



Project in Microengineering (10 credits)
Microengineering Section, Master Semester 4
Spring 2021

Proximal Operators for Nonnegative Inverse Problems

Author

Alejandro Noguerón Arámburu

Supervised by

Dr. Pol del Aguila Pla

Professor

Prof. Michael Unser

October 15, 2021

Contents

1	Introduction	1
1.1	Imaging as an Inverse Problem	1
2	Background	3
2.1	Image Regularization	3
2.2	The Proximal Operator	6
2.2.1	Closed-form Solutions of Functions of Interest	7
2.3	Proximal Optimization	8
2.3.1	Example 1: Proximal Minimization	8
2.3.2	Example 2: Proximal Gradient Method	9
2.3.3	Example 3: ADMM	10
2.4	Scope of the Project	11
3	Methods	11
3.1	Computational Environment	11
3.2	Evaluating Combination of \mathcal{R} with $\delta_{R_+^N}$	11
3.3	Experiments	12
4	Results	12
5	Conclusion	17
6	Acknowledgements	18
	Appendices	19
A	GitHub Repository	19

1 Introduction

The present semester project has the aim of exploring properties of the proximal operators of common image regularizers -used to promote a desirable property like sparsity or smoothness in reconstructed images-, which play a critical role in image reconstruction across different fields of science, engineering and medicine. I will study their behaviour when they are combined with nonnegativity constraints, a term that will penalize the reconstruction infinitely when there is any negative term in the result. This is of particular importance because in scientific imaging, images are formed from physical signals constrained to a known range. Specifically, for a given regularizer $\mathcal{R} : \mathbb{R}^N \rightarrow \mathbb{R}$ we want to test whether

$$\begin{aligned} \text{prox}_{\mathcal{R}+\delta_{\mathbb{R}_+^N}}(\mathbf{v}) &= \text{prox}_{\mathcal{R}}(\text{prox}_{\delta_{\mathbb{R}_+^N}}(\mathbf{v})), \text{ or} \\ \text{prox}_{\mathcal{R}+\delta_{\mathbb{R}_+^N}}(\mathbf{v}) &= \text{prox}_{\delta_{\mathbb{R}_+^N}}(\text{prox}_{\mathcal{R}}(\mathbf{v})). \end{aligned} \tag{1}$$

Any of the two fulfilled conditions can potentially improve computational performance when using proximal optimization algorithms for image reconstruction.

In this section I will describe the general imaging problem, which is recovering a representation of an object from noisy measurements. Furthermore, I will describe how optimization and regularization play a key role in solving it. In Section 2, I will describe some relevant image regularizers, along with the concept of proximal operator and its applicability in image reconstruction problems. Moreover, I will present some examples of optimization algorithms that rely on the proximal operator. In Section 3 I will give a description of the methods and the computational environment I used throughout the project. Finally, in Section 4, I will present and discuss the results of the project.

1.1 Imaging as an Inverse Problem

An imaging problem [1, 2] takes the form

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{2}$$

where $\mathbf{y} \in \mathbb{R}^M$ corresponds to discrete measurements recorded by an imaging system, and $\mathbf{x} \in \mathbb{R}^N$ is the representation of a d -dimensional, continuously defined object. \mathbf{x} and \mathbf{y} are therefore related through the linear operator $\mathbf{H} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ that models the imaging system (for example, the Radon transform in the case of computer tomography). Then, $\mathbf{H}\mathbf{x}$ corresponds to a noiseless representation that reaches a detector with the added noise $\mathbf{w} \in \mathbb{R}^M$. The noise \mathbf{w} is assumed to be independent and identically distributed (i.i.d.), and it is usually modelled as Gaussian.

Therefore, the imaging task consists of recovering \mathbf{x} from the measurement \mathbf{y} . The *forward* model is given by \mathbf{H} (it is assumed to be known, at least up to some variation), and the problem of recovering \mathbf{y} is referred to as an *inverse* problem.

The most immediate solution is to solve the *least squares* optimization problem

$$\arg \min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2 \tag{3}$$

Which makes $\mathbf{H}\mathbf{x}$ as similar to \mathbf{y} as possible, where \mathbf{x} is an optimization variable whose optimal value should be a representation of the original object. Note that, since \mathbf{H} is known, a value of 0

for this objective is equivalent to having a noiseless measurement. Given that the noise is assumed to be i.i.d, solving (3) corresponds to using the maximum likelihood estimation.

The solution $\hat{\mathbf{x}}_{\text{LS}}$ to the least squares problem is available in closed-form as

$$\hat{\mathbf{x}}_{\text{LS}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}. \quad (4)$$

However, inverse problems are generally ill-posed, which means, among other things, that there are many solutions $\hat{\mathbf{x}}_i$ for which $\hat{\mathbf{x}}_1 = \hat{\mathbf{x}}_2$ yields $\mathbf{H}\hat{\mathbf{x}}_1 = \mathbf{H}\hat{\mathbf{x}}_2$, and this often results in non-desirable artifacts. Moreover there is no guarantee that $\hat{\mathbf{x}}_{\text{LS}}$ will be similar to the true \mathbf{x} . To *guide* the solution, it is necessary to include previous knowledge about the object in \mathbf{x} . For the scope of this project, we will look at two terms that can be added to the minimization objective of (3).

The first additional term is used to restrict the solution to nonnegative elements, and it is given by

$$\delta_{\mathbb{R}_+^N}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathbb{R}_+^N, \\ +\infty & \text{if } \mathbf{x} \notin \mathbb{R}_+^N. \end{cases} \quad (5)$$

Here, $\delta : \mathbb{R}^N \rightarrow \mathbb{R}_+ \cup \{+\infty\}$ is an indicator function that forces the solution into a particular set, in this case, the set of nonnegative vectors \mathbb{R}_+^N . Since the values are limited to $\{0, \infty\}$, this term guarantees that the solution will not have any nonnegative element, as the value of the objective function goes to ∞ . On the other hand, inside the valid set, it will have no further effect.

The second term that is added to the objective function is given by:

$$\mathcal{R}(\mathbf{x}) = R(\mathbf{L}\mathbf{x}) \quad (6)$$

In (6), $\mathcal{R} : \mathbb{R}^N \rightarrow \mathbb{R}$ is an operator that *measures* a transform $\mathbf{L} : \mathbb{R}^N \rightarrow \mathbb{R}^Q$ of \mathbf{x} through the scalar function $R : \mathbb{R}^Q \rightarrow \mathbb{R}$. Thus, \mathcal{R} is a scalar quantity that represents a property of \mathbf{x} that is desired to be small. In the case of imaging problems, previous knowledge can often be encoded as sparsity, either directly on \mathbf{x} (the case in which $\mathbf{L} = \mathbf{I}$) or on a transform $\mathbf{L}\mathbf{x}$. Thus, a common choice for R is the ℓ_1 norm, but there are many sophisticated choices (see Section 2.1 for more details).

Finally, it is important to say that while it is necessary to have a measure of similarity between $\mathbf{H}\mathbf{x}$ and \mathbf{y} , the sum of squares is by no means the only choice. Thus, we will denote the measure of similarity between forward model and measurements by $\mathcal{D}(\mathbf{H}\mathbf{x}, \mathbf{y})$, where $\mathcal{D} : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$. However, the choice of \mathcal{D} is outside of the scope of this project.

With that, we arrive at the most common optimization problem in imaging,

$$\arg \min_{\mathbf{x} \in \mathbb{R}} \mathcal{D}(\mathbf{H}\mathbf{x}, \mathbf{y}) + \mathcal{R}(\mathbf{L}\mathbf{x}) + \delta_{\mathbb{R}_+^N}(\mathbf{x}). \quad (7)$$

This minimization problem -evidently more complex than the one presented in (3)- has several difficulties. One of them is that there is no closed form solution as in (3), so the solution has to be found through iterative algorithms. These algorithms usually require first-order information (e.g. the gradient) or second-order information (e.g. the Hessian, which is used in iterative algorithms to achieve faster convergence). However, biomedical imaging has the constraint that even if second derivatives are available, the size of $\nabla^2 f(\mathbf{x}) \in \mathbb{R}^{N^2}$ is the squared of the size of \mathbf{x} . In an image of 256×256 pixels, this is in the order of 10^{10} , which means that it cannot be easily stored. Thus, there is a need to restrict to first-order methods, otherwise the computational memory requirements are simply too high. Moreover, image regularizers in general are or include norms, which are non-smooth and non-differentiable, whilst the nonnegativity constraint is highly non-smooth. Therefore,

the optimization problem (7) is not solvable through algorithms designed for smooth optimization like gradient descent, and proximal algorithms are required.

To solve the objective function, algorithms like the alternating direction method of multipliers (ADMM) are used. ADMM is a common and representative example of an iterative algorithm that can solve the optimization problem (7), but there are several options. ADMM performs one splitting per term in the objective function, and solves its proximal operator, a process that is guaranteed to converge but increments the computational cost and memory requirements for each split. Thus, investigating the proximal operator of common regularizers in combination with nonnegativity constraints could provide significant speed-ups and memory savings in image reconstruction tasks.

2 Background

In this Section I will deepen on the theoretical background required for the project. First, in Section 2.1, I will describe ℓ_1 and ℓ_2 norm regularization and common image regularizers like total variation (TV), group sparsity and the Hessian-Schatten norm. Then, in Section 2.2, I will describe the proximal operator and some relevant optimization algorithms for image reconstruction.

2.1 Image Regularization

As mentioned in Section 1, sparsity is a well known property of real life images and objects. In particular, sparsity on transforms that measure variation in the vicinity of pixels (e.g., the wavelet transform [3], or derivative operators [1, 4]) promote smoothness on an image, and in many cases they are a good indicator of low noise and high fidelity with the true object. Moreover, the ℓ_1 norm is a good objective function to promote sparsity, which is why common image regularizers are a combination of ℓ_1 norms measuring a transform \mathbf{L} that measures a property that is desired to be small.

Other desirable properties in image regularizers are convexity -to make optimization feasible- and invariance to rotation, translation, and scaling, so that they do not depend on any specific coordinate axis.

ℓ_p regularization

As aforementioned, ℓ_1 and ℓ_2 norms are common choices of \mathbf{R} for regularization. When acting on a raw measurement, ℓ_p regularization consists in choosing the point within a validity region (which can be determined by a scaling parameter λ multiplying \mathcal{R} in (7)) that touches the smallest norm ball of the chosen ℓ_p norm. This is exemplified in Figure 1 (reused with permission from [2]) for the ℓ_1 and ℓ_2 norms.

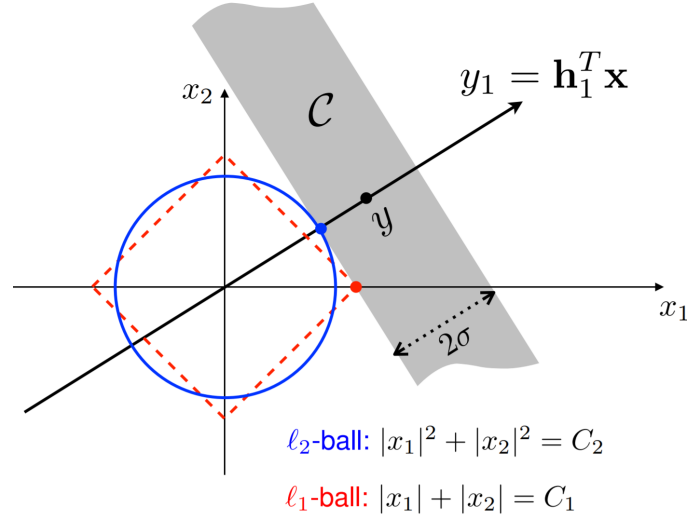


Figure 1: ℓ_1 and ℓ_2 norm regularization on a low dimensional example. The regularization of \mathbf{x} consists in finding the points (in red for the ℓ_1 norm and blue for the ℓ_2 norm) as optimal reconstruction. The gray area corresponds to the data fidelity region. Figure reused with permission from [2].

In Figure 1, in an example of a 2-dimensional inverse problem, the properties of ℓ_1 and ℓ_2 regularization are exemplified. The solution of ℓ_2 norm regularization lies within the space spanned by \mathbf{h}_1^T , but does not necessarily promote sparsity. On the other hand, ℓ_1 regularization promotes sparsity in the solution -under some conditions in \mathbf{h}_1^T .

Group Sparsity

In images where there is prior information on the structure of the image in the form of pixel labelling (e.g., many cell images where structural information can be known [5], or using a spatio-temporal representation of a single image [6, 7]), ℓ_p norm regularization can be applied independently (without overlap) to the different structures. This results in a vector of measurements of each region, and a second norm can be taken over this new vector. This technique has been proven to be successful in image reconstruction tasks, and is known as group sparsity. It is also referred to as $\|\cdot\|_{p,q}$ mixed-norm, and is formalized by

$$\|\mathbf{x}\|_{p,q} = \left(\sum_{i=1}^S \|\mathbf{x}_i\|_p^q \right)^{\frac{1}{q}}, \quad (8)$$

where S is the total number of groups, and \mathbf{x}_i represents the i^{th} group. In short, (8) shows that the ℓ_p norm is taken on the different groups individually, and the ℓ_q norm is taken across the group norms. The different groups \mathbf{x}_i need not have the same size.

Total Variation [4, 8]

Total variation (TV) is a tool that has been used extensively in image reconstruction. It combines the gradient operator $\nabla_n = [\mathbf{D}_1, \dots, \mathbf{D}_n]$ with the mixed $\|\cdot\|_{i,j}$ norm. In this context and applied to an image $\mathbf{x} \in \mathbb{R}^{N \times M}$, $\nabla_2 \mathbf{x} \in \mathbb{R}^{2 \times N \times M}$ is a 3-dimensional array, where the derivatives along the two

original dimensions are stacked to form the third dimension. So, i stands for the norm applied to ∇_n along the 1st dimension -along the values of the gradients in the same position-. After applying this norm, the result has the original dimensions of \mathbf{x} , $\mathbb{R}^{N \times M}$. j is then the norm applied to this matrix to result in a scalar. For example, TV with mixed $\|\cdot\|_{1,1}$ norm is written explicitly as

$$\mathcal{R}(\mathbf{x}) = \sum_{n=1}^N \|\mathbf{D}_1 \mathbf{x}\|_n + \|\mathbf{D}_2 \mathbf{x}\|_n. \quad (9)$$

In (9) it is shown explicitly that 2 different transforms (\mathbf{D}_1 and \mathbf{D}_2) are taken. Inside the summation, the ℓ_1 norm along the first dimension is taken by summing absolute values at each element n . The summation stands for the outer norm, taken over the resulting $\mathbb{R}^{N \times M}$ array. Of course this is equivalent to taking the ℓ_1 norm over the original $\mathbb{R}^{2 \times N \times M}$ array, but the notation serves for illustration purposes.

(9) is sometimes referred to as nonanisotropic TV, since it is not invariant to rotation. On the other hand, TV with mixed $\|\cdot\|_{2,1}$ norm is referred to as isotropic TV, as rotations in the coordinate axis do not affect solution of the minimization of the ℓ_2 norm. It is written explicitly as

$$\mathcal{R}(\mathbf{x}) = \sum_{n=1}^N \sqrt{[\mathbf{D}_1 \mathbf{x}]_n^2 + [\mathbf{D}_2 \mathbf{x}]_n^2}. \quad (10)$$

In (10), it is necessary to write explicitly the two different norms. While this norm is invariant to rotation, its minimization is more complex than the minimization of the nonisotropic TV, and in practice both produce good results. However, there are some limitations with TV, regardless of the choice of i, j . Since it promotes vanishing first order derivatives, the results are usually piece-wise constant, an effect that is known as *staircase effect* (see for example, Figure 5 in Section 4).

Hessian Schatten Norm

As an alternative to TV and a solution for the staircase effect, the Hessian-Schatten norm was proposed in [9]. Instead of using first order derivatives, it uses second order derivatives. It computes the $\|\cdot\|_{S_p,1}$ norm of the Hessian operator $\mathbf{HS} = [\mathbf{D}_{i,j}]_{1 \leq i,j \leq N}$ applied to $x \in \mathbb{R}^N$

$$\mathcal{R}(\mathbf{x}) = \sum_{n=1}^N \left\| \begin{bmatrix} [D_{11}\mathbf{x}]_n & [D_{12}\mathbf{x}]_n \\ [D_{21}\mathbf{x}]_n & [D_{22}\mathbf{x}]_n \end{bmatrix} \right\|_{S_p}, \quad (11)$$

where $\|\cdot\|_{S_p}$ denotes the Schatten norm. It is defined as

$$\|\mathbf{X}\|_{S_p} = \left(\sum_{k=1}^{\min(n_1, n_2)} \sigma_k^p(\mathbf{X}) \right)^{\frac{1}{p}}, \quad (12)$$

where $\sigma_k^p(\mathbf{X})$ denotes the k_{th} singular value of \mathbf{X} .

In contrast to TV, the use of second order derivatives results in piece-wise linear images, far more realistic than the piece-wise constant results of TV. Moreover, the Hessian Schatten norm preserves the properties of convexity and invariance.

None of TV nor the Hessian Schatten norm are differentiable. However, their proximal operators can be taken efficiently, which means that the algorithm of choice for the optimization problem described in (7) is ADMM (see Section 2.3.3).

2.2 The Proximal Operator

The proximal (or proximity) operator [10] of f with parameter λ is defined as

$$\text{prox}_{\lambda f}(\mathbf{v}) = \arg \min_{\mathbf{x}} (f(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{v}\|_2^2), \quad (13)$$

where $f : \mathbb{R}^n \rightarrow \bigcup\{+\infty\}$ is a closed convex function, and $\|\cdot\|$ denotes the Euclidean norm. It has a strong connection with the Moreau envelope of f , defined by

$$M_{\lambda f}(\mathbf{v}) = \inf_{\mathbf{x}} (f(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{v}\|_2^2). \quad (14)$$

So that $\text{prox}_{\lambda f}(\mathbf{v})$ finds the minimal argument of the Moreau envelope. Take as an example the 1-dimensional function $f = \frac{x^2}{2} + \delta_{\mathbb{R}_+}$, a convex function, yet neither smooth not differentiable. Its Moreau envelope for different values of λ is shown in Figure 2.

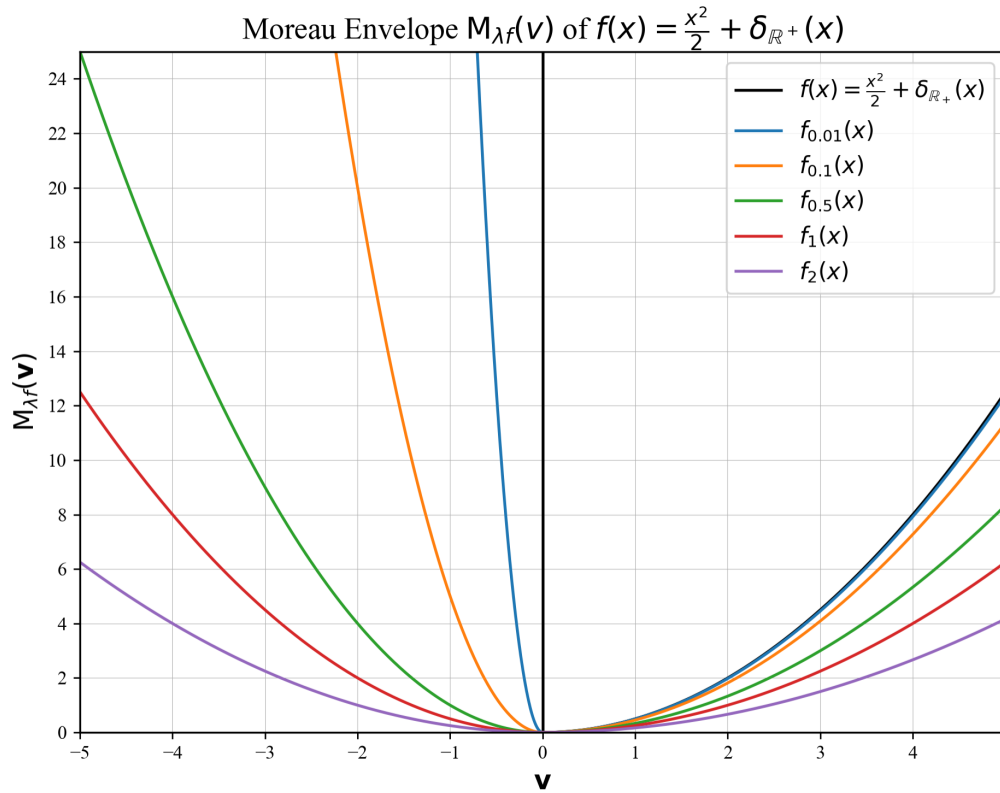


Figure 2: Moreau envelope $M_{\lambda f}$ for $f = \frac{x^2}{2} + \delta_{\mathbb{R}_+}$ and different values of λ .

Figure 2 shows how $M_{\lambda f}$, by adding a quadratic term to f is basically a smoothed version of f , where the parameter λ is a trade-off between fidelity to f and smoothness. Furthermore $M_{\lambda f}$ has several desirable qualities. Its domain is the whole set of real numbers \mathbb{R}^n , regardless of the domain of f . Moreover, it is continuously differentiable, even if f is not, and the set of minimizers of f and $M_{\lambda f}$ is the same, which makes the minimization of $M_{\lambda f}$ and f equivalent problems. The advantage is that minimizing $M_{\lambda f}$ is always a smooth problem, even though $M_{\lambda f}$ can be difficult to evaluate.

2.2.1 Closed-form Solutions of Functions of Interest

Indicator Function

One particularly interesting case is the indicator function, for example the one given in (5). Its proximal operator has a closed-form solution, and it is given by

$$\text{prox}_{\lambda\delta_{\mathbb{R}_+^N}}(\mathbf{v}) = \arg \min_{\mathbf{x}} (\delta_{\mathbb{R}_+^N} + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{v}\|_2^2) = \arg \min_{\mathbf{x} \in \mathbb{R}_+^N} \left(\frac{1}{2} \|\mathbf{x} - \mathbf{v}\|_2^2 \right). \quad (15)$$

As (15) demonstrates, the proximal operator of indicator functions is simply the projection from an input vector into the domain imposed by the function. For (5), this domain is the set of real nonnegative numbers \mathbb{R}_+^N .

Norms

To evaluate the proximal operator of norms, the property known as *Moreau decomposition* [10] is helpful. This property states that

$$\mathbf{v} = \text{prox}_f(\mathbf{v}) + \text{prox}_{f^*}(\mathbf{v}), \quad (16)$$

where f^* is the conjugate function of f . This is very useful to evaluate proximal operators of norms, since their conjugate function is the indicator function $\mathcal{I}_{\mathcal{B}}$, where \mathcal{B} is the unit ball of the norm. As aforementioned, the proximal operator of indicator functions has a closed form solution, and the problem of evaluating the proximal operator of norms reduces to evaluating the projection onto \mathcal{B} . Thus, (16) gives us a simple form of evaluating $\text{prox}_{\lambda f}$. It follows that $\text{prox}_f(\mathbf{v})$ parametrized with λ is written as

$$\text{prox}_{\lambda\ell_p}(\mathbf{v}) = \mathbf{v} - \lambda \Pi_{\mathcal{B}_p}\left(\frac{\mathbf{v}}{\lambda}\right) \quad (17)$$

And finding $\text{prox}_{\lambda f}(\mathbf{v})$ can be done by finding $\Pi_{\mathcal{B}}(\frac{\mathbf{v}}{\lambda})$.

Figure 3 shows some element-wise plots of the closed-form solutions of functions of interest. Explicit derivations can be found in [11].

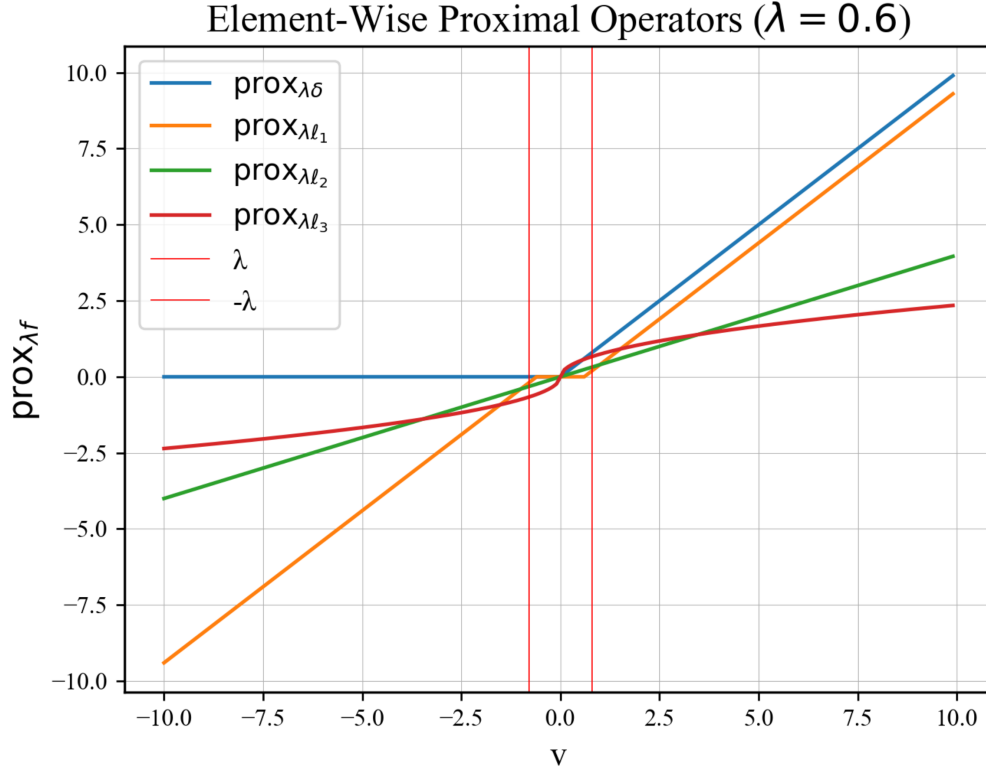


Figure 3: Closed form solutions of element-wise proximal operators: $f = \delta_{\mathbb{R}_+^N}$, $f = \ell_1$, $f = \ell_2$, and $f = \ell_3$. Note that $f = \ell_2$, and $f = \ell_3$ have been parametrized so that $\|\mathbf{v}\| = 1$.

In this simple functions, the effect of the proximal operator of ℓ_p norms and of $\delta_{\mathbb{R}_+^N}$ (defined in (5)) is exemplified. For the nonnegativity constraints, where the proximal operator is the projection into \mathbb{R}_+^N , we can see how nonnegative elements are left unchanged and negative elements are set to 0. On the other hand, the proximal operator of the ℓ_1 norm acts equivalently on all elements, bringing them closer to 0 -without crossing it-, an operation called *soft thresholding*. The proximal operator of the ℓ_2 norm applies a linear transformation that depends on the value of the element v_i , parametrized by the norm of \mathbf{v} (normalized to 1 in Figure 3) and of course by λ . Norms of higher order have different behaviours for different values of \mathbf{v} , but we can see that in general they are element-wise operators.

2.3 Proximal Optimization

In this Section I will describe some representative optimization algorithms that rely on the proximal operator. and are relevant in the context of image reconstruction.

2.3.1 Example 1: Proximal Minimization

The simplest algorithm is proximal minimization, where the update rule

$$\mathbf{x}^{k+1} := \text{prox}_{\lambda f}(\mathbf{x}^k) \quad (18)$$

is simply applying the proximal operator on the current value. In (18), f is still a closed convex scalar function, \mathbf{x} is the optimization variable, and k is the iteration counter. One step of the proximal optimization algorithm is shown in figure 4.

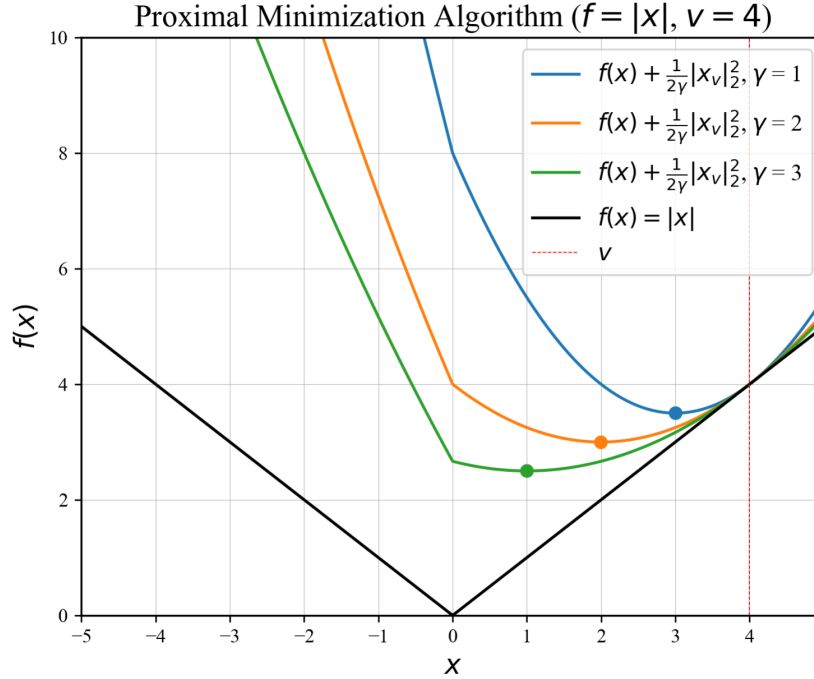


Figure 4: Proximal minimization algorithm (18). The image shows one step of the proximal minimization algorithm applied to the simple function $f(x) = |x|$ (in black). Three different values of $\lambda \in \{1, 2, 3\}$ are shown. The whole objective function is shown, and its minimum -the result of the proximal operator- is indicated in the plot.

In figure 4, the function $f(x) = |x|$ is minimized via the proximal minimization algorithm, arbitrarily choosing $v = 4$ as starting point. The objective function of the proximal operator is plotted for $\lambda \in \{1, 2, 3\}$. Note how, since $f(x)$ is the ℓ_1 norm of a scalar, figure 4 is in complete agreement with figure 3, where it is shown the element wise operation of $\text{prox}_{\lambda\ell_1}$ is soft thresholding, with λ as a threshold.

2.3.2 Example 2: Proximal Gradient Method

An algorithm that has found further applications is the *proximal gradient method*. It acts on a convex objective function $f + g$, where $f : \mathbb{R}^N \rightarrow \mathbb{R}$ is smooth and differentiable, but $g : \mathbb{R}^N \rightarrow \mathbb{R} \cup \{+\infty\}$ is not. Note that since g includes ∞ it can be the indicator function in (7).

The proximal gradient method update rule is given by:

$$\mathbf{x}^{k+1} := \text{prox}_{\lambda g}(\mathbf{x}^k - \lambda^k \nabla f(\mathbf{x}^k)), \quad (19)$$

where the minimization objective is split into two, f and g . Then, an iteration over the whole objective includes a gradient step in f -note that f is both smooth and differentiable-, and a

proximal minimization step in g . If g is an indicator function, then as (15) shows, the minimization step over g is the projection on the valid set, and the proximal gradient method becomes equivalent to projected gradient descent.

2.3.3 Example 3: ADMM

The alternating direction method of multipliers (ADMM) [12] is a method for minimizing objective functions of the form.

$$\sum_{n=1}^N F_n(\mathbf{H}_n \mathbf{x}), \quad (20)$$

where, as before, $\mathbf{x} \in \mathbb{R}^N$, and each term of the summation has a linear operator $\mathbf{H}_n : \mathbb{R}^N \rightarrow \mathbb{R}^{Q_n}$ acting on \mathbf{x} and a scalar function $F_n : \mathbb{R}^{Q_n} \rightarrow \mathbb{R}$ that has $\mathbf{H}_n \mathbf{x}$ as input. This form clearly represents (7), since it is formed out of three separable functions, each with its own transforms and functions. For $n = 1$, the transform $\mathbf{H}_1 = \mathbf{H}$ is the forward model and $F_1 = \mathcal{D}$ is the data fidelity metric. For $n = 2$, shown in (6), $\mathbf{H}_2 = \mathbf{L}$ and $F_2 = \mathbf{R}$, and for $n = 3$, the nonnegativity constraint, $\mathbf{H}_3 = \mathbf{I}$ is the identity operator and $F_n = \delta_{\mathbb{R}_+^N}$. Moreover, F_n for all n needs to be convex, but need not be neither differentiable nor smooth. Therefore, from the examples we have seen, ADMM has the broadest applicability and is one of the most relevant algorithms for image reconstruction tasks.

The principle of ADMM is to find a saddle point of the scaled augmented Lagrangian formulation of (20), given by

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_n, \mathbf{w}_1, \dots, \mathbf{w}_n) = \sum_{n=1}^N \frac{1}{2} \rho_n \left\| \mathbf{H}_n \mathbf{x} - \mathbf{y}_n + \frac{\mathbf{w}_n}{\rho_n} \right\|^2 + F_n(\mathbf{y}_n), \quad (21)$$

through the update rules

$$\begin{aligned} \mathbf{x}^{k+1} &:= \arg \min_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \mathbf{y}_1^k, \dots, \mathbf{y}_n^k, \mathbf{w}_1^k \dots \mathbf{w}_n^k) \\ \mathbf{y}_1^{k+1} &:= \arg \min_{\mathbf{y}_1} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{y}_1, \dots, \mathbf{y}_n, \mathbf{w}_1^k \dots \mathbf{w}_n^k) \\ &\vdots \\ \mathbf{y}_n^{k+1} &:= \arg \min_{\mathbf{y}_n} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{y}_1, \dots, \mathbf{y}_n, \mathbf{w}_1^k \dots \mathbf{w}_n^k) \\ \mathbf{w}_1^{k+1} &:= \mathbf{w}_1^k + \rho(\mathbf{x}^{k+1} - \mathbf{y}_1^{k+1}) \\ &\vdots \\ \mathbf{w}_n^{k+1} &:= \mathbf{w}_n^k + \rho(\mathbf{x}^{k+1} - \mathbf{y}_n^{k+1}). \end{aligned} \quad (22)$$

Since \mathcal{L}_ρ is separable accross n , an update step only acts on the terms of the corresponding summation. As such, the explicit update step of \mathbf{y}_n is

$$\mathbf{y}_n^{k+1} := \arg \min_{\mathbf{y}_n} \left(F_n(\mathbf{y}_n) + \frac{\rho}{2} \left\| \mathbf{H}_n \mathbf{x} - \mathbf{y}_n - \frac{1}{\rho} \mathbf{w}_n^k \right\|^2 \right) = \text{prox}_{F_n \lambda}(\mathbf{x}^{k+1} - \mathbf{w}_n^k), \quad (23)$$

Where we have made $\lambda = 1/\rho$.

Therefore, ADMM is a common choice when the proximal operator of $f = F_n$, $\text{prox}_{F_n \lambda}(\mathbf{v})$ is known, but the proximal operator of $f = F_n + F_{n+1}$, $\text{prox}_{(F_n + F_{n+1}) \lambda}(\mathbf{v})$, is not known or it is hard to evaluate, as it is the case for combinations norms with nonnegativity constraints.

2.4 Scope of the Project

One of the drawbacks of solving an inverse imaging problem through (7) is the splitting required to use ADMM. This splitting results in the need to optimize two additional variables of the same size of the original per extra function. Consequently, this project aims to mitigate this issue by investigating the combination of the proximal operator of common image regularizers with nonnegativity constraints.

In other words, we will study the $\text{prox}_{\lambda f}(x)$ for

$$f(\mathbf{x}) = \mathcal{R}(\mathbf{x}) + \delta_{\mathbb{R}_+^N}(\mathbf{x}), \quad (24)$$

and by combination it is meant taking first the proximal of one of the two functions in f , then the proximal of the other function, and then check whether they are equal to prox_f . The goal is to find regularizers for which one of the following two conditions

$$\text{prox}_{\mathcal{R}+\delta_{\mathbb{R}_+^N}}(\mathbf{x}) = \text{prox}_{\delta_{\mathbb{R}_+^N}}(\text{prox}_{\mathcal{R}}(\mathbf{x})), \text{ or} \quad (25)$$

$$\text{prox}_{\mathcal{R}+\delta_{\mathbb{R}_+^N}}(\mathbf{x}) = \text{prox}_{\mathcal{R}}(\text{prox}_{\delta_{\mathbb{R}_+^N}}(\mathbf{x})) \quad (26)$$

are met. This has been exploited before in [6, 7] and the conditions for it to happen is a topic of active research in convex analysis [13].

If, for a given regularizer, any of the two equations 25 and 26 hold, it is possible to reconstruct an image using the right-hand side of the valid equation. This means that there is no need to treat \mathcal{R} and δ separately, which has the advantage of not having to store the extra variables generated during ADMM splitting. Therefore, this study has the potential of significantly reducing the computational and memory load of solving imaging problems.

3 Methods

3.1 Computational Environment

All the experiments were carried out in Jupyter Notebooks, with Python version 3.7.9, using standard libraries like NumPy, SciPy and Matplotlib. To obtain most proximal operators, I used CVXPY, a Python modelling library to solve convex optimization problems, using the solver MOSEK. This adapts well to image regularizers, since, as it was explained in Section 2.1, they are convex functions.

Specific library versions are found in the requirements file of the project's Github repository. Furthermore, all the Jupyter Notebooks are found in the same repository.

3.2 Evaluating Combination of \mathcal{R} with $\delta_{\mathbb{R}_+^N}$

To validate each of (25) and (26), I first created a random array (1-dimensional or 2-dimensional as required by the regularizer), and then used CVXPY to calculate the right-hand and the left-hand sides of both equations. Then, I created an error array, to calculate and store the maximum and average errors. The process was repeated 100 times on different arrays (of the same size and sampled

from the same distribution) and the average and maximum errors were taken over the results of the 100 arrays.

To generate the input random arrays, I used Gaussian (with $\mu = 0$ and $\sigma \in [1, 255]$) and uniform distributions in several ranges ($[0, 1]$, $[0, 100]$, $[0, 255]$). Since CVXPY solves optimization problems numerically, there is never a complete equality of both arrays, and an arbitrary decision has to be taken on whether (25) and (26) are valid or not. As a general trend, the more complex the regularizer (i.e. the more CVXPY functions are used) the lower accuracy, and the larger the arrays the bigger the accuracy. This of course results in computational constraints on the size of the arrays, a constraint that is more relevant for complex regularizers. Consequently the decision on whether two expressions are evaluated to be equal will be explained in a case by case basis in Section 4.

3.3 Experiments

For some complex regularizers of interest, and to compare the results and behaviour on real images, I performed some experiments. These were done on much larger images than the ones use during the experimental procedure, in which the true behaviour of the regularizer is clearly visible. The downside of this experiments is that they are too computationally demanding to perform many of them.

In particular, I did 2 experiments, one for nonisotropic TV regularization and one for Hessian-Schatten norm regularization. In both cases, I took a standard test image, normalized it to the range $[0, 1]$, and added white noise to it. Then, I used it as the input of the left-hand side and right hand side of (25) and (26). For an analysis on the results, I compared them visually, calculated the numerical differences and calculated the SNR to the original.

4 Results

In this section I will present the results of the evaluation of (25) and (26), for *simple* regularizers (e.g. norms acting on the raw arrays) to more complex, state-of-the-art image regularizers. In these cases, I will also present some experiments on standard test images. A summary of the results is presented in Table 1.

Dimension	\mathbf{L}	\mathbf{R}	$\text{prox}_\delta(\text{prox}_{\mathcal{R}}(\mathbf{v}))$	$\text{prox}_{\mathcal{R}}(\text{prox}_\delta(\mathbf{v}))$	Parameters
1, 2	I	$\ \cdot\ _1$	$\checkmark(10^{-5})$	$\checkmark(10^{-7})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{100}$
1, 2	I	$\ \cdot\ _2$	$\times(10^{-3})$	$\checkmark(10^{-6})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{100}$
1, 2	I	$\ \cdot\ _p$	$\times(10^{-3})$	$\checkmark(10^{-7})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{100}$
1, 2	I	$\ \cdot\ _p^p$	$\checkmark(10^{-6})$	$\checkmark(10^{-6})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{100}$
2	I	$\ \cdot\ _{S_1}$	$\times(10^{-2})$	$?(10^{-3})$	$\mathcal{N}(0, 100), x \in \mathbb{R}^{2 \times 2}$
2	I	$\ \cdot\ _{S_2}$	$\times(10^{-2})$	$?(10^{-3})$	$\mathcal{N}(0, 100), x \in \mathbb{R}^{2 \times 2}$
2	I	$\ \cdot\ _{S_3}$	$\times(10^{-2})$	$?(10^{-3})$	$\mathcal{N}(0, 100), x \in \mathbb{R}^{2 \times 2}$
1	∇_1	$\ \cdot\ _1$	$\checkmark(10^{-6})$	$\times(10^{-1})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{50}$
1	∇_1	$\ \cdot\ _p$	$\times(10^{-2})$	$\times(10^{-2})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{50}$
1	∇_1	$\ \cdot\ _p^p$	$\times(10^{-2})$	$\times(10^{-2})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{50}$
2	∇_2	$\ \cdot\ _{1,1}$	$\checkmark(10^{-7})$	$\times(10^{-1})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{40 \times 40}$
2	∇_2	$\ \cdot\ _{p,q}$	$\times(10^{-2})$	$\times(10^{-1})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{40 \times 40}$
2	∇_2	$\ \cdot\ _{p,q}^{p,q}$	$\times(10^{-1})$	$\times(10^{-1})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{40 \times 40}$
1, 2	I	$\ \cdot\ _{1,1}$	$\checkmark(10^{-7})$	$\checkmark(10^{-7})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{40 \times 40}$
1, 2	I	$\ \cdot\ _{p,q}$	$\times(10^{-2})$	$\checkmark(10^{-6})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{40 \times 40}$
1, 2	I	$\ \cdot\ _{p,q}^{p,q}$	$\times(10^{-4})$	$\checkmark(10^{-7})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{40 \times 40}$
1, 2	I	$\ \cdot\ _{1,q}$	$\checkmark(10^{-7})$	$\checkmark(10^{-7})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{40 \times 40}$
1, 2	I	$\ \cdot\ _{p,1}^{p,q}$	$\checkmark(10^{-7})$	$\checkmark(10^{-7})$	$\mathcal{N}(0, 1), x \in \mathbb{R}^{40 \times 40}$
2	HS	$\ \cdot\ _{S_{1,p}}$	$\times(10^{-1})$	$\times(10^{-1})$	$\mathcal{N}(0, 10), x \in \mathbb{R}^{15 \times 15}$
2	HS	$\ \cdot\ _{S_{2,p}}$	$\times(10^{-1})$	$\times(10^{-1})$	$\mathcal{N}(0, 10), x \in \mathbb{R}^{15 \times 15}$
2	HS	$\ \cdot\ _{S_{\infty,p}}$	$\times(10^{-1})$	$\times(10^{-1})$	$\mathcal{N}(0, 10), x \in \mathbb{R}^{15 \times 15}$

Table 1: Experimental Results. The first three columns describe the regularizer \mathcal{R} , while the fourth and fifth columns present the results obtained with the average absolute error. The last column shows the parameters used to obtain the presented results. In all cases, 100 different arrays were sampled and tested, and the average absolute error is presented over the hundred samples. The first two rows show results that were previously known, and are therefore useful to test the setup of the project. The rest of the lines separate the table by transform applied to the original signal/image. The second block corresponds to simple norms, the third block to TV regularizers, the fourth one to Group Sparsity regularizers and the fifth block to Hessian-Schatten norm regularizers.

As explained in section 3.1, I used CVXPY to evaluate all image regularizers, constructing them in terms of combinations of simple convex functions. In particular, for the derivatives needed to calculate TV and the Hessian-Schatten norm, I used only the valid region of the resulting array, to avoid boundary artifacts.

ℓ_1, ℓ_2 norm

The ℓ_1 and ℓ_2 norms were studied with the purpose of validating the experimental setup, as these results are already known in the literature [7]. Moreover, as presented in section 2.2.1, and depicted in figure 3, there is a closed-form solution. However, I still used the experimental setup as a validation of the methodology.

As shown in table 1, the results agree with the literature. In the case of the ℓ_1 norm, both (25) and (26) are valid for $\mathcal{R} = \ell_1$, with average absolute errors in the order of 10^{-5} for (26) and of 10^{-7}

for (25). In the case of $\mathcal{R} = \ell_2$, I also recovered the results presented in [7]. I found that (26) is valid with average absolute errors in the order of 10^{-6} , while (25) does not hold true, with average absolute errors in the order of 10^{-3} .

This error rates, applied on simple regularizers and with 1-dimensional and 2-dimensional, reasonably big arrays, provide guidelines on what I will consider to validate or not equations (25) and (26).

ℓ_p and ℓ_p^p norms

After validating the experimental methodology, I made an investigation of the ℓ_p norm (for $p \in \mathbb{Z}^+$). The results show that the properties found in [7, 6] for the ℓ_2 norm can be generalized, which means that any norm of the ℓ_p norm family - where p is a positive integer- satisfies (26), with average absolute errors in the order of 10^{-7} .

Following the previous results and looking at how both (26) and (25) hold true for the ℓ_1 norm (a particular case where $\ell_p^p = \ell_p$), I looked further at the family of norms ℓ_p^p . The results are very promising, as I found that both (26) and (25) hold true for all ℓ_p^p norms. Moreover, all average absolute errors are in the order of 10^{-6} , an error comparable to the one found for the ℓ_1 and ℓ_2 norms.

Schatten Norms

While they are not used as image regularizers by themselves, I also studied the family of Schatten Norms for $p \in \{1, 2, \infty\}$, mostly motivated by their use in the Hessian-Schatten norm. I studied different combinations of parameters (matrices of sizes ranging from 2×2 to 20×20 , Gaussian distributions with σ in the range $[0, 100]$), and in every case got similar results to the ones presented in table 1. In the results I show, the ratio $\frac{\sigma}{\epsilon}$ (where ϵ is the average absolute error) is high enough to suggest that the Schatten (for $p \in \{1, 2, \infty\}$) norm might fulfill (26), and encouraging enough to study the Hessian-Schatten image regularizer. However, the experimental setup used in this project is unable to arrive to a definitive conclusion, and a close-form study should be done.

As discussed in Section 3.2, in general, bigger arrays have lower error rates. In this regard, the Schatten norms are the first functions to have a computational limitation, as the results presented were taken on arrays much smaller than the arrays used to evaluate ℓ_p . This is because evaluating the Schatten norm of larger matrices involves expensive singular value computations, and it was unfeasible to perform an experiment of the same size as the one performed for the ℓ_p norm. Therefore, I arrive to the conclusion that further study is required to determine the results for the Schatten norm.

Group Sparsity

The evaluation of group sparsity was done over both 1-dimensional and 2-dimensional arrays. As presented in table 1, the results were the same in both cases, which is completely expected since we are taking p -norms, and the dimensionality of the array does not play a role. The average absolute errors are in the order of 10^{-7} for the cases that I judged satisfactory. In particular, in every single studied case (all possible combinations of $\|\cdot\|_{p,q}$ and $\|\cdot\|_{p,q}^{p,q}$, where $\|\cdot\|_{p,q}^{p,q}$ stands for taking an inner ℓ_p^p and an outer ℓ_q^q norm), (26) holds true. This was expected from the previous results found on norms, and even though not every single case of (25) holds true, as explained in section 1 only one of both equations is required to avoid splitting during reconstruction. This encourages the use of group sparsity as regularizer, as there are no particular conditions to be met during its use to allow

for reduced splitting.

Total Variation

As described in section 2.1, TV is a regularizer of utmost importance in the signal and image processing fields. For the 1-dimensional cases, following the results previously presented, I studied $\mathcal{R} = \|\mathbf{D}\mathbf{x}\|_p$ and $\mathcal{R} = \|\mathbf{D}\mathbf{x}\|_p^p$. As table 1 shows, only the case of $\mathcal{R} = \|\mathbf{D}\mathbf{x}\|_1$ makes (25) valid, with an average absolute error in the order of 10^{-6} . On all the other cases ($\mathcal{R} = \|\mathbf{D}\mathbf{x}\|_p$, and $\mathcal{R} = \|\mathbf{D}\mathbf{x}\|_p^p$, both for $p \in \mathbb{Z}^+ \setminus \{1\}$), average absolute errors are in the order of 10^{-2} , which demonstrate that neither of (25) and (26) hold true.

In the case of 2-dimensional arrays or images, the results obtained are very similar results. I tried several combinations of the $\|\cdot\|_{p,q}$ and the $\|\cdot\|_{p,q}^{p,q}$ mixed norms. Table 1 shows that only the combination $p = 1$ and $q = 1$ satisfies (25), with average absolute errors in the order of 10^{-7} . As mentioned in section 2.1, this combination is known as nonisotropic TV and it is a well known, widely used regularizer. The rest of the combinations show average absolute errors in the range of $[10^{-1}, 10^{-3}]$, depending on the combination.

To further explore the nonisotropic TV (the explicit expression is given in (9) and, as explained above, has several desirable properties, with (25) being an additional one), I decided to test on a toy image. I choose the Shepp-Logan phantom, as it is a standard test image composed of piece-wise constant segments, so it is optimal for exemplifying the effects of TV minimization.

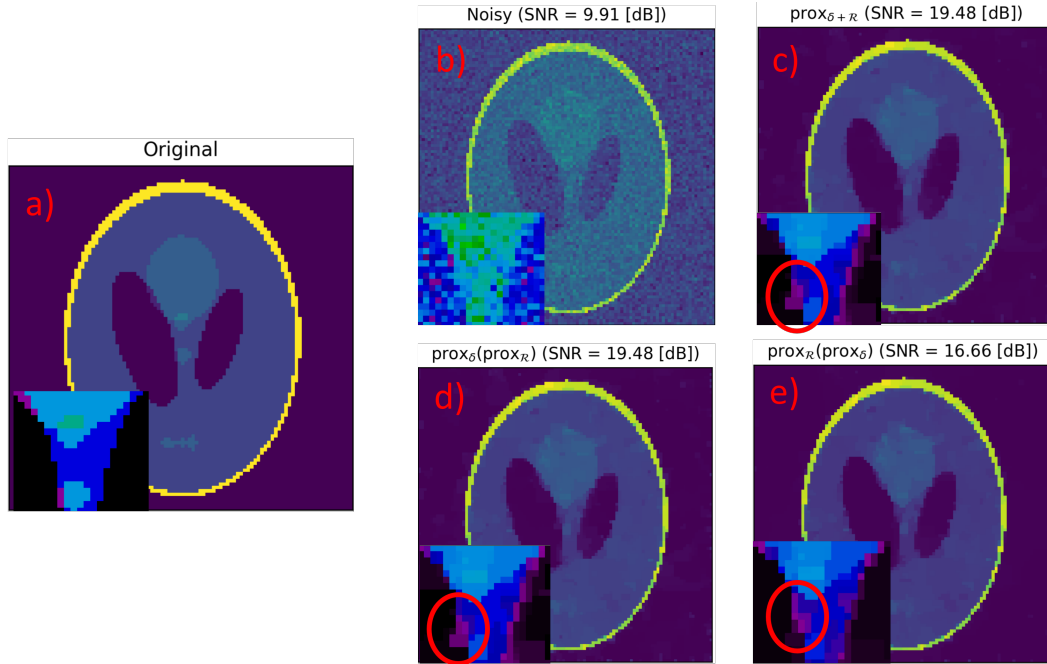


Figure 5: Results of Total Variation with nonnegativity constraints regularization on the Shepp Logan Phantom with Gaussian noise. a) shows original image, b) with Gaussian noise added, c) $\text{prox}_{\mathcal{R} + \delta_{\mathbb{R}_+^N}}(\mathbf{v})$ (ground truth), d) $\text{prox}_{\delta_{\mathbb{R}_+^N}}(\text{prox}_{\mathcal{R}}(\mathbf{v}))$ (found in experiments -see table 1- to be equal to c), and e) $\text{prox}_{\mathcal{R}}(\text{prox}_{\delta_{\mathbb{R}_+^N}}(\mathbf{v}))$. In each of the images, the center region is zoomed-in, to show some of the errors. Moreover, the SNR is shown in [dB].

In Figure 5 we can see (25) and (26) applied to a real image. The original image was set to a size of 128×128 , and normalized to the range $[0, 1]$. The added noise was sampled from the distribution $\mathcal{N}(0, 0.08)$. Just as in the random array scenario (see Table 1), between figures 5c) and 5d) there is an average absolute error in the order of 10^{-7} , and the maximum absolute error is in the order of 10^{-5} . Moreover, both visual inspection (see zoomed-in region for detailed zones) and SNR in [dB] show that both results are equal (and that there are indeed some differences between 5c) and 5e)). This promising result, that to the best of our knowledge is previously unknown, encourages the investigation of a closed-form solution of $\text{prox}_{\text{TV} + \delta_{\mathbb{R}_+^N}}(\mathbf{v})$ and $\text{prox}_{\text{TV}}(\text{prox}_{\delta_{\mathbb{R}_+^N}}(\mathbf{v}))$, as well as the use of this result in image reconstruction libraries, as it has the potential to speed up image reconstruction.

Hessian-Schatten Norm

For evaluation of the Hessian Schatten Norm, as for the evaluation of the simple Schatten norm, the results are not compelling enough to jump to a conclusion, and further investigation is required. Moreover, there are several technical constraints that hinder the proper evaluation of the Hessian Schatten Norm using CVXPY. As aforementioned, larger arrays with a higher σ seem to have slightly more exact results, but the computational requirements of evaluating the Hessian-Schatten Norm make it unfeasible to evaluate bigger arrays than the ones presented in table 1. So while the results presented do not completely rule out the Hessian-Schatten norm for reduced splitting, a more efficient evaluation of the proximal operator of the Hessian-Schatten norm with nonnegativity constraints (an evaluation of the Hessian Schatten norm is presented in [14]) is required to test for reduced splitting.

For further evaluation, I performed an experiment on the standard test image **peppers**. The image was first normalized to the range $[0, 1]$, transformed to have a size of 64×64 and finally had Gaussian noise added, sampled from $\mathcal{N}(0, 0.1)$. Regularization was done with $p = 1$, which is the nuclear norm. The results are shown on figure 6

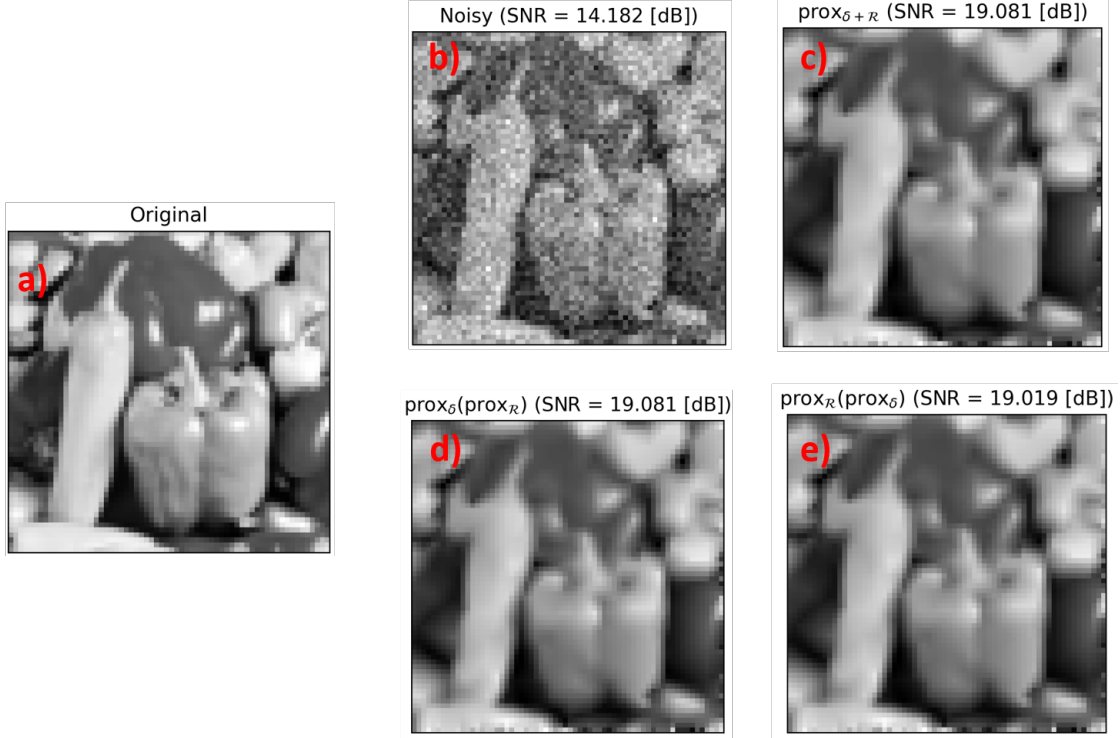


Figure 6: Results of Hessian-Schatten norm with nonnegativity constraints regularization on standard *Peppers* image with added Gaussian noise. a) shows original image in the range $[0, 1]$, b) with Gaussian noise added, sampled from $\mathcal{N}(0, 0.1)$, c) $\text{prox}_{\mathcal{R}+\delta_{\mathbb{R}_+^N}}(\mathbf{v})$ (ground truth), d) $\text{prox}_{\delta_{\mathbb{R}_+^N}}(\text{prox}_{\mathcal{R}}(\mathbf{v}))$, and e) $\text{prox}_{\mathcal{R}}(\text{prox}_{\delta_{\mathbb{R}_+^N}}(\mathbf{v}))$. Moreover, the *SNR* is shown in *dB*.

The results of the experiment on a standard test image, shown on Figure 6 show the effectiveness of Hessian-Schatten norm regularization for denoising. Moreover, the experiment encourages the investigation of equation (25), as the maximum error between images 6c) and 6d) -that is, the left-hand and right-hand sides of (25)- is in the order of 10^{-6} , and the *SNR* in [dB] is the same up to the 6th digit. On the other hand, the maximum error between images 6c) and 6e) is in the order of 10^{-2} . While this error is not highly visible in the image, it does rule out (26).

While the results of the experiment are encouraging, they are by no means compelling enough, as this experiment is just 1, out of the 100 (sampled from different distributions) that are used for the results presented in table 1. For further exploration, the same kind of experiment can be carried out by evaluating the proximal operator of the regularizer, instead of using a numerical approximation, as done here.

5 Conclusion

In this project I have presented a thorough study of the combination of the proximal operator of common image regularizers with nonnegativity constraints. This is of interest because current algorithms of choice like ADMM perform variable splitting, which has a heavy computational cost.

Therefore a good combination can reduce the computational load of state-of-the-art image reconstruction techniques.

Specifically, my experimental results suggest that all the family of norms $\|\cdot\|_p$ and $\|\cdot\|_p^p$ combine well with nonnegativity constraints, a result of interest in the domains of signal and image reconstructions. Furthermore, I found that group sparsity -or the $\|\cdot\|_{p,q}$ mixed-norm- also has good combination, as all studied cases comply with (26). In the case of more complex image regularizers, I have found that nonisotropic TV combines well with nonnegativity constraints. Finally, I studied the Hessian-Schatten norm, and while the result was not compelling enough to jump to a conclusion, it is promising enough to further investigate the topic.

All of these cases are of interest because they can have a substantial effect on the performance of image reconstruction methods, and they are all commonly used regularizers. The next step is to prove the results found here using closed-form solutions, though it is out of the scope of this project. Moreover, as further work and before proving the results for closed form solutions, direct testing in an optimization library could be made. This would also allow testing on more varied settings, and images of different kinds.

6 Acknowledgements

I would like to thank my supervisor, Dr. Pol del Aguila Pla, for constantly solving all kinds of doubts, from technical to theoretical. I also want to thank Pakshal Bohra, who provided me with MATLAB code to perform a similar 1-dimensional TV experiment. Finally, from EPFL's Biomedical Imaging Group, I would like to thank Thomas Debarre, and Prof. Michael Unser for fruitful discussions and feedback on the project.

Appendices

A GitHub Repository

All the Jupyter notebooks, Python modules, and code in general that was used to generate the results presented in this report can be found in the Github repository *Alejandro-1996/ProximalOperatorsProject* ([link here](#)).

References

- [1] E. Soubies, F. Soulez, M. T. McCann, T.-a. Pham, L. Donati, T. Debarre, D. Sage, and M. Unser, “Pocket guide to solve inverse problems with GlobalBioIm,” *Inverse Problems*, vol. 35, p. 104006, Sept. 2019. 1, 3
- [2] M. Unser and P. del Aguila Pla, “Tutorial: Biomedical image reconstruction—From foundations to deep neural networks,” *International Conference in Acoustic, Speech and Signal Processing*, May 2020. 1, 3, 4
- [3] C. Vonesch and M. Unser, “A fast multilevel algorithm for wavelet-regularized image restoration,” *IEEE Transactions on Image Processing*, vol. 18, pp. 509–523, Mar. 2009. 3
- [4] A. Chambolle and P.-L. Lions, “Image recovery via total variation minimization and related problems,” *Numerische Mathematik*, vol. 76, pp. 167–188, Apr. 1997. 3, 4
- [5] J.-C. Baritau, K. Hassler, M. Bucher, S. Sanyal, and M. Unser, “Sparsity-driven reconstruction for FDOT with anatomical priors,” *IEEE Transactions on Medical Imaging*, vol. 30, pp. 1143–1153, May 2011. Conference Name: IEEE Transactions on Medical Imaging. 4
- [6] P. del Aguila Pla and J. Jaldén, “Cell detection by functional inverse diffusion and non-negative group sparsity—Part I: Modeling and inverse problems,” *IEEE Transactions on Signal Processing*, vol. 66, pp. 5407–5421, Oct. 2018. 4, 11, 14
- [7] P. del Aguila Pla and J. Jaldén, “Cell detection by functional inverse diffusion and non-negative group sparsity—Part II: Proximal optimization and performance evaluation,” *IEEE Transactions on Signal Processing*, vol. 66, pp. 5422–5437, Oct. 2018. 4, 11, 13, 14
- [8] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock, “An introduction to total variation for image analysis.” Nov. 2009. 4
- [9] S. Lefkimmiatis, J. P. Ward, and M. Unser, “Hessian Schatten-norm regularization for linear inverse problems,” *IEEE Transactions on Image Processing*, vol. 22, pp. 1873–1888, May 2013. 5
- [10] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1, pp. 127–239, Jan. 2014. 6, 7
- [11] P. L. Combettes and J.-C. Pesquet, “proximal thresholding algorithm for minimization over orthonormal bases,” *SIAM Journal on Optimization*, vol. 18, pp. 1351–1376, Nov. 2007. Publisher: Society for Industrial and Applied Mathematics. 7
- [12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, pp. 1–122, Jan. 2011. 10
- [13] S. Adly, L. Bourdin, and F. Caubet, “On a decomposition formula for the proximal operator of the sum of two convex functions,” *Journal of Convex Analysis*, vol. 26, no. 3, pp. 699–718, 2019. 11
- [14] S. Lefkimmiatis and M. Unser, “Poisson image reconstruction with Hessian Schatten-norm regularization,” *IEEE Transactions on Image Processing*, vol. 22, pp. 4314–4327, Nov. 2013. 16