{Extremum} \\
0 & -\frac{2}{r_i+r_{i-1}} \lambda & \frac{3(r_i^2 - r_{i-1}^2)}{r_i^2 - r_{i-1}^2} \lambda & \text{Step} \\
\frac{1}{x_i-x_{i-1}} \lambda & -\frac{2r_i}{r_i^2 - r_{i-1}^2} \lambda & \frac{3r_i^2 + 1}{r_i^2 - r_{i-1}^2} \lambda & \text{Boundary} \\
\end{array}
\]

\(^9\)i.e. non-blind deconvolution with a delta function
\(^10\)see [22] for more detail
Figure 25: Total variation nonblind deconvolution. 10% noise was added to the blurred function.
Figure 26: Total variation nonblind deconvolution. 10% noise was added to the blurred function.
Where in the 2d and 3d case \( r_i \) is the radius. Note that the change in contrast i.e. \( \delta_i \) does only depend on the current region. This shows the local nature of the TV regularization, although it is a global approach. This is particular interesting because of the high costs of the minimization for a large number of unknowns. In order to choose the appropriate subregion to split the problem in smaller pieces it would be necessary to know the location of the edges also this result applies only on piece wise smooth functions. Nevertheless splitting the problem could significantly reduce the complexity of the problem. It would also enable us to choose different regularization parameters in each subregion. More research in this direction has to be done.

6.2 Blind deconvolution

The blind deconvolution problem is then solved by minimizing the functional

\[
J(f, h) = \|f * h - g\|_1 + \lambda_f \|\nabla f\|_1 + \lambda_h \|\nabla h\|_1
\]

(34)

It is also possible to mix total variation and Tikhonov regularization e.g. TV regularization on \( f \) but Tikhonov regularization on \( h \).

Compared to the Tikhonov regularization the TV regularization is able to recover a much better estimate of the original signal. This can be seen by comparing the figure 28 of the TV restoration with the Tikhonov solution shown in figure 22. Although the Tikhonov regularized blind deconvolution was started at the exact solution, it can be seen that the TV regularized deconvolution shows a better fit to the original signal even when starting at the given data and a rough estimate of the psf. However the TV regularization also suffers the same problems as the Tikhonov regularization in terms of the local minima as figure 29 shows. It may also be pointed out that the minimum shown in figure 29 depends strongly on the parameters \( \lambda_f \) and \( \lambda_h \). Increasing one of them enough will eventually make the minimum unstable and a new minimum will be found. Of course by increasing the parameters particularly \( \lambda_f \) will lead also to the loss of contrast described in the previous section.
Figure 28: TV regularized blind deconvolution started at the given data and a rough estimation of the psf.
Figure 29: TV regularized blind deconvolution started at the given data and a delta function.

7 Numerical implementation and examples

To actually carry out the reconstruction on a computer we have to find a discrete representation of (30) or (34) and their gradients.

7.1 1 dimensional signals

In the one dimensional case $f$, $g$ and $h$ are represented by vectors i.e. $f, g \in \mathbb{R}^N$ and $h \in \mathbb{R}^M$. (30) can be written as

$$J(f, h) = \|f * h - g * \delta\|_2^2 + \lambda_f \|L_f f\|_p^p + \lambda_h \|L_h h\|_q^q \quad \text{(35)}$$

Here we assume the full convolution i.e. the convolution is defined as

$$(f * h)_{i,j} = \sum_k \sum_l f_{k,l} h_{i,j,k,l} \quad \text{(36)}$$

where we assume the $f$ and $h$ are zero for non-valid indices. This results in a vector of the size $N + M - 1$. Therefore we have to adjust the size of $g$ to fit the size of $f * h$. This is done by convolving $g$ with an appropriate discrete representation of the delta function. The vector $\delta \in \mathbb{R}^M$ has zeros in each component except in the middle of the vector there is a 1. The position of the 1 indicates the center of the point spread function. As we will assume symmetry of the psf this choice is is appropriate.

The matrices $L_f \in \mathbb{R}^{N-1 \times N}$ and $L_h \in \mathbb{R}^{M-1 \times M}$ are finite difference approximations of the 1st order derivative operator e.g. given by (28). The function get1 in [12] provides such a matrix.

38
We also need to supply the gradient of (35) to efficiently implement the minimization. All the following indices will be in Matlab notation i.e. starting with 1.

**Proposition 1 (Gradient of the fit-to-data functional)** If we assume that \( f, h \) and \( g \) are 0 for all non-valid indices, then the gradient of the fit-to-data functional

\[
J_{ftd}(f, h) := \| f * h - g * \delta \|_2^2
\]

with respect to \( f \) is given by:

\[
\frac{\partial}{\partial f_m} = 2 \sum_k r_k h_{k+1-m}
\]

and with respect to \( h \) by

\[
\frac{\partial}{\partial h_m} = 2 \sum_k r_k f_{k+1-m}
\]

where \( r \) is the residual \( r = f * h - g * \delta \).

**Proof**

For \( \bar{g} = g * \delta \) and fixed \( h \):

\[
J_{ftd}(f) := J_{ftd}(f, h) = \| f * h - \bar{g} \|_2^2 = \sum_k ((f * h)_k - \bar{g}_k)^2 = \sum_k ((\sum_j f_j h_{k+1-j}) - \bar{g}_k)^2.
\]

Now

\[
\frac{\partial}{\partial f_m} J_{ftd}(f) = \frac{\partial}{\partial f_m} \sum_k ((\sum_j f_j h_{k+1-j}) - \bar{g}_k)^2 = 2 \sum_k \sum_j \frac{\partial}{\partial f_m} f_j h_{k+1-j} = 2 \sum_k r_k f_{k+1-m}
\]

(38) follows immediately from the fact \( f * h = h * f \).

**Proposition 2 (Gradient of the Tikhonov regularization term)** The gradient of the Tikhonov regularization term

\[
J_{tik}(f) = \| Lf \|_2^2
\]

where \( L \) is the derivative matrix (28), is given by

\[
\begin{pmatrix}
\frac{\partial J_{tik}(f)}{\partial f_1} \\
\vdots \\
\frac{\partial J_{tik}(f)}{\partial f_N}
\end{pmatrix} = 2L^T L f
\]

**Proof**

\[
J_{tik}(f) = \| Lf \|_2^2 = \sum_k (L f)_k^2 = \sum_k (\sum_j L_{k,j} f_j)^2.
\]

Now, with \( df := Lf \)

\[
\frac{\partial}{\partial f_m} J_{tik}(f) = \sum_k \frac{\partial}{\partial f_m} (\sum_j L_{k,j} f_j)^2 = 2 \sum_k (\sum_j L_{k,j} f_j) (\sum_j \frac{\partial}{\partial f_m} L_{k,j} f_j) = 2 \sum_k d_{f_k} L_{k,m} = 2 (L^T df)_m
\]
The total variation term requires the computation of the absolute value i.e. $|x| = \sqrt{x^2}$. This term is not differentiable at the point 0. Thus we have to replace it by an approximation that is differentiable. A possible choice for such a approximation is $\psi(x) = \sqrt{x^2 + \beta}$ with a small $\beta > 0$ [23].

**Proposition 3 (Gradient of the TV term)** the Gradient of the total variation term

$$J_{TV}(f) = \sum_k \psi(Lf)_k$$

is given by

$$\left( \begin{array}{c} \frac{\partial J_{ijk}(f)}{\partial f_1} \\ \vdots \\ \frac{\partial J_{ijk}(f)}{\partial f_N} \end{array} \right) = L^T \Psi'(Lf)$$

(40)

where $\Psi'$ applies $\psi'$ to each component and $\beta > 0$.

**Proof**

$$J_{TV}(f) = \|Lf\|_1 = \sum_k |\sum_j L_{k,j} f_j|.$$  

Now, with $df := Lf$

$$\frac{\partial}{\partial f_m} J_{TV}(f) = \sum_k \frac{\partial}{\partial f_m} |\sum_j L_{k,j} f_j| = \sum_k \psi'(df_k) \sum_j \frac{\partial}{\partial f_m} L_{k,j} f_j = \sum_k \phi'(df_k) L_{k,m} = \left( L^T \phi'(df) \right)_m$$

$$L_{k,j} \quad m = j$$

$$L_{k,m} \quad 0 \quad m \neq j$$

Now we can try some numerical examples. All 1-d examples are done with the Matlab program `blind.m`\textsuperscript{11}. It performs either blind or non blind deconvolution using Tikhonov or TV regularization on $f$ and $h$.

To minimize (35) the program uses an alternating minimization, with the Matlab function `fminunc`. The number of steps of the minimization in the direction of $f$ and $h$ can be chosen at run time, as well as the type of regularization the parameters and the initial estimate of $f$ and $h$. For the psf non-negativity and symmetry are assumed.

The extreme non linearity of the TV regularization causes the standard Matlab minimization to converge relatively slowly, for real applications, specifically for this kind of problem, special minimization schemes have to be used\textsuperscript{12}. However to demonstrate the potential and the difficulties of the blind deconvolution the chosen method is good enough and allows an easy switch between the regularization methods.

### 7.1.1 1-d slice of a MRI scan

Figure 30 shows the reconstruction of a 1d slice of a simulated\textsuperscript{13} MRI scan. The data was taken as a 1-d slice of the picture in figure 9, namely also 9% noise assumed.

\textsuperscript{11} see the attached disk

\textsuperscript{12} see [6] and [5]

\textsuperscript{13} the data was downloaded from http://www.bic.mni.mcgill.ca/brainweb/. See also [8, 17, 18, 9]
Figure 30: Example of a practical problem. The given data is taken from a simulation of a MRI scan. The true psf in this case is not available. The true signal is also not available and in the figure only an approximation of the true signal.
In this example the true psf is unknown. Also the true signal is unknown. However the tissue type for each pixel is known. To construct the true signal in figure 30 i.e. the gray level for each tissue type the mean value of the gray level for each tissue type in the 2-d slice was taken. This is only an estimate of the true signal, but the reconstruction should be constant wherever the true signal is constant thus giving a hint of the quality of the restoration.

In reconstruction in figure 30 Tikhonov regularization was used for \( h \) because we expect a round shaped psf and TV regularization for \( f \) because we expect a piece wise constant signal. As we can see the blind deconvolution seems to work reasonably well, it was able to recover most of the edges and results in a practically noiseless signal.

7.2 Restoration of images

In the 2-d case we are dealing with matrices \( f \in \mathbb{R}^{N_1 \times N_2}, h \in \mathbb{R}^{M_1 \times M_2} \) and \( g \in \mathbb{R}^{N_1 \times N_2} \). It is possible to convert the 2-d problem to an equivalent matrix-vector equation by rearranging the matrices \( f \) and \( g \) into vectors \( f, g \) and \( h \) into a appropriate matrix \( [h] \) such that if \( \bar{g} = f * h \) then \( \bar{g} = [h]f \). Also the matrix \( L \) has to be chosen appropriately. It is however easier to stay with the matrices. Note that in this case the gradient of \( J(f,h) \) is also a matrix i.e. e.g. w.r.t. \( f \): \( G_{i,j} = \frac{\partial}{\partial f_{i,j}} J(f,h) \).

Next we have to discretize \( \|\nabla f\|_1 \) and \( \|\nabla f\|_2 \). In the continuous case \( \|\nabla f\|_1 \) is given by \( \int \int |\text{grad}(f(x,y))|dxdy \) i.e.

\[
\int \int \sqrt{f_x(x,y)^2 + f_y(x,y)^2} dxdy
\]

and \( \|\nabla f\|_2 \) is given by

\[
\sqrt{\int \int (f_x(x,y)^2 + f_y(x,y)^2) dxdy}
\]

It seems reasonable to replace the partial derivative operators by their discrete representation (28).

To avoid problems with the gradient we compute \( \|\nabla f\|_2^2 \) instead of \( \|\nabla f\|_2 \). Also, as in the 1-d case we replace \( \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \) by a differentiable approximation \( \psi(f(x,y)) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} + \beta \), with a small \( \beta > 0 \).

**Proposition 4 (gradient of the 2-d fit-to-data functional)** The gradient of

\[
J_{ftd}(f,h) = \|f * h - g * \delta\|_2^2
\]

with respect to \( f \) is given by

\[
\frac{\partial}{\partial f_{m,n}} J_{ftd}(f) = 2 \sum_{k,l} r_{k,l} h_{k-m+1,l-n+1}
\]

(41)

where \( r = f * h - g * \delta \) is the residual, and w.r.t. \( h \) by

\[
\frac{\partial}{\partial h_{m,n}} J_{ftd}(h) = 2 \sum_{k,l} r_{k,l} f_{k-m+1,l-n+1}
\]

(42)

**Proof**

With \( g = g * \delta \) and fixed \( h \)
\[ J_{\text{ftd}}(f,h) = \|f * h - g * \delta\|_2^2 = \sum_{k,l} \left( \sum_{i,j} f_{i,j} h_{k-i+1,l-j+1} - \bar{g}_{k,l} \right)^2 \]

Now

\[
\frac{\partial}{\partial f_{m,n}} J_{\text{ftd}}(f) = 2 \sum_{k,l} \left( \sum_{i,j} f_{i,j} h_{k-i+1,l-j+1} - \bar{g}_{k,l} \right) \frac{\partial}{\partial f_{m,n}} \left( f_{i,j} h_{k-i+1,l-j+1} - \bar{g}_{k,l} \right) \]

\[
= 2 \sum_{k,l} \frac{\partial}{\partial f_{m,n}} \left( f_{i,j} h_{k-i+1,l-j+1} - \bar{g}_{k,l} \right) \frac{\partial}{\partial f_{m,n}} \left( f_{i,j} h_{k-i+1,l-j+1} - \bar{g}_{k,l} \right) \]

\[
= 2 \sum_{k,l} \left( \sum_{i,j} f_{i,j} h_{k-i+1,l-j+1} - \bar{g}_{k,l} \right) \frac{\partial}{\partial f_{m,n}} \left( f_{i,j} h_{k-i+1,l-j+1} - \bar{g}_{k,l} \right) \]

Then (42) follows from \( f * h = h * f \)

Proposition 5 (Gradient of the 2-d Tikhonov term) The gradient of

\[ J_{\text{tik}}(f) = \|\nabla f\|_2^2 \]

is given by

\[
\left( \begin{array}{cccc}
\frac{\partial}{\partial f_{1,1}} & \cdots & \frac{\partial}{\partial f_{1,N_2}} \\
\vdots & \ddots & \vdots \\
\frac{\partial}{\partial f_{N_1,1}} & \cdots & \frac{\partial}{\partial f_{N_1,N_2}} \\
\end{array} \right) J_{\text{tik}}(f) = 2(L^T f_y + f_x L) \quad (43)
\]

where \( f_y = Lf \) and \( f_x = fL^T \).

Proof

\[ J_{\text{tik}}(f) = \sum_{k,l} \left( (Lf)_{k,l}^2 + (fL)_{k,l}^2 \right) = \sum_{k,l} \left( \sum_i (L_{k,i} f_{i,l})^2 + (f_{k,l} L_{i,l})^2 \right). \]

Now

\[
\frac{\partial}{\partial f_{m,n}} J_{\text{tik}} = \sum_{k,l} \left( \frac{\partial}{\partial f_{m,n}} \left( \sum_i L_{k,i} f_{i,l} \right)^2 + \frac{\partial}{\partial f_{m,n}} \left( \sum_i f_{k,l} L_{i,l} \right)^2 \right)
\]

\[
= \sum_{k,l} \left( \sum_i L_{k,i} f_{i,l} \right) \frac{\partial}{\partial f_{m,n}} L_{k,i} f_{i,l} + \sum_i \left( \sum_i f_{k,l} L_{i,l} \right) \frac{\partial}{\partial f_{m,n}} f_{k,l} L_{i,l} \]

\[
= 2 \sum_k (f_y)_{k,n} L_{k,m} + 2 \sum_i (f_x)_{m,i} L_{i,n} = 2(L^T f_y)_{m,n} + 2(f_x L)_{m,n}
\]

Proposition 6 (Gradient of the 2-d total variation term) The gradient of

\[ J_{\text{TV}}(f) = \sum_{k,l} \psi((\nabla f)_{k,l}) \]
for \(|\psi((f_x)_{i,j})| \neq 0 \) and \(|(f_y)_{i,j}| \neq 0 \) for all valid \((i,j)\) is given by

\[
\left( \frac{\partial}{\partial f_{1,1}} \cdots \frac{\partial}{\partial f_{1,N_2}} \right) J_{TV}(f) = L^T \tilde{f}_y + \tilde{f}_x L
\]

(44)

where \((\tilde{f}_x)_{i,j} = \frac{(f_x)_{i,j}}{\psi((\nabla f)_{i,j})}\), \((\tilde{f}_y)_{i,j} = \frac{(f_y)_{i,j}}{\psi((\nabla f)_{i,j})}\) and \(f_y = Lf\), \(f_x = fL^T\).

Proof

\[
J_{TV}(f) = \sum_{k,l} \psi((\nabla f)_{k,l}) = \sum_{k,l} \sqrt{(Lf)_{k,l}^2 + (fL^T)_{l,i}^2} + \beta = \sum_{k,l} \sqrt{\left( \sum_i L_{k,i} f_{i,l} \right)^2 + \left( \sum_l f_{k,i} L_{l,i} \right)^2} + \beta.
\]

Now

\[
\frac{\partial}{\partial f_{m,n}} J_{TV}(f) = \sum_{k,l} \frac{1}{2 \psi((\nabla f)_{i,j})} \left( 2(Lf)_{k,l} \sum_l \frac{\partial}{\partial f_{m,n}} L_{k,i} f_{i,l} + 2(fL^T)_{l,i} \sum_i \frac{\partial}{\partial f_{m,n}} f_{k,i} L_{l,i} \right)
\]

\[
= \sum_k \frac{(f_y)_{k,m}}{\psi((\nabla f)_{i,j})} L_{k,m} + \sum_l \frac{(f_x)_{m,l}}{\psi((\nabla f)_{i,j})} L_{l,m}
\]

\[
\square
\]

7.2.1 Restoration in Fourier space

It can be beneficial to restore the image not in physical space but in Fourier space. In the 1d discrete Fourier transform \(fft(f)\) of a vector \(f \in \mathbb{C}^n\) can be written as \(fft(f) = Df\) with \(D \in \mathbb{C}^{n \times n}\). Note that \(D\) is a symmetric Matrix i.e. \(D^T = D\). The 2d discrete Fourier transform can then be written as \(fft2(F) = D_1 F D_2\) where \(F \in \mathbb{C}^{m \times n}\), \(D_1 \in \mathbb{C}^{n \times n}\) and \(D_2 \in \mathbb{C}^{m \times m}\).

In order to restore the image in Fourier space we need a Fourier space version of the objective functions \(J_{TK}\) and \(J_{TV}\) as well as their gradients. The fit-to-data functional can be easily computed in Fourier space because of the Convolution Theorem i.e. \(\mathfrak{F}(f * h) = \mathfrak{F}(f) \mathfrak{F}(h)\) and Parseval’s Theorem. Thus

\[
J_{fitd} = \| \hat{f} \hat{h} - \hat{g} \|^2_2
\]

(45)

where \(\hat{\cdot}\) denotes the discrete Fourier coefficient. In order to minimize (45) we regard a complex number \(c \in \mathbb{C}\) as a vector \(\vec{c} \in \mathbb{R}^2\) i.e. the real and imaginary part are treated separately. In this context the gradient of the fit-to-data functional can be written as

\[
G = 2 \left( \hat{r} \cdot \hat{h} R + \hat{t} \cdot \hat{h} I - i(\hat{r} \cdot \hat{h} I + \hat{t} \cdot \hat{h} R) \right)
\]

where \(\cdot\) operates on componentwise and \(x^R\) denotes the real part of \(x\) and \(x^I\) denotes the imaginary part of \(x\).

To compute the Tikhonov and total variation functional we have to go back into physical space i.e. \(J(f) = \| \nabla D_1 \tilde{f} D_2 \|^2_2\) where this time \(D_{1,2}\) is the Matrix corresponding to the inverse discrete Fourier transform. In the same context as above the gradient is given by

\[
G = D_1 (L^T \tilde{f}_y) D_2 + D_1 (f_x L) D_2
\]

where \(\tilde{f}_y = f_y\) in case of Tikhonov regularization and \((f_y)_{k,l} = \frac{(f_y)_{k,l}}{(f_y)_{k,l}^2 + (f_x L)_{l,i}^2 + \beta}\) in case of total variation with \(f_y = Lf\) and \(f\) is physical space i.e. \(f = D_1 \tilde{f} D_2\).
7.2.2 2d examples

The following examples are computed using TV regularization on \( f \) and Tikhonov regularization on the PSF \( h \). \( f \) is reconstructed in the frequency domain and \( h \) in physical space. The minimization is done in an alternating scheme i.e. first minimize w.r.t. \( f \) \((n_f)\)-iterations and then minimize w.r.t. \( h \) \((n_h)\)-iterations. The minimization w.r.t. \( h \) is done using the Matlab function \texttt{fminunc} and the minimization w.r.t. \( f \) is done by a CG with line search. The CG minimization converges much slower to the solution then \texttt{fminunc} which uses a Newton-like method but unlike the Newton-like methods the CG method does not depend on the Hessian-matrix which is for a \( 512 \times 512 \) picture a \( 260,000 \times 260,000 \) Matrix which can not be handled by a standard desktop computer.

Figure 32 shows the observed data after blurring the Shepp-Logan Phantom (Figure 31). Figure 33 and 34 show two restorations of figure 32 using non-blind deconvolution with TV-regularization. These examples show the impact of different choices of \( \lambda_f \) on the result of the restoration. It is not surprising that some of details are lost because of the high noise level. They also show the artifacts to be expected if the parameters of the restoration are chosen improperly.

An example of a blind deconvolution is shown in Figure 35. The algorithm was initialized with the blurred picture and a gauss-shape PSF with \( \sigma = 8 \), where the picture in figure 31 was blurred by a gauss PSF with \( \sigma = 3 \). Figure 36 shows the psf derived during the restoration.

The blind deconvolution can also be used for restoration problems in other fields like astronomy as the following example demonstrates. Figure 37 shows a Photo of Jupiter\textsuperscript{14}. The picture shows what can be observed after a long time exposure through the atmosphere. Figure 38 shows the restoration of the image. Figure 39 shows the psf retrieved during the process. Note that in order to achieve this result no a priori knowledge is necessary. The “shadow” on the right side of the picture and the ring around Jupiter is most an artifact or the restoration.

\textsuperscript{14}captured by Howard C. Anderson (http://www.astroshow.com)
Figure 32: Shepp-Logan Phantom blurred by a gauss psf with noise of a level of about 20%

Figure 33: Shepp-Logan Phantom deblurred by non blind deconvolution with TV regularization. $\lambda_f = 100$
Figure 34: Shepp-Logan Phantom deblurred by non blind deconvolution with TV regularization. \( \lambda_f = 1000 \)

Figure 35: Shepp-Logan Phantom deblurred by blind deconvolution with TV regularization. \( \lambda_f = 1000 \)
Figure 36: point spread function of the restoration shown in Figure 35

Figure 37: Unprocessed image of Jupiter.

Figure 38: Restored image of Jupiter using blind deconvolution.
8 Conclusions

The blind deconvolution is a powerful tool to restore images with little or no a priori knowledge about the PSF that blurred the image. Especially blind deconvolution with TV regularization seems to be very suitable since it is able to restore signals with edges in them.

However there are some significant disadvantages of the method

choice of the regularization Some knowledge is necessary to choose the right regularization method i.e. Tikhonov or total variation. In most cases total variation can be chosen for the picture itself. Yet the 1d examples showed that it is useful to use Tikhonov regularization on the PSF if the PSF is expected to be round shaped. On the other hand if a PSF has edges in it TV regularization for the PSF is a better choice.

complexity Although a f-evaluation and the evaluation of the gradient of the objective function can be implemented fairly quickly using Fast-Fourier-transforms, the method is very expensive due to its iterative nature. The reconstruction can take many minutes to hours on current desktop computers.

choice of the parameters The method is very sensitive towards the choice of the parameters. Although the noise-level can be used as a guide to choose the parameters it is very difficult to find the best set of parameters. The reconstruction qualities reach from totally unreasonable to good results depending on the choice of the parameters. Since most applications are repetitive we can assume that a set parameters that works good on one problem will also work good on a nother problem of the same kind.

multiple minima The problems also suffer from the existence of multiple minima. This means that it is very important to start the minimazation close enough to the “real” solution. This also can result in reconstructions that are completely unreasonable results, even if for the proper choice of the starting point the reconstruction is very good.

Artifacts Numerical results also tend to show different kinds of artifacts in the restoration. In particular the water like transformation of the noise seen in Figure 33 and Gibbs Ringing which appear if the minimization is interrupted before it is completed. Because of the nature of the objective function this is often neccessery to keep the restoration within a reasonable time frame.
The lack of robustness of the method due to the above seems to be a major obstacle for implementation in practice. Never the less the high potential of the blind deconvolution certainly justifies more research in both the analytical and practical aspect of the blind deconvolution. Particularly, since most applications are repetitive - thus it can be worth investigating if for a particular problem the very general blind deconvolution can be replaced by a more specific problem e.g. a general Gaussian blur for MRI.

Future work in the analytical field includes the analysis of the nature of multiple minima as well as the convergence analysis. Due to the extreme nonlinearity of the problem this will be very difficult if not impossible.

Future work in the application is to find more efficient implementation of the minimization. There is also a method to reduce the Gibbs Rimming artifact by restoring the signal in smooth subregions using Gegenbauer polynomials. In this method the fourier coefficients of a signal can be used to derive a set of coefficients for a family of orthogonal polynomials, the normalized Gegenbauer polynomials [3, 10].
References


