PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado

Novel Edge-Preserving Filtering Model Based on the Quadratic Envelope of the ℓ_0 Gradient Regularization

Tesis para obtener el grado académico de Magíster en Procesamiento de Señales e Imágenes Digitales que presenta:

Eduar Aníbal Vásquez Ortiz

Asesor:

Paul Antonio Rodríguez Valderrama

Lima, 2022

Informe de Similitud

Yo, Paul Antonio Rodríguez Valderrama, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada Novel Edge-Preserving Filtering Model Based on the Quadratic Envelope of the Gradient Regularization del autor Eduar Aníbal Vásquez Ortiz, dejo constancia de la siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 12%. Así lo consigna el reporte de similitud emitido por el software Turnitin el 15/12/2022.
- He revisado con detalle dicho reporte y la Tesis, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas

Lugar y fecha: Lima, 15 de diciembre de 2022.

Asesor: Rodriguez Valderrama, Paul Antonio	
DNI: 07754238	Firma:
ORCID: 0000-0002-8501-0907	

Abstract

In image processing, the ℓ_0 gradient regularization (ℓ_0-grad) is an inverse problem which penalizes the ℓ_0 norm of the reconstructed image's gradient. Current state-of-the art algorithms for solving this problem are based on the alternating direction method of multipliers (ADMM). ℓ_0-grad however, reconstructs images poorly in cases where the noise level is large, giving images with plain regions and abrupt changes between them, that look very distorted. This happens because it prioritizes keeping the main edges but risks losing important details when the images are too noisy. Furthermore, since $\|\nabla \mathbf{u}\|_0$ is a non-continuous and non-convex regularizer, ℓ_0-grad can not be directly solved by methods like the accelerated proximal gradient (APG).

This thesis presents a novel edge-preserving filtering model ($Q\ell_0$-grad) that uses a relaxed form of the quadratic envelope of the ℓ_0 norm of the gradient. This enables us to control the level of details that can be lost during denoising and deblurring. The $Q\ell_0$-grad model can be seen as a mixture of the Total Variation and ℓ_0-grad models. The results for the denoising and deblurring problems show that our model sharpens major edges while strongly attenuating textures. When it was compared to the ℓ_0-grad model, it reconstructed images with flat, texture-free regions that had smooth changes between them, even for scenarios where the input image was corrupted with a large amount of noise. Furthermore the averages of the differences between the obtained metrics with $Q\ell_0$-grad and ℓ_0-grad were +0.96 dB SNR (signal to noise ratio), +0.96 dB PSNR (peak signal to noise ratio) and +0.03 SSIM (structural similarity index measure). An early version of the model was presented in the paper *Fast gradient-based algorithm for a quadratic envelope relaxation of the ℓ_0 gradient regularization* which was published in the international and indexed conference proceedings of the XXIII Symposium on Image, Signal Processing and Artificial Vision.

Keywords

Accelerated proximal gradient, ℓ_0 gradient minimization, quadratic envelope.

Contents

Chapter 1

Introduction

In this chapter, we present a brief explanation of the ℓ_0 gradient regularization (ℓ_0-grad) and the problem it has when working with very noisy images. This issue was the starting point for developing a new filtering model based on ℓ_0-grad. Also, we show the main objective and the specific ones.

1.1 Motivation

The ℓ_0 gradient regularization image smoothing method [1] consists of minimizing the sum of the ℓ_0 norm of the gradient and a quadratic data-fidelity term:

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 + \lambda\|\nabla\mathbf{u}\|_0. \tag{1.1}$$

There exist several methods to solve this minimization problem with a good approximation: introducing auxiliary variables to expand the original terms and use half-quadratic splitting [1], using coordinate descent with region fusion [2, 3], or via the Alternating Direction Method of Multipliers algorithm (ADMM) [4, 5], which is currently the state of the art solution.

ℓ_0-grad applications include edge enhancement and extraction, layer decomposition based manipulation for detail magnification, non-photo realistic rendering like image abstraction and pencil sketching, and clip-art restoration [1]. In medical imaging it can be used in X-ray computed tomography reconstruction [6].

However, ℓ_0-grad reconstructs images poorly in cases where the noise level is large, giving images with plain regions and abrupt changes between them, that look very distorted compared to the original ones. The problem happens because, given the nature of the ℓ_0 norm, ℓ_0-grad removes almost all the low-amplitude structures of the original images, clearly prioritizing keeping the main edges but risking losing important details. An example of this can be seen in Fig. 1.1 where a Fibonacci search was performed to find the λ in (1.1) that obtained the best value for signal to noise ratio (SNR). Figure 1.1 also shows that, despite the described problem, ℓ_0-grad can obtain fairly good values for SNR and structural similarity index measure (SSIM).

Due to the drawbacks summarized above, we propose using a quadratic envelope approximation of the ℓ_0 norm of the gradient as a regularizer term, instead of the actual ℓ_0 norm. By doing this we will be able to control the level of detail that we can afford to lose during denoising or deblurring, thus developing a new edge-preserving filtering model. Furthermore, unlike ℓ_0-grad, this filtering model can be solved using the accelerated proximal gradient (APG) method [7], which solves non-smooth convex optimization problems presented as the sum of two convex functions.

(a) Original greyscale image

(b) Noisy greyscale image with $\sigma = 0.5$: SNR=-5.69, SSIM=0.12

(c) Reconstructed greyscale image using ℓ_0-grad: SNR=7.38, SSIM=0.58

Figure 1.1: Reconstruction of a noisy greyscale image with Gaussian additive noise of $\sigma = 0.5$ using ℓ_0-grad.

1.2 Main Objective

To develop a novel edge-preserving filtering model based on the quadratic envelope approximation of the ℓ_0 gradient regularization.

1.3 Specific Objectives

- To compute the quadratic envelope of the ℓ_0 norm of the gradient.
- To develop an APG-based algorithm for the quadratic envelope approximation of the ℓ_0 gradient regularization.
- To develop and to implement a program in Python for an edge-preserving filtering model based on the quadratic envelope approximation of the ℓ_0 gradient regularization using the APG-based algorithm.
- To determine the performance of the novel edge-preserving filtering model, comparing it with ℓ_0-grad and Total Variation (ℓ_2-TV) for denoising and deblurring problems.
- To compare the APG-based algorithm performance with an ADMM-based algorithm.

Chapter 2

Background

In this chapter, we present the definitions of entrywise matrix norms and gradient of a digital image and the general forms of ADMM and APG. Furthermore, the ℓ_0 gradient minimization and Total Variation models are explained, as well as some of the methods that have been used to solve them. Finally, we describe how to obtain the quadratic envelope of a function and how to perform a Fibonacci search to maximize the value of a function.

2.1 Entrywise matrix norm

It is defined in [8] as a type of norm that treats an $m \times n$ matrix as a column vector of size mn and use one of the familiar vector norms. Let $\mathbf{a} = [a_{11}, \cdots, a_{m1}, a_{12}, \cdots, a_{m2}, \cdots, a_{1n}, \cdots, a_{mn}]^T = \mathrm{vec}(\mathbf{A})$ be an $mn \times 1$ elongated vector of the $m \times n$ matrix $\mathbf{A}$. If we use the l_p-norm definition of the elongated vector $\mathbf{a}$ then we obtain the l_p-norm of the matrix $\mathbf{A}$ as follows:

$$\|\mathbf{A}\|_p \stackrel{\text{def}}{=} \|\mathbf{a}\|_p = \|\,\mathrm{vec}(\mathbf{A})\|_p = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^p \right)^{1/p}. \tag{2.1}$$

The Frobenius norm is a special case when $p = 2$ as defined in [9]:

$$\|\mathbf{A}\|_F = \|\mathbf{A}\|_2 = \sqrt{\mathrm{Tr}(\mathbf{A}^T\mathbf{A})} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}, \quad \mathbf{A} \in \mathbb{R}^{m \times n}. \tag{2.2}$$

2.2 Gradient of a digital image

The gradient of a digital image f at location (x, y) is defined in [10] as the vector :

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x,y) \\ \frac{\partial f}{\partial y}(x,y) \end{bmatrix}, \tag{2.3}$$

in this research we used the discrete approximation:

$$\nabla f(m,n) = \begin{bmatrix} f(m+1,n) - f(m,n) \\ f(m,n+1) - f(m,n) \end{bmatrix}. \tag{2.4}$$

2.3 ℓ_0 gradient minimization

For [11] the ℓ_0 norm is a very simple and intuitive measure of sparsity of a vector $\mathbf{v}$, counting the number of nonzero entries in $\mathbf{v}$. The term ℓ_0 norm is misleading, as this function does not satisfy all the axiomatic requirements of a norm. They denote the ℓ_0 norm as:

$$\|\mathbf{v}\|_0 = \lim_{p\to 0}\|\mathbf{v}\|_p^p = \lim_{p\to 0}\sum_{k=1}^{m}|v_k|^p = \#\{i : v_i \neq 0\}. \tag{2.5}$$

The ℓ_0 gradient minimization was introduced in [1] as an image editing method effective for sharpening major edges while eliminating a manageable degree of low-amplitude structures. This method is formulated as follows:

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 + \lambda\|\nabla\mathbf{u}\|_0, \tag{2.6}$$

where A represents a blurring operator, $\mathbf{b}$ is the observed noisy data, $\mathbf{u}$ is the expected data and $\lambda > 0$ is a weight controlling the significance of the ℓ_0 norm of the gradient which is defined as:

$$\|\nabla\mathbf{u}\|_0 = \|T\left(D_x\mathbf{u}\right) + T\left(D_y\mathbf{u}\right)\|_0, \tag{2.7}$$

where D_x and D_y represent the discrete operator that approximates the derivatives in the x and y directions respectively, and $T(M)$ is the matrix of the absolute values of each element of M.

2.4 ADMM

The Alternating Direction Method of Multipliers algorithm, known as ADMM was introduced in [12], the following explanation can be found in [13]. ADMM can solve convex optimization problems of the form:

$$\min_{\mathbf{x},\mathbf{z}}(f(\mathbf{x}) + g(\mathbf{z})) \text{ subject to } A\mathbf{x} + B\mathbf{z} = \mathbf{c}, \tag{2.8}$$

with $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$, where $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times n}$ and $\mathbf{c} \in \mathbb{R}^p$. The functions f and g are usually assumed to be convex and $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^m \to \mathbb{R}$. Thus, the augmented Lagrangian will be

$$L_\rho(\mathbf{x},\mathbf{y},\mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(A\mathbf{x} + B\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|A\mathbf{x} + B\mathbf{z} - \mathbf{c}\|_2^2 \tag{2.9}$$

where $\mathbf{y} \in \mathbb{R}^p$ is the dual variable or Lagrange multiplier.

ADMM consists of the following iterations:

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} L_\rho(\mathbf{x},\mathbf{z}_k,\mathbf{y}_k) \tag{2.10}$$

$$\mathbf{z}_{k+1} = \operatorname*{argmin}_{\mathbf{z}} L_\rho(\mathbf{x}_{k+1},\mathbf{z},\mathbf{y}_k) \tag{2.11}$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \rho(A\mathbf{x}_{k+1} + B\mathbf{z}_{k+1} - \mathbf{c}), \tag{2.12}$$

where $\rho > 0$ is called the penalty parameter. The algorithm consists of two minimization steps of $\mathbf{x}$ and $\mathbf{z}$ respectively, and a dual variable $(\mathbf{y})$ update.

ADMM can be written in a different form by combining the linear and quadratic terms in the augmented Lagrangian and scaling the dual variable. Defining the residual $\mathbf{r} = A\mathbf{x} + B\mathbf{z} - \mathbf{c}$, we have

$$\mathbf{y}^T\mathbf{r} + \frac{\rho}{2}\|\mathbf{r}\|_2^2 = \frac{\rho}{2}\left\|\mathbf{r} + \frac{1}{\rho}\mathbf{y}\right\|_2^2 - \frac{1}{2\rho}\|\mathbf{y}\|_2^2, \tag{2.13}$$

if $\mathbf{u} = \frac{1}{\rho}\mathbf{y}$, then:

$$\mathbf{y}^T\mathbf{r} + \frac{\rho}{2}\|\mathbf{r}\|_2^2 = \frac{\rho}{2}\|\mathbf{r} + \mathbf{u}\|_2^2 - \frac{\rho}{2}\|\mathbf{u}\|_2^2. \tag{2.14}$$

Using the scaled dual variable $\mathbf{u}$, ADMM can be expressed as:

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}}\left(f(\mathbf{x}) + \frac{\rho}{2}\|A\mathbf{x} + B\mathbf{z}_k - \mathbf{c} + \mathbf{u}_k\|_2^2\right) \tag{2.15}$$

$$\mathbf{z}_{k+1} = \operatorname*{argmin}_{\mathbf{z}}\left(g(\mathbf{z}) + \frac{\rho}{2}\|A\mathbf{x}_{k+1} + B\mathbf{z} - \mathbf{c} + \mathbf{u}_k\|_2^2\right) \tag{2.16}$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k + A\mathbf{x}_{k+1} + B\mathbf{z}_{k+1} - \mathbf{c}. \tag{2.17}$$

The vector $A\mathbf{x}_{k+1}$ can be replaced with

$$\beta A\mathbf{x}_{k+1} - (1-\beta)(B\mathbf{z}_k - \mathbf{c}), \tag{2.18}$$

where $\beta \in [0,2]$ is a relaxation parameter [13], then (2.16) and (2.17) become:

$$\mathbf{z}_{k+1} = \underset{\mathbf{z}}{\operatorname{argmin}} \left(g(\mathbf{z}) + \frac{\rho}{2} \|\beta A\mathbf{x}_{k+1} - (1-\beta)B\mathbf{z}_k + B\mathbf{z} - \beta\mathbf{c} + \mathbf{u}_k\|_2^2 \right) \tag{2.19}$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta A\mathbf{x}_{k+1} - (1-\beta)B\mathbf{z}_k + B\mathbf{z}_{k+1} - \beta\mathbf{c}. \tag{2.20}$$

The method starts with randomly selected values for $\mathbf{z}_0$ and $\mathbf{u}_0$, which are usually $\mathbf{0}$.

2.5 APG

The accelerated proximal gradient (APG) method, as explained in [7], is used to solve the following optimization problem:

$$\min_{\mathbf{x}} \left(f(\mathbf{x}) + g(\mathbf{x}) \right), \tag{2.21}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are closed proper convex and the gradient of f is L-Lipschitz continuous. Its step is:

$$\mathbf{x}_k = \operatorname{prox}_{\alpha_k, g}(\mathbf{y}_k - \alpha_k \nabla f(\mathbf{y}_k)) \tag{2.22}$$

where $\alpha_k > 0$ is a step size, prox is the proximal function, which is defined as:

$$\operatorname{prox}_{\alpha,g}(\mathbf{z}) = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ g(\mathbf{x}) + \frac{1}{2\alpha} \|\mathbf{x} - \mathbf{z}\|_2^2 \right\}, \tag{2.23}$$

and $\mathbf{y}_k$ is obtained as follows

$$\mathbf{y}_{k+1} = \mathbf{x}_k + \omega_k(\mathbf{x}_k - \mathbf{x}_{k-1}), \tag{2.24}$$

and $\omega_k = \left(\frac{k-1}{k+2}\right)$ is an extrapolation parameter.

This method can converge with rate $O(1/k^2)$ when a fixed step size $\alpha_k \in (0, 1/L]$ is used; with L being the Lipschitz constant of ∇f. The step-size α_k can be either a constant or computed for every iteration, which could be via exact or inexact line search, the Cauchy method, the Barzilai-Borwein method or other alternatives. The APG method initializes with $\mathbf{x_0} = \mathbf{y_1} = \mathbf{0}$.

2.6 TV-based deblurring model

It was stated in [14] that the proper norm for images is the total variation (TV) norm and not the ℓ_2 norm. TV norms are essentially ℓ_1 norms of derivatives. For the discrete TV norm, two popular choices are given in [15], with $\mathbf{u} \in \mathbb{R}^{m\times n}$: the isotropic TV norm defined by

$$\mathbf{TV}_I(\mathbf{u}) = \left\| \left[\sqrt{(D_x u_{ij})^2 + (D_y u_{ij})^2} \right] \right\|_1, \tag{2.25}$$

and the ℓ_1-based, anisotropic TV norm defined by

$$\mathbf{TV}_{\ell_1}(\mathbf{u}) = \|D_x \mathbf{u}\|_1 + \|D_y \mathbf{u}\|_1. \tag{2.26}$$

Usually, for (2.25) or (2.26) reflexive boundary conditions are assumed

$$u_{m+1,j} - u_{m,j} = 0, \quad \forall j \quad \text{and} \quad u_{i,n+1} - u_{i,n} = 0, \quad \forall i. \tag{2.27}$$

The TV-based deblurring model was introduced in [16] as a regularization approach capable of handling properly edges and removing noise in a given image. The discrete penalized version of the TV-based deblurring model consists of solving an unconstrained convex minimization problem of the form:

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2+\lambda\mathbf{TV}(\mathbf{u}), \tag{2.28}$$

where, similar to section 2.3, A represents a blurring operator, $\mathbf{b}$ is the observed noisy data, $\mathbf{u}$ is the expected data and $\lambda > 0$ is a parameter that controls the significance of the TV-norm of $\mathbf{u}$, using its isotropic (direction independent) or anisotropic (direction dependent) formula.

2.7 Fast-gradient based algorithm for constrained Total Variation (TV) denoising

This algorithm was originally described in [17] based on the unconstrained solution given in [15]. The original problem for constrained Total Variation (TV) denoising can be written as:

$$\min_{\mathbf{u}\in\mathcal{C}} \frac{1}{2}\|\mathbf{u}-\mathbf{b}\|_2^2+\lambda\mathbf{TV}(\mathbf{u}) \tag{2.29}$$

with $\mathcal{C}$ being a closed convex subset of $\mathbb{R}^{m\times n}$, and $\mathbf{TV}(\mathbf{u})$ being either the isotropic TV norm (2.25) or the anisotropic TV norm (2.26). Using the facts that:

$$\sqrt{x^2+y^2}=\max_{p,q}\ (p\cdot x+q\cdot y) \quad \text{s.t.} \quad p^2+q^2\leq 1, \tag{2.30}$$

$$|x|+|y|=\max_{p,q}\ (p\cdot x+q\cdot y) \quad \text{s.t.} \quad |p|\leq 1 \wedge |q|\leq 1, \tag{2.31}$$

then (2.29) can be recast as:

$$\min_{(\mathbf{p},\mathbf{q})\in\mathcal{P}} \frac{1}{2}\|\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q})\|_2^2-\frac{1}{2}\|P_{\mathcal{C}}(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q}))-(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q}))\|_2^2 \tag{2.32}$$

where $\mathcal{L}(\mathbf{p},\mathbf{q})=D_x^T\mathbf{p}+D_y^T\mathbf{q}$, $P_{\mathcal{C}}$ represents the orthogonal projection on the set $\mathcal{C}$, and $\mathcal{P}$ is the set defined by (2.30) or (2.31) depending on the type of TV norm in (2.29).

In order to solve (2.32), APG is applied with its step being:

$$(\mathbf{p}_k,\mathbf{q}_k)=P_{\mathcal{P}}\left[(\mathbf{r}_k,\mathbf{s}_k)+\frac{1}{8\lambda}\mathcal{L}^T\left(P_{\mathcal{C}}\left(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p}_k,\mathbf{q}_k)\right)\right)\right], \tag{2.33}$$

and the orthogonal projection on $\mathcal{P}$ for $\mathbf{TV}_I(\mathbf{u})$ is:

$$P_{\mathcal{P}}(\mathbf{p},\mathbf{q})=(\mathbf{r},\mathbf{s})=\begin{cases} r_{i,j}=\frac{p_{i,j}}{\max\{1,\sqrt{p_{i,j}^2+q_{i,j}^2}\}} \\ s_{i,j}=\frac{q_{i,j}}{\max\{1,\sqrt{p_{i,j}^2+q_{i,j}^2}\}}\end{cases}, \tag{2.34}$$

or for $\mathbf{TV}_{\ell_1}(\mathbf{u})$ is:

$$P_{\mathcal{P}}(\mathbf{p},\mathbf{q})=(\mathbf{r},\mathbf{s})=\begin{cases} r_{i,j}=\frac{p_{i,j}}{\max\{1,|p_{i,j}|\}} \\ s_{i,j}=\frac{q_{i,j}}{\max\{1,|q_{i,j}|\}}\end{cases}, \tag{2.35}$$

with this, the optimal $(\mathbf{p},\mathbf{q})$ for (2.32) can be found, and the optimal solution of (2.29) is $\mathbf{u}=P_{\mathcal{C}}\left(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q})\right)$.

2.8 ADMM-based solution for the ℓ_0 gradient minimization problem

To solve the ℓ_0 gradient minimization problem via ADMM, we rewrite (2.6) as:

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2+\lambda\|\mathbf{v}\|_0 \quad \text{s.t} \quad \mathbf{v}=\begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u} \tag{2.36}$$

with $\mathbf{v}=\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$, $\mathbf{w}=\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}$ and using the scaled form of ADMM given in [13]. The steps of each iteration are:

$$\mathbf{u}_{k+1}=\operatorname*{argmin}_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2+\frac{\rho}{2}\left\|\begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u}-\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k+\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}_k\right\|_2^2 \tag{2.37}$$

$$\mathbf{v}_{k+1}=\operatorname*{argmin}_{\mathbf{v}} \lambda\|\mathbf{v}\|_0+\frac{\rho}{2}\left\|\beta\begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u}_{k+1}+(1-\beta)\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k-\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}+\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}_k\right\|_2^2 \tag{2.38}$$

$$\mathbf{w}_{k+1}=\mathbf{w}_k+\beta\begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u}_{k+1}+(1-\beta)\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k-\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_{k+1}. \tag{2.39}$$

For (2.37) we find the gradient and equal that to zero, obtaining:

$$A^TA\mathbf{u}-A^T\mathbf{b}+\rho\left(D_x^TD_x+D_y^TD_y\right)\mathbf{u}+\rho D_x^T(\mathbf{w}_1-\mathbf{v}_1)_k+\rho D_y^T(\mathbf{w}_2-\mathbf{v}_2)_k=\mathbf{0}. \tag{2.40}$$

The equation in (2.40) can be solved in the spatial or frequency domain; from a computational perspective, the latter is usually preferred. For (2.38) we use the proximal of $\lambda\|.\|_0$:

$$\operatorname{prox}_{\lambda\|\cdot\|_0}\left(\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}\right)=\operatorname*{argmin}_{\mathbf{v}}\left(\lambda\left\|\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}\right\|_0+\frac{1}{2}\left\|\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}-\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}\right\|_2^2\right)=\begin{cases} \begin{bmatrix} z_{1i,j} \\ z_{2i,j} \end{bmatrix} & \left\|\begin{bmatrix} z_{1i,j} \\ z_{2i,j} \end{bmatrix}\right\|_2^2 \geq 2\lambda \\ \mathbf{0} & \left\|\begin{bmatrix} z_{1i,j} \\ z_{2i,j} \end{bmatrix}\right\|_2^2 < 2\lambda \end{cases}. \tag{2.41}$$

2.9 Quadratic envelope

The quadratic envelope (Q_γ) is defined in [18] for any $[0,\infty]$-valued lower semi-continuous functional f with $\gamma > 0$ by:

$$Q_\gamma(f)(\mathbf{x})=\sup_{\mathbf{y}}\left[\inf_{\mathbf{w}}\left(f(\mathbf{w})+\frac{\gamma}{2}\|\mathbf{w}-\mathbf{y}\|_2^2\right)-\frac{\gamma}{2}\|\mathbf{x}-\mathbf{y}\|_2^2\right], \tag{2.42}$$

where γ is the parameter that controls the maximum negative curvature of $Q_\gamma(f)$.

The indicator function ($|x|_0$), defined by:

$$|x|_0=\begin{cases} 1 & \text{if } x\neq 0 \\ 0 & \text{if } x=0 \end{cases} \qquad \forall x\in\mathbb{R}, \tag{2.43}$$

has a quadratic envelope which was first determined in [19], and it can easily be obtained using (2.42). Since $x\in\mathbb{R}$, we can write:

$$Q_\gamma(\lambda|x|_0)=\sup_{y}\left[\inf_{w}\left(\lambda|w|_0+\frac{\gamma}{2}(w-y)^2\right)-\frac{\gamma}{2}(x-y)^2\right],$$

to calculate $Q_\gamma(\lambda|x|_0)$, first we find the infimum with respect to w, then the supremum with respect to y, and using

$$I_{[\text{condition}]}=\begin{cases} 1 & \text{if condition is true} \\ 0 & \text{if condition is false} \end{cases}$$

we have:

1. $\inf_w \left(\lambda|w|_0 + \frac{\gamma}{2}(w-y)^2\right)$

$$w^* = \begin{cases} 0 & |y| \leq \sqrt{\frac{2\lambda}{\gamma}} \\ y & |y| > \sqrt{\frac{2\lambda}{\gamma}} \end{cases} \rightarrow \inf_w = \begin{cases} \frac{\gamma}{2}y^2 & |y| \leq \sqrt{\frac{2\lambda}{\gamma}} \\ \lambda & |y| > \sqrt{\frac{2\lambda}{\gamma}} \end{cases}$$

$$\inf_w = I_{\left[|y|>\sqrt{\frac{2\lambda}{\gamma}}\right]}\left(\lambda - \frac{\gamma}{2}y^2\right) + \frac{\gamma}{2}y^2$$

2. $\sup_y \left(I_{\left[|y|>\sqrt{\frac{2\lambda}{\gamma}}\right]}\left(\lambda - \frac{\gamma}{2}y^2\right) + \frac{\gamma}{2}y^2 - \frac{\gamma}{2}(x-y)^2\right)$

$$\sup_y \left(I_{\left[|y|>\sqrt{\frac{2\lambda}{\gamma}}\right]}\left(\lambda - \frac{\gamma}{2}y^2\right) - \frac{\gamma}{2}x^2 + \gamma xy\right)$$

here we have two options:

- $|y| > \sqrt{\frac{2\lambda}{\gamma}}$

$$\sup_y \left(\lambda - \frac{\gamma}{2}(x-y)^2\right)$$

$$y^* = x \rightarrow \sup_y = \lambda$$

- $|y| \leq \sqrt{\frac{2\lambda}{\gamma}}$

$$\sup_y \left(-\frac{\gamma}{2}x^2 + \gamma xy\right)$$

$$y^* = \text{sign}(x)\sqrt{\frac{2\lambda}{\gamma}} \rightarrow \sup_y = \sqrt{2\lambda\gamma}|x| - \frac{\gamma}{2}x^2$$

with this, the quadratic envelope of $\lambda|x|_0$ is defined by:

$$Q_\gamma(\lambda|x|_0) = \begin{cases} \sqrt{2\lambda\gamma}|x| - \frac{\gamma}{2}x^2 & |x| \leq \sqrt{\frac{2\lambda}{\gamma}} \\ \lambda & |x| > \sqrt{\frac{2\lambda}{\gamma}} \end{cases}. \tag{2.44}$$

Finally, using (2.44), if $f(\mathbf{x}) = \lambda\|\mathbf{x}\|_0$ and $\mathbf{x} \in \mathbb{R}^N$ then due to separability:

$$Q_\gamma(\lambda\|\mathbf{x}\|_0) = Q_\gamma(\lambda|x_1|_0) + Q_\gamma(\lambda|x_2|_0) + \cdots + Q_\gamma(\lambda|x_N|_0) \tag{2.45}$$

2.10 Fibonacci Search

Fibonacci search, which was introduced in [20], is a technique that can be used to find an approximation of the maximum of an unimodal function. Denoting the i-th number in the Fibonacci sequence as F_i, where $F_0 = 1$, we have the following algorithm for a function $f(x)$:

Algorithm 1 Fibonacci search

Require: a, b, n
  $\triangleright x \in [a,b]$
  $\triangleright n-2$ is the number of function evaluations.
  $L = b - a$
  $k = 2$
  repeat
    $L_k^* = \left(\frac{F_{n-k+1}}{F_{n+1}}\right) L$
    $y = a + L_k^*$
    $z = b - L_k^*$
    if $f(y) < f(z)$ **then**
      $a = y$
      $x = z$
    if $f(y) > f(z)$ **then**
      $b = z$
      $x = y$
    if $f(y) = f(z)$ **then**
      $a = y$
      $b = z$
      $x = z$
    $k = k + 1$
  until $k = n$
  return x

Chapter 3

Methodology

Here we compute the quadratic envelope of $\lambda\|\nabla\mathbf{u}\|_0$ and propose a quadratic envelope approximation of the ℓ_0 gradient regularization. We extend on what has been published in [21] by giving an exact solution to $Q_\gamma(\lambda\|\nabla\mathbf{u}\|_0)$ and adding an ADMM-based solution to the obtained minimization problem.

3.1 Quadratic envelope of $\lambda\|\nabla\mathbf{u}\|_0$

Let $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$, $\|\mathbf{z}\|_0 = \||z_1|+|z_2|\|_0$ with $z_1, z_2 \in \mathbb{R}$ (see (2.7) and the definition of the indicator function given in (2.43)), then we seek to find:

$$Q_\gamma(\lambda\|\mathbf{z}\|_0) = \sup_{\mathbf{y}}\left(\inf_{\mathbf{w}}\left(\lambda\|\mathbf{w}\|_0 + \frac{\gamma}{2}\|\mathbf{w}-\mathbf{y}\|_2^2\right) - \frac{\gamma}{2}\|\mathbf{z}-\mathbf{y}\|_2^2\right).$$

Similar to what we did for $Q_\gamma(\lambda|x|_0)$, we first find the infimum with respect to $\mathbf{w}$ and then the supremum with respect to $\mathbf{y}$:

1. $\inf_{\mathbf{w}}\left(\lambda\|\mathbf{w}\|_0 + \frac{\gamma}{2}\|\mathbf{w}-\mathbf{y}\|_2^2\right)$

$$\mathbf{w}^* = \begin{cases} \mathbf{0} & \|\mathbf{y}\|_2^2 \leq \frac{2\lambda}{\gamma} \\ \mathbf{y} & \|\mathbf{y}\|_2^2 > \frac{2\lambda}{\gamma} \end{cases} \rightarrow \inf_{\mathbf{w}}\left(\lambda\|\mathbf{w}\|_0 + \frac{\gamma}{2}\|\mathbf{w}-\mathbf{y}\|_2^2\right) = \begin{cases} \frac{\gamma}{2}\|\mathbf{y}\|_2^2 & \|\mathbf{y}\|_2^2 \leq \frac{2\lambda}{\gamma} \\ \lambda & \|\mathbf{y}\|_2^2 > \frac{2\lambda}{\gamma} \end{cases}$$

$$\inf_{\mathbf{w}}\left(\lambda\|\mathbf{w}\|_0 + \frac{\gamma}{2}\|\mathbf{w}-\mathbf{y}\|_2^2\right) = I_{\left[\|\mathbf{y}\|_2^2 > \frac{2\lambda}{\gamma}\right]}\left(\lambda - \frac{\gamma}{2}\|\mathbf{y}\|_2^2\right) + \frac{\gamma}{2}\|\mathbf{y}\|_2^2$$

2. $\sup_{\mathbf{y}}\left(I_{\left[\|\mathbf{y}\|_2^2 > \frac{2\lambda}{\gamma}\right]}\left(\lambda - \frac{\gamma}{2}\|\mathbf{y}\|_2^2\right) + \frac{\gamma}{2}\|\mathbf{y}\|_2^2 - \frac{\gamma}{2}\|\mathbf{z}-\mathbf{y}\|_2^2\right)$

$$\sup_{\mathbf{y}}\left(I_{\left[\|\mathbf{y}\|_2^2 > \frac{2\lambda}{\gamma}\right]}\left(\lambda - \frac{\gamma}{2}\|\mathbf{y}\|_2^2\right) - \frac{\gamma}{2}\|\mathbf{z}\|_2^2 + \gamma\langle\mathbf{z},\mathbf{y}\rangle\right),$$

   here we have two options:

   - $\|\mathbf{y}\|_2^2 > \frac{2\lambda}{\gamma}$

$$\sup_{\mathbf{y}}\left(\lambda - \frac{\gamma}{2}\|\mathbf{z}-\mathbf{y}\|_2^2\right)$$

$$\mathbf{y}^* = \mathbf{z} \rightarrow \sup_{\mathbf{y}}\left(\lambda - \frac{\gamma}{2}\|\mathbf{z}-\mathbf{y}\|_2^2\right) = \lambda.$$

- $\|\mathbf{y}\|_2^2 \leq \frac{2\lambda}{\gamma}$

$$\sup_{\mathbf{y}} \left(-\frac{\gamma}{2}\|\mathbf{z}\|_2^2 + \gamma\langle \mathbf{z}, \mathbf{y}\rangle\right),$$

for this supremum we can find an approximate solution [21] and an exact solution using Karush–Kuhn–Tucker (KKT) conditions.

Case I: An approximate solution
Clearly the solution is of the form $y_n = \operatorname{sign}(z_n) \cdot c_n$; if we consider that $|z_1| \approx |z_2|$, then $c_1 = c_2 = \sqrt{\frac{\lambda}{\gamma}}$ and thus, the supremum is given by:

$$\sup_{\mathbf{y}} \left(-\frac{\gamma}{2}\|\mathbf{z}\|_2^2 + \gamma\langle \mathbf{z}, \mathbf{y}\rangle\right) = \sqrt{\lambda\gamma}\|\mathbf{z}\|_1 - \frac{\gamma}{2}\|\mathbf{z}\|_2^2,$$

with this we obtain:

$$Q_\gamma(\lambda\|\mathbf{z}\|_0) = \begin{cases} \sqrt{\lambda\gamma}\|\mathbf{z}\|_1 - \frac{\gamma}{2}\|\mathbf{z}\|_2^2 & \|\mathbf{z}\|_2^2 \leq \frac{2\lambda}{\gamma} \\ \lambda & \|\mathbf{z}\|_2^2 > \frac{2\lambda}{\gamma} \end{cases}.$$

If we consider $\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} D_x u \\ D_y u \end{bmatrix} = \nabla u$, then:

$$Q_\gamma(\lambda\|\nabla u\|_0) = \begin{cases} \sqrt{\lambda\gamma}\|\nabla u\|_1 - \frac{\gamma}{2}\|\nabla u\|_2^2 & \|\nabla u\|_2^2 \leq \frac{2\lambda}{\gamma} \\ \lambda & \|\nabla u\|_2^2 > \frac{2\lambda}{\gamma} \end{cases}. \tag{3.1}$$

Case II: An exact solution
The supremum is with respect to $\mathbf{y}$, so we can evaluate only for $\langle \mathbf{z}, \mathbf{y}\rangle$. We know that:

$$\sup_{\mathbf{y}} (\langle \mathbf{z}, \mathbf{y}\rangle) = -\inf_{\mathbf{y}} (-\langle \mathbf{z}, \mathbf{y}\rangle) \tag{3.2}$$

We will evaluate for the infimum:

$$\mathbf{y}^* = \operatorname*{argmin}_{\mathbf{y}} (-\langle \mathbf{z}, \mathbf{y}\rangle)$$

Using the Lagrangian with the condition $\frac{2\lambda}{\gamma} - y_1^2 - y_2^2 \geq 0$, we obtain:

$$L(\mathbf{y}, \beta) = -\langle \mathbf{z}, \mathbf{y}\rangle - \beta\left(\frac{2\lambda}{\gamma} - y_1^2 - y_2^2\right) \tag{3.3}$$

Considering the KKT conditions:

$$\nabla_{\mathbf{y}} L(\mathbf{y}^*, \beta^*) = \mathbf{0} \tag{3.4}$$

$$\beta^*\left(\frac{2\lambda}{\gamma} - (y_1^*)^2 - (y_2^*)^2\right) = 0 \tag{3.5}$$

Using (3.4),

$$\begin{bmatrix} -z_1 \\ -z_2 \end{bmatrix} + \beta^* \begin{bmatrix} 2y_1^* \\ 2y_2^* \end{bmatrix} = \mathbf{0} \tag{3.6}$$

from this we find β^*

$$\beta^* = \frac{z_1}{2y_1^*} = \frac{z_2}{2y_2^*}$$

then we can relate y_1^* and y_2^* by:

$$\frac{z_1}{2y_1^*} = \frac{z_2}{2y_2^*}$$

$$\frac{z_1}{y_1^*} = \frac{z_2}{y_2^*}$$

$$\frac{z_1}{z_2}y_2^* = y_1^* \tag{3.7}$$

Evaluating (3.5) and using the relation in (3.7)

$$\beta^*\left(\frac{2\lambda}{\gamma} - \left(\frac{z_1}{z_2}y_2^*\right)^2 - (y_2^*)^2\right) = 0$$

$$(y_2^*)^2 = \frac{2\lambda}{\gamma}\frac{1}{1+\left(\frac{z_1}{z_2}\right)^2} = \frac{2\lambda}{\gamma}\frac{z_2^2}{\|\mathbf{z}\|_2^2}$$

using (3.6), and the fact that $\beta \geq 0$, we know that y_2^* and z_2 have the same sign, then:

$$y_2^* = \sqrt{\frac{2\lambda}{\gamma}}\frac{z_2}{\|\mathbf{z}\|_2} \tag{3.8}$$

and similarly for y_1^*

$$y_1^* = \sqrt{\frac{2\lambda}{\gamma}}\frac{z_1}{\|\mathbf{z}\|_2} \tag{3.9}$$

Replacing (3.8) and (3.9) in (3.2):

$$\sup_{\mathbf{y}}(\langle \mathbf{z},\mathbf{y}\rangle) = \sqrt{\frac{2\lambda}{\gamma}}\frac{z_1^2+z_2^2}{\|\mathbf{z}\|_2}$$

$$\sup_{\mathbf{y}}(\langle \mathbf{z},\mathbf{y}\rangle) = \sqrt{\frac{2\lambda}{\gamma}}\|\mathbf{z}\|_2 \tag{3.10}$$

with (3.10), we obtain the supremum of the original function:

$$\sup_{\mathbf{y}}\left(-\frac{\gamma}{2}\|\mathbf{z}\|_2^2 + \gamma\langle \mathbf{z},\mathbf{y}\rangle\right) = \sqrt{2\lambda\gamma}\|\mathbf{z}\|_2 - \frac{\gamma}{2}\|\mathbf{z}\|_2^2$$

then, the quadratic envelope is:

$$Q_\gamma(\lambda\|\mathbf{z}\|_0) = \begin{cases} \sqrt{2\lambda\gamma}\|\mathbf{z}\|_2 - \frac{\gamma}{2}\|\mathbf{z}\|_2^2 & \|\mathbf{z}\|_2^2 \leq \frac{2\lambda}{\gamma} \\ \lambda & \|\mathbf{z}\|_2^2 > \frac{2\lambda}{\gamma} \end{cases}.$$

Similar to the previous case, if $\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} D_x u \\ D_y u \end{bmatrix} = \nabla u$, then:

$$Q_\gamma(\lambda\|\nabla u\|_0) = \begin{cases} \sqrt{2\lambda\gamma}\|\nabla u\|_2 - \frac{\gamma}{2}\|\nabla u\|_2^2 & \|\nabla u\|_2^2 \leq \frac{2\lambda}{\gamma} \\ \lambda & \|\nabla u\|_2^2 > \frac{2\lambda}{\gamma} \end{cases} \tag{3.11}$$

Finally, due to separability, with $\mathbf{u} \in \mathbb{R}^{m\times n}$, $mn = N$, and using either (3.1) or (3.11), the quadratic envelope of the ℓ_0 norm of the gradient can be obtained by:

$$Q_\gamma(\lambda\|\nabla\mathbf{u}\|_0) = Q_\gamma(\lambda\|\nabla u_1\|_0) + Q_\gamma(\lambda\|\nabla u_2\|_0) + \cdots + Q_\gamma(\lambda\|\nabla u_N\|_0). \tag{3.12}$$

3.2 Quadratic envelope approximation of the ℓ_0 gradient regularization

We use the quadratic envelope obtained in section 3.1 as the ℓ_0 norm of the gradient in (2.6), with this we can rewrite the original problem as:

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u} - \mathbf{b}\|_2^2 + Q_\gamma(\lambda\|\nabla\mathbf{u}\|_0). \tag{3.13}$$

If $(D_x u_{ij})^2 + (D_y u_{ij})^2 \leq \frac{2\lambda}{\gamma}$ holds true for every component of $\begin{bmatrix} D_x\mathbf{u} \\ D_y\mathbf{u} \end{bmatrix}$, then using the approximate solution, (3.13) becomes the following minimization problem (see (2.26)):

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 + \sqrt{\gamma\lambda}\mathbf{TV}_{\ell_1}(\mathbf{u}) - \frac{\gamma}{2}\|\nabla\mathbf{u}\|_2^2, \tag{3.14}$$

similarly for the exact solution (see (2.25)):

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 + \sqrt{2\gamma\lambda}\mathbf{TV}_{I}(\mathbf{u}) - \frac{\gamma}{2}\|\nabla\mathbf{u}\|_2^2. \tag{3.15}$$

To determine (3.14) and (3.15) we have only considered the case when $(D_x u_{ij})^2 + (D_y u_{ij})^2 \leq \frac{2\lambda}{\gamma}$, but since the quadratic envelope is defined for all values, we would need to include both conditions, representing them as a matrix we would have:

$$M = (m_{ij}) = \begin{cases} 1 & (D_x u_{ij})^2 + (D_y u_{ij})^2 \leq \frac{2\lambda}{\gamma} \\ 0 & (D_x u_{ij})^2 + (D_y u_{ij})^2 > \frac{2\lambda}{\gamma} \end{cases}, \tag{3.16}$$

however due to the nature of γ in (3.16), we can always find a value for which M is exactly an all-ones matrix. Moreover, if an element in M is zero, it is not clear how the resulting minimization problem should be solved because a direct implementation would lead to $u_{i,j} = (A^{-1}\mathbf{b})_{ij}$, but at the same time the condition $((D_x u_{ij})^2 + (D_y u_{ij})^2 > \frac{2\lambda}{\gamma})$ must be satisfied, unfortunately this depends on the values of the adjacent pixels and can not be carried out in a local fashion.

Given the above consideration, we propose (3.14) or (3.15) as a quadratic envelope approximation of the ℓ_0 gradient regularization which can be solved by APG and ADMM.

1. **APG-based solution**

   For (3.14), let $f(\mathbf{u}) = \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 - \frac{\gamma}{2}\|\nabla\mathbf{u}\|_2^2$ and $g(\mathbf{u}) = \sqrt{\gamma\lambda}\mathbf{TV}_{\ell_1}(\mathbf{u})$, then:

   $$\nabla f(\mathbf{u}) = A^T A\mathbf{u} - A^T\mathbf{b} - \gamma\nabla^T(\nabla\mathbf{u}), \tag{3.17}$$

   with this we have the following step of APG:

   $$\mathbf{u}_k = \operatorname*{argmin}_{\mathbf{u}} \frac{1}{2\alpha_k}\|\mathbf{u} - (\mathbf{v}_k - \alpha_k\nabla f(\mathbf{v}_k))\|_2^2 + \sqrt{\gamma\lambda}\mathbf{TV}_{\ell_1}(\mathbf{u}), \tag{3.18}$$

   and for (3.15), $\nabla f(\mathbf{u})$ is the same as (3.17), $g(\mathbf{u}) = \sqrt{2\gamma\lambda}\mathbf{TV}_I(\mathbf{u})$ and the step of APG will be:

   $$\mathbf{u}_k = \operatorname*{argmin}_{\mathbf{u}} \frac{1}{2\alpha_k}\|\mathbf{u} - (\mathbf{v}_k - \alpha_k\nabla f(\mathbf{v}_k))\|_2^2 + \sqrt{2\gamma\lambda}\mathbf{TV}_I(\mathbf{u}), \tag{3.19}$$

   both of them can be solved using the method described in Section 2.7.

2. **ADMM-based solution**

   For the exact solution of $Q_\gamma(\lambda\|\nabla\mathbf{u}\|_0)$, with $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$, $\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}$ and using the scaled form of ADMM, the method iterates the following three steps:

$$\mathbf{u}_{k+1} = \operatorname*{argmin}_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 - \frac{\gamma}{2}\|\nabla\mathbf{u}\|_2^2 + \frac{\rho}{2}\left\|\begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u} - \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k + \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}_k\right\|_2^2 \tag{3.20}$$

$$\mathbf{v}_{k+1} = \operatorname*{argmin}_{\mathbf{v}} \sqrt{2\lambda\gamma}\|\mathbf{v}\|_1 + \frac{\rho}{2}\left\|\beta\begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u}_{k+1} + (1-\beta)\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k - \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}_k\right\|_2^2 \tag{3.21}$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \beta\begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u}_{k+1} + (1-\beta)\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k - \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_{k+1}. \tag{3.22}$$

To solve (3.20), we simply equate its gradient to zero:

$$A^T A\mathbf{u} - A^T\mathbf{b} + (\rho - \gamma)\nabla^T\nabla\mathbf{u} + \rho\nabla^T(\mathbf{w}_k - \mathbf{v}_k) = \mathbf{0}, \tag{3.23}$$

which can be solved in the spatial or frequency domain. For (3.21) we use the proximal function of the ℓ_1 norm associated with the isotropic TV-norm (see (2.25)):

$$\mathrm{prox}_{\lambda\|\cdot\|_1}\left(\begin{bmatrix}\mathbf{z}_1\\\mathbf{z}_2\end{bmatrix}\right) = \underset{\mathbf{v}}{\mathrm{argmin}}\left(\lambda\left\|\begin{bmatrix}\mathbf{v}_1\\\mathbf{v}_2\end{bmatrix}\right\|_1 + \frac{1}{2}\left\|\begin{bmatrix}\mathbf{v}_1\\\mathbf{v}_2\end{bmatrix} - \begin{bmatrix}\mathbf{z}_1\\\mathbf{z}_2\end{bmatrix}\right\|_2^2\right)$$

$$\mathrm{prox}_{\lambda\|\cdot\|_1}\left(\begin{bmatrix}\mathbf{z}_1\\\mathbf{z}_2\end{bmatrix}\right) = \begin{cases}\dfrac{\begin{bmatrix}z_{1i,j}\\z_{2i,j}\end{bmatrix}}{\left\|\begin{bmatrix}z_{1i,j}\\z_{2i,j}\end{bmatrix}\right\|_2}\left(\left\|\begin{bmatrix}z_{1i,j}\\z_{2i,j}\end{bmatrix}\right\|_2 - \lambda\right) & \left\|\begin{bmatrix}z_{1i,j}\\z_{2i,j}\end{bmatrix}\right\|_2^2 \geq \lambda\\[2ex] \mathbf{0} & \left\|\begin{bmatrix}z_{1i,j}\\z_{2i,j}\end{bmatrix}\right\|_2^2 < \lambda\end{cases}. \tag{3.24}$$

Chapter 4

Results

We developed a new edge-preserving filtering model, which we called $Q\ell_0$-grad, using the quadratic envelope approximation of the ℓ_0 gradient regularization, this model is formulated as:

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 + \sqrt{2\gamma\lambda}\mathbf{TV}_I(\mathbf{u}) - \frac{\gamma}{2}\|\nabla\mathbf{u}\|_2^2. \tag{4.1}$$

In order to evaluate the performance of the filtering model for denoising and deblurring of images, we compared it with ℓ_0-grad (4.2) and ℓ_2-TV (4.3):

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 + \lambda\|\nabla\mathbf{u}\|_0, \tag{4.2}$$

$$\min_{\mathbf{u}} \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 + \lambda\mathbf{TV}_I(\mathbf{u}). \tag{4.3}$$

Only for denoising we also compared against BM3D [22], since it is currently one of the state of the art methods for noise reduction in images.

The programs were written in Python on a Jupyter environment, which can be found in [23], and were run in a Lenovo Ideapad 110 Laptop with Intel core i7-6498DU @ 2.5 GHz processor, 8 GB of RAM and Ubuntu 18.04 operating system. We used four 512×512 test images: Lena, Barbara, Mandrill and Peppers, both in greyscale and in color. The results for the denoising and deblurring problems with these images were organized as follows:

- Section 4.1: denoising.
- Section 4.2: deblurring with an average filter.
- Section 4.3: deblurring with a Gaussian filter.

For deblurring we applied a 5×5 average filter and a 9×9 Gaussian filter, and for the three different problems we added Gaussian additive noise with 0 mean and different values of standard deviation $\sigma = \{0.05, 0.1, 0.25, 0.5\}$. The peak signal to noise ratio (PSNR), SNR and SSIM [24] metrics of the obtained noisy images are shown in Tables A.1 and A.2.

The values of λ in (4.1), (4.2) and (4.3) used in every case were found by performing a Fibonacci search (see section 2.10) with SNR as the function to be evaluated. For $Q\ell_0$-grad we chose to keep $\gamma = 0.01$ fixed, with this value we obtained fairly good metrics on some preview tests and it was easier to have just one parameter to be optimized. For APG implementations we used a Cauchy lagged stepsize and for ADMM we chose $\rho = 20$ and $\beta = 1.5$, this last value was chosen because according to [13] it can improve convergence.

In section 4.4, we compared the performance of the APG-based and ADMM-based algorithms for $Q\ell_0$-grad. Finally in section 4.5 we showed how, changing the value of γ while keeping a fixed λ, influences the restoration quality of the reconstructed images for denoising and deblurring with our proposed method.

4.1 Results for denoising

For $Q\ell_0$-grad we used APG with 20 outer and 20 inner iterations; these inner iterations were done because of the Total Variation denoising problem that it is obtained (3.19); on the other hand, for ℓ_0-grad and ℓ_2-TV we used ADMM with 40 iterations. Then, to find the optimal λ that maximized the SNR metric for each method, we performed a Fibonacci search with eight function evaluations and the bounds given in Table A.3. For BM3D we had a direct implementation that only required the value of standard deviation of the additive Gaussian noise as an input. We used the four methods to reconstruct the greyscale and color images that were corrupted with additive Gaussian noise and evaluated their PSNR, SNR and SSIM metrics, and computation time; we did these simulations 10 times and obtained their respective averages and standard deviations (tables A.4 to A.7).

The graphs, in figures 4.1 to 4.8, of the averages of each metric and time with respect to the value of standard deviation of the noise (σ), show that the $Q\ell_0$-grad performance for PSNR, SNR and SSIM was between ℓ_0-grad and ℓ_2-TV, being closer to the latter and better than ℓ_0-grad with an average improvement of +1.56 dB PSNR, +1.56 dB SNR and +0.06 SSIM. In the greyscale simulations (Fig. 4.1 to 4.4) sometimes $Q\ell_0$-grad and ℓ_2-TV overlapped each other as it happened with the metrics for Lena in Fig. 4.1 and in the color simulations (Fig. 4.5 to 4.8) they were a little more separated. We can also see that BM3D had almost always the best performance for every metric, but there were cases like the obtained values of SSIM for greyscale Peppers (Fig. 4.4) where $Q\ell_0$-grad and ℓ_2-TV surpass it for $\sigma = 0.5$. Computation time remained constant regardless of the value of σ, nonetheless the color images took more time to be processed than the greyscale ones. APG for $Q\ell_0$-grad was the slowest for all the simulations, however we clarify that during the simulations for the greyscale cases with $\sigma = 0.1$, there were some background processes in the computer used which influenced on the obtained time.

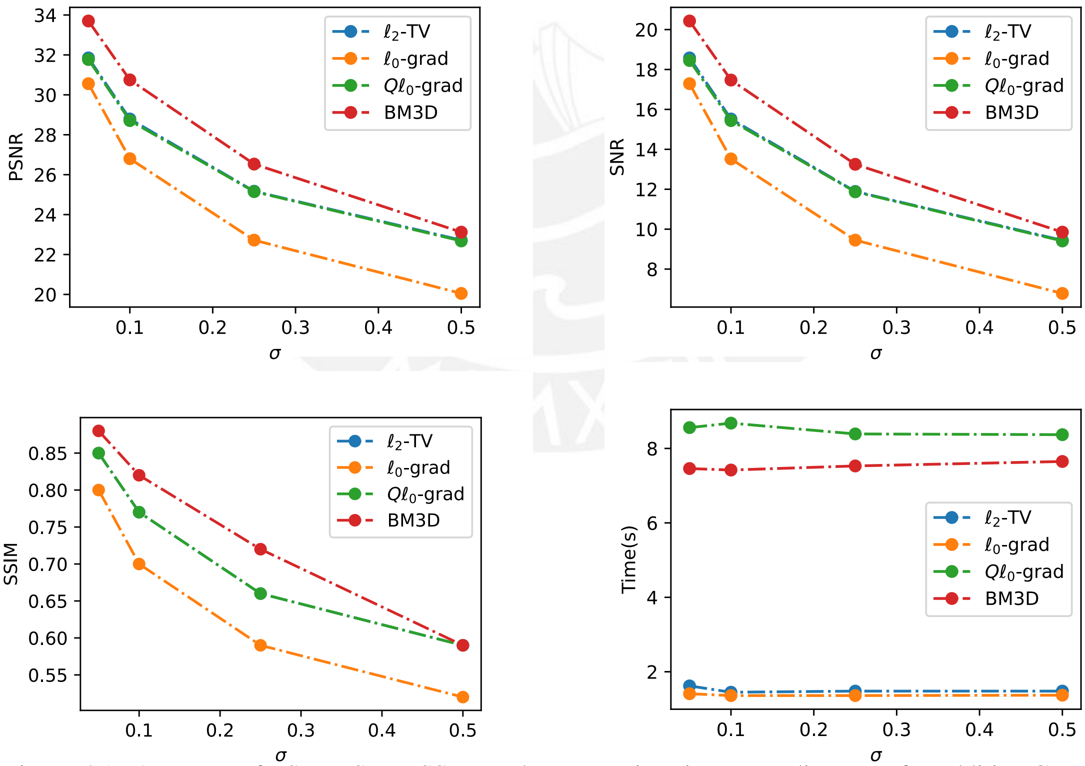


Figure 4.1: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for denoising greyscale Lena using the 4 methods.

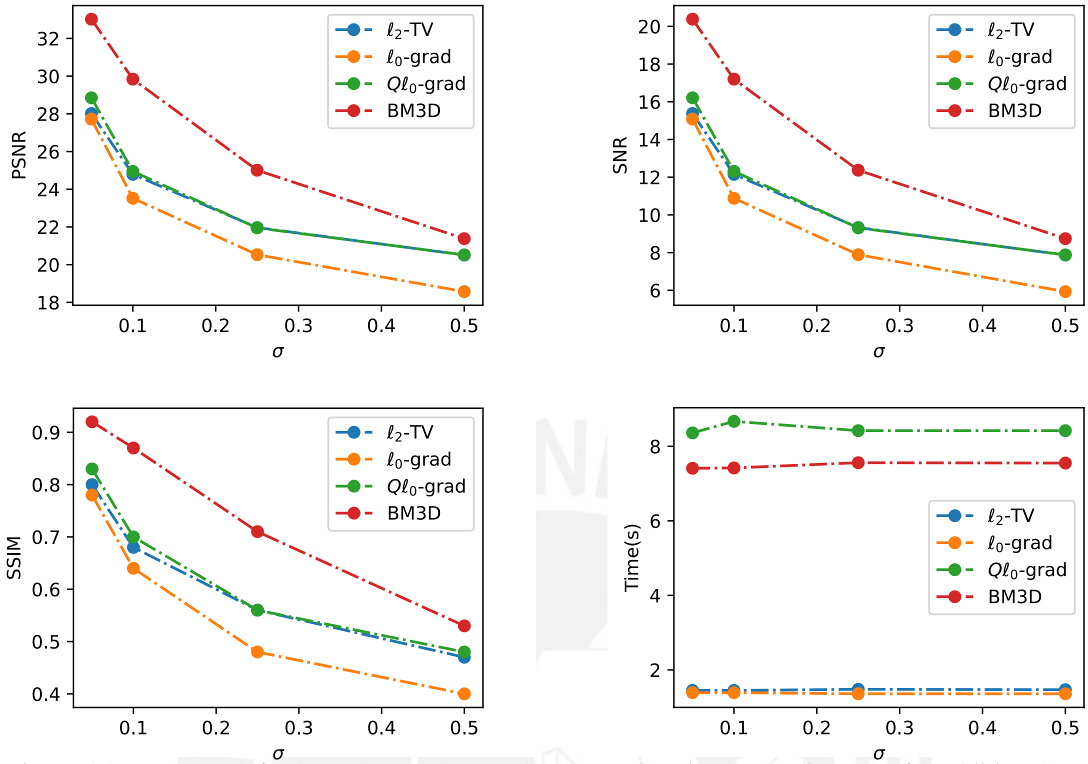


Figure 4.2: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for denoising greyscale Barbara using the 4 methods.

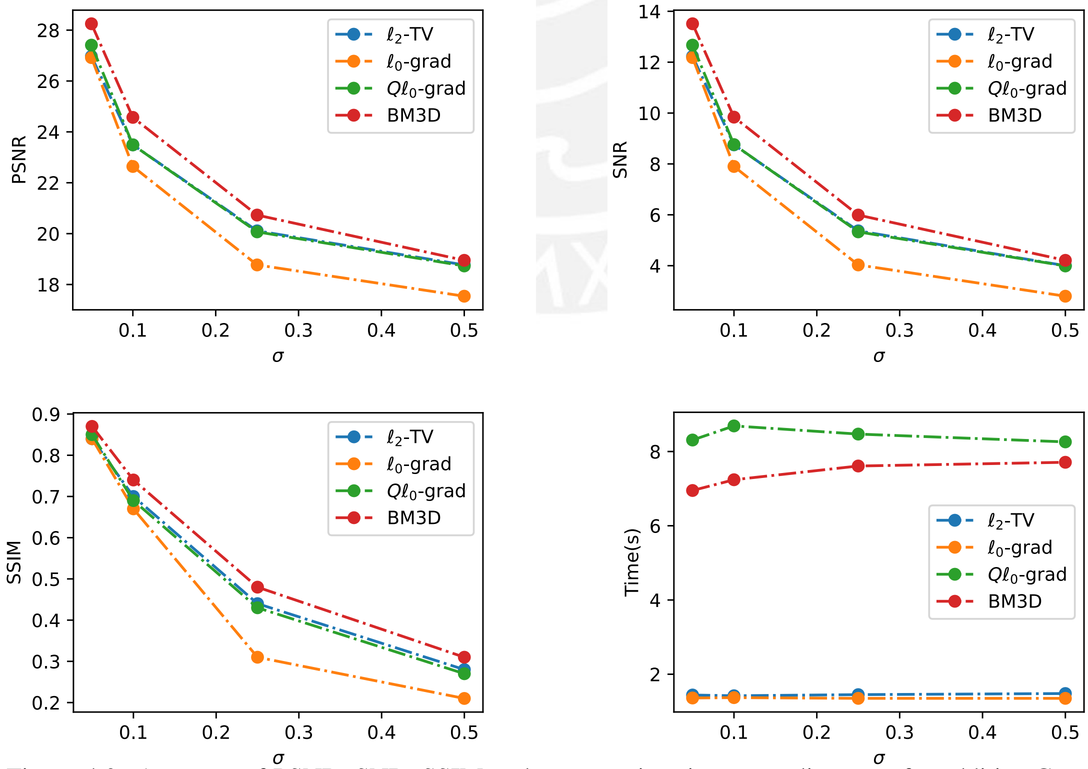


Figure 4.3: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for denoising greyscale Mandrill using the 4 methods.

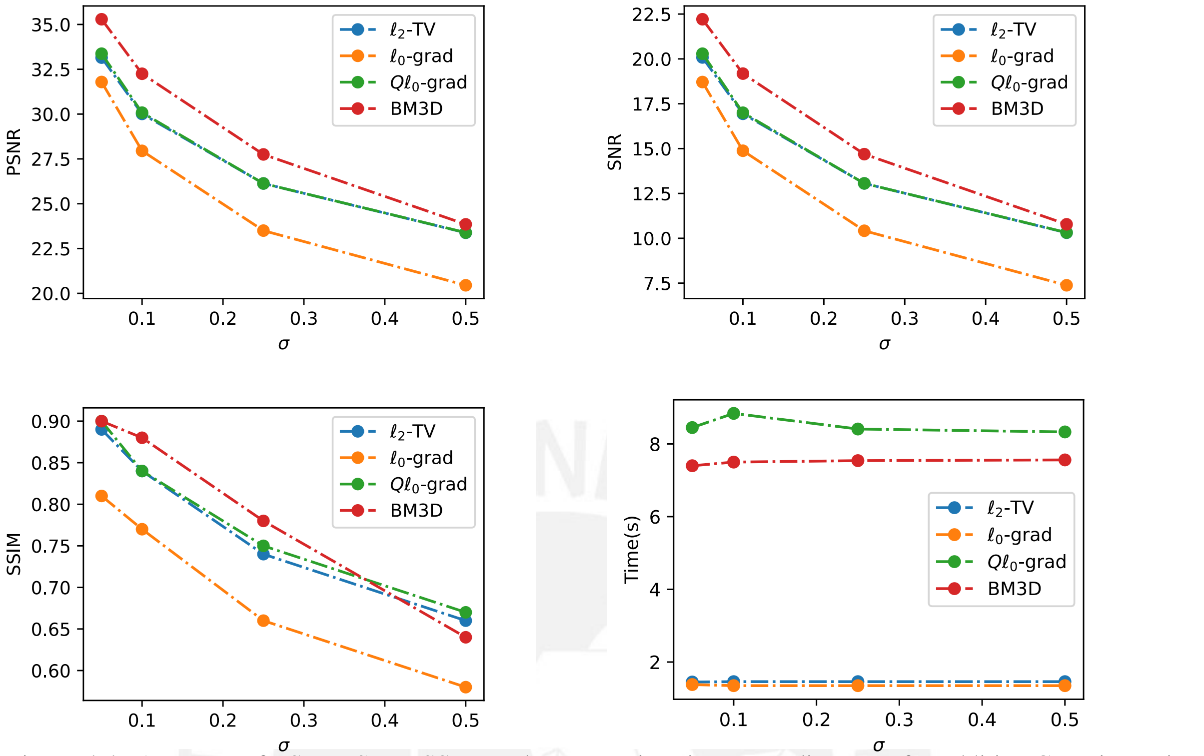


Figure 4.4: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for denoising greyscale Peppers using the 4 methods.

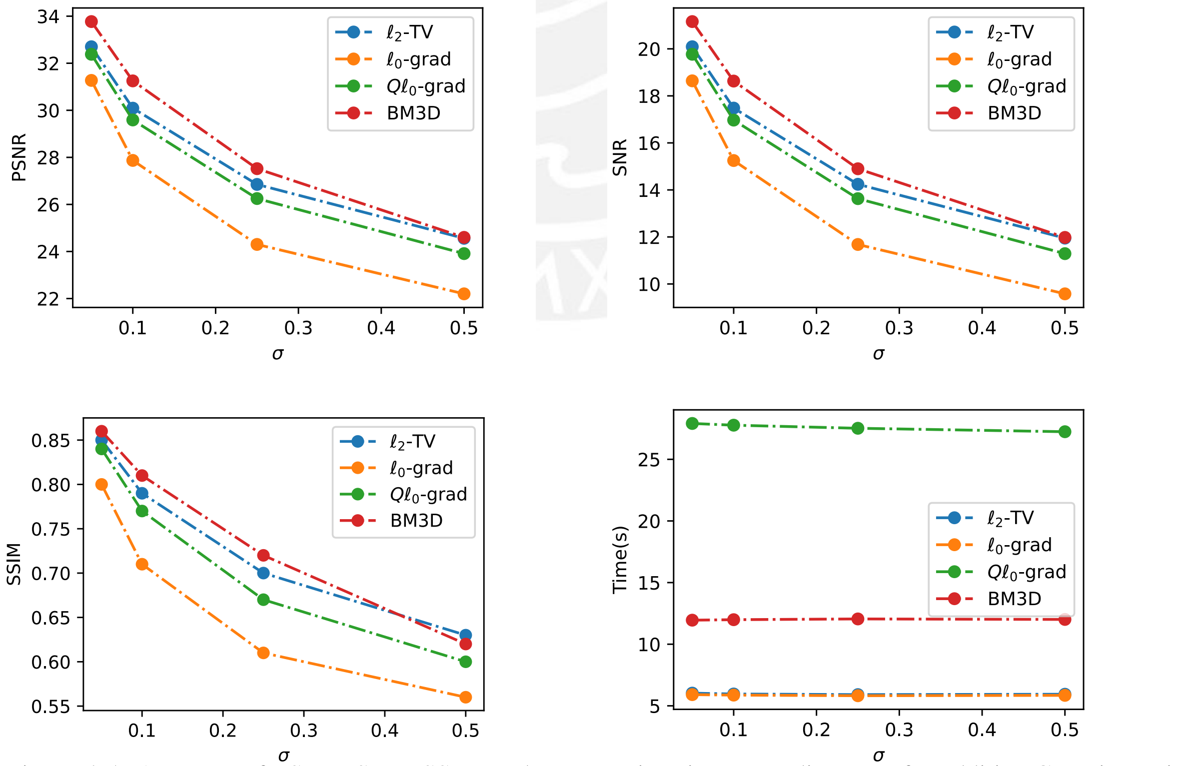


Figure 4.5: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for denoising color Lena using the 4 methods.

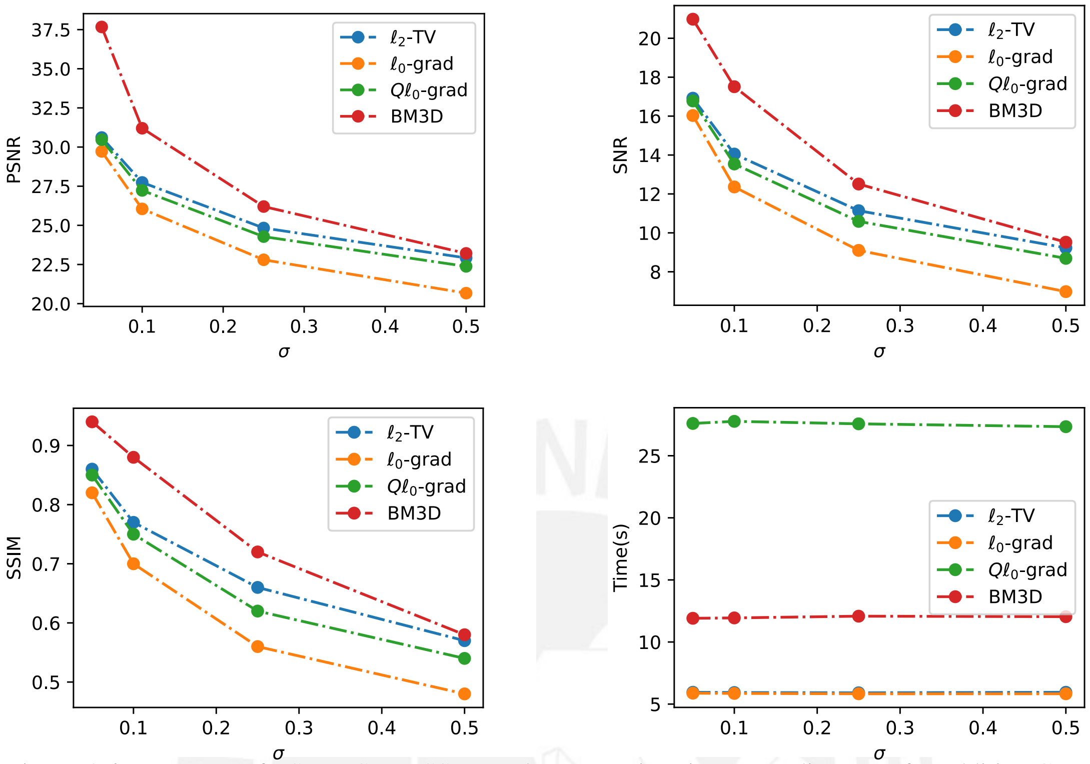


Figure 4.6: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for denoising color Barbara using the 4 methods.

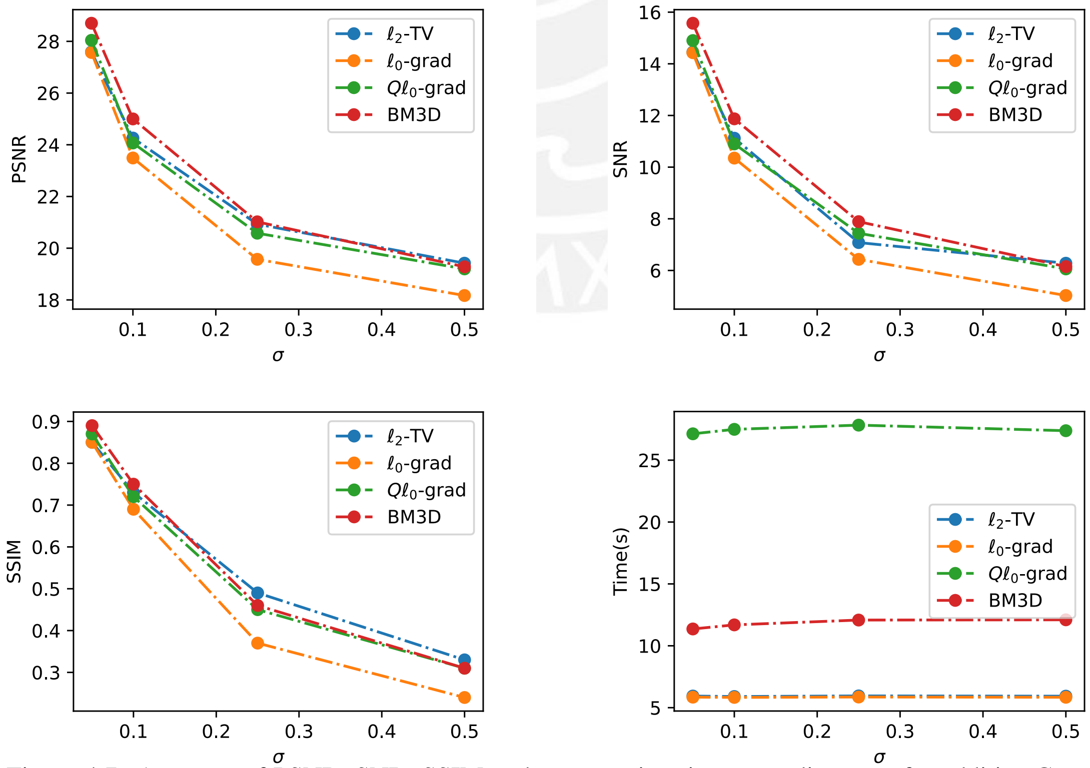


Figure 4.7: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for denoising color Mandrill using the 4 methods.

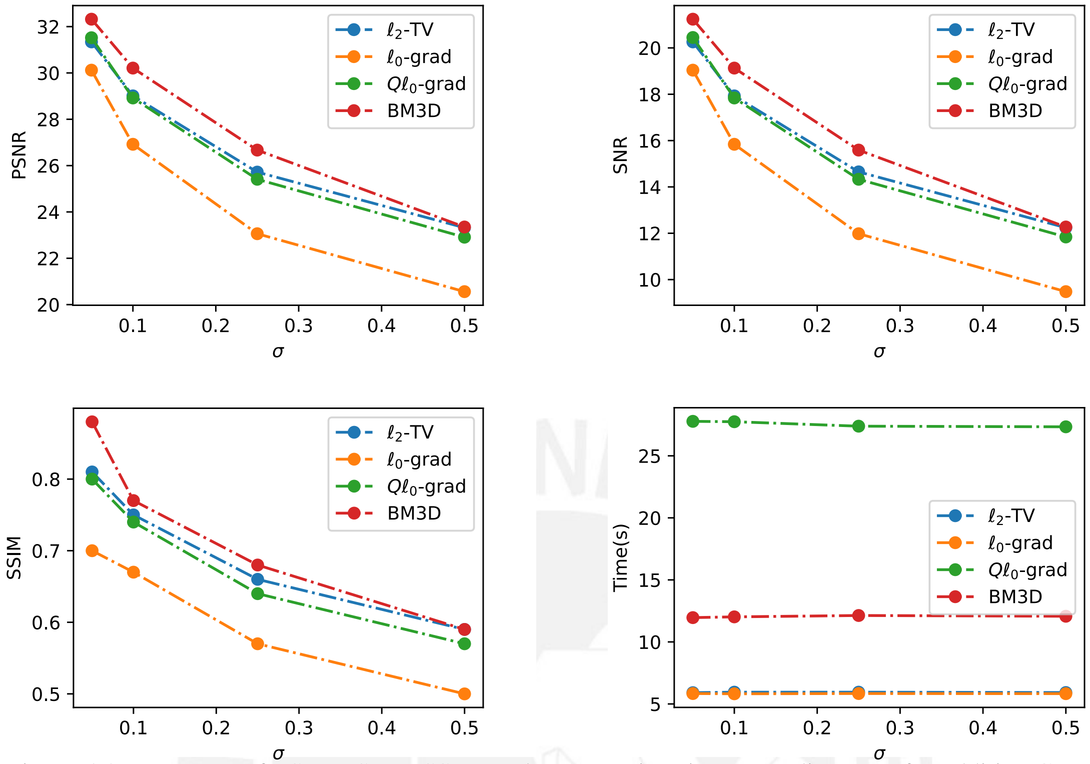


Figure 4.8: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for denoising color Peppers using the 4 methods.

The images presented in figures 4.9, 4.10 and 4.11 are some examples of what was obtained during the simulations. We can see that for $\sigma = 0.05$ and $\sigma = 0.1$ the processed images look fairly similar, because there was not that much noise to be reduced; with $\sigma = 0.25$ and $\sigma = 0.5$ the difference between methods became more visible. ℓ_0-grad returned images with plain regions and abrupt changes reducing some of the details of the originals. ℓ_2-TV and $Q\ell_0$-grad worked very similar preserving the edges, reducing the noise and some texture but keeping the overall structure and detail. BM3D gave images that, looking closer, were a little distorted because it works by block matching but they looked more like the original images than the ones obtained with the other methods.

(a) $\sigma = 0.05$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad (e) BM3D

(f) $\sigma = 0.1$ (g) ℓ_2-**TV** (h) ℓ_0-grad (i) $Q\ell_0$-grad (j) BM3D

(k) $\sigma = 0.25$ (l) ℓ_2-**TV** (m) ℓ_0-grad (n) $Q\ell_0$-grad (o) BM3D

Figure 4.9: Reconstructed images with ℓ_2-TV, ℓ_0-grad, $Q\ell_0$-grad and BM3D from a noisy image with different σ for additive Gaussian noise.

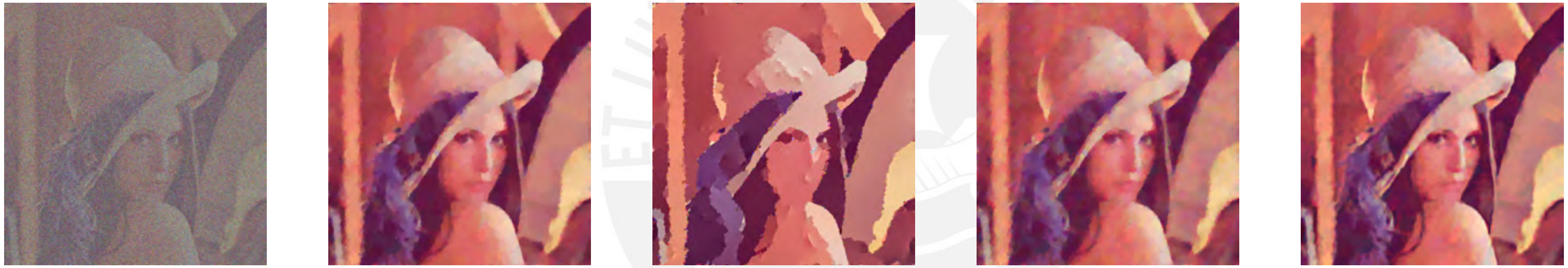
(a) $\sigma = 0.5$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad (e) BM3D

(f) $\sigma = 0.05$ (g) ℓ_2-**TV** (h) ℓ_0-grad (i) $Q\ell_0$-grad (j) BM3D

(k) $\sigma = 0.1$ (l) ℓ_2-**TV** (m) ℓ_0-grad (n) $Q\ell_0$-grad (o) BM3D

Figure 4.10: Reconstructed images with ℓ_2-TV, ℓ_0-grad, $Q\ell_0$-grad and BM3D from a noisy image with different σ for additive Gaussian noise.

(a) $\sigma = 0.25$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad (e) BM3D



(f) $\sigma = 0.5$ (g) ℓ_2-**TV** (h) ℓ_0-grad (i) $Q\ell_0$-grad (j) BM3D

Figure 4.11: Reconstructed images with ℓ_2-TV, ℓ_0-grad, $Q\ell_0$-grad and BM3D from a noisy image with different σ for additive Gaussian noise.

4.2 Results for deblurring with a 5×5 average filter

In this section, the images were filtered with a 5×5 average filter and then corrupted with additive Gaussian noise with different standard deviations ($\sigma = \{0.05, 0.1, 0.25, 0.5\}$). Similar to the denoising section, we used the same number of iterations for APG and ADMM respectively, and obtained the optimal λ by a Fibonacci search, but with different bounds, which are presented in Table A.8. We reconstructed the images 10 times with each of the 3 different methods: $Q\ell_0$-grad, ℓ_0-grad and ℓ_2-TV, and obtained the averages and standard deviations of the PSNR, SNR, SSIM metrics and computation time (tables A.9 to A.13).

In the graphs of the averages of the metrics and computation time with respect to σ, in figures 4.12 to 4.19, we can see that for $\sigma < 0.1$, in general no method was better than the others, because every one of them had at least some cases where its performance was the best one. In the greyscale cases for $\sigma \geq 0.1$, ℓ_2-TV and $Q\ell_0$-grad achieved nearly equal metrics, with their lines in the graphs sometimes overlapping, they were the most separated in the results for Peppers (Fig. 4.15), ℓ_0-grad most of the times got the worst metrics, the exceptions being for $\sigma = 0.1$ in Mandrill (Fig. 4.14) and Peppers (Fig. 4.15). For the color results with $\sigma \geq 0.1$, ℓ_2-TV was definitely the best one, followed by $Q\ell_0$-grad and then ℓ_0-grad, with some exceptions for $\sigma = 0.1$: $Q\ell_0$-grad performed the best for PSNR and SNR of Lena (Fig. 4.16) and ℓ_0-grad obtained the highest SSIM of Mandrill (Fig. 4.18). When compared to ℓ_0-grad, our model obtained an average improvement of +0.55 dB PSNR, +0.55 dB SNR and +0.01 SSIM. Computation time remained almost constant no matter the value of σ for the three methods and APG for $Q\ell_0$-grad was still the slowest during the simulations. Also, since $Q\ell_0$-grad had the same number of inner and outer iterations for deblurring with an average filter and for denoising, they both had the same computation time, 8 seconds for greyscale and 28 seconds for color approximately.

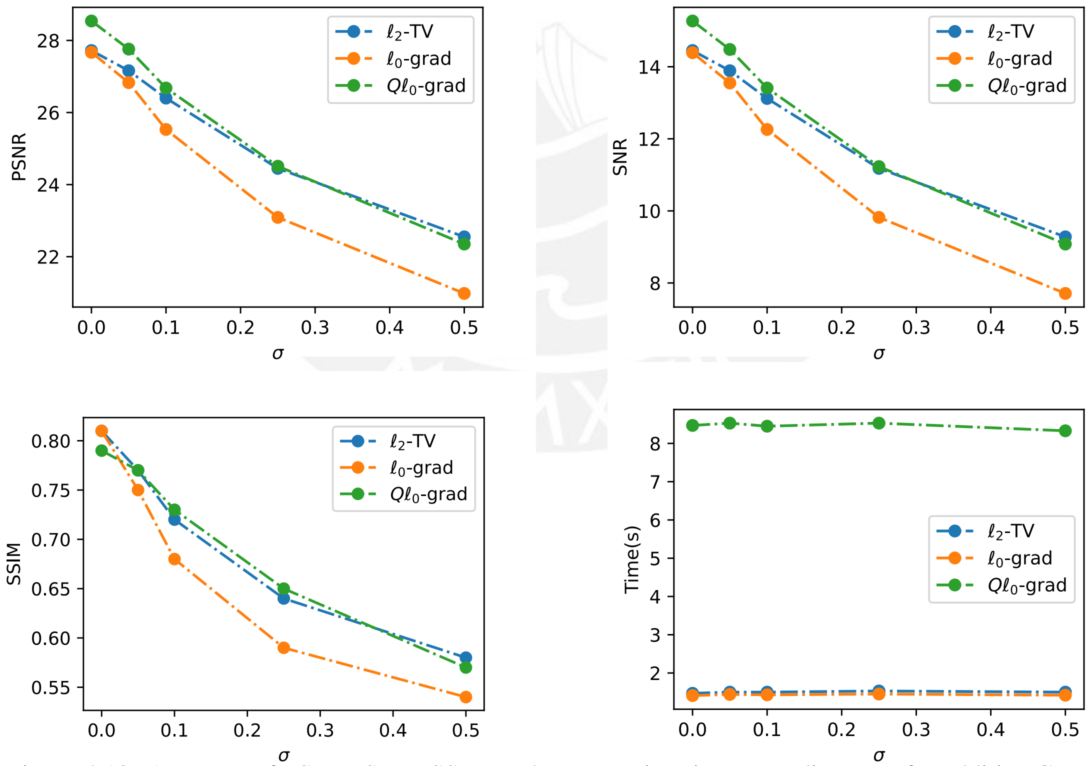


Figure 4.12: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Lena with a 5×5 average filter using the 3 methods

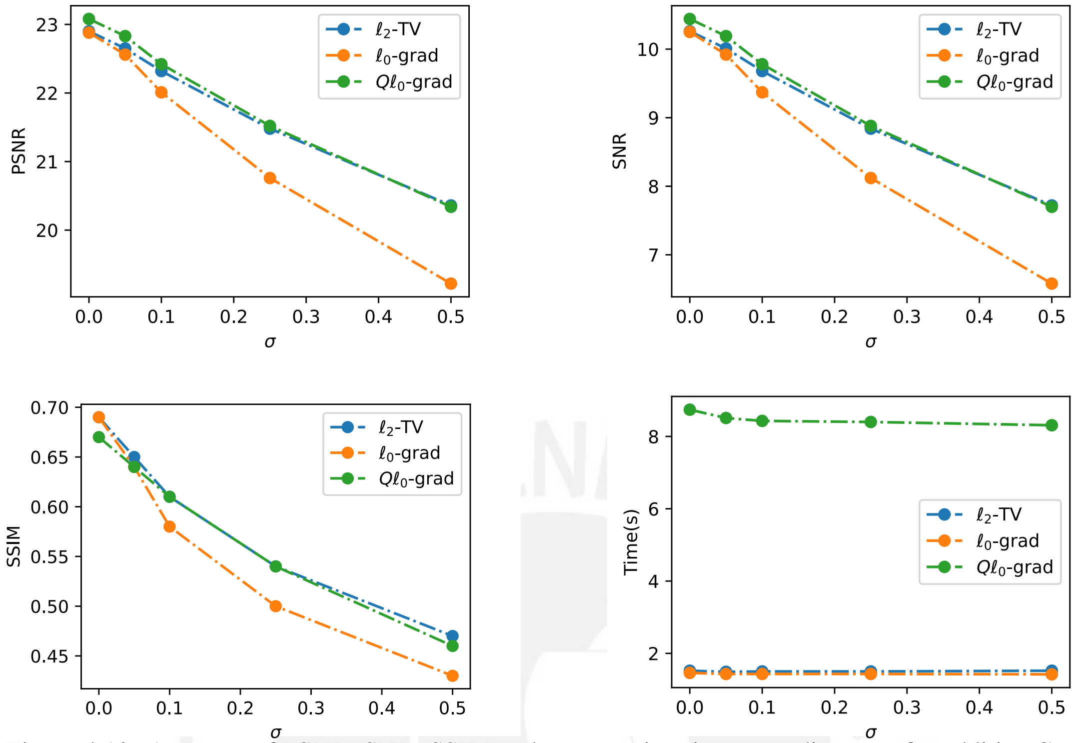


Figure 4.13: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Barbara with a 5×5 average filter using the 3 methods.

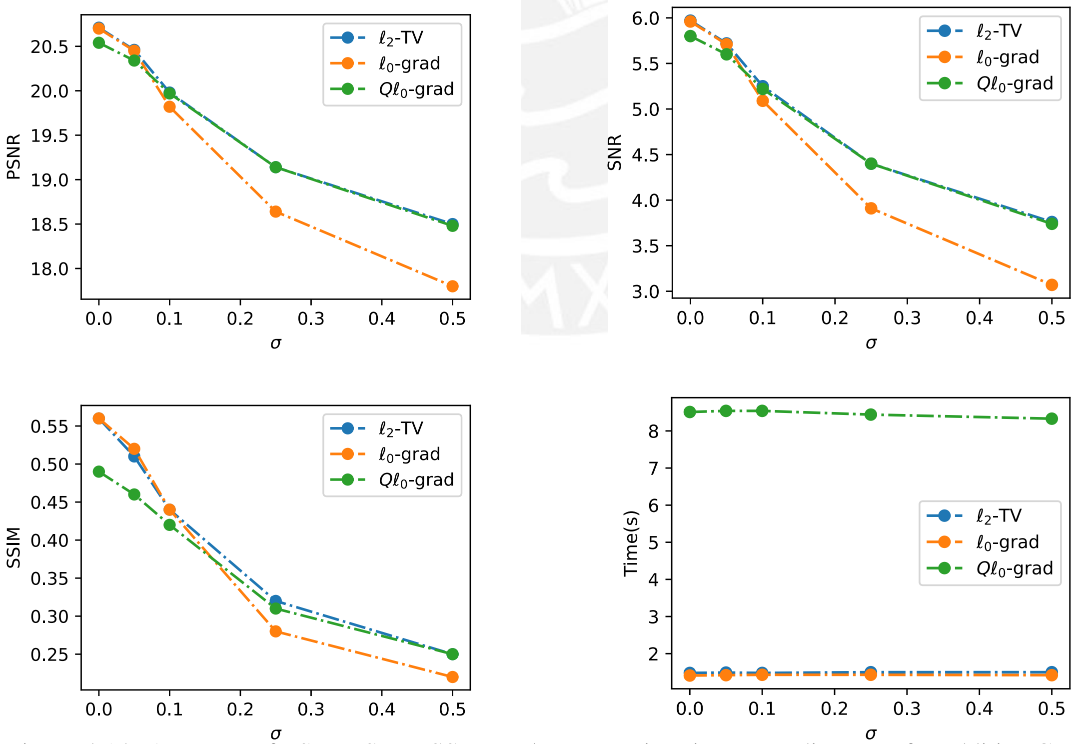


Figure 4.14: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Mandrill with a 5×5 average filter using the 3 methods.

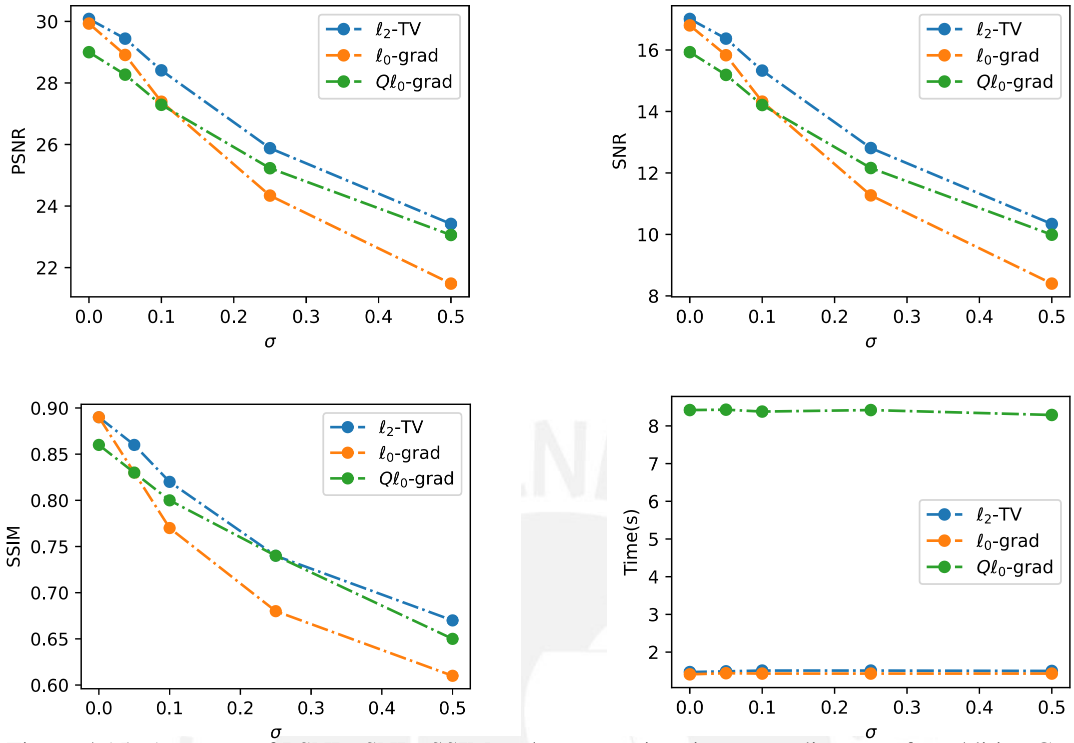


Figure 4.15: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Peppers with a 5×5 average filter using the 3 methods.

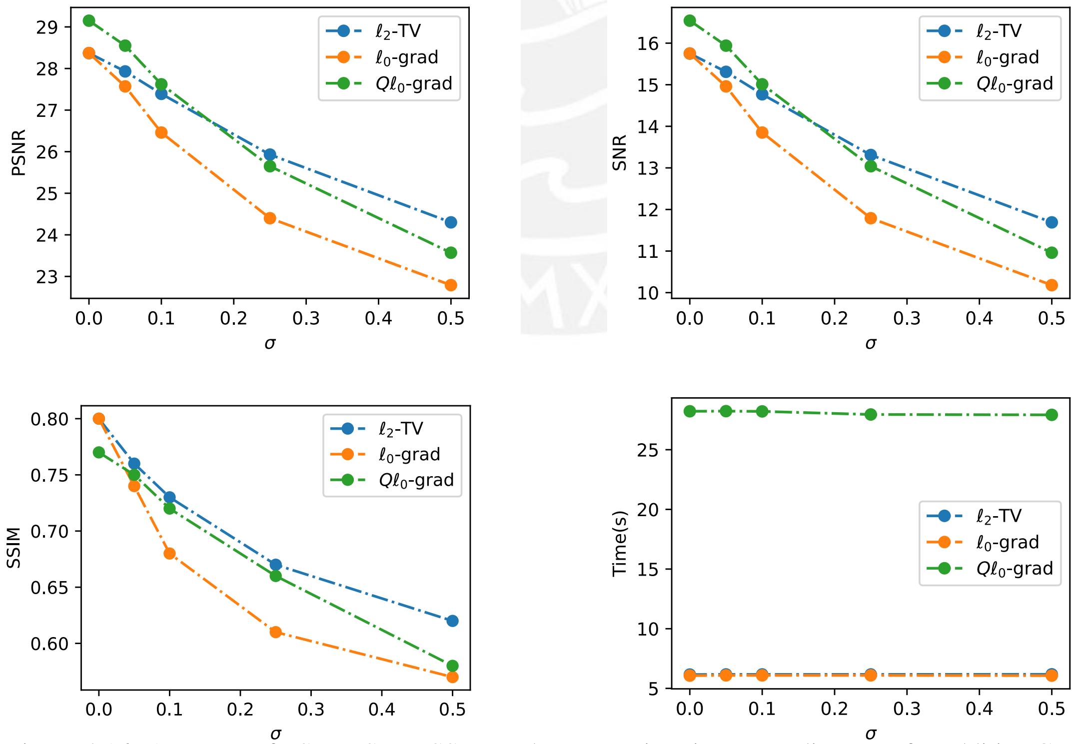


Figure 4.16: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring color Lena with a 5×5 average filter using the 3 methods.

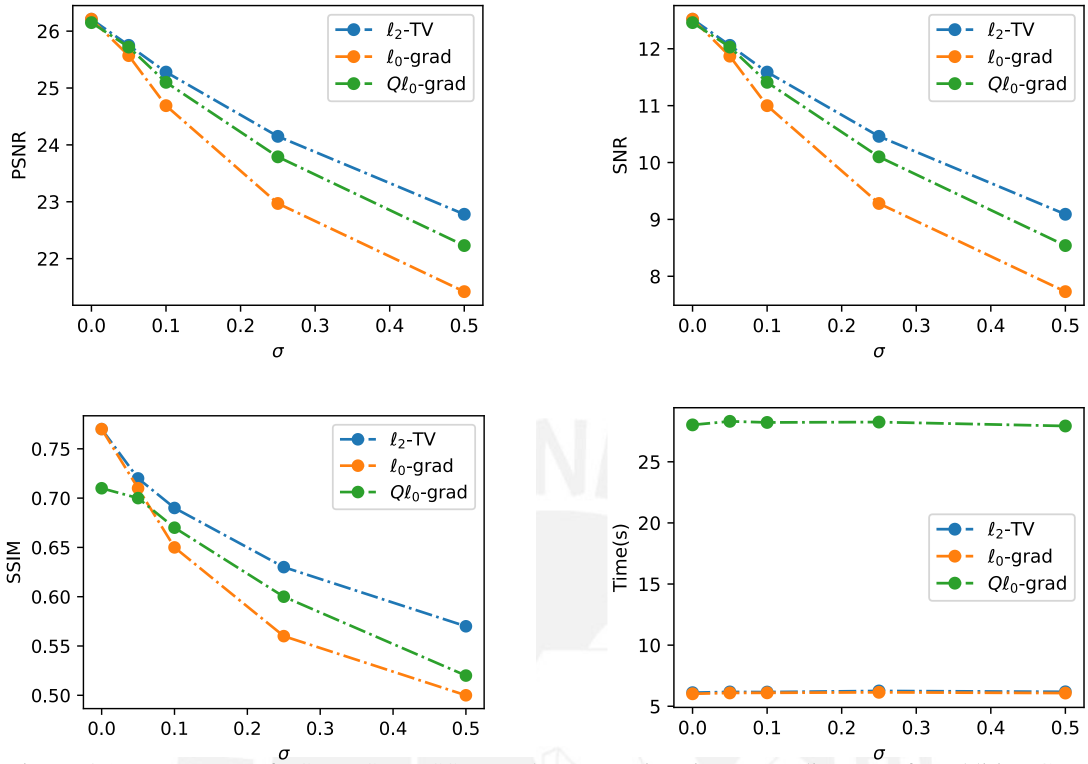


Figure 4.17: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring color Barbara with a 5×5 average filter using the 3 methods.

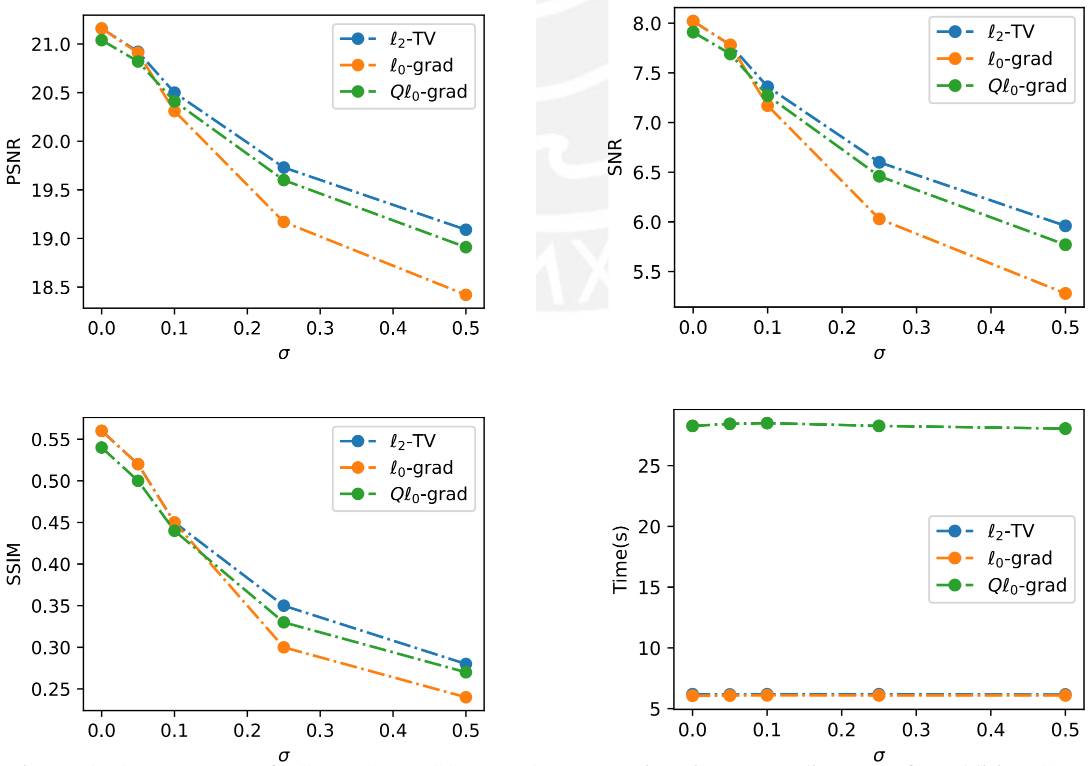


Figure 4.18: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring color Mandrill with a 5×5 average filter using the 3 methods.

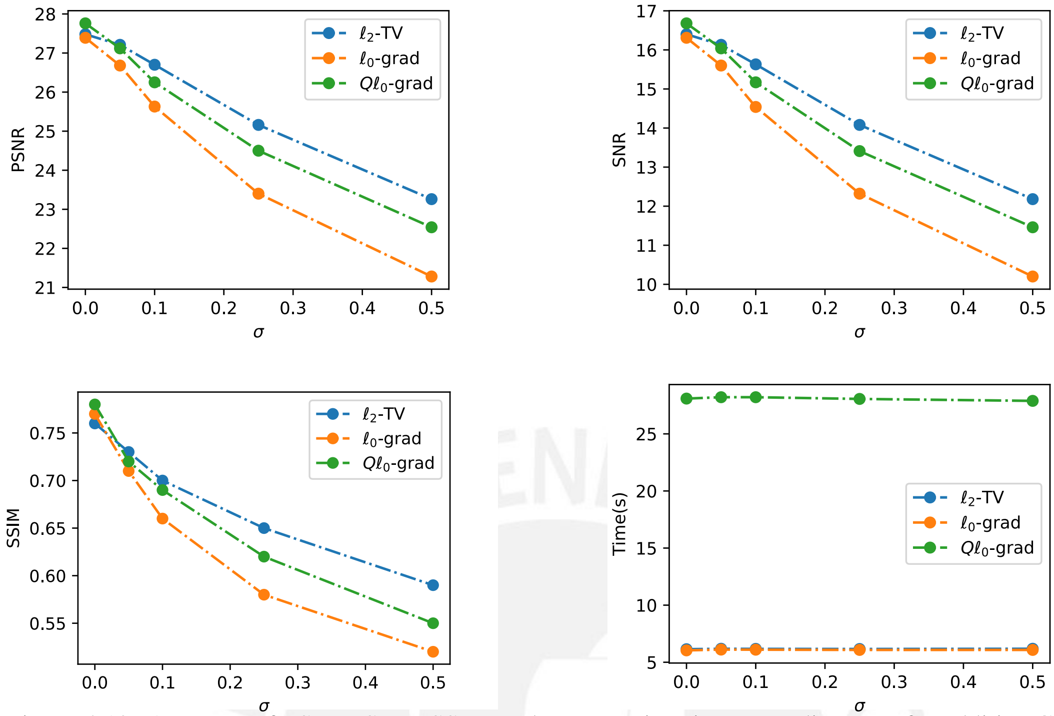
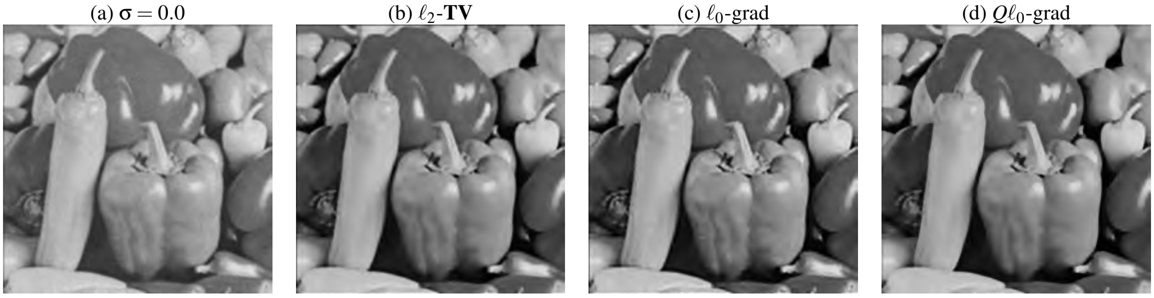


Figure 4.19: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring color Peppers with a 5×5 average filter using the 3 methods.

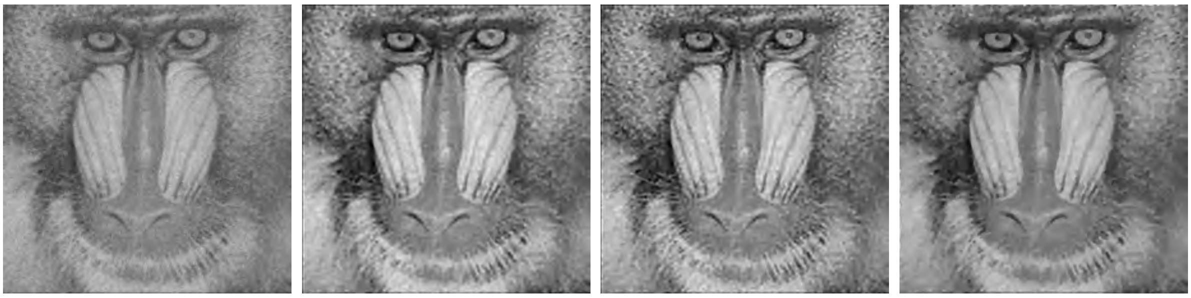
Figures 4.20, 4.21 and 4.22 show that for $\sigma \leq 0.1$ there was almost no difference between the three reconstructed images, all the methods kept the edges and reduced the noise. For $\sigma = 0.25$ the obtained images with $Q\ell_0$-grad and ℓ_2-TV were almost similar, but with $\sigma = 0.5$ the reconstructed image with $Q\ell_0$-grad had smoother changes between regions. Meanwhile, ℓ_0-grad with $\sigma = 0.25$ and $\sigma = 0.5$ reduced the noise significantly but distorted the image because it gave priority to having plain regions.

(a) $\sigma = 0.0$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad



(e) $\sigma = 0.05$ (f) ℓ_2-**TV** (g) ℓ_0-grad (h) $Q\ell_0$-grad

Figure 4.20: Reconstructed images with ℓ_2-TV, ℓ_0-grad and $Q\ell_0$-grad from a noisy image with different σ for additive Gaussian noise and filtered with an average 5×5 filter.



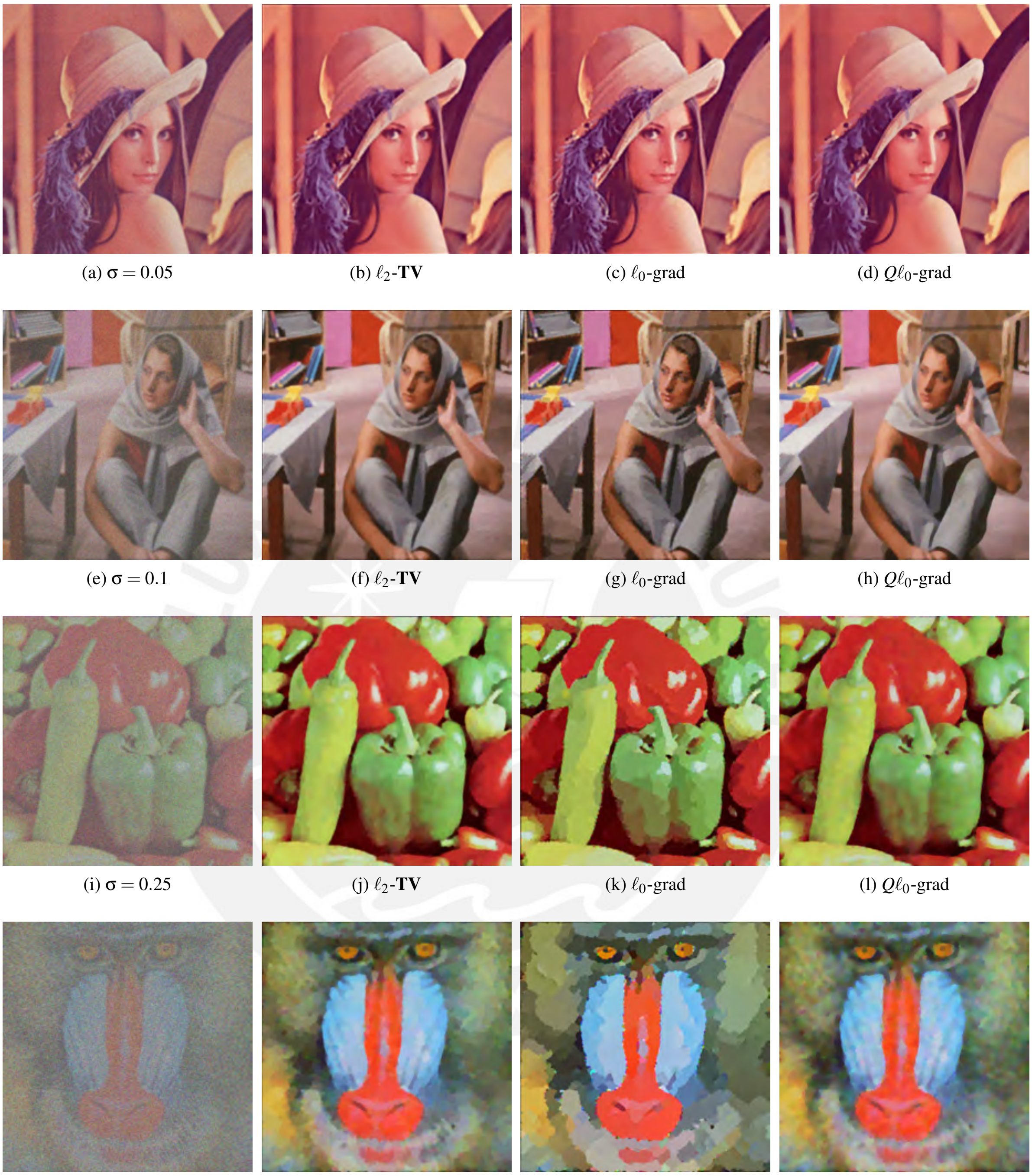
(a) $\sigma = 0.1$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad

(e) $\sigma = 0.25$ (f) ℓ_2-**TV** (g) ℓ_0-grad (h) $Q\ell_0$-grad

(i) $\sigma = 0.5$ (j) ℓ_2-**TV** (k) ℓ_0-grad (l) $Q\ell_0$-grad

(m) $\sigma = 0.0$ (n) ℓ_2-**TV** (o) ℓ_0-grad (p) $Q\ell_0$-grad

Figure 4.21: Reconstructed images with ℓ_2-TV, ℓ_0-grad and $Q\ell_0$-grad from a noisy image with different σ for additive Gaussian noise and filtered with an average 5×5 filter.



(a) $\sigma = 0.05$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad

(e) $\sigma = 0.1$ (f) ℓ_2-**TV** (g) ℓ_0-grad (h) $Q\ell_0$-grad

(i) $\sigma = 0.25$ (j) ℓ_2-**TV** (k) ℓ_0-grad (l) $Q\ell_0$-grad

(m) $\sigma = 0.5$ (n) ℓ_2-**TV** (o) ℓ_0-grad (p) $Q\ell_0$-grad

Figure 4.22: Reconstructed images with ℓ_2-TV, ℓ_0-grad and $Q\ell_0$-grad from a noisy image with different σ for additive Gaussian noise and filtered with an average 5×5 filter.

4.3 Results for deblurring with a 9×9 Gaussian filter

Here, we kept using 20 ADMM iterations for ℓ_2-TV and ℓ_0-grad, however, unlike denoising and deblurring with an average filter, for APG we used 20 outer and 30 inner iterations. The increase on the number of inner iterations was done because the Gaussian filter was more aggressive than the average filter. We also performed a Fibonacci search similar to the previous two sections with the bounds for λ given in Table A.14. We performed 10 simulations, using the 3 methods on the test images filtered with a 9×9 Gaussian filter and corrupted with additive Gaussian noise, and calculated the averages and standard deviations of the PSNR, SNR, SSIM metrics and computation time (tables A.15 to A.19).

The following graphs (Fig. 4.23 to 4.30) were made with the averages of the metrics and computation time. $Q\ell_0$-grad obtained the best PSNR and SNR for $\sigma \leq 0.1$ with the exceptions of greyscale Peppers (Fig. 4.26) and color Barbara (Fig. 4.28), where ℓ_2-TV was superior by a small margin for $\sigma = 0.1$. For SSIM with $\sigma \leq 0.1$, even though $Q\ell_0$-grad performed fairly good for all the images, being the best sometimes, for example both greyscale and color cases of Lena (Fig. 4.23 and 4.27), other times it started as the worst but then it improved its performance like in color Mandrill (Fig. 4.29). For bigger σ values, the methods behaved like in denoising and deblurring with an average filter, this means that the performance of $Q\ell_0$-grad was between ℓ_0-grad and ℓ_2-TV, being better than ℓ_0-grad. For the greyscale cases $Q\ell_0$-grad was closer to ℓ_2-TV, even overlapping each other for the Mandrill image (Fig. 4.25). Moreover, the averages of the differences between the obtained metrics with $Q\ell_0$-grad and ℓ_0-grad were +0.75 dB PSNR, +0.75 dB SNR and +0.02 SSIM. Computation time stayed almost constant during the simulations, like in the other previous two cases. Also, since our method now had 30 inner iterations, it reached more than 40 seconds for color images.

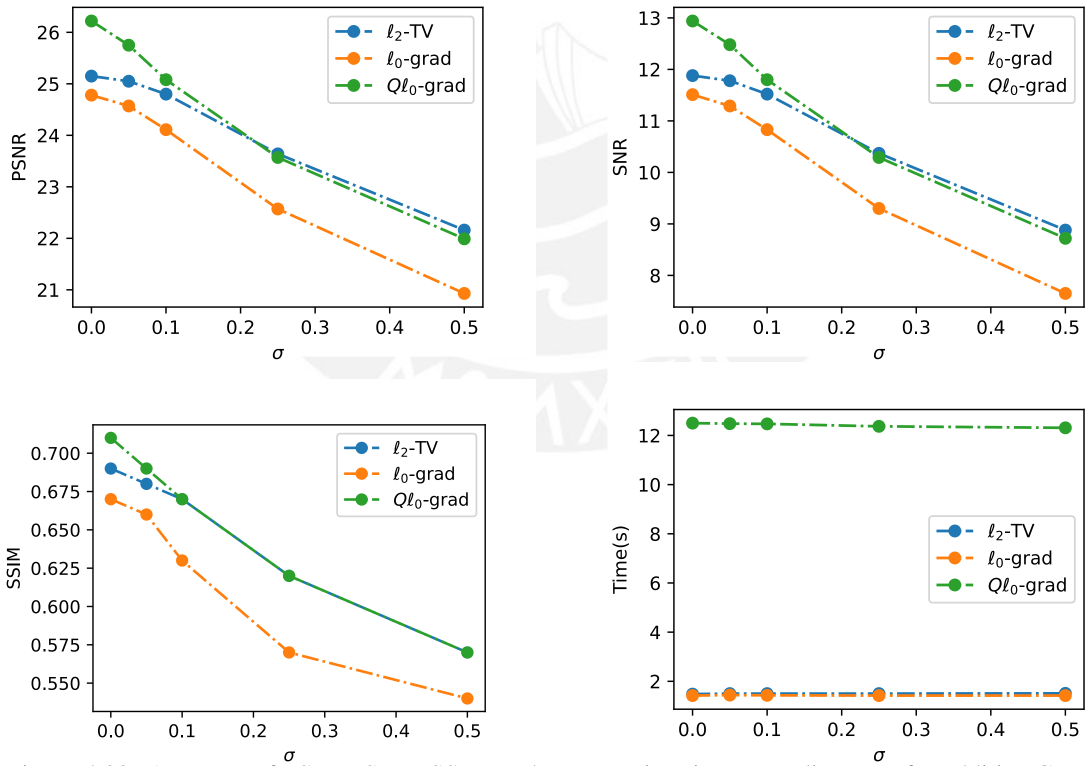


Figure 4.23: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Lena with a Gaussian filter using the 3 methods.

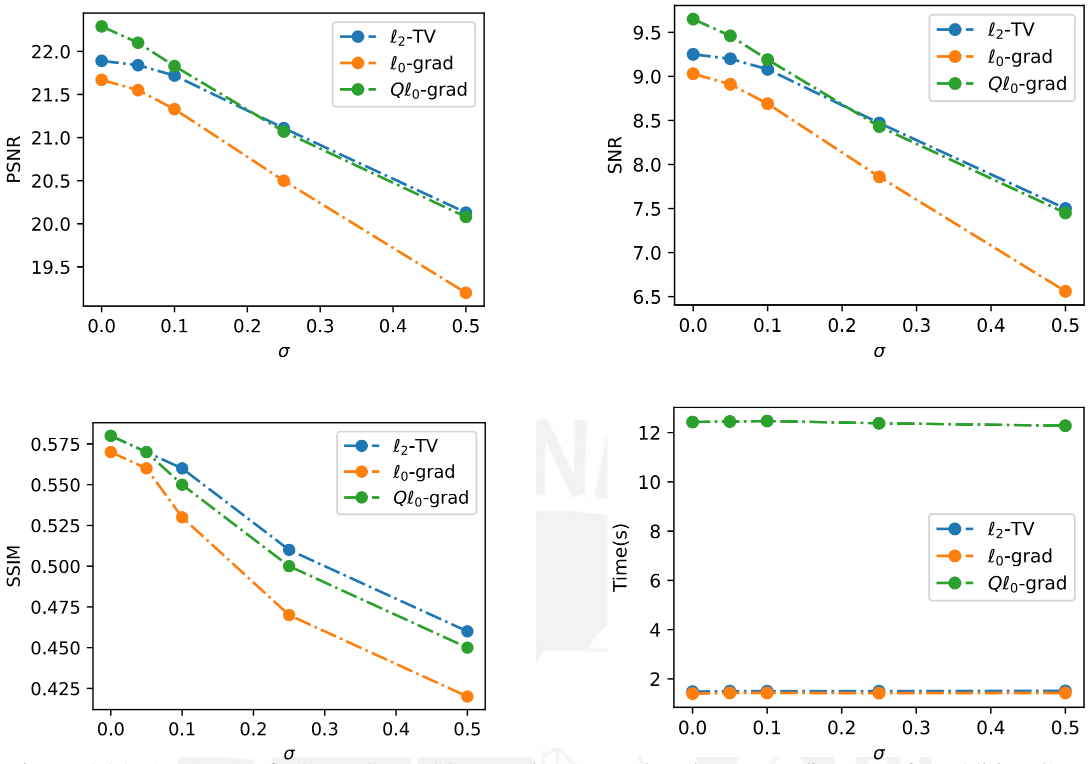


Figure 4.24: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Barbara with a Gaussian filter using the 3 methods.

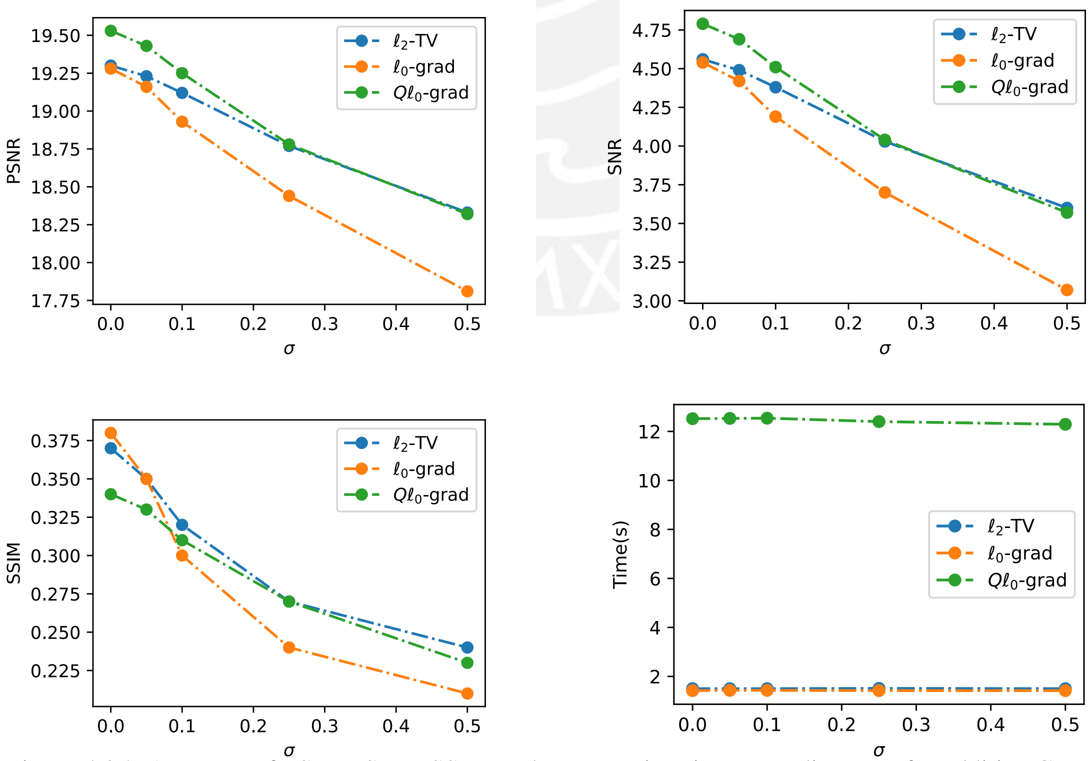


Figure 4.25: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Mandrill with a Gaussian filter using the 3 methods.

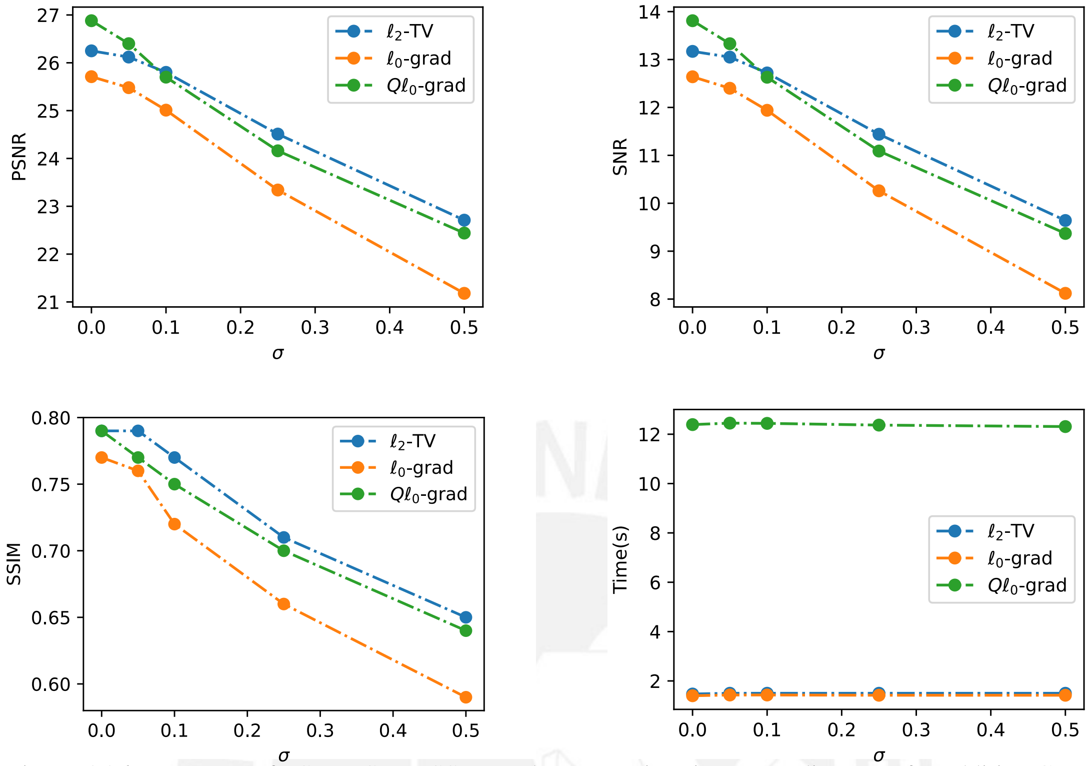


Figure 4.26: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Peppers with a Gaussian filter using the 3 methods.

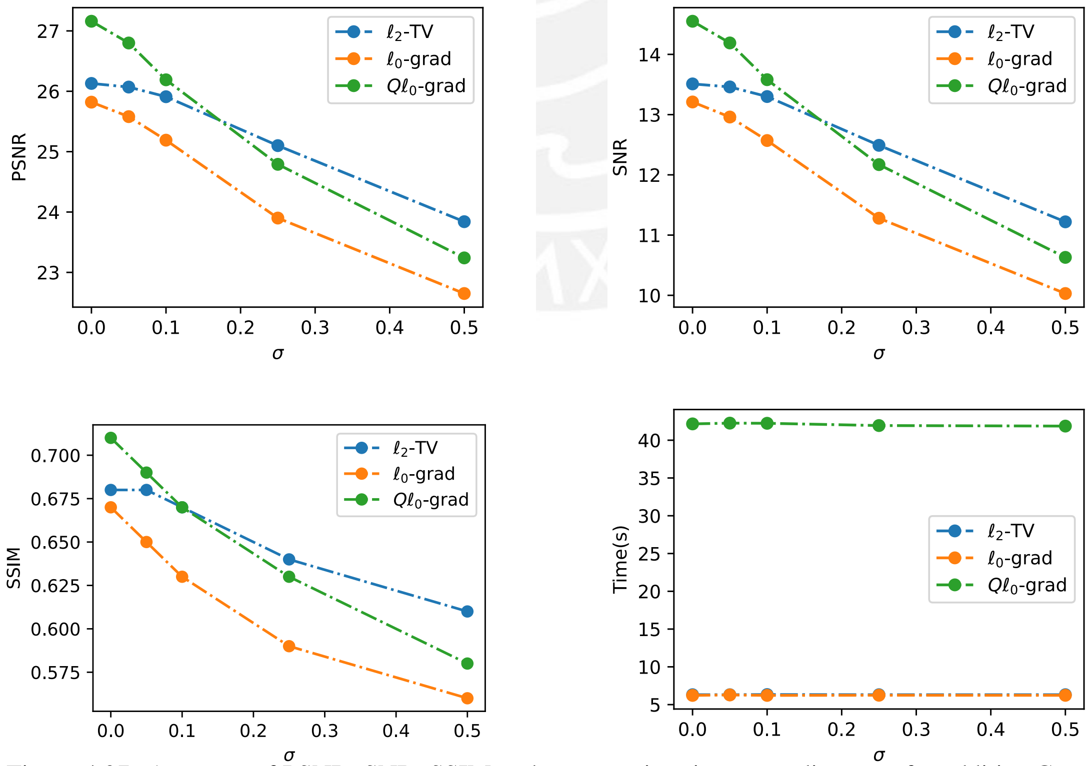


Figure 4.27: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring color Lena with a Gaussian filter using the 3 methods.

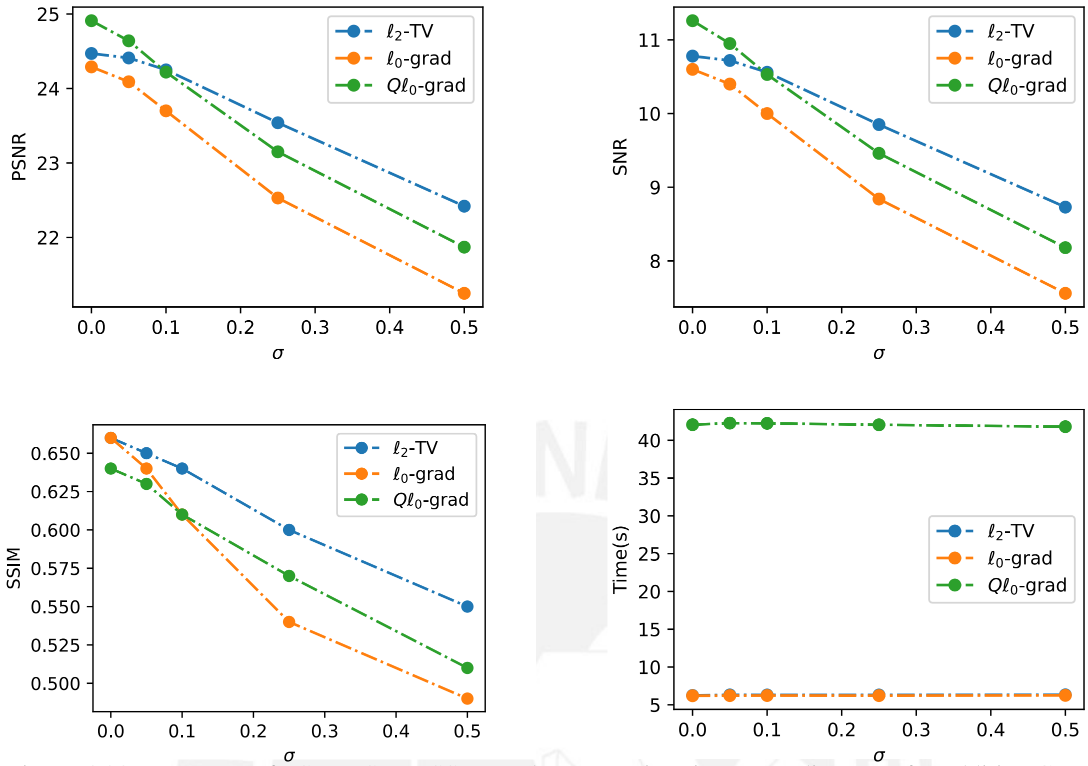


Figure 4.28: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring color Barbara with a Gaussian filter using the 3 methods.

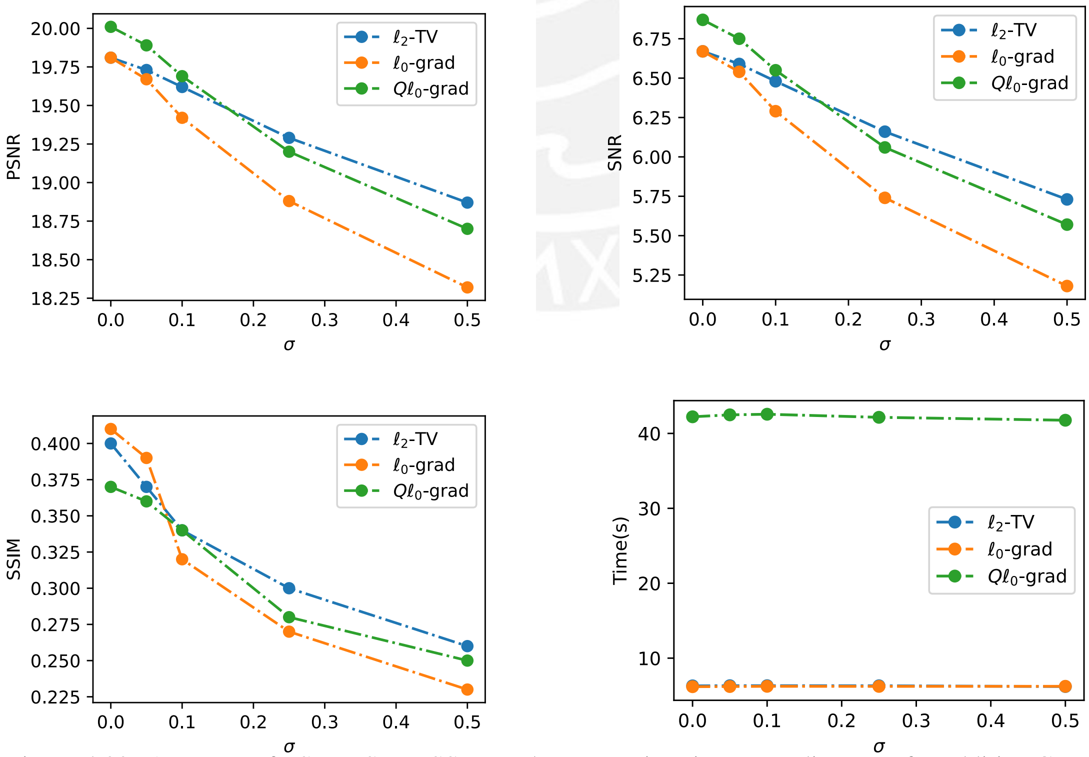


Figure 4.29: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring color Mandrill with a Gaussian filter using the 3 methods.

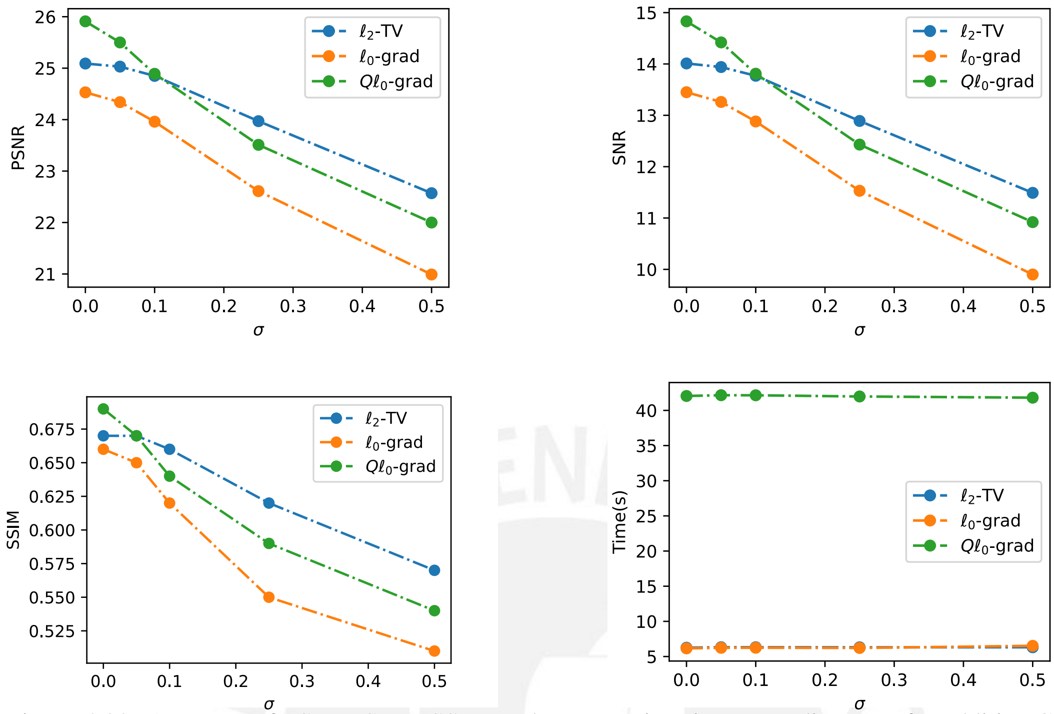
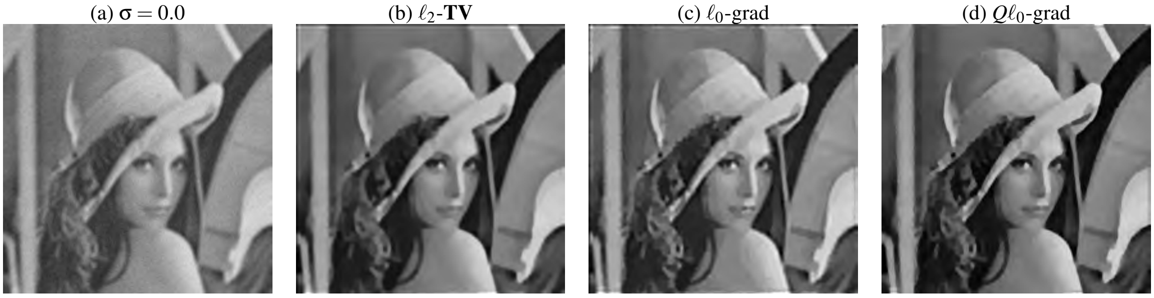


Figure 4.30: Averages of PSNR, SNR, SSIM and computation time according to σ for additive Gaussian noise, after 10 experiments, for deblurring color Peppers with a Gaussian filter using the 3 methods.

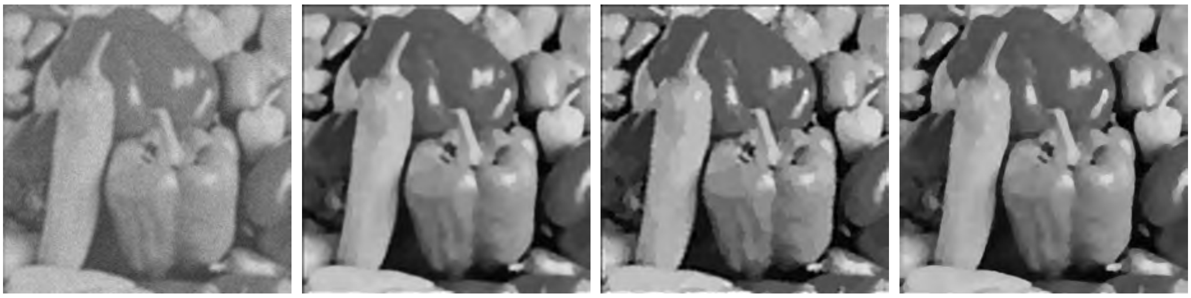
In figures 4.31, 4.32 and 4.33, we observe that for $\sigma = 0$, $\sigma = 0.05$ and $\sigma = 0.1$ the reconstructed images with $Q\ell_0$-grad had better sharpness and kept the edges well-defined. For $\sigma = 0.25$ and $\sigma = 0.5$, ℓ_0-grad gave distorted images with plain regions but preserving some of the edges. Meanwhile, the reconstructed images with ℓ_2-TV and $Q\ell_0$-grad were very similar, however the changes between adjacent regions with $Q\ell_0$-grad were smoother than the ones obtained with ℓ_2-TV.

(a) $\sigma = 0.0$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad



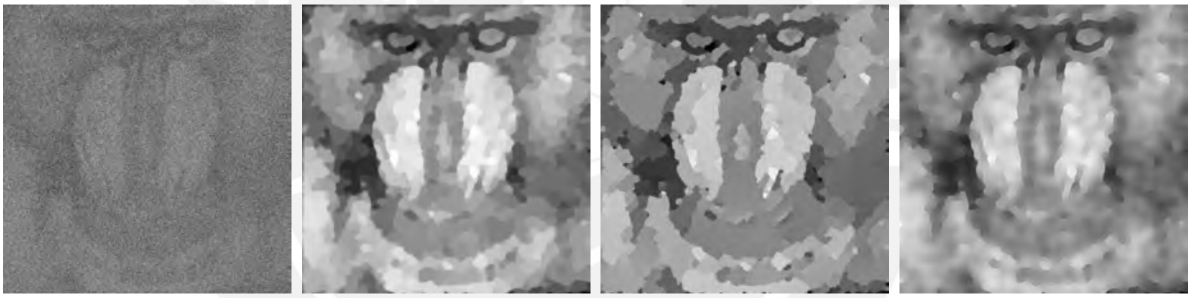
(e) $\sigma = 0.05$ (f) ℓ_2-**TV** (g) ℓ_0-grad (h) $Q\ell_0$-grad

Figure 4.31: Reconstructed images with ℓ_2-TV, ℓ_0-grad and $Q\ell_0$-grad from a noisy image with different σ for additive Gaussian noise and a Gaussian 9×9 filter.



(a) $\sigma = 0.1$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad

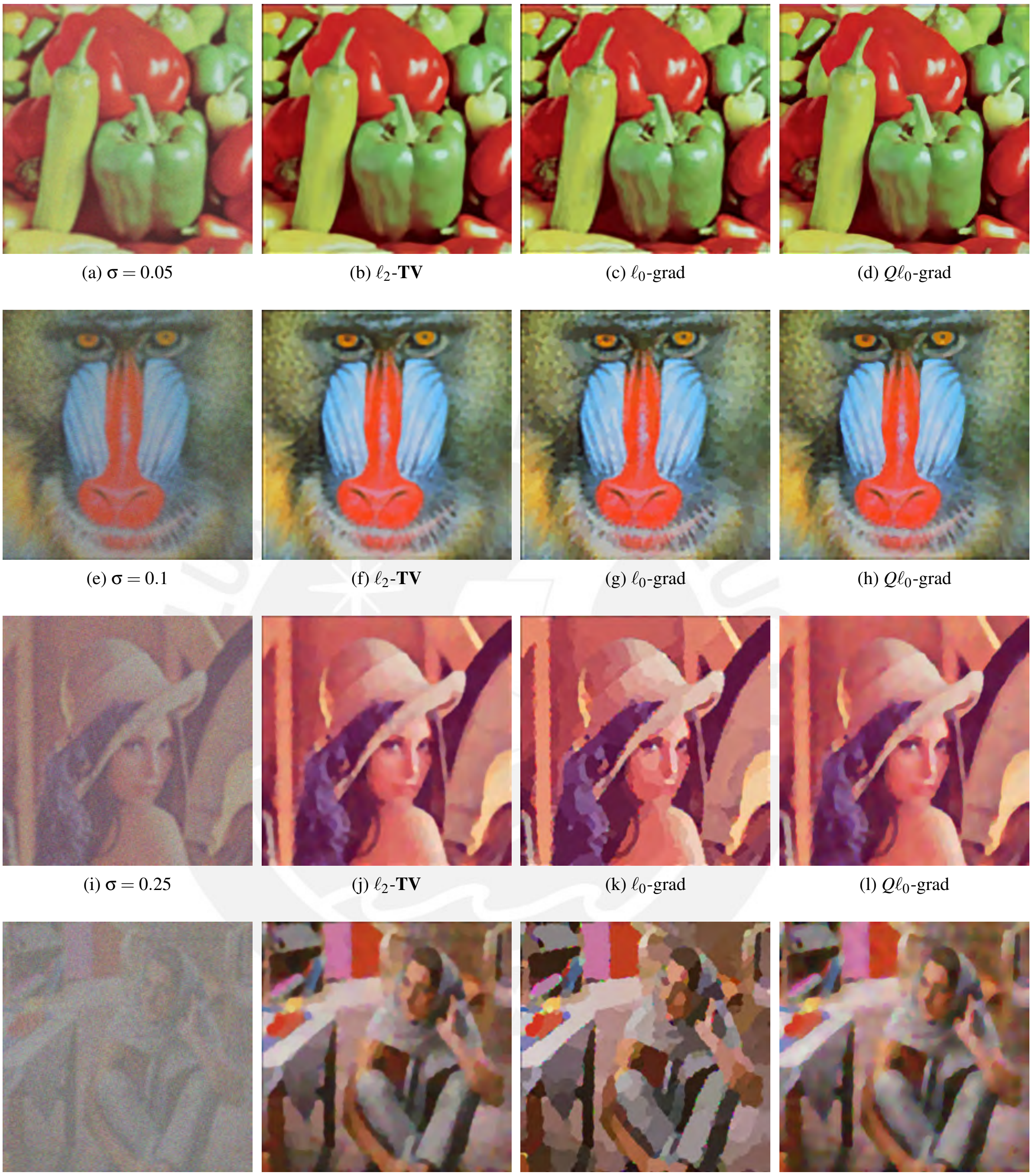
(e) $\sigma = 0.25$ (f) ℓ_2-**TV** (g) ℓ_0-grad (h) $Q\ell_0$-grad



(i) $\sigma = 0.5$ (j) ℓ_2-**TV** (k) ℓ_0-grad (l) $Q\ell_0$-grad

(m) $\sigma = 0.0$ (n) ℓ_2-**TV** (o) ℓ_0-grad (p) $Q\ell_0$-grad

Figure 4.32: Reconstructed images with ℓ_2-TV, ℓ_0-grad and $Q\ell_0$-grad from a noisy image with different σ for additive Gaussian noise and a Gaussian 9×9 filter.



(a) $\sigma = 0.05$ (b) ℓ_2-**TV** (c) ℓ_0-grad (d) $Q\ell_0$-grad

(e) $\sigma = 0.1$ (f) ℓ_2-**TV** (g) ℓ_0-grad (h) $Q\ell_0$-grad

(i) $\sigma = 0.25$ (j) ℓ_2-**TV** (k) ℓ_0-grad (l) $Q\ell_0$-grad

(m) $\sigma = 0.5$ (n) ℓ_2-**TV** (o) ℓ_0-grad (p) $Q\ell_0$-grad

Figure 4.33: Reconstructed images with ℓ_2-TV, ℓ_0-grad and $Q\ell_0$-grad from a noisy image with different σ for additive Gaussian noise and a Gaussian 9×9 filter.

4.4 Comparison between APG and ADMM algorithms for $Q\ell_0$-grad

We compared the performance of both algorithms analyzing the SNR, PSNR and SSIM metrics, the evolution of the cost function $\left(f(\mathbf{u}) = \frac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 + \sqrt{2\gamma\lambda}\mathbf{TV}_I(\mathbf{u}) - \frac{\gamma}{2}\|\nabla\mathbf{u}\|_2^2\right)$ and computation time. For ADMM we used $\rho = 20$ and $\beta = 1.5$ and for APG a Cauchy lagged stepsize, and for both of them we used the values of λ found by the Fibonacci searches performed for the previous sections, with $\gamma = 0.01$. The pseudo-codes of both algorithms are given in section D of the Annexes.

4.4.1 Greyscale comparison

1. **Denoising**

   For denoising we chose the greyscale Barbara image with Gaussian additive noise of $\sigma = 0.25$, we then ran 40 iterations of ADMM and 20 outer, with 20 inner, iterations of APG. The graphs presented in Fig.4.34 show that even though the evolutions of the cost function were very different, their values at the end were similar. We can see that in terms of iterations, APG needed less to arrive to a steady minimum, however ADMM was much faster when it came to time, performing all of its iterations even before APG stabilized.

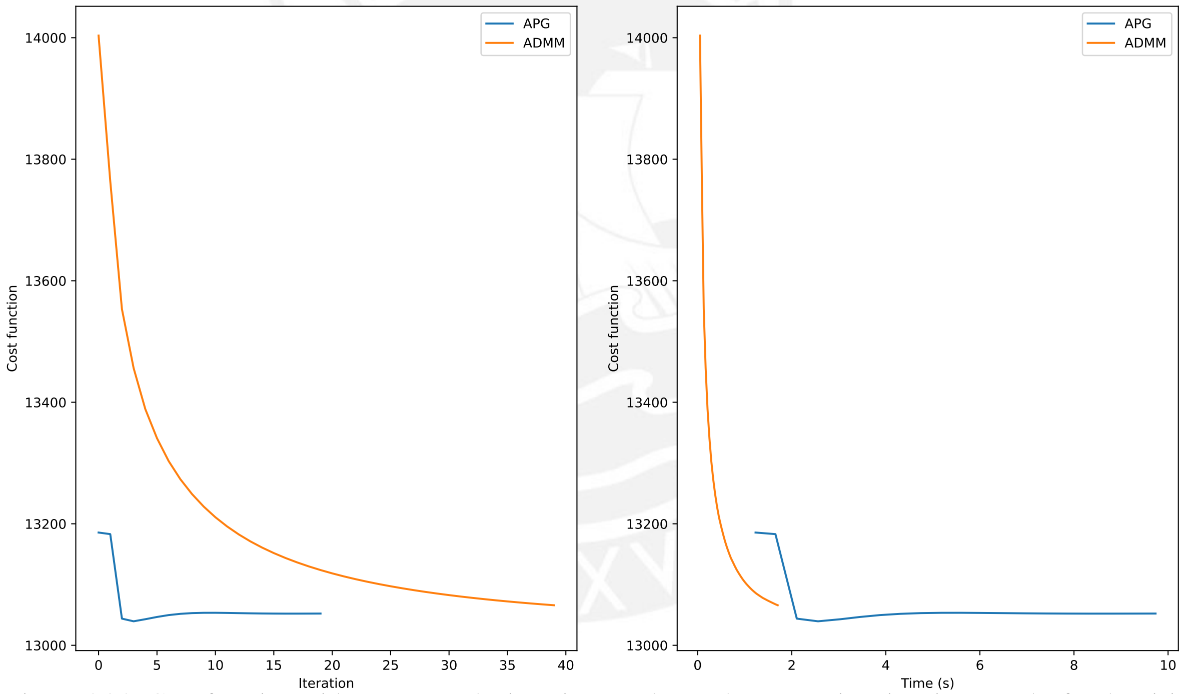


Figure 4.34: Cost function with respect to the iteration number and computation time in seconds, for denoising greyscale Barbara with Gaussian additive noise of $\sigma = 0.25$ using the APG and ADMM algorithms for $Q\ell_0$-grad.

   The resulting images and their respective metrics presented in Fig. 4.35 were very similar for the two algorithms, showing that we could use any of them for comparisons with the other methods, since they both clearly reduced the amount of noise and preserved the edges.

(a) Noisy greyscale image with $\sigma = 0.25$: PSNR=11.30, SNR=-1.35, SSIM=0.25

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=21.96, SNR=9.32, SSIM=0.55

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=21.99, SNR=9.35, SSIM=0.56

Figure 4.35: Reconstruction of a noisy greyscale image with Gaussian additive noise of $\sigma = 0.25$ using $Q\ell_0$-grad with APG and ADMM.

2. **Deblurring with an average filter**

   For this, we chose the greyscale Mandrill image filtered with a 5×5 average filter and Gaussian additive noise of $\sigma = 0.5$, we then ran 40 iterations of ADMM and 20 outer, with 20 inner, iterations of APG. The graphs presented in Fig.4.36 show that ADMM, unlike APG, first increased the value of the cost function before decreasing it; however, it obtained a lower minimum than APG. Analyzing the computation time, it can be seen that ADMM was still faster.

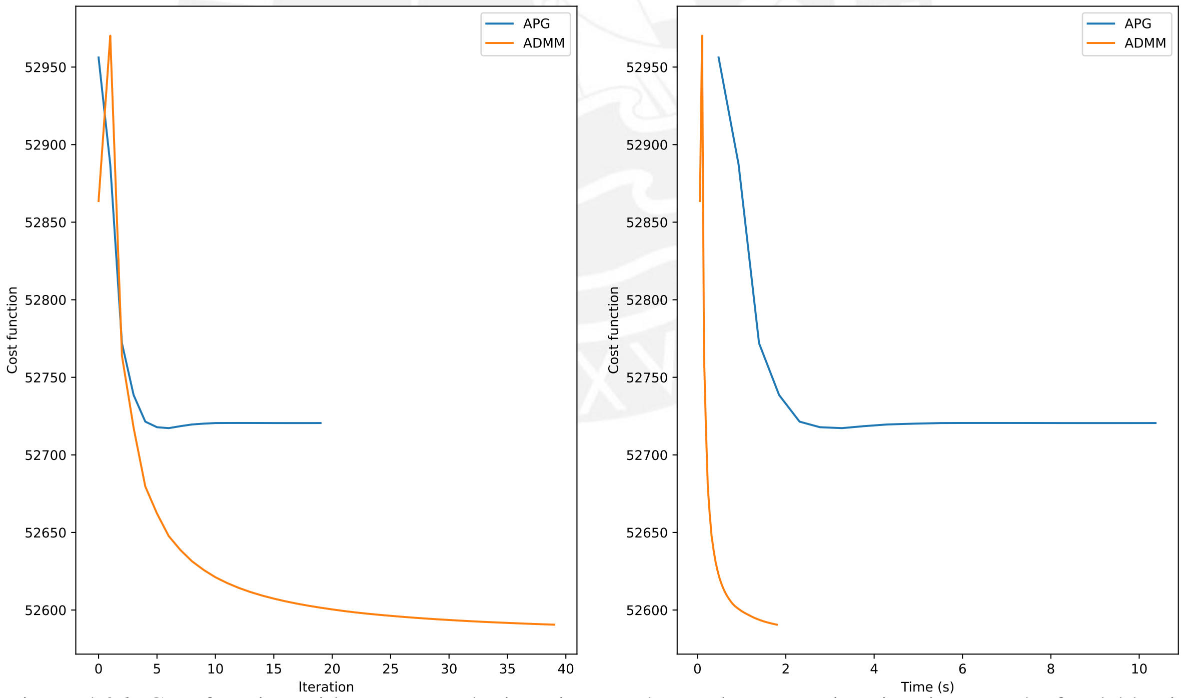


Figure 4.36: Cost function with respect to the iteration number and computation time in seconds, for deblurring greyscale Mandrill with a 5×5 average filter and Gaussian additive noise of $\sigma = 0.25$ using the APG and ADMM algorithms for $Q\ell_0$-grad.

The similarity in the metrics in Fig. 4.37 for the reconstructed images with both algorithms show that we could use any of the two for comparisons. The images also show how the noise was considerably lowered and the edges were preserved, with APG giving a slightly smoother result than ADMM.

(a) Noisy greyscale image with $\sigma = 0.5$: PSNR=5.03, SNR=-9.70, SSIM=0.08

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=18.47, SNR=3.73, SSIM=0.25

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=18.50, SNR=3.76, SSIM=0.25

Figure 4.37: Reconstruction of a noisy greyscale image with Gaussian additive noise of $\sigma = 0.5$ and an average 5×5 filter using $Q\ell_0$-grad with APG and ADMM.

3. **Deblurring with a Gaussian filter**

   For deblurring with a Gaussian filter, we chose the greyscale Lena image filtered with a 9×9 Gausian filter and Gaussian additive noise of $\sigma = 0.5$, we then ran 40 iterations of ADMM and 20 outer, with 30 inner, iterations of APG. We increased the number of inner iterations for APG because the Gaussian filter was more aggressive than the average one.

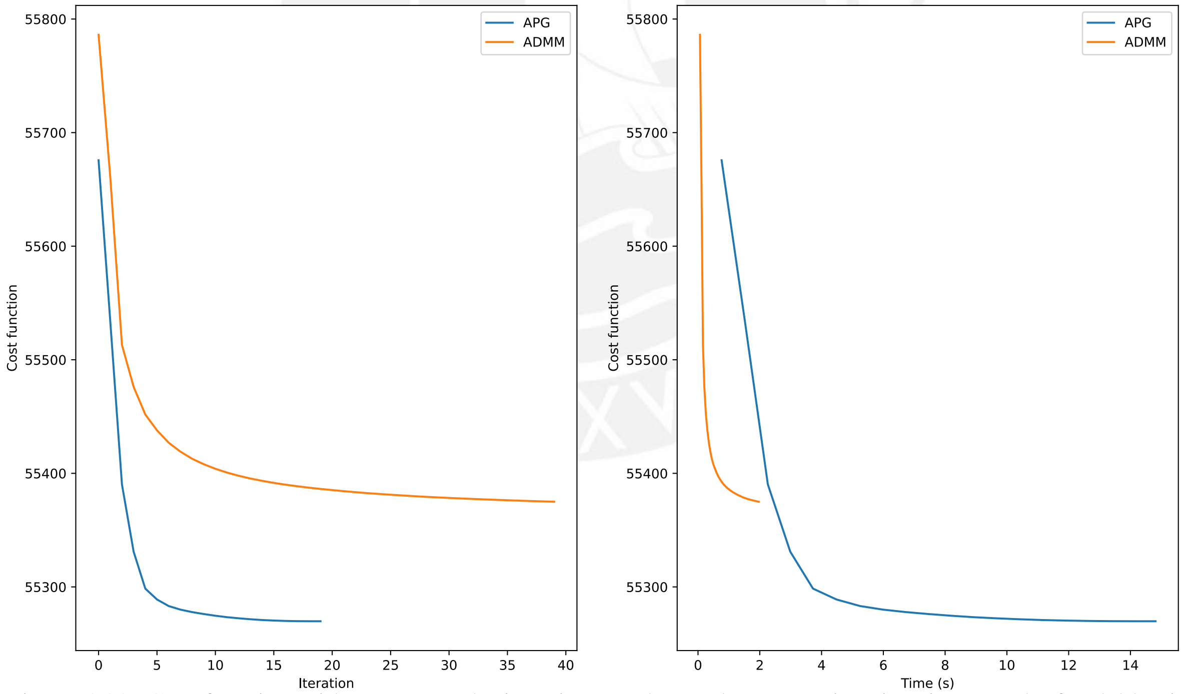


Figure 4.38: Cost function with respect to the iteration number and computation time in seconds, for deblurring greyscale Lena with a 9×9 Gaussian filter and Gaussian additive noise of $\sigma = 0.55$ using the APG and ADMM algorithms for $Q\ell_0$-grad.

The graphs presented in Fig. 4.38 show that APG obtained a lower value of the cost function than ADMM. Furthermore, the images and their metrics in Fig. 4.39 show that they both did a very good job in removing the noise and preserving the main edges. ADMM reconstructed the image with slightly better PSNR and SNR, and similar to the other two cases, it was clearly faster.

(a) Noisy greyscale image with $\sigma = 0.5$: PSNR=4.74, SNR=-8.53, SSIM=0.08

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=21.91, SNR=8.64, SSIM=0.56

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=22.08, SNR=8.81, SSIM=0.56

Figure 4.39: Reconstruction of a noisy greyscale image with Gaussian additive noise of $\sigma = 0.5$ and a Gaussian 9×9 filter using $Q\ell_0$-grad with APG and ADMM.

4.4.2 Color Comparison

For the greyscale cases, we observed that there was not any significant difference in the results obtained when using either ADMM or APG, except for the computation time; however, when working with color images, this was not the case. For denoising and deblurring of color images, we used the same parameters and number of iterations as the greyscale comparisons and the λ obtained via the Fibonacci searches, but with two different values of γ.

1. **Denoising**

   For this we used the color Lena image with Gaussian additive noise of $\sigma = 0.25$.

   - $\boldsymbol{\gamma = 0.01}$

(a) Noisy color image with $\sigma = 0.25$: PSNR=11.94, SNR=-0.67, SSIM=0.20

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=26.26, SNR=13.64, SSIM=0.67

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=25.27, SNR=12.65, SSIM=0.55

Figure 4.40: Reconstruction of a noisy color image with Gaussian additive noise of $\sigma = 0.25$ using $Q\ell_0$-grad with APG and ADMM, with $\gamma = 0.01$.

The metrics given in Fig. 4.40 showed that APG clearly obtained better PSNR, SNR and SSIM than ADMM; also, the reconstructed image obtained with ADMM did not have almost any plain regions and the unwanted low-amplitude structures were still kept. By observing the evolution of the cost functions in Fig. 4.41, we could see that this happened because when using ADMM the function actually diverged, increasing its value instead of decreasing.

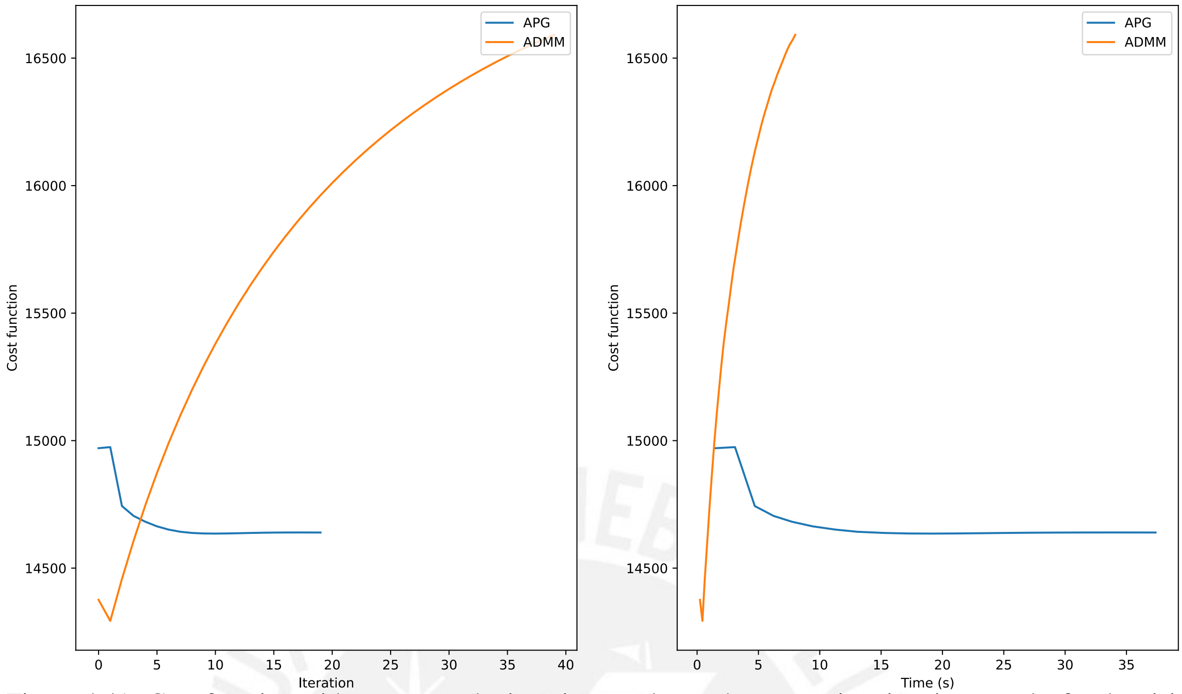
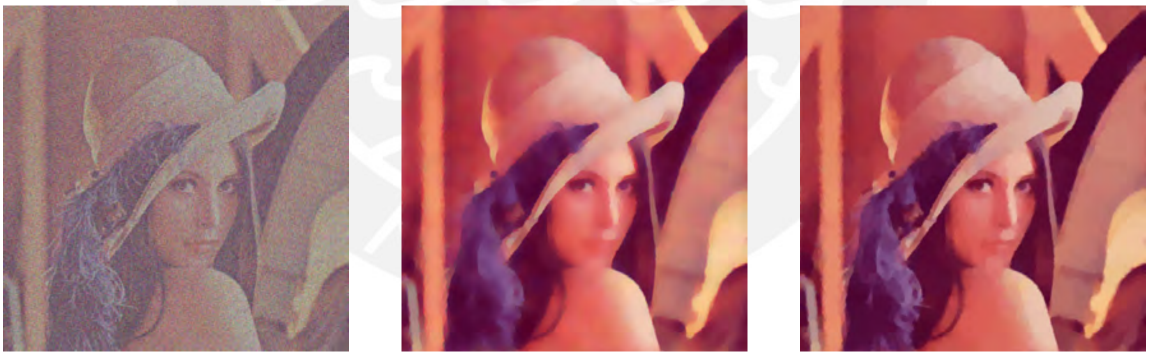


Figure 4.41: Cost function with respect to the iteration number and computation time in seconds, for denoising color Lena with Gaussian additive noise of $\sigma = 0.25$ using the APG and ADMM algorithms for $Q\ell_0$-grad with $\gamma = 0.01$

- $\gamma = \mathbf{0.04}$



(a) Noisy color image with $\sigma = 0.25$: PSNR=11.94, SNR=-0.67, SSIM=0.20

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=24.63, SNR=12.02, SSIM=0.64

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=26.12, SNR=13.51, SSIM=0.69

Figure 4.42: Reconstruction of a noisy color image with Gaussian additive noise of $\sigma = 0.25$ using $Q\ell_0$-grad with APG and ADMM, with $\gamma = 0.04$.

We can see in Fig. 4.42 how both APG and ADMM reconstructed the image with sharp edges and plain regions, furthermore, the low-amplitude structures were greatly reduced, and the obtained metrics with ADMM were the best ones. Fig. 4.43 show that, after changing γ to 0.04, ADMM did manage to converge the cost function to a minimum, even to a lower value than the one obtained with APG.

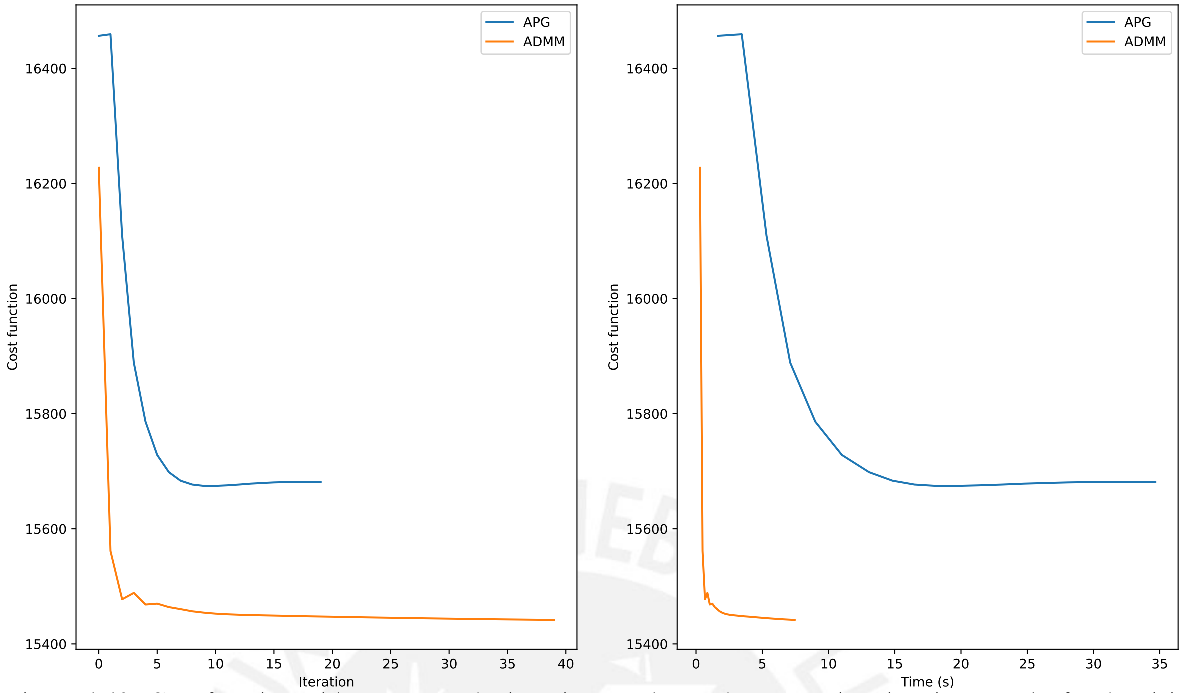


Figure 4.43: Cost function with respect to the iteration number and computation time in seconds, for denoising color Lena with Gaussian additive noise of $\sigma = 0.25$ using the APG and ADMM algorithms for $Q\ell_0$-grad with $\gamma = 0.04$.

2. **Deblurring with an average filter**

   We used the Barbara image filtered with a 5×5 average filter and Gaussian additive noise of $\sigma = 0.25$.

   - $\gamma = \mathbf{0.01}$

(a) Noisy color image with $\sigma = 0.25$: PSNR=11.85, SNR=-1.84, SSIM=0.15

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=23.79, SNR=10.10, SSIM=0.60

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=24.00, SNR=10.31, SSIM=0.61

Figure 4.44: Reconstruction of a noisy color image with Gaussian additive noise of $\sigma = 0.25$ and an average 5×5 filter using $Q\ell_0$-grad with APG and ADMM, with $\gamma = 0.01$.

Fig. 4.44 show that the difference between the reconstructed images obtained with ADMM and APG was similar to the denoising case: ADMM obtained images with more low amplitude structures. However, ADMM got better metrics, this happened because even though it kept unwanted noise, it also kept some fine details of the original image. We can see in the cost function comparisons given in Fig. 4.45, that both methods increased the function value but they converged quickly, nonetheless, ADMM still reached a greater value.

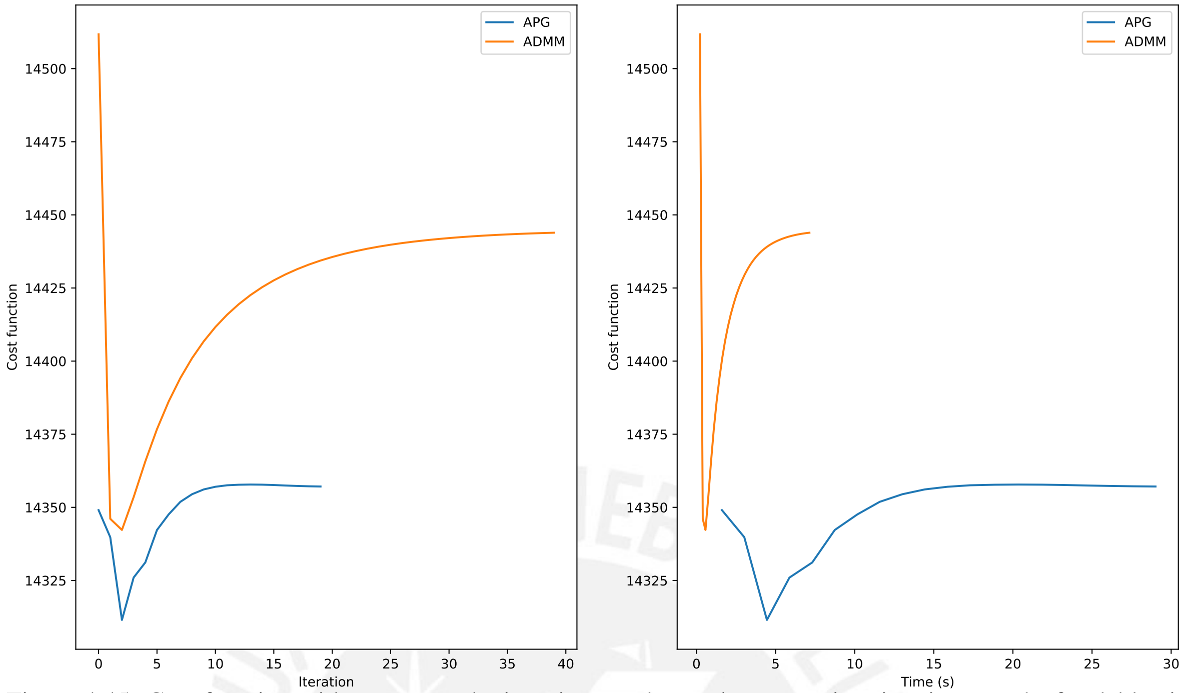


Figure 4.45: Cost function with respect to the iteration number and computation time in seconds, for deblurring color Barbara with a 5×5 average filter and Gaussian additive noise of $\sigma = 0.25$ using the APG and ADMM algorithms for $Q\ell_0$-grad with $\gamma = 0.01$.

- $\gamma = \mathbf{0.04}$

(a) Noisy color image with $\sigma = 0.25$: PSNR=11.85, SNR=-1.84, SSIM=0.15

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=23.29, SNR=9.60, SSIM=0.58

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=24.03, SNR=10.34, SSIM=0.63

Figure 4.46: Reconstruction of a noisy color image with Gaussian additive noise of $\sigma = 0.25$ and an average 5×5 filter using $Q\ell_0$-grad with APG and ADMM, with $\gamma = 0.04$.

In Fig. 4.46, we can see how, with $\gamma = 0.04$, ADMM improved its performance in reconstructing the noisy image, eliminating more low amplitude structures than with $\gamma = 0.01$. Meanwhile, the image obtained with APG became a little bit blurry but still had sharp edges. The graphs in Fig. 4.47 show that ADMM reached a considerable lower value than APG, explaining its better metrics.

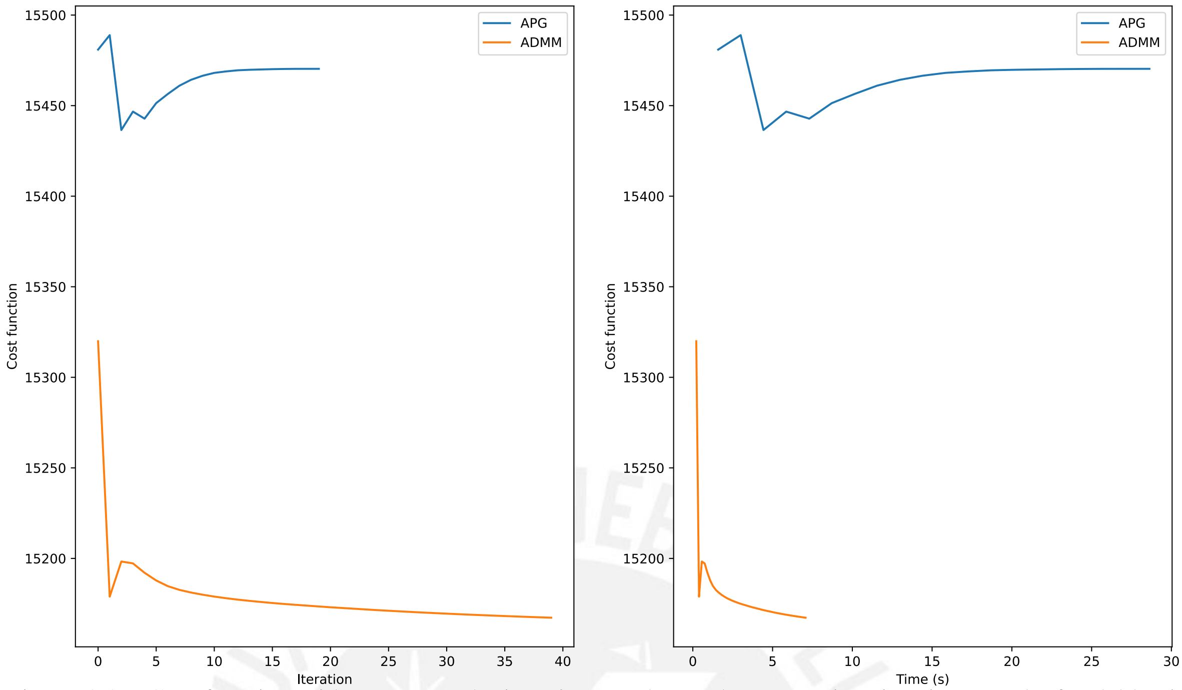


Figure 4.47: Cost function with respect to the iteration number and computation time in seconds, for deblurring color Barbara with a 5×5 average filter and Gaussian additive noise of $\sigma = 0.25$ using the APG and ADMM algorithms for $Q\ell_0$-grad with $\gamma = 0.04$.

3. **Deblurring with a Gaussian filter**

   We used the Peppers image filtered with a 9×9 Gaussian filter and Gaussian additive noise of $\sigma = 0.25$.

   - $\boldsymbol{\gamma = 0.01}$

(a) Noisy color image with $\sigma = 0.25$: PSNR=11.19, SNR=0.11, SSIM=0.12

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=23.48, SNR=12.40, SSIM=0.59

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=23.81, SNR=12.72, SSIM=0.61

Figure 4.48: Reconstruction of a noisy color image with Gaussian additive noise of $\sigma = 0.25$ and a Gaussian 9×9 filter using $Q\ell_0$-grad with APG and ADMM, with $\gamma = 0.01$.

In Fig. 4.48, the reconstructed image with APG obtained sharper edges and greatly reduced the amount of low amplitude structures, but the one obtained with ADMM achieved better PSNR, SNR and SSIM metrics; a similar behavior to the deblurring with an average filter case. However, we can see in Fig. 4.49 that both methods clearly decreased until getting to a minimum; ADMM stabilized faster than APG, but the latter obtained a lower value of cost function.

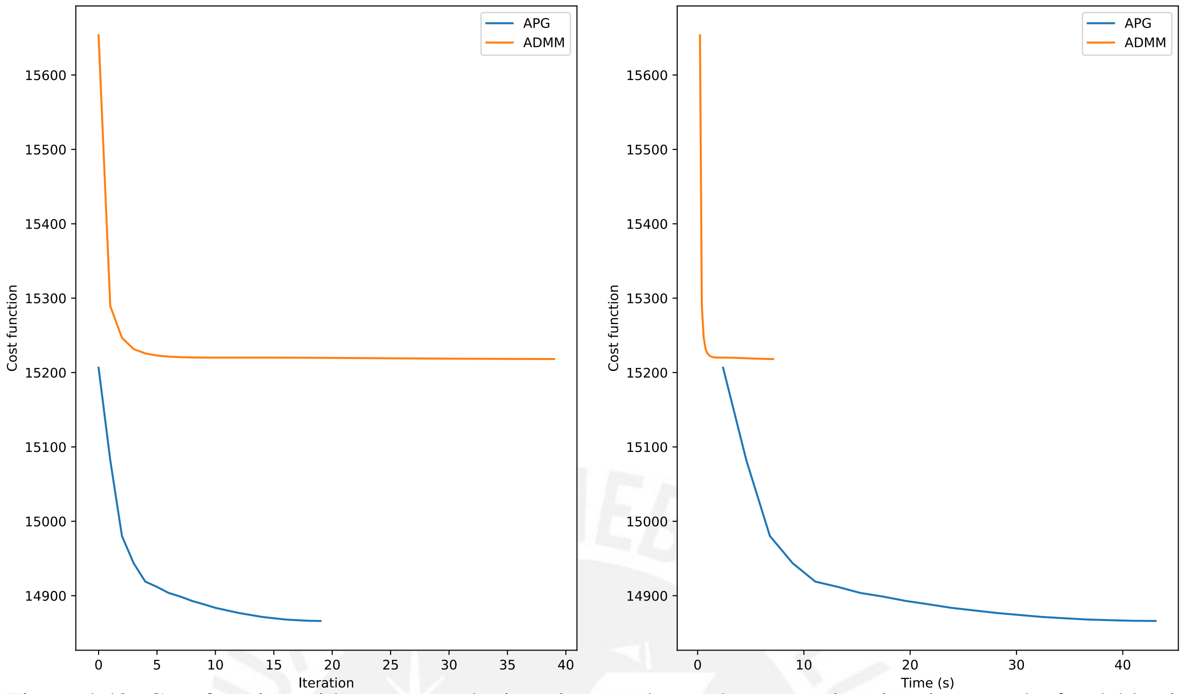


Figure 4.49: Cost function with respect to the iteration number and computation time in seconds, for deblurring color Peppers with a 9×9 Gaussian filter and Gaussian additive noise of $\sigma = 0.25$ using the APG and ADMM algorithms for $Q\ell_0$-grad with $\gamma = 0.01$.

- $\gamma = \mathbf{0.04}$

(a) Noisy color image with $\sigma = 0.25$: PSNR=11.19, SNR=0.11, SSIM=0.12

(b) Reconstructed image using $Q\ell_0$-grad with APG: PSNR=23.01, SNR=11.93, SSIM=0.58

(c) Reconstructed image using $Q\ell_0$-grad with ADMM: PSNR=23.81, SNR=12.73, SSIM=0.62

Figure 4.50: Reconstruction of a noisy color image with Gaussian additive noise of $\sigma = 0.25$ and a Gaussian 9×9 filter using $Q\ell_0$-grad with APG and ADMM, with $\gamma = 0.04$.

The difference between the reconstructed images in Fig. 4.50 was not as significant as with $\gamma = 0.01$, still they had the same behavior (APG better in removing low amplitude structures and ADMM better in metrics). Observing the graphs in Fig. 4.51, there was clearly an improvement, since now, the difference between the values of the cost functions of both methods was smaller than before.

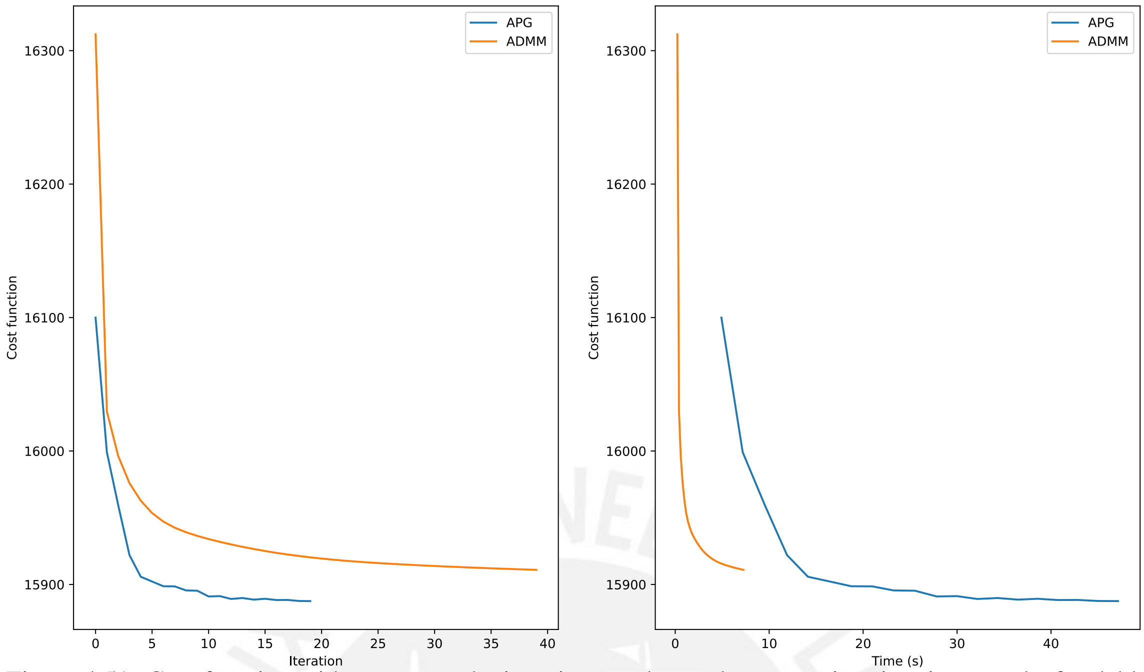


Figure 4.51: Cost function with respect to the iteration number and computation time in seconds, for deblurring color Peppers with a 9×9 Gaussian filter and Gaussian additive noise of $\sigma = 0.25$ using the APG and ADMM algorithms for $Q\ell_0$-grad with $\gamma = 0.04$.

In general, ADMM was much faster than APG for denoising and both cases of deblurring, however, when working with color, ADMM could diverge depending on the value of γ. Nonetheless, when ADMM did work, it achieved images very similar to the ones achieved with APG, even better, and in considerably lower time. From this we can say that ADMM could be used to solve $Q\ell_0$-grad with fairly good results, but APG was more robust to changes in the γ parameter. Also, it should be mentioned that the reason why APG took more time to have a value of the cost function was because it first had to run the initial 20 or 30 inner iterations.

4.5 Results for denoising and deblurring of one greyscale image with $Q\ell_0$-grad using different values of γ and a fixed λ

In the first three sections, we focused on the comparisons of our edge-preserving filtering model with other methods and used a fixed $\gamma = 0.01$ obtaining similar results than the ones with ℓ_2-TV. However, we also wanted to be able to control the level of detail that we could lose during both denoising and deblurring. For this, we needed to change the special parameter γ, this is why in this section we analyzed how, changing γ while keeping a fixed λ, influenced the output image and the metrics obtained.

We used one greyscale image for each of the three cases: denoising and deblurring with an average filter and a Gaussian filter. Similar to the sections before, we corrupted the images with additive Gaussian noise with different values of standard deviation $\sigma = \{0.05, 0.1, 0.25, 0.5\}$. We reconstructed the images using the APG implementation for $Q\ell_0$-grad with $\gamma = \{0.005, 0.0075, 0.01, 0.025, 0.05\}$, but maintaining the other parameters and the number of iterations; and then obtained their PSNR, SNR and SSIM metrics.

- **Denoising:** For this part we chose the greyscale Lena image. In Fig. 4.52 we can see the noisy image obtained with $\sigma = 0.1$ and how changes of γ changed the appearance of the reconstructed image; increasing γ sharpened the borders and reduced the noise considerably but the image lost some details, while decreasing it, kept the small details of the image but the edges were not sharp and it also maintained a great amount of noise. For the other values of σ the reconstructed images had a similar behavior, these images are presented in figures A.1 to A.3.

(a) $\sigma = 0.1$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure 4.52: Reconstructed images from a noisy greyscale Lena with $\sigma = 0.1$ for additive Gaussian noise using $Q\ell_0$-grad and different values of γ.

We repeated the tests 10 times for each of the different values of σ and obtained the graphs in Fig. 4.53 with their respective metrics means. Having $\gamma = 0.01$ as the middle value, we can see that for PSNR and SNR, whether γ increased or decreased, both of their values decreased; however, analyzing the SSIM for $\sigma = 0.5$ lowering γ decreased greatly its performance, while increasing γ to 0.025 slightly improved it. For the complete numeric values of the metrics, see Table A.20.

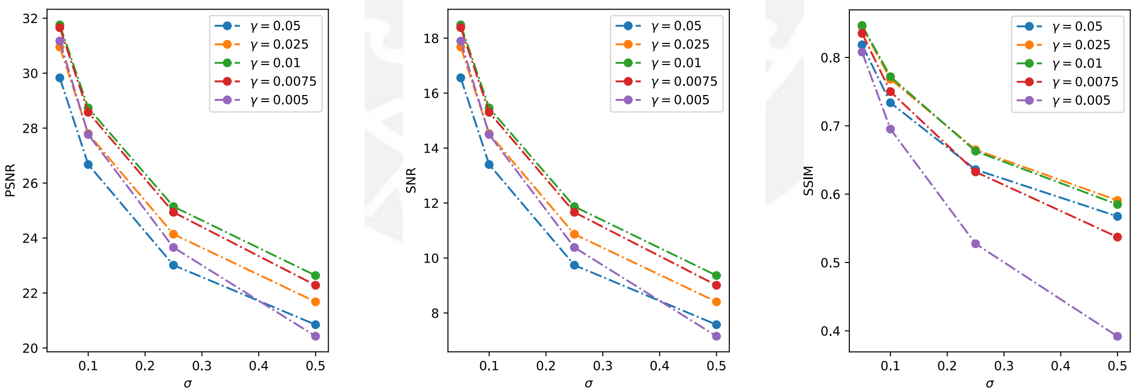


Figure 4.53: PSNR, SNR and SSIM according to σ for additive Gaussian noise, after 10 experiments, for denoising greyscale Lena with $Q\ell_0$-grad using 5 different values of γ.

- **Deblurring with an average filter:** We did the same as in the denoising section with the Barbara greyscale image filtered with a 5×5 average filter. The case for $\sigma = 0.1$ is presented in Fig. 4.54.

(a) $\sigma = 0.1$

(b) $\gamma = 0.05$

(c) $\gamma = 0.025$

(d) $\gamma = 0.01$

(e) $\gamma = 0.0075$

(f) $\gamma = 0.005$

Figure 4.54: Reconstructed images from a noisy greyscale Barbara with $\sigma = 0.1$ for additive Gaussian noise and an average 5×5 filter, using $Q\ell_0$-grad and different values of γ.

The reconstructed images behaved similar to those of denoising. In this case, for $\gamma = 0.05$ the tablecloth lines completely disappeared but the noise was practically gone, on the contrary for $\gamma = 0.005$ the lines could be seen perfectly but the amount of noise in the image was still large. For the reconstructed images of the other values of σ see Fig. A.4 to A.7. Analyzing the graphs in Fig. 4.55 we observe that for $\sigma \leq 0.1$, using a smaller γ got the best metrics, $\gamma = 0.005$ even having the best SSIM for the first two values of σ, however, as σ increased, the best performance was obtained with $\gamma = 0.01$, and both bigger and smaller γ obtained lower metrics. For a full detailed list of the metric values see Table A.21.

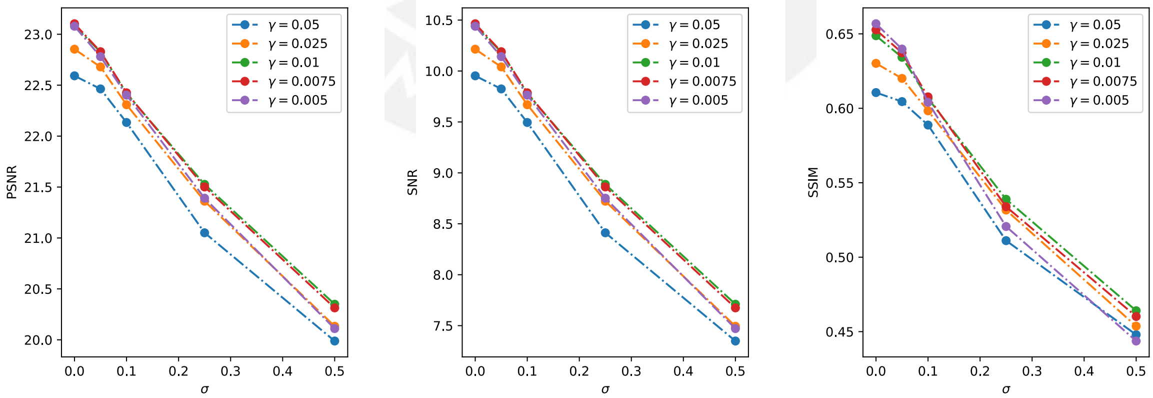
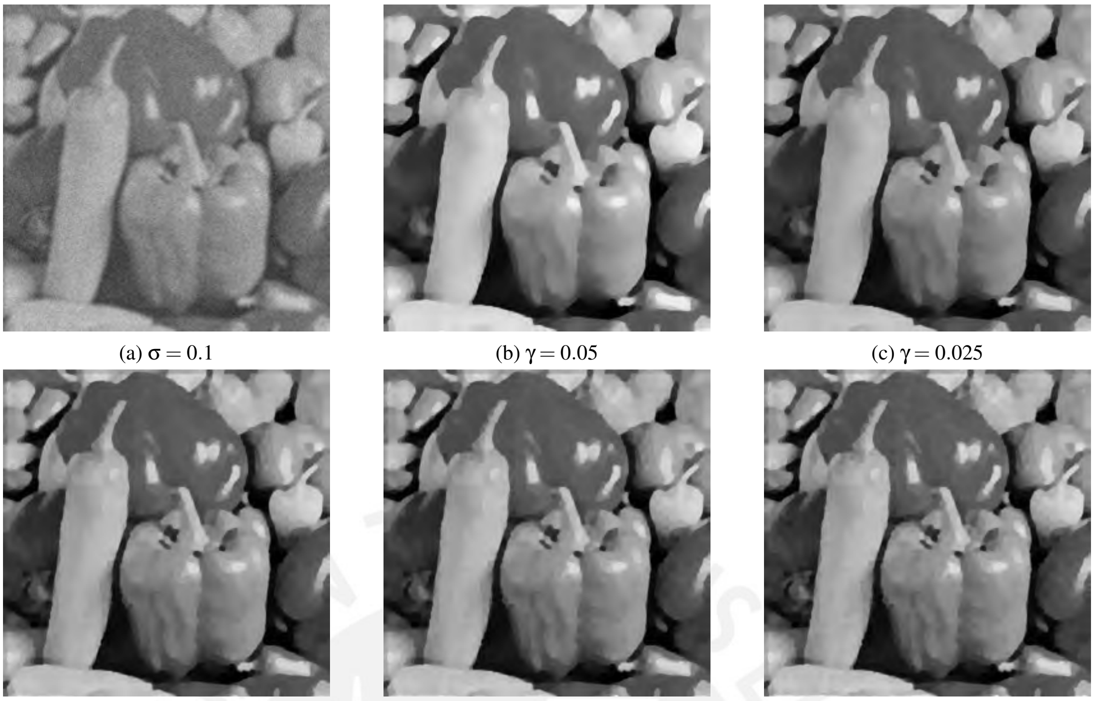


Figure 4.55: PSNR, SNR and SSIM according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Barbara with a 5×5 average filter using $Q\ell_0$-grad and 5 different values of γ.

- **Deblurring with a Gaussian filter:** For this case we worked with the Peppers greyscale image filtered with a 9×9 Gaussian filter, which was then corrupted with additive Gaussian noise.



(a) $\sigma = 0.1$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure 4.56: Reconstructed images from a noisy greyscale Peppers with $\sigma = 0.1$ for additive Gaussian noise and a Gaussian 9×9 filter, using $Q\ell_0$-grad and different values of γ.

In Fig. 4.56 we show the reconstructed images for $\sigma = 0.1$, here, a bigger γ gave sharper edges with plainer regions, while a smaller one returned edges which were not quite well defined but the image was more similar to the original. This is emphasized by the fact that the graphs in Fig. 4.57 show that $\gamma = 0.05$ obtained the worst metrics while $\gamma = 0.005$ achieved one of the best performances, there we can also see that for $\sigma = 0.5$, SSIM behaved differently than PSNR and SNR because $\gamma = 0.05$ had the lowest value followed close by $\gamma = 0.5$. The specific values that formed these graphs are presented in Table A.22 and the reconstructed images for $\sigma = \{0.0, 0.05, 0.25, 0.5\}$ are given in Fig A.8 to A.11.

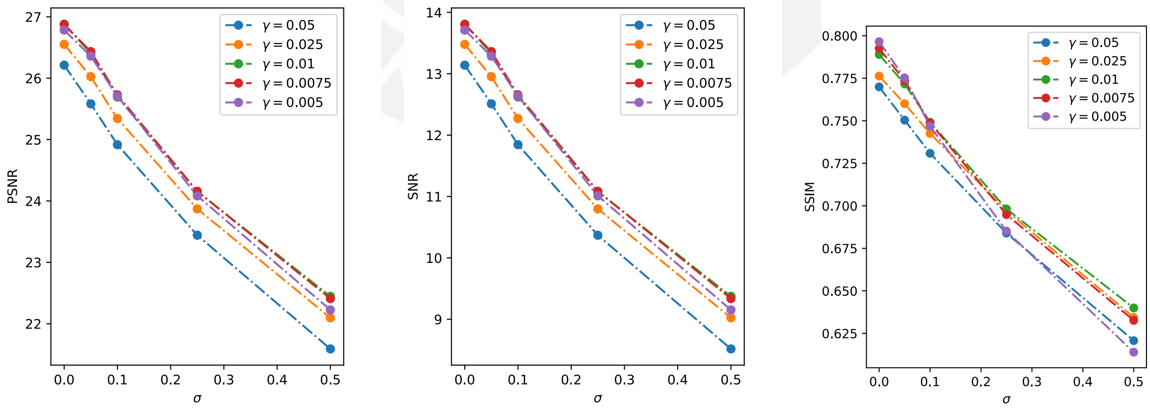


Figure 4.57: PSNR, SNR and SSIM according to σ for additive Gaussian noise, after 10 experiments, for deblurring greyscale Peppers with a 9×9 Gaussian filter using $Q\ell_0$-grad and 5 different values of γ (Table A.22).

After analyzing all the reconstructed images for the three cases, we can say that when we increased the value of γ the noise was significantly reduced, the borders were preserved but the image lost some details, when we decreased it, the borders were still preserved and the image kept the details of the original but it also maintained a considerable amount of noise. Furthermore, the parameter γ indeed controlled how close the quadratic envelope was to the real ℓ_0 norm of the gradient, meaning that for greater values of γ the results would be closer to the ones obtained with the ℓ_0-grad model, giving images with texture-free regions and sharp edges, but they could become blurry if the amount of noise was large.

Chapter 5

Conclusions and recommendations

5.1 Conclusions

We began this work by computing the quadratic envelope of the ℓ_0 norm of the gradient, however the obtained function had two different definitions depending on the actual value of the gradient (see Section 3.1), making it unsuitable to be used in a minimization problem similar to ℓ_2-TV or ℓ_0-grad. Due to this fact, we instead chose to use a relaxed form of such envelope, by doing this we developed a novel edge-preserving filtering model ($Q\ell_0$-grad).

To solve the resulting optimization problem of the $Q\ell_0$-grad model, we developed an APG-based algorithm, as well as an ADMM-based algorithm (see Section 3.2) and implemented them using Python; they achieved similar results when compared with each other, but ADMM was considerably faster and APG was more robust to changes in the γ parameter which controlled how close the initial quadratic envelope was to the ℓ_0 norm of the gradient. We used APG to make the comparisons with ℓ_2-TV and ℓ_0-grad for denoising and deblurring. The results showed that the $Q\ell_0$-grad model sharpened major edges while strongly attenuating textures; furthermore, when compared to the ℓ_0-grad model, it reconstructed images with better qualitative characteristics (flat, texture-free regions, with smooth changes between adjacent regions, even in the large noise scenario) and better metrics, obtaining an improvement on average of +0.96 dB SNR , +0.96 dB PSNR and +0.03 SSIM.

In the presented results, our method and ℓ_2-TV had very similar metrics, but the main difference between $Q\ell_0$-grad and existing models was the introduction of the parameter γ; concluding that: increasing its value would reduce a great amount of noise and some details, giving images that are similar to the ones obtained with the ℓ_0-grad method, but that look smoother and have no abrupt changes between adjacent regions; decreasing its value would make the results be closer to ℓ_2-TV, because the quadratic envelope would be closer to the TV-norm, with images that do not lose too much detail but still keep some noise.

An early version of the model was presented in the paper *Fast gradient-based algorithm for a quadratic envelope relaxation of the ℓ_0 gradient regularization* [21], which had an approximate solution to the quadratic envelope of the ℓ_0 norm of the gradient and an APG-based solution to the minimization problem of the $Q\ell_0$-grad model.

5.2 Recommendations

- The value of γ should be optimized according to the desired application. If it is required to greatly reduce the noise, even if it means losing some details, then its value should be higher, but if keeping the details is more important than reducing the noise, γ should be lower.
- One of the applications of ℓ_0-grad is X-ray computed tomography reconstruction; and since our method and ℓ_0-grad share some similar characteristics, further investigations could be made comparing the performance of the $Q\ell_0$-grad model with other state of the art X-ray CT image reconstruction methods.

Appendix A

Annexes

Table A.1: Averages and standard deviations of PSNR, SNR and SSIM of the noisy greyscale images without processing according to the filter used (identity, Gaussian and average) and the standard deviation (σ) of the Gaussian additive noise, after 10 experiments.

		IDENTITY			**GAUSSIAN**			**AVERAGE**		
σ		PSNR	SNR	SSIM	PSNR	SNR	SSIM	PSNR	SNR	SSIM
0.0	Lena	-	-	-	25.03	11.76	0.73	28.4	15.12	0.84
	Barbara	-	-	-	21.96	9.32	0.59	22.66	10.02	0.67
	Mandrill	-	-	-	19.33	4.59	0.34	20.37	5.63	0.51
	Peppers	-	-	-	25.5	12.43	0.79	28.68	15.61	0.89
0.05	Lena	24.77±0.018	11.50±0.018	0.55±0.004	21.90±0.007	8.62±0.007	0.34±0.002	23.21±0.008	9.94±0.008	0.42±0.003
	Barbara	25.27±0.010	12.64±0.010	0.67±0.002	20.30±0.007	7.65±0.007	0.32±0.002	20.76±0.007	8.12±0.007	0.38±0.002
	Mandrill	25.15±0.011	10.41±0.011	0.80±0.001	18.32±0.006	3.58±0.006	0.20±0.001	19.12±0.008	4.39±0.008	0.33±0.002
	Peppers	25.60±0.012	12.53±0.012	0.56±0.005	22.54±0.012	9.47±0.012	0.39±0.003	23.87±0.013	10.80±0.013	0.46±0.004
0.1	Lena	18.75±0.011	5.48±0.011	0.35±0.003	17.84±0.009	4.57±0.009	0.21±0.004	18.31±0.015	5.04±0.015	0.26±0.003
	Barbara	19.25±0.008	6.61±0.008	0.47±0.002	17.39±0.010	4.75±0.010	0.21±0.003	17.62±0.010	4.98±0.011	0.25±0.003
	Mandrill	19.14±0.005	4.40±0.005	0.61±0.001	16.22±0.010	1.49±0.010	0.13±0.002	16.70±0.009	1.96±0.009	0.22±0.001
	Peppers	19.58±0.010	6.51±0.010	0.35±0.005	18.60±0.011	5.52±0.011	0.24±0.005	19.07±0.009	6.00±0.009	0.29±0.003
0.25	Lena	10.80±0.013	-2.47±0.013	0.19±0.003	10.63±0.014	-2.64±0.014	0.11±0.006	10.73±.0012	-2.55±0.012	0.14±0.004
	Barbara	11.30±0.011	-1.35±0.011	0.25±0.003	10.94±0.015	-1.70±0.015	0.11±0.002	10.98±0.014	-1.66±0.014	0.13±0.003
	Mandrill	11.19±0.12	-3.55±0.012	0.32±0.001	10.56±0.014	-4.17±0.014	0.08±0.003	10.70±0.012	-4.05±0.012	0.12±0.003
	Peppers	11.62±0.013	-1.45±0.013	0.19±0.004	11.45±0.013	-1.63±0.013	0.13±0.004	11.54±0.016	-1.54±0.016	0.15±0.004
0.5	Lena	4.78±0.014	-8.50±0.014	0.11±0.003	4.74±0.010	-8.53±0.010	0.08±0.004	4.76±0.010	-8.51±0.010	0.09±0.003
	Barbara	5.28±0.014	-7.36±0.014	0.15±0.003	5.19±0.010	-7.45±0.010	0.08±0.002	5.19±0.013	-7.45±0.013	0.09±0.002
	Mandrill	5.16±0.009	-9.58±0.009	0.19±0.003	5.00±0.007	-9.74±0.007	0.06±0.002	5.03±0.018	-9.70±0.018	0.08±0.003
	Peppers	5.61±0.015	-7.46±0.015	0.12±0.005	5.55±0.008	-7.52±0.008	0.09±0.007	5.58±0.006	-7.49±0.006	0.010±0.003

Table A.2: Averages and standard deviations of PSNR, SNR and SSIM of the noisy color images without processing according to the filter used (identity, Gaussian and average) and the standard deviation (σ) of the Gaussian additive noise, after 10 experiments.

		IDENTITY			**GAUSSIAN**			**AVERAGE**		
σ		PSNR	SNR	SSIM	PSNR	SNR	SSIM	PSNR	SNR	SSIM
0.0	Lena	-	-	-	26.22	13.6	0.71	29.17	16.55	0.81
	Barbara	-	-	-	24.22	10.53	0.63	26.1	12.41	0.74
	Mandrill	-	-	-	19.7	6.56	0.34	20.74	7.6	0.48
	Peppers	-	-	-	24.6	13.53	0.67	27.23	16.15	0.76
0.05	Lena	25.92±0.006	13.30±0.006	0.60±0.003	23.06±0.004	10.44±0.004	0.38±0.003	24.25±0.007	11.62±0.007	0.45±0.002
	Barbara	25.99±0.006	12.30±0.006	0.66±0.003	22.00±0.005	8.31±0.005	0.36±0.002	23.03±0.006	9.34±0.006	0.44±0.002
	Mandrill	26.02±0.007	12.88±0.007	0.81±0.001	18.79±0.003	5.65±0.003	0.22±0.001	19.61±0.004	6.47±0.004	0.33±0.001
	Peppers	25.38±0.006	14.30±0.006	0.58±0.002	21.96±0.006	10.88±0.006	0.37±0.003	23.20±0.008	12.12±0.008	0.42±0.004
0.1	Lena	19.90±0.007	7.28±0.007	0.39±0.003	18.98±0.006	6.37±0.006	0.24±0.002	19.41±0.006	6.80±0.006	0.28±0.005
	Barbara	19.97±0.006	6.28±0.006	0.45±0.004	18.58±0.007	4.89±0.007	0.23±0.004	19.02±0.005	5.33±0.005	0.28±0.004
	Mandrill	20.00±0.009	6.86±0.009	0.62±0.004	16.83±0.007	3.70±0.007	0.15±0.0001	17.35±0.005	4.21±0.005	0.23±0.001
	Peppers	19.36±0.007	8.28±0.007	0.37±0.004	18.23±0.007	7.15±0.007	0.23±0.003	18.71±0.008	7.62±0.008	0.26±0.003
0.25	Lena	11.94±0.007	-0.67±0.007	0.20±0.006	11.78±0.007	-0.83±0.007	0.013±0.003	11.86±0.006	-0.76±0.006	0.15±0.003
	Barbara	12.01±0.006	-1.68±0.006	0.23±0.003	11.75±0.004	-1.94±0.004	0.12±0.005	11.84±0.009	-1.85±0.009	0.15±0.002
	Mandrill	12.05±0.004	-1.09±0.004	0.34±0.001	11.35±0.008	-1.78±0.008	0.09±0.002	11.50±0.005	-1.65±0.005	0.13±0.002
	Peppers	11.40±0.010	0.32±0.010	0.19±0.002	11.20±0.004	0.12±0.004	0.12±0.003	11.30±0.005	0.21±0.005	0.14±0.002
0.5	Lena	5.92±0.006	-6.70±0.006	0.13±0.005	5.88±0.007	-6.75±0.007	0.09±0.005	5.89±0.006	-6.72±0.006	0.10±0.003
	Barbara	5.99±0.006	-7.70±0.006	0.15±0.003	5.92±0.008	-7.77±0.008	0.09±0.004	5.95±0.006	-7.74±0.006	0.10±0.004
	Mandrill	6.02±0.008	-7.12±0.008	0.20±0.002	5.84±0.005	-7.30±0.005	0.07±0.002	5.89±0.006	-7.26±0.006	0.08±0.001
	Peppers	5.39±0.009	-5.69±0.010	0.12±0.003	5.33±0.006	-5.75±0.006	0.08±0.003	5.36±0.008	-5.72±0.008	0.10±0.003

A Tables for denoising

Table A.3: Bounds of λ used in the Fibonacci searches (Section 2.10 Fibonacci search) for finding the optimal value that maximizes the SNR metric for the denoising tests, according to the standard deviation (σ) of the additive Gaussian noise and method.

		Greyscale		**Color**	
σ	**Method**	**Lower bound**	**Upper bound**	**Lower bound**	**Upper bound**
0.05	ℓ_2-TV	0.007	0.04	0.008	0.1
	ℓ_0-grad	0.0003	0.004	0.003	0.05
	$Q\ell_0$-grad	0.01	0.1	0.01	0.1
0.1	ℓ_2-TV	0.01	0.15	0.03	0.25
	ℓ_0-grad	0.002	0.015	0.01	0.15
	$Q\ell_0$-grad	0.08	0.5	0.1	1.0
0.25	ℓ_2-TV	0.15	0.3	0.2	0.6
	ℓ_0-grad	0.03	0.15	0.1	1.0
	$Q\ell_0$-grad	1.5	3.5	2.0	6.0
0.5	ℓ_2-TV	0.4	0.6	0.6	1.0
	ℓ_0-grad	0.2	0.4	1.0	2.0
	$Q\ell_0$-grad	14.0	16.0	12.0	16.0

Table A.4: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad, ℓ_2-**TV** and BM3D) for denoising with $\sigma = 0.05$ of standard deviation for additive Gaussian noise, after 10 experiments.

	Image	**Method**	**PSNR**	**SNR**	**SSIM**	**Time**
GREYSCALE	Lena	ℓ_2-TV	31.85±0.009	18.58±0.009	0.85±0.001	1.62±0.420
		ℓ_0-grad	30.56±0.019	17.29±0.019	0.80±0.001	1.41±0.052
		$Q\ell_0$-grad	31.77±.009	18.45±0.009	0.85±0.002	8.56±0.307
		BM3D	33.70±0.021	20.43±0.021	0.88±0.004	7.46±0.169
	Barbara	ℓ_2-TV	28.02±0.016	15.38±0.016	0.80±0.001	1.45±0.028
		ℓ_0-grad	27.72±0.018	15.08±0.018	0.78±0.001	1.39±0.029
		$Q\ell_0$-grad	28.85±0.015	16.21±0.015	0.83±0.001	8.36±0.183
		BM3D	33.02±0.016	20.38±0.016	0.92±0.001	7.41±0.097
	Mandrill	ℓ_2-TV	26.97±0.013	12.23±0.013	0.84±0.000	1.44±0.023
		ℓ_0-grad	26.92±0.012	12.18±0.012	0.84±0.000	1.36±0.026
		$Q\ell_0$-grad	27.41±0.014	12.67±0.014	0.85±0.002	8.31±0.147
		BM3D	28.25±0.0007	13.51±0.007	0.87±0.002	6.95±0.014
	Peppers	ℓ_2-TV	33.15±0.026	20.08±0.026	0.89±0.001	1.45±0.017
		ℓ_0-grad	31.79±0.019	18.71±0.019	0.81±0.001	1.38±0.021
		$Q\ell_0$-grad	33.37±0.026	20.29±0.026	0.90±0.001	8.45±0.156
		BM3D	35.29±0.021	22.21±0.021	0.92±0.000	7.40±0.024
COLOR	Lena	ℓ_2-TV	32.70±0.006	20.09±0.006	0.85±0.000	6.03±0.191
		ℓ_0-grad	31.27±0.10	18.64±0.010	0.80±0.001	5.91±0.077
		$Q\ell_0$-grad	32.38±0.007	19.77±0.007	0.84±0.000	27.91±0.347
		BM3D	33.77±0.010	21.16±0.010	0.86±0.000	11.95±0.037
	Barbara	ℓ_2-TV	30.61±0.008	16.92±0.008	0.86±0.000	5.95±0.028
		ℓ_0-grad	29.72±0.011	16.03±0.011	0.82±0.001	5.88±0.038
		$Q\ell_0$-grad	30.47±0.009	16.78±0.009	0.85±0.000	27.60±0.188
		BM3D	37.67±0.011	20.98±0.011	0.94±0.000	11.91±0.030
	Mandrill	ℓ_2-TV	27.59±0.000	14.45±0.005	0.85±0.000	5.94±0.026
		ℓ_0-grad	27.57±0.006	14.43±0.006	0.85±0.000	5.85±0.035
		$Q\ell_0$-grad	28.04±0.005	14.90±0.005	0.87±0.000	27.13±0.084
		BM3D	28.70±0.007	15.57±0.007	0.89±0.000	11.36±0.246
	Peppers	ℓ_2-TV	31.34±0.011	20.26±0.011	0.81±0.001	5.91±0.018
		ℓ_0-grad	30.12±0.010	19.04±0.010	0.76±0.000	5.84±0.023
		$Q\ell_0$-grad	31.52±0.010	20.45±0.010	0.80±0.001	27.78±0.136
		BM3D	32.32±0.011	21.24±0.011	0.88±0.001	11.96±0.030

Table A.5: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad, ℓ_2-**TV** and BM3D) for denoising with $\sigma = 0.1$ of standard deviation for additive Gaussian noise, after 10 experiments.

	Image		**PSNR**	**SNR**	**SSIM**	**Time**
GREYSCALE	Lena	ℓ_2-TV	28.79±0.020	15.52±0.020	0.77±0.004	1.45±0.021
		ℓ_0-grad	26.80±0.019	13.52±0.019	0.70±0.001	1.36±0.023
		$Q\ell_0$-grad	28.71±0.015	15.44±0.015	0.77±0.006	8.68±0.342
		BM3D	30.75±0.026	17.47±0.026	0.82±0.005	7.42±0.015
	Barbara	ℓ_2-TV	24.79±0.021	12.15±0.021	0.68±0.002	1.45±0.022
		ℓ_0-grad	23.52±0.022	10.88±0.022	0.64±0.002	1.39±0.011
		$Q\ell_0$-grad	24.95±0.017	12.31±0.017	0.70±0.001	8.67±0.275
		BM3D	29.84±0.051	17.20±0.051	0.87±0.003	7.42±0.037
	Mandrill	ℓ_2-TV	23.49±0.015	8.75±0.015	0.70±0.002	1.42±0.027
		ℓ_0-grad	22.64±0.016	7.90±0.016	0.67±0.001	1.37±0.018
		$Q\ell_0$-grad	23.49±0.012	8.76±0.012	0.69±0.002	8.69±0.245
		BM3D	24.57±0.015	9.84±0.015	0.74±0.003	7.24±0.033
	Peppers	ℓ_2-TV	30.01±0.026	16.94±0.026	0.84±0.002	1.46±0.022
		ℓ_0-grad	27.95±0.030	14.88±0.030	0.77±0.001	1.35±0.025
		$Q\ell_0$-grad	30.08±0.035	17.01±0.035	0.84±0.002	8.84±0.129
		BM3D	32.25±0.036	19.18±0.036	0.88±0.001	7.50±0.181
COLOR	Lena	ℓ_2-TV	30.09±0.012	17.48±0.012	0.79±0.001	5.96±0.101
		ℓ_0-grad	27.87±0.019	15.25±0.019	0.71±0.002	5.87±0.058
		$Q\ell_0$-grad	29.59±0.010	16.97±0.010	0.77±0.001	27.77±0.071
		BM3D	31.25±0.012	18.63±0.012	0.81±0.000	11.99±0.037
	Barbara	ℓ_2-TV	27.74±0.015	14.05±0.015	0.77±0.001	5.93±0.038
		ℓ_0-grad	26.05±0.011	12.36±0.011	0.70±0.001	5.86±0.044
		$Q\ell_0$-grad	27.23±0.014	13.54±0.014	0.75±0.001	27.77±0.162
		BM3D	31.20±0.030	17.51±0.030	0.88±0.001	11.94±0.043
	Mandrill	ℓ_2-TV	24.26±0.010	11.12±0.010	0.73±0.001	5.90±0.021
		ℓ_0-grad	23.49±0.006	10.35±0.006	0.69±0.001	5.83±0.019
		$Q\ell_0$-grad	24.07±0.010	10.90±0.010	0.72±0.001	27.49±0.156
		BM3D	25.00±0.010	11.87±0.010	0.75±0.001	11.69±0.039
	Peppers	ℓ_2-TV	29.01±0.012	17.93±0.012	0.75±0.001	5.95±0.104
		ℓ_0-grad	26.92±0.016	15.84±0.016	0.67±0.001	5.82±0.025
		$Q\ell_0$-grad	28.92±0.015	17.84±0.014	0.74±0.001	27.75±0.076
		BM3D	30.21±0.014	19.13±0.014	0.77±0.001	12.02±0.033

Table A.6: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad, ℓ_2-**TV** and BM3D) for denoising with $\sigma = 0.25$ of standard deviation for additive Gaussian noise, after 10 experiments.

	Image		**PSNR**	**SNR**	**SSIM**	**Time**
	Lena	ℓ_2-TV	25.17±0.033	11.89±0.033	0.66±0.003	1.48±0.027
		ℓ_0-grad	22.72±0.046	9.45±0.046	0.59±0.002	1.36±0.023
		$Q\ell_0$-grad	25.15±0.031	11.87±0.031	0.66±0.002	8.39±0.159
G		BM3D	26.53±0.030	13.25±0.030	0.72±0.002	7.53±0.076
R	Barbara	ℓ_2-TV	21.97±0.011	9.33±0.011	0.56±0.002	1.48±0.029
E		ℓ_0-grad	20.53±0.026	7.89±0.026	0.48±0.002	1.36±0.021
Y		$Q\ell_0$-grad	21.95±0.010	9.31±0.010	0.56±0.002	8.42±0.152
S		BM3D	25.01±0.030	12.37±0.030	0.71±0.003	7.56±0.070
C	Mandrill	ℓ_2-TV	20.11±0.014	5.37±0.014	0.44±0.002	1.45±0.025
A		ℓ_0-grad	18.77±0.020	4.03±0.020	0.31±0.002	1.35±0.020
L		$Q\ell_0$-grad	20.07±0.014	5.33±0.014	0.43±0.002	8.47±0.166
E		BM3D	20.73±0.024	5.99±0.024	0.48±0.004	7.61±0.075
	Peppers	ℓ_2-TV	26013±0.044	13.06±0.044	0.74±0.002	1.46±0.020
		ℓ_0-grad	23.50±0.051	10.42±0.051	0.66±0.002	1.35±0.024
		$Q\ell_0$-grad	26.13±0.042	13.06±0.042	0.75±0.002	8.41±0.177
		BM3D	27.75±0.054	14.69±0.054	0.78±0.001	7.54±0.032
	Lena	ℓ_2-TV	26.85±0.020	14.24±0.020	0.70±0.001	5.91±0.030
		ℓ_0-grad	24.30±0.027	11.68±0.027	0.61±0.001	5.82±0.024
		$Q\ell_0$-grad	26.25±0.022	13.63±0.022	0.67±0.001	27.52±0.113
		BM3D	27.52±0.022	14.90±0.022	0.72±0.001	12.05±0.028
	Barbara	ℓ_2-TV	24.83±0.015	11.14±0.015	0.66±0.001	5.91±0.018
C		ℓ_0-grad	22.80±0.027	9.10±0.027	0.56±0.001	5.83±0.027
O		$Q\ell_0$-grad	24.28±0.013	10.59±0.013	0.62±0.001	27.57±0.047
L		BM3D	26.20±0.038	12.51±0.038	0.72±0.002	12.08±0.213
O	Mandrill	ℓ_2-TV	20.93±0.011	7.08±0.011	0.49±0.001	5.95±0.042
R		ℓ_0-grad	19.57±0.012	6.43±0.012	0.37±0.002	5.86±0.023
		$Q\ell_0$-grad	20.58±0.009	7.44±0.009	0.45±0.001	27.83±0.168
		BM3D	21.02±0.018	7.89±0.018	0.46±0.002	12.08±0.025
	Peppers	ℓ_2-TV	25.72±0.025	14.65±0.025	0.66±0.001	5.95±0.028
		ℓ_0-grad	23.06±0.041	11.98±0.041	0.57±0.001	5.84±0.037
		$Q\ell_0$-grad	25.41±0.022	14.33±0.022	0.64±0.001	27.39±0.059
		BM3D	26.67±0.033	15.59±0.033	0.68±0.001	12.13±0.279

Table A.7: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad, ℓ_2-**TV** and BM3D) for denoising with $\sigma = 0.5$ of standard deviation for additive Gaussian noise, after 10 experiments.

	Image		**PSNR**	**SNR**	**SSIM**	**Time**
GREYSCALE	Lena	ℓ_2-TV	22.72±0.044	9.44±0.044	0.59±0.002	1.48±0.023
		ℓ_0-grad	20.05±0.075	6.78±0.075	0.52±0.004	1.37±0.014
		$Q\ell_0$-grad	22.68±0.041	9.41±0.041	0.59±0.002	8.37±0.113
		BM3D	23.12±0.038	9.85±0.038	0.59±0.003	7.65±0.239
	Barbara	ℓ_2-TV	20.51±0.032	7.87±0.032	0.47±0.002	1.47±0.017
		ℓ_0-grad	18.57±0.066	5.93±0.066	0.40±0.004	1.36±0.020
		$Q\ell_0$-grad	20.52±0.031	7.88±0.031	0.48±0.002	8.42±0.116
		BM3D	21.38±0.055	8.74±0.054	0.53±0.005	7.55±0.011
	Mandrill	ℓ_2-TV	18.77±0.012	4.00±0.012	0.28±0.001	1.48±0.018
		ℓ_0-grad	17.54±0.029	2.80±0.029	0.21±0.002	1.35±0.022
		$Q\ell_0$-grad	18.73±0.013	3.99±0.013	0.27±0.001	8.26±0.089
		BM3D	18.95±0.028	4.21±0.028	0.31±0.003	7.71±0.263
	Peppers	ℓ_2-TV	23.38±0.026	10.31±0.026	0.66±0.002	1.46±0.024
		ℓ_0-grad	20.45±0.062	7.38±0.062	0.58±0.003	1.35±0.023
		$Q\ell_0$-grad	23.39±0.028	10.32±0.028	0.67±0.003	8.33±0.168
		BM3D	23.85±0.100	10.78±0.100	0.64±0.004	7.56±0.027
COLOR	Lena	ℓ_2-TV	24.56±0.046	11.95±0.046	0.63±0.002	5.94±0.030
		ℓ_0-grad	22.20±0.040	9.58±0.040	0.56±0.002	5.85±0.088
		$Q\ell_0$-grad	23.91±0.044	11.29±0.044	0.60±0.002	27.24±0.075
		BM3D	24.60±0.048	11.98±0.048	0.62±0.002	12.01±0.035
	Barbara	ℓ_2-TV	22.92±0.028	9.23±0.028	0.57±0.001	5.95±0.027
		ℓ_0-grad	20.67±0.043	6.98±0.043	0.48±0.002	5.83±0.023
		$Q\ell_0$-grad	22.38±0.023	8.69±0.023	0.54±0.001	27.34±0.172
		BM3D	23.21±0.053	9.52±0.053	0.58±0.003	12.04±0.019
	Mandrill	ℓ_2-TV	19.42±0.006	6.28±0.006	0.33±0.001	5.93±0.024
		ℓ_0-grad	18.17±0.016	5.03±0.016	0.24±0.001	5.84±0.061
		$Q\ell_0$-grad	19.20±0.007	6.06±0.007	0.31±0.001	27.38±0.102
		BM3D	19.28±0.012	6.15±0.012	0.31±0.001	12.10±0.029
	Peppers	ℓ_2-TV	23.32±0.024	12.24±0.024	0.59±0.002	5.92±0.023
		ℓ_0-grad	20.56±0.030	9.48±0.029	0.50±0.002	5.83±0.037
		$Q\ell_0$-grad	22.91±0.023	11.84±0.023	0.57±0.002	27.34±0.161
		BM3D	23.35±0.040	12.27±0.040	0.59±0.002	12.07±0.018

B Tables for deblurring with a 5×5 average filter

Table A.8: Bounds of λ used in the Fibonacci searches (Section 2.10 Fibonacci search) for finding the optimal value that maximizes the SNR metric for the deblurring with a 5×5 average filter tests, according to the standard deviation (σ) of the additive Gaussian noise and method.

		Greyscale		**Color**	
σ	**Method**	**Lower bound**	**Upper bound**	**Lower bound**	**Upper bound**
0.0	ℓ_2-TV	0.0005	0.05	0.0	0.03
	ℓ_0-grad	0.00005	0.005	0.0	0.005
	$Q\ell_0$-grad	0.005	0.05	0.0	0.05
0.05	ℓ_2-TV	0.0005	0.05	0.001	0.1
	ℓ_0-grad	0.0001	0.0015	0.0001	0.05
	$Q\ell_0$-grad	0.005	0.08	0.005	0.1
0.1	ℓ_2-TV	0.01	0.1	0.01	0.2
	ℓ_0-grad	0.0007	0.01	0.003	0.1
	$Q\ell_0$-grad	0.01	0.15	0.01	0.5
0.25	ℓ_2-TV	0.05	0.25	0.1	0.5
	ℓ_0-grad	0.01	0.1	0.05	0.4
	$Q\ell_0$-grad	0.5	1.5	0.3	3.0
0.5	ℓ_2-TV	0.2	0.6	0.2	1.0
	ℓ_0-grad	0.05	0.15	0.3	1.0
	$Q\ell_0$-grad	3.5	6.5	2.5	10.0

Table A.9: PSNR, SNR, SSIM and averages and standard deviations of computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 5×5 average filter, after 10 experiments.

	Image	Method	PSNR	SNR	SSIM	Time
G R E Y S C A L E	Lena	ℓ_2-TV	27.72	14.45	0.81	1.47±0.025
		ℓ_0-grad	27.67	14.39	0.81	1.41±0.008
		$Q\ell_0$-grad	28.54	15.27	0.79	8.47±0.202
	Barbara	ℓ_2-TV	22.90	10.26	0.69	1.52±0.043
		ℓ_0-grad	22.88	10.25	0.69	1.46±0.038
		$Q\ell_0$-grad	23.08	10.44	0.67	8.74±0.240
	Mandrill	ℓ_2-TV	20.71	5.97	0.56	1.48±0.008
		ℓ_0-grad	20.70	5.96	0.56	1.41±0.005
		$Q\ell_0$-grad	20.54	5.80	0.49	8.51±0.082
	Peppers	ℓ_2-TV	30.08	17.01	0.89	1.47±0.003
		ℓ_0-grad	29.93	16.80	0.89	1.41±0.004
		$Q\ell_0$-grad	29.00	15.93	0.86	8.42±0.092
C O L O R	Lena	ℓ_2-TV	28.37	15.75	0.80	6.16±0.070
		ℓ_0-grad	28.37	15.75	0.80	6.07±0.056
		$Q\ell_0$-grad	29.15	16.54	0.77	28.22±0.175
	Barbara	ℓ_2-TV	26.21	12.52	0.77	6.11±0.023
		ℓ_0-grad	26.21	12.52	0.77	6.02±0.042
		$Q\ell_0$-grad	26.15	12.46	0.71	28.02±0.089
	Mandrill	ℓ_2-TV	21.16	8.02	0.56	6.18±0.067
		ℓ_0-grad	21.16	8.02	0.56	6.06±0.019
		$Q\ell_0$-grad	21.04	7.91	0.54	28.27±0.053
	Peppers	ℓ_2-TV	27.48	16.40	0.76	6.14±0.020
		ℓ_0-grad	27.39	16.31	0.77	6.04±0.023
		$Q\ell_0$-grad	27.76	16.68	0.78	28.09±0.095

Table A.10: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 5×5 average filter and $\sigma = 0.05$ of standard deviation for additive Gaussian noise, after 10 experiments.

	Image	Method	PSNR	SNR	SSIM	Time
GREYSCALE	Lena	ℓ_2-TV	27.16±0.013	13.89±0.013	0.77±0.001	1.50±0.026
		ℓ_0-grad	26.83±0.018	13.55±0.018	0.75±0.004	1.44±0.017
		$Q\ell_0$-grad	27.76±0.028	14.48±0.028	0.77±0.003	8.53±0.113
	Barbara	ℓ_2-TV	22.65±0.005	10.01±0.005	0.65±0.001	1.49±0.006
		ℓ_0-grad	22.56±0.006	9.92±0.006	0.64±0.002	1.43±0.005
		$Q\ell_0$-grad	22.83±0.006	10.19±0.006	0.64±0.003	8.51±0.086
	Mandrill	ℓ_2-TV	20.46±0.007	5.72±0.007	0.51±0.003	1.49±0.007
		ℓ_0-grad	20.45±0.006	5.71±0.006	0.52±0.003	1.42±0.005
		$Q\ell_0$-grad	20.34±0.008	5.60±0.008	0.46±0.001	8.54±0.089
	Peppers	ℓ_2-TV	29.44±0.017	16.37±0.017	0.86±0.001	1.49±0.005
		ℓ_0-grad	28.91±0.014	15.83±0.014	0.83±0.001	1.44±0.056
		$Q\ell_0$-grad	28.27±0.017	15.19±0.017	0.83±0.001	8.43±0.018
COLOR	Lena	ℓ_2-TV	27.93±0.003	15.31±0.003	0.76±0.000	6.17±0.015
		ℓ_0-grad	27.57±0.003	14.96±0.003	0.74±0.000	6.09±0.008
		$Q\ell_0$-grad	28.55±0.008	15.94±0.008	0.75±0.000	28.23±0.105
	Barbara	ℓ_2-TV	25.75±0.005	12.06±0.005	0.72±0.000	6.17±0.012
		ℓ_0-grad	25.57±0.005	11.87±0.005	0.71±0.000	6.09±0.014
		$Q\ell_0$-grad	25.72±0.004	12.03±0.004	0.70±0.004	28.31±0.110
	Mandrill	ℓ_2-TV	20.92±0.002	7.78±0.002	0.52±0.001	6.16±0.018
		ℓ_0-grad	20.91±0.002	7.78±0.002	0.52±0.001	6.08±0.018
		$Q\ell_0$-grad	20.82±0.005	7.69±0.004	0.50±0.001	28.44±0.095
	Peppers	ℓ_2-TV	27.21±0.005	16.13±0.005	0.73±0.000	6.18±0.054
		ℓ_0-grad	26.68±0.006	15.60006±0.	0.71±0.000	6.11±0.035
		$Q\ell_0$-grad	27.12±0.010	16.04±0.010	0.72±0.000	28.21±0.070

Table A.11: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 5×5 average filter and $\sigma = 0.1$ of standard deviation for additive Gaussian noise, after 10 experiments.

	Image	Method	PSNR	SNR	SSIM	Time
GREYSCALE	Lena	ℓ_2-TV	26.40±0.022	13.12±0.022	0.72±0.001	1.50±0.006
		ℓ_0-grad	25.53±0.025	12.26±0.025	0.68±0.002	1.43±0.007
		$Q\ell_0$-grad	26.68±0.031	13.41±0.031	0.73±0.002	8.45±0.085
	Barbara	ℓ_2-TV	22.32±0.007	9.68±0.007	0.61±0.002	1.50±0.037
		ℓ_0-grad	22.01±0.005	9.37±0.005	0.58±0.001	1.43±0.024
		$Q\ell_0$-grad	22.42±0.009	9.78±0.009	0.61±0.003	8.43±0.020
	Mandrill	ℓ_2-TV	19.98±0.007	5.25±0.007	0.44±0.002	1.48±0.008
		ℓ_0-grad	19.82±0.009	5.09±0.009	0.44±0.002	1.43±0.006
		$Q\ell_0$-grad	19.97±0.007	5.22±0.007	0.42±0.001	8.54±0.087
	Peppers	ℓ_2-TV	28.41±0.023	15.33±0.023	0.82±0.001	1.51±0.037
		ℓ_0-grad	27.39±0.026	14.32±0.026	0.77±0.002	1.43±0.004
		$Q\ell_0$-grad	27.29±0.018	14.21±0.018	0.80±0.001	8.38±0.025
COLOR	Lena	ℓ_2-TV	27.39±0.010	14.77±0.010	0.73±0.001	6.17±0.025
		ℓ_0-grad	26.46±0.016	13.85±0.016	0.68±0.001	6.09±0.013
		$Q\ell_0$-grad	27.62±0.013	15.01±0.013	0.72±0.001	28.21±0.104
	Barbara	ℓ_2-TV	25.28±0.006	11.59±0.006	0.69±0.001	6.16±0.015
		ℓ_0-grad	24.69±0.010	11.00±0.010	0.65±0.001	6.10±0.031
		$Q\ell_0$-grad	25.10±0.006	11.41±0.006	0.67±0.001	28.22±0.130
	Mandrill	ℓ_2-TV	20.50±0.004	7.36±0.004	0.45±0.001	6.18±0.038
		ℓ_0-grad	20.31±0.006	7.17±0.006	0.45±0.001	6.10±0.027
		$Q\ell_0$-grad	20.41±0.004	7.27±0.004	0.44±0.001	28.50±0.126
	Peppers	ℓ_2-TV	26.71±0.010	15.63±0.010	0.70±0.001	6.17±0.015
		ℓ_0-grad	25.63±0.014	14.54±0.014	0.66±0.000	6.10±0.023
		$Q\ell_0$-grad	26.25±0.016	15.17±0.016	0.69±0.001	28.21±0.099

Table A.12: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 5×5 average filter and $\sigma = 0.25$ of standard deviation for additive Gaussian noise, after 10 experiments.

	Image	Method	PSNR	SNR	SSIM	Time
GREYSCALE	Lena	ℓ_2-TV	24.45±0.041	11.18±0.041	0.64±0.004	1.53±0.065
		ℓ_0-grad	23.09±0.055	9.82±0.055	0.59±0.002	1.45±0.061
		$Q\ell_0$-grad	24.51±0.047	11.23±0.047	0.65±0.003	8.53±0.290
	Barbara	ℓ_2-TV	21.48±0.013	8.84±0.013	0.54±0.004	1.50±0.010
		ℓ_0-grad	20.76±0.022	8.12±0.022	0.50±0.002	1.43±0.014
		$Q\ell_0$-grad	21.52±0.012	8.88±0.012	0.54±0.003	8.40±0.092
	Mandrill	ℓ_2-TV	19.14±0.011	4.40±0.011	0.32±0.002	1.50±0.007
		ℓ_0-grad	18.64±0.016	3.91±0.016	0.28±0.001	1.43±0.008
		$Q\ell_0$-grad	19.14±0.016	4.40±0.016	0.31±0.005	8.44±0.100
	Peppers	ℓ_2-TV	25.88±0.034	12.81±0.034	0.74±0.003	1.51±0.015
		ℓ_0-grad	24.34±0.030	11.27±0.030	0.68±0.002	1.43±0.027
		$Q\ell_0$-grad	25.23±0.045	12.16±0.045	0.74±0.004	8.42±0.159
COLOR	Lena	ℓ_2-TV	25.93±0.018	13.31±0.018	0.67±0.002	6.17±0.041
		ℓ_0-grad	24.40±0.027	11.79±0.027	0.61±0.001	6.09±0.039
		$Q\ell_0$-grad	25.65±0.026	13.04±0.026	0.66±0.001	27.95±0.078
	Barbara	ℓ_2-TV	24.15±0.013	10.46±0.013	0.63±0.001	6.25±0.129
		ℓ_0-grad	22.97±0.039	9.28±0.031	0.56±0.001	6.14±0.120
		$Q\ell_0$-grad	23.79±0.009	10.10±0.009	0.60±0.001	28.25±0.566
	Mandrill	ℓ_2-TV	19.73±0.006	6.60±0.006	0.35±0.001	6.18±0.015
		ℓ_0-grad	19.17±0.009	6.03±0.009	0.30±0.001	6.09±0.026
		$Q\ell_0$-grad	19.60±0.007	6.46±0.007	0.33±0.001	28.27±0.175
	Peppers	ℓ_2-TV	25.16±0.019	14.08±0.019	0.65±0.002	6.16±0.025
		ℓ_0-grad	23.40±0.030	12.32±0.030	0.58±0.002	6.08±0.017
		$Q\ell_0$-grad	24.50±0.021	13.41±0.021	0.62±0.001	28.06±0.128

Table A.13: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 5×5 average filter and $\sigma = 0.5$ of standard deviation for additive Gaussian noise, after 10 experiments.

	Image	**Method**	**PSNR**	**SNR**	**SSIM**	**Time**
GREYSCALE	Lena	ℓ_2-TV	22.55±0.046	9.28±0.046	0.58±0.006	1.50±0.008
		ℓ_0-grad	20.98±0.063	7.71±0.063	0.54±0.003	1.42±0.011
		$Q\ell_0$-grad	22.35±0.045	9.08±0.045	0.57±0.007	8.33±0.090
	Barbara	ℓ_2-TV	20.36±0.035	7.72±0.035	0.47±0.006	1.52±0.038
		ℓ_0-grad	19.22±0.045	6.58±0.045	0.43±0.004	1.42±0.007
		$Q\ell_0$-grad	20.34±0.030	7.70±0.030	0.46±0.004	8.31±0.045
	Mandrill	ℓ_2-TV	18.50±0.025	3.76±0.025	0.25±0.002	1.50±0.010
		ℓ_0-grad	17.80±0.034	3.07±0.034	0.22±0.002	1.42±0.008
		$Q\ell_0$-grad	18.48±0.023	3.74±0.023	0.25±0.002	8.33±0.095
	Peppers	ℓ_2-TV	23.42±0.080	10.34±0.080	0.67±0.004	1.50±0.008
		ℓ_0-grad	21.48±0.113	8.40±0.113	0.61±0.004	1.43±0.046
		$Q\ell_0$-grad	23.06±0.077	9.99±0.077	0.65±0.003	8.29±0.036
COLOR	Lena	ℓ_2-TV	24.30±0.030	11.69±0.030	0.62±0.001	6.18±0.011
		ℓ_0-grad	22.79±0.085	10.18±0.085	0.57±0.003	6.07±0.011
		$Q\ell_0$-grad	23.57±0.028	10.96±0.028	0.58±0.001	27.92±0.115
	Barbara	ℓ_2-TV	22.78±0.033	9.09±0.033	0.57±0.002	6.18±0.017
		ℓ_0-grad	21.42±0.034	7.73±0.034	0.50±0.002	6.08±0.013
		$Q\ell_0$-grad	22.23±0.025	8.54±0.025	0.52±0.001	27.93±0.0.79
	Mandrill	ℓ_2-TV	19.09±0.008	5.96±0.008	0.28±0.001	6.16±0.022
		ℓ_0-grad	18.42±0.017	5.28±0.017	0.24±0.001	6.09±0.055
		$Q\ell_0$-grad	18.91±0.010	5.77±0.010	0.27±0.001	28.05±0.483
	Peppers	ℓ_2-TV	23.26±0.049	12.18±0.050	0.59±0.002	6.18±0.053
		ℓ_0-grad	21.28±0.041	10.20±0.041	0.52±0.002	6.08±0.016
		$Q\ell_0$-grad	22.54±0.047	11.46±0.045	0.55±0.002	27.89±0.056

C Tables for deblurring with a 9×9 Gaussian filter

Table A.14: Bounds of λ used in the Fibonacci searches (Section 2.10 Fibonacci search) for finding the optimal value that maximizes the SNR metric for the deblurring with a 9×9 Gaussian filter tests, according to the standard deviation (σ) of the additive Gaussian noise and method.

		Greyscale		Color	
σ	Method	Lower bound	Upper bound	Lower bound	Upper bound
0.0	ℓ_2-TV	0.005	0.05	0.0	0.1
	ℓ_0-grad	0.0001	0.005	0.0	0.02
	$Q\ell_0$-grad	0.005	0.025	0.0	0.05
0.05	ℓ_2-TV	0.01	0.07	0.005	0.1
	ℓ_0-grad	0.0005	0.005	0.0001	0.05
	$Q\ell_0$-grad	0.01	0.05	0.005	0.05
0.1	ℓ_2-TV	0.02	0.1	0.01	0.15
	ℓ_0-grad	0.001	0.01	0.001	0.1
	$Q\ell_0$-grad	0.01	0.1	0.01	0.1
0.25	ℓ_2-TV	0.05	0.15	0.05	0.5
	ℓ_0-grad	0.005	0.025	0.05	0.5
	$Q\ell_0$-grad	0.25	1.0	0.1	1.5
0.5	ℓ_2-TV	0.2	0.3	0.1	1.0
	ℓ_0-grad	0.03	0.1	0.2	1.0
	$Q\ell_0$-grad	2.0	4.5	2.0	4.0

Table A.15: PSNR, SNR, SSIM and averages and standard deviations of computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 9×9 Gaussian filter, after 10 experiments.

	Image	**Method**	**PSNR**	**SNR**	**SSIM**	**Time**
	Lena	ℓ_2-TV	25.15	11.88	0.69	1.48±0.028
G		ℓ_0-grad	24.78	11.51	0.67	1.42±0.047
R		$Q\ell_0$-grad	26.22	12.94	0.71	12.50±0.204
E	Barbara	ℓ_2-TV	21.89	9.25	0.58	1.48±0.005
Y		ℓ_0-grad	21.67	9.03	0.57	1.40±0.005
S		$Q\ell_0$-grad	22.29	9.65	0.58	12.43±0.098
C	Mandrill	ℓ_2-TV	19.3	4.56	0.37	1.50±0.038
A		ℓ_0-grad	19.28	4.54	0.38	1.42±0.033
L		$Q\ell_0$-grad	19.53	4.79	0.34	12.52±0.101
E	Peppers	ℓ_2-TV	26.25	13.17	0.79	1.47±0.005
		ℓ_0-grad	25.71	12.64	0.77	1.40±0.006
		$Q\ell_0$-grad	26.88	13.81	0.79	12.39±0.094
	Lena	ℓ_2-TV	26.13	13.51	0.68	6.31±0.093
		ℓ_0-grad	25.82	13.21	0.67	6.22±0.138
		$Q\ell_0$-grad	27.16	14.55	0.71	42.16±0.200
C	Barbara	ℓ_2-TV	24.47	10.78	0.66	6.24±0.025
O		ℓ_0-grad	24.29	10.6	0.66	6.20±0.100
L		$Q\ell_0$-grad	24.91	11.26	0.64	42.06±0.125
O	Mandrill	ℓ_2-TV	19.81	6.67	0.4	6.30±0.081
R		ℓ_0-grad	19.81	6.67	0.41	6.19±0.019
		$Q\ell_0$-grad	20.01	6.87	0.37	42.22±0.069
	Peppers	ℓ_2-TV	25.09	14.01	0.67	6.25±0.017
		ℓ_0-grad	24.53	13.45	0.66	6.16±0.017
		$Q\ell_0$-grad	25.91	14.83	0.69	42.05±0.187

Table A.16: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 9×9 Gaussian filter and $\sigma = 0.05$ for additive Gaussian noise, after 10 experiments.

	Image	**Method**	**PSNR**	**SNR**	**SSIM**	**Time**
GREYSCALE	Lena	ℓ_2-TV	25.05±0.011	11.78±0.011	0.68±0.001	1.50±0.010
		ℓ_0-grad	24.57±0.019	11.29±0.019	0.66±0.002	1.44±0.041
		$Q\ell_0$-grad	25.75±0.031	12.48±0.031	0.69±0.003	12.48±0.111
	Barbara	ℓ_2-TV	21.84±0.006	9.20±0.006	0.57±0.001	1.50±0.011
		ℓ_0-grad	21.55±0.012	8.91±0.012	0.56±0.001	1.43±0.012
		$Q\ell_0$-grad	22.10±0.011	9.46±0.011	0.57±0.002	12.45±0.101
	Mandrill	ℓ_2-TV	19.23±0.003	4.49±0.003	0.35±0.001	1.50±0.010
		ℓ_0-grad	19.16±0.005	4.42±0.005	0.35±0.002	1.43±0.007
		$Q\ell_0$-grad	19.43±0.004	4.69±0.004	0.33±0.001	12.53±0.091
	Peppers	ℓ_2-TV	26.12±0.015	13.05±0.015	0.79±0.002	1.50±0.009
		ℓ_0-grad	25.48±0.018	12.40±0.018	0.76±0.001	1.43±0.008
		$Q\ell_0$-grad	26.40±0.030	13.33±0.030	0.77±0.003	12.45±0.093
COLOR	Lena	ℓ_2-TV	26.07±0.007	13.46±0.007	0.68±0.000	6.31±0.017
		ℓ_0-grad	25.58±0.017	12.96±0.017	0.65±0.001	6.28±0.014
		$Q\ell_0$-grad	26.80±0.018	14.19±0.018	0.69±0.001	42.26±0.172
	Barbara	ℓ_2-TV	24.41±0.005	10.72±0.005	0.65±0.002	6.31±0.014
		ℓ_0-grad	24.09±0.009	10.40±0.009	0.64±0.001	6.22±0.019
		$Q\ell_0$-grad	24.64±0.004	10.95±0.004	0.63±0.003	42.27±0.217
	Mandrill	ℓ_2-TV	19.73±0.003	6.59±0.003	0.37±0.000	6.32±0.019
		ℓ_0-grad	19.67±0.003	6.54±0.003	0.39±0.000	6.21±0.013
		$Q\ell_0$-grad	19.89±0.003	6.75±0.003	0.36±0.000	42.48±0.166
	Peppers	ℓ_2-TV	25.03±0.005	13.94±0.005	0.67±0.000	6.32±0.053
		ℓ_0-grad	24.34±0.009	13.26±0.009	0.65±0.001	6.24±0.039
		$Q\ell_0$-grad	25.50±0.009	14.42±0.009	0.67±0.000	42.16±0.044

Table A.17: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 9×9 Gaussian filter and $\sigma = 0.1$ for additive Gaussian noise, after 10 experiments.

	Image	Method	PSNR	SNR	SSIM	Time
GREYSCALE	Lena	ℓ_2-TV	24.80±0.020	11.52±0.020	0.67±0.001	1.50±0.007
		ℓ_0-grad	24.11±0.037	10.83±0.037	0.63±0.002	1.43±0.010
		$Q\ell_0$-grad	25.08±0.034	11.80±0.034	0.67±0.002	12.47±0.091
	Barbara	ℓ_2-TV	21.72±0.009	9.08±0.010	0.56±0.003	1.50±0.009
		ℓ_0-grad	21.33±0.020	8.69±0.020	0.53±0.002	1.43±0.005
		$Q\ell_0$-grad	21.83±0.016	9.19±0.016	0.55±0.006	12.47±0.090
	Mandrill	ℓ_2-TV	19.12±0.005	4.38±0.005	0.32±0.002	1.50±0.013
		ℓ_0-grad	18.93±0.011	4.19±0.011	0.30±0.002	1.43±0.010
		$Q\ell_0$-grad	19.25±0.007	4.51±0.007	0.31±0.002	12.54±0.094
	Peppers	ℓ_2-TV	25.80±0.024	12.72±0.024	0.77±0.002	1.50±0.009
		ℓ_0-grad	25.01±0.024	11.94±0.024	0.72±0.001	1.43±0.008
		$Q\ell_0$-grad	25.70±0.039	12.63±0.039	0.75±0.002	12.44±0.088
COLOR	Lena	ℓ_2-TV	25.91±0.011	13.30±0.011	0.67±0.000	6.33±0.044
		ℓ_0-grad	25.19±0.017	12.57±0.017	0.63±0.001	6.22±0.015
		$Q\ell_0$-grad	26.19±0.022	13.58±0.022	0.67±0.001	42.24±0.151
	Barbara	ℓ_2-TV	24.25±0.007	10.56±0.007	0.64±0.001	6.30±0.022
		ℓ_0-grad	23.70±0.013	10.00±0.013	0.61±0.001	6.23±0.020
		$Q\ell_0$-grad	24.22±0.008	10.53±0.008	0.61±0.004	42.24±0.193
	Mandrill	ℓ_2-TV	19.62±0.003	6.48±0.003	0.34±0.000	6.32±0.018
		ℓ_0-grad	19.42±0.007	6.29±0.007	0.32±0.001	6.24±0.013
		$Q\ell_0$-grad	19.69±0.005	6.55±0.005	0.34±0.001	42.56±0.141
	Peppers	ℓ_2-TV	24.85±0.004	13.77±0.004	0.66±0.000	6.32±0.017
		ℓ_0-grad	23.96±0.016	12.88±0.016	0.62±0.001	6.25±0.081
		$Q\ell_0$-grad	24.89±0.017	13.81±0.017	0.64±0.001	42.15±0.063

Table A.18: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 9×9 Gaussian filter and $\sigma = 0.25$ for additive Gaussian noise, after 10 experiments.

	Image	Method	PSNR	SNR	SSIM	Time
GREYSCALE	Lena	ℓ_2-TV	23.64±0.047	10.37±0.047	0.62±0.003	1.50±0.007
		ℓ_0-grad	22.57±0.050	9.30±0.050	0.57±0.002	1.42±0.005
		$Q\ell_0$-grad	23.57±0.041	10.29±0.041	0.62±0.005	12.37±0.086
	Barbara	ℓ_2-TV	21.11±0.013	8.47±0.014	0.51±0.004	1.50±0.006
		ℓ_0-grad	20.50±0.016	7.86±0.016	0.47±0.002	1.42±0.007
		$Q\ell_0$-grad	21.07±0.017	8.43±0.017	0.50±0.003	12.38±0.085
	Mandrill	ℓ_2-TV	18.77±0.009	4.03±0.009	0.27±0.003	1.51±0.0.21
		ℓ_0-grad	18.44±0.014	3.70±0.014	0.24±0.002	1.42±0.006
		$Q\ell_0$-grad	18.78±0.004	4.04±0.004	0.27±0.003	12.40±0.062
	Peppers	ℓ_2-TV	24.51±0.038	11.44±0.038	0.71±0.002	1.50±0.011
		ℓ_0-grad	23.34±0.072	10.26±0.072	0.66±0.003	1.42±0.008
		$Q\ell_0$-grad	24.16±0.044	11.09±0.044	0.70±0.002	12.37±0.087
COLOR	Lena	ℓ_2-TV	25.10±0.020	12.49±0.020	0.64±0.001	6.31±0.063
		ℓ_0-grad	23.90±0.040	11.28±0.040	0.59±0.001	6.24±0.064
		$Q\ell_0$-grad	24.79±0.025	12.17±0.012	0.63±0.001	41.94±0.037
	Barbara	ℓ_2-TV	23.54±0.026	9.85±0.023	0.60±0.002	6.30±0.014
		ℓ_0-grad	22.53±0.027	8.84±0.027	0.54±0.002	6.22±0.017
		$Q\ell_0$-grad	23.15±0.019	9.46±0.019	0.57±0.002	42.05±0.154
	Mandrill	ℓ_2-TV	19.29±0.006	6.16±0.006	0.30±0.001	6.31±0.014
		ℓ_0-grad	18.88±0.013	5.74±0.013	0.27±0.002	6.23±0.022
		$Q\ell_0$-grad	19.20±0.007	6.06±0.007	0.28±0.001	42.15±0.172
	Peppers	ℓ_2-TV	23.97±0.019	12.89±0.019	0.62±0.001	6.31±0.021
		ℓ_0-grad	22.61±0.025	11.53±0.025	0.55±0.002	6.22±0.015
		$Q\ell_0$-grad	23.51±0.023	12.43±0.023	0.59±0.001	41.98±0.150

Table A.19: Averages and standard deviations of PSNR, SNR, SSIM and computation time in seconds, according to method ($Q\ell_0$-grad, ℓ_0-grad and ℓ_2-**TV**) for deblurring with a 9×9 Gaussian filter and $\sigma = 0.5$ for additive Gaussian noise, after 10 experiments.

	Image	**Method**	**PSNR**	**SNR**	**SSIM**	**Time**
GREYSCALE	Lena	ℓ_2-TV	22.16±0.052	8.88±0.052	0.57±0.008	1.51±0.010
		ℓ_0-grad	20.93±0.062	7.65±0.062	0.54±0.004	1.42±0.011
		$Q\ell_0$-grad	21.99±0.050	8.72±0.050	0.57±0.006	12.31±0.096
	Barbara	ℓ_2-TV	20.13±0.028	7.50±0.028	0.46±0.004	1.51±0.028
		ℓ_0-grad	19.20±0.048	6.56±0.048	0.42±0.002	1.43±0.039
		$Q\ell_0$-grad	20.08±0.026	7.45±0.026	0.45±0.004	12.28±0.020
	Mandrill	ℓ_2-TV	18.33±0.013	3.60±0.013	0.24±0.002	1.50±0.011
		ℓ_0-grad	17.81±0.035	3.07±0.035	0.21±0.001	1.42±0.006
		$Q\ell_0$-grad	18.31±0.011	3.57±0.011	0.23±0.002	12.29±0.098
	Peppers	ℓ_2-TV	22.71±0.048	9.64±0.048	0.65±0.006	1.50±0.007
		ℓ_0-grad	21.18±0.062	8.12±0.063	0.59±0.003	1.42±0.004
		$Q\ell_0$-grad	22.44±0.048	9.37±0.048	0.64±0.005	12.31±0.086
COLOR	Lena	ℓ_2-TV	23.84±0.043	11.22±0.043	0.61±0.002	6.31±0.015
		ℓ_0-grad	22.65±0.052	10.03±0.052	0.56±0.002	6.23±0.010
		$Q\ell_0$-grad	23.24±0.037	10.63±0.037	0.58±0.002	41.87±0.166
	Barbara	ℓ_2-TV	22.42±0.030	8.73±0.030	0.55±0.002	6.32±0.018
		ℓ_0-grad	21.25±0.028	7.56±0.028	0.49±0.002	6.25±0.088
		$Q\ell_0$-grad	21.87±0.029	8.18±0.029	0.51±0.001	41.80±0.046
	Mandrill	ℓ_2-TV	18.87±0.012	5.73±0.012	0.26±0.001	6.21±0.026
		ℓ_0-grad	18.32±0.015	5.18±0.015	0.23±0.002	6.24±0.106
		$Q\ell_0$-grad	18.70±0.011	5.57±0.011	0.25±0.001	41.76±0.076
	Peppers	ℓ_2-TV	22.57±0.036	11.49±0.036	0.57±0.002	6.32±0.061
		ℓ_0-grad	20.99±0.036	9.90±0.036	0.51±0.002	6.52±0.026
		$Q\ell_0$-grad	22.00±0.042	10.92±0.042	0.54±0.002	41.81±0.106

D Pseudo-codes of the APG and ADMM algorithm for $Q\ell_0$-grad

Before the pseudo-codes are given, the following should be taken into consideration:

- **b** is the observed image.
- H is the filter used, where H***b**=A**b**.
- n is the number of outer iterations in APG and normal iterations in ADMM.
- m is the number of inner iterations in APG.
- For ADMM $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}$.
- For APG $P_{\mathcal{P}}(\mathbf{p},\mathbf{q}) = (\mathbf{r},\mathbf{s}) = \begin{cases} r_{i,j} = \frac{p_{i,j}}{\max\{1,\sqrt{p_{i,j}^2+q_{i,j}^2}\}} \\ s_{i,j} = \frac{q_{i,j}}{\max\{1,\sqrt{p_{i,j}^2+q_{i,j}^2}\}} \end{cases}$.
- If $\mathbf{b} \in \mathbb{S}$ then $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{w}_1$, $\mathbf{w}_2 \in \mathbb{S}$, where $\mathbb{S}$ could be a two dimension or a three dimension space depending on if the image is greyscale or color.

Algorithm 2 ADMM

Require: H, **b**, λ, ρ, γ, β, n

$k = 0$

$\mathbf{v}_0 = \mathbf{0}$

$\mathbf{w}_0 = \mathbf{0}$

repeat

$$\mathbf{u}_{k+1} = \text{argmin}_{\mathbf{u}} \tfrac{1}{2}\|A\mathbf{u}-\mathbf{b}\|_2^2 - \tfrac{\gamma}{2}\|\nabla\mathbf{u}\|_2^2 + \tfrac{\rho}{2}\left\| \begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u} - \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k + \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}_k \right\|_2^2$$

$$\mathbf{v}_{k+1} = \text{argmin}_{\mathbf{v}} \sqrt{2\gamma\lambda}\|\mathbf{v}\|_1 + \tfrac{\rho}{2}\left\|\beta \begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u}_{k+1} + (1-\beta)\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k - \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}_k \right\|_2^2$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \beta \begin{bmatrix} D_x \\ D_y \end{bmatrix}\mathbf{u}_{k+1} + (1-\beta)\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_k - \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}_{k+1}$$

$k = k+1$

until $k = n$

return $\mathbf{u}_n$

Algorithm 3 APG

Require: H, $\mathbf{b}$, λ, γ, n

$k = 0$
$l = 0$
$\mathbf{u}_0 = \mathbf{0}$
$\mathbf{v}_1 = \mathbf{0}$
$(\mathbf{p}, \mathbf{q})_0 = (\mathbf{0}, \mathbf{0})$
$(\mathbf{r}, \mathbf{s})_1 = (\mathbf{0}, \mathbf{0})$
$G_1 = -A^T \mathbf{b}$
$\alpha = \frac{\|-A^T \mathbf{b}\|_2^2}{\|-AA^T \mathbf{b}\|_2^2}$
$D = \nabla(\alpha A^T \mathbf{b})$
repeat
  if $k > 0$ **then**
    $G_1 = A^T A \mathbf{v}_{k+1} - A^T \mathbf{b} - \gamma \nabla^T \nabla \mathbf{v}_{k+1}$
    if $k\%2 = 0$ **then**
      $\alpha = \frac{\|G_1\|_2^2}{\|AG_1\|_2^2 + \|\nabla G_1\|_2^2}$
    $D = \nabla(\mathbf{v}_{k+1} - \alpha G_1)$
    $(\mathbf{p}, \mathbf{q})_0 = (\mathbf{0}, \mathbf{0})$
    $(\mathbf{r}, \mathbf{s})_1 = (\mathbf{0}, \mathbf{0})$
  repeat
    $G_2 = \alpha \sqrt{2\lambda\gamma} \nabla \nabla^T (\mathbf{r}, \mathbf{s})_{l+1} - D$
    $(\mathbf{p}, \mathbf{q})_{l+1} = (\mathbf{r}, \mathbf{s})_{l+1} - \frac{1}{8\alpha\sqrt{2\lambda\gamma}} G_2$
    $(\mathbf{p}, \mathbf{q})_{l+1} = P_{\mathcal{P}}((\mathbf{p}, \mathbf{q})_{l+1})$
    $w_i = \frac{l-1}{l+2}$
    $l = l + 1$
    $(\mathbf{r}, \mathbf{s})_{l+1} = (\mathbf{p}, \mathbf{q})_l + w_i((\mathbf{p}, \mathbf{q})_l - (\mathbf{p}, \mathbf{q})_{l-1})$
  until $l = m$
$l = 0$
$\mathbf{u}_{k+1} = \mathbf{v}_{k+1} - \alpha(G_1 + \sqrt{2\lambda\gamma} \nabla^T (\mathbf{r}, \mathbf{s})_{m+1})$
$w_e = \frac{k-1}{k+2}$
$k = k + 1$
$\mathbf{v}_{k+1} = \mathbf{u}_k + w_e(\mathbf{u}_k - \mathbf{u}_{k-1})$
until $k = n$
return $\mathbf{u}_n$

E Tables and reconstructed images with $Q\ell_0$-grad using different values of γ and a fixed λ

Table A.20: Averages and standard deviations of PSNR, SNR and SSIM according to standard deviation (σ) of additive Gaussian noise and γ used in the $Q\ell_0$-grad method for denoising greyscale Lena after 10 experiments.

σ	γ	PSNR	SNR	SSIM
0.05	0.05	29.83±0.016	16.56±0.016	0.82±0.002
	0.025	30.95±0.015	17.68±0.015	0.85±0.001
	0.01	31.76±0.016	18.48±0.016	0.85±0.003
	0.0075	31.66±0.018	18.38±0.018	0.84±0.004
	0.005	31.17±0.019	17.89±0.019	0.81±0.004
0.1	0.05	26.68±0.017	13.40±0.017	0.73±0.003
	0.025	27.80±0.019	14.53±0.019	0.77±0.004
	0.01	28.73±0.025	15.46±0.025	0.77±0.005
	0.0075	28.58±0.025	15.30±0.025	0.75±0.003
	0.005	27.77±0.028	14.50±0.028	0.70±0.002
0.25	0.05	23.01±0.035	9.74±0.035	0.64±0.002
	0.025	24.14±0.034	10.87±0.034	0.66±0.004
	0.01	25.14±0.038	11.87±0.038	0.66±0.002
	0.0075	24.94±0.047	11.66±0.047	0.63±0.003
	0.005	23.76±0.049	10.38±0.049	0.53±0.003
0.5	0.05	20.85±0.058	7.57±0.058	0.57±0.001
	0.025	21.68±0.072	8.41±0.072	0.59±0.002
	0.01	22.64±0.049	9.36±0.049	0.58±0.002
	0.0075	22.28±0.026	9.01±0.026	0.54±0.002
	0.005	20.43±0.020	7.15±0.020	0.39±0.002

(a) $\sigma = 0.05$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.1: Reconstructed images from a noisy greyscale Lena with $\sigma = 0.05$ for additive Gaussian noise using $Q\ell_0$-grad and different values of γ.

(a) $\sigma = 0.25$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.2: Reconstructed images from a noisy greyscale Lena with $\sigma = 0.25$ for additive Gaussian noise using $Q\ell_0$-grad and different values of γ.

(a) $\sigma = 0.5$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.3: Reconstructed images from a noisy greyscale Lena with $\sigma = 0.5$ for additive Gaussian noise using $Q\ell_0$-grad and different values of γ.

Table A.21: Averages and standard deviations of PSNR, SNR and SSIM according to standard deviation (σ) of additive Gaussian noise and γ used in the $Q\ell_0$-grad method for deblurring greyscale Barbara with a 5×5 average filter after 10 experiments.

σ	γ	**PSNR**	**SNR**	**SSIM**
0.0	0.05	22.59	9.95	0.61
	0.025	22.85	10.21	0.63
	0.01	23.08	10.44	0.65
	0.0075	23.10	10.46	0.65
	0.005	23.08	10.44	0.66
0.05	0.05	22.46±0.006	9.82±0.006	0.60±0.004
	0.025	22.68±0.006	10.04±0.006	0.62±0.003
	0.01	22.82±0.006	10.19±0.006	0.63±0.002
	0.0075	22.83±0.006	10.19±0.006	0.64±0.002
	0.005	22.78±0.008	10.14±0.008	0.64±0.003
0.1	0.05	22.13±0.009	9.49±0.009	0.59±0.001
	0.025	22.31±0.008	9.67±0.008	0.60±0.003
	0.01	22.42±0.009	9.78±0.009	0.61±0.003
	0.0075	22.43±0.009	9.79±0.009	0.61±0.002
	0.005	22.40±0.009	9.76±0.009	0.60±0.002
0.25	0.05	21.05±0.017	8.41±0.017	0.51±0.001
	0.025	21.36±0.014	8.72±0.014	0.53±0.002
	0.01	21.53±0.014	8.89±0.014	0.54±0.003
	0.0075	21.50±0.014	8.86±0.014	0.53±0.003
	0.005	21.39±0.016	8.75±0.016	0.52±0.002
0.5	0.05	19.99±0.029	7.35±0.029	0.45±0.003
	0.025	20.13±0.031	7.49±0.031	0.45±0.002
	0.01	20.34±0.032	7.71±0.032	0.46±0.003
	0.0075	20.31±0.033	7.67±0.033	0.46±0.003
	0.005	20.11±0.035	7.47±0.035	0.44±0.003

(a) $\sigma = 0.0$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.4: Reconstructed images from a noisy greyscale Barbara with $\sigma = 0.0$ for additive Gaussian noise and an average 5×5 filter, using $Q\ell_0$-grad and different values of γ.

(a) $\sigma = 0.05$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.5: Reconstructed images from a noisy greyscale Barbara with $\sigma = 0.05$ for additive Gaussian noise and an average 5×5 filter, using $Q\ell_0$-grad and different values of γ.

(a) $\sigma = 0.25$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.6: Reconstructed images from a noisy greyscale Barbara with $\sigma = 0.25$ for additive Gaussian noise and an average 5×5 filter, using $Q\ell_0$-grad and different values of γ.

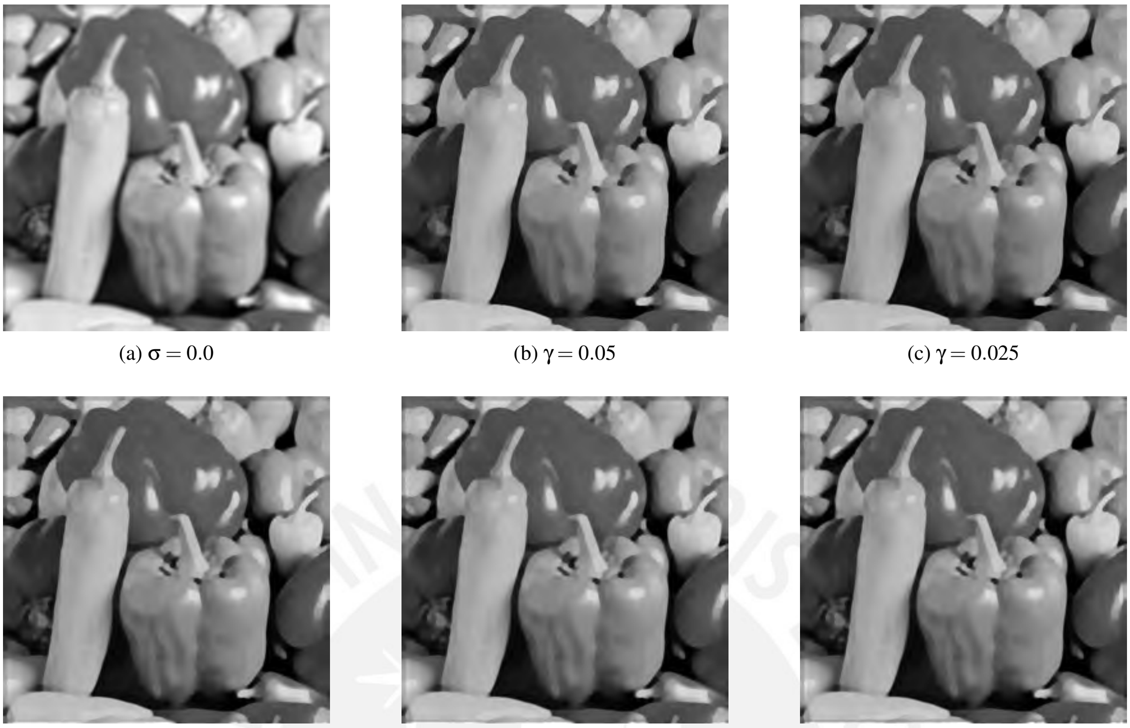
(a) $\sigma = 0.5$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.7: Reconstructed images from a noisy greyscale Barbara with $\sigma = 0.5$ for additive Gaussian noise and an average 5×5 filter, using $Q\ell_0$-grad and different values of γ.

Table A.22: Averages and standard deviations of PSNR, SNR and SSIM according to standard deviation (σ) of additive Gaussian noise and γ used in the $Q\ell_0$-grad method for deblurring greyscale Peppers with a 9×9 Gaussian filter after 10 experiments.

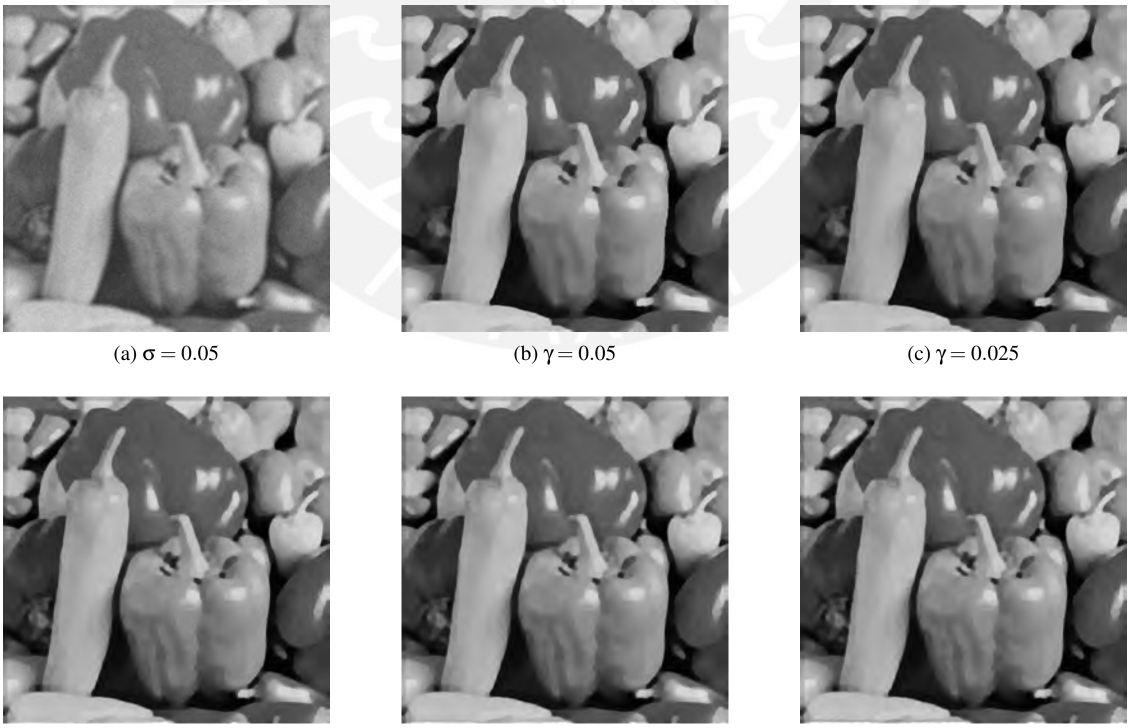
σ	γ	**PSNR**	**SNR**	**SSIM**
0.0	0.05	26.21	13.14	0.77
	0.025	26.55	13.48	0.78
	0.01	26.88	13.81	0.79
	0.0075	26.88	13.80	0.79
	0.005	26.78	13.71	0.80
0.05	0.05	25.58±0.022	12.51±0.022	0.75±0.002
	0.025	26.02±0.028	12.95±0.028	0.76±0.002
	0.01	26.40±0.027	13.33±0.027	0.77±0.002
	0.0075	26.43±0.029	13.36±0.029	0.77±0.002
	0.005	26.36±0.033	13.29±0.033	0.78±0.002
0.1	0.05	24.92±0.033	11.84±0.033	0.73±0.001
	0.025	25.34±0.026	12.27±0.026	0.74±0.002
	0.01	25.69±0.025	12.62±0.025	0.75±0.003
	0.0075	25.73±0.025	12.66±0.025	0.75±0.002
	0.005	25.70±0.024	12.64±0.024	0.75±0.002
0.25	0.05	23.44±0.037	10.37±0.037	0.68±0.002
	0.025	23.87±0.038	10.80±0.038	0.70±0.002
	0.01	24.15±0.032	11.08±0.032	0.70±0.002
	0.0075	24.16±0.032	11.08±0.032	0.69±0.002
	0.005	24.08±0.036	11.01±0.036	0.69±0.002
0.5	0.05	21.59±0.035	8.52±0.035	0.62±0.003
	0.025	22.10±0.053	9.02±0.053	0.63±0.004
	0.01	22.45±0.070	9.37±0.070	0.64±0.003
	0.0075	22.41±0.070	9.34±0.070	0.63±0.003
	0.005	22.22±0.070	9.15±0.070	0.61±0.004



(a) $\sigma = 0.0$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.8: Reconstructed images from a noisy greyscale Peppers with $\sigma = 0.0$ for additive Gaussian noise and a Gaussian 9×9 filter, using $Q\ell_0$-grad and different values of γ.



(a) $\sigma = 0.05$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

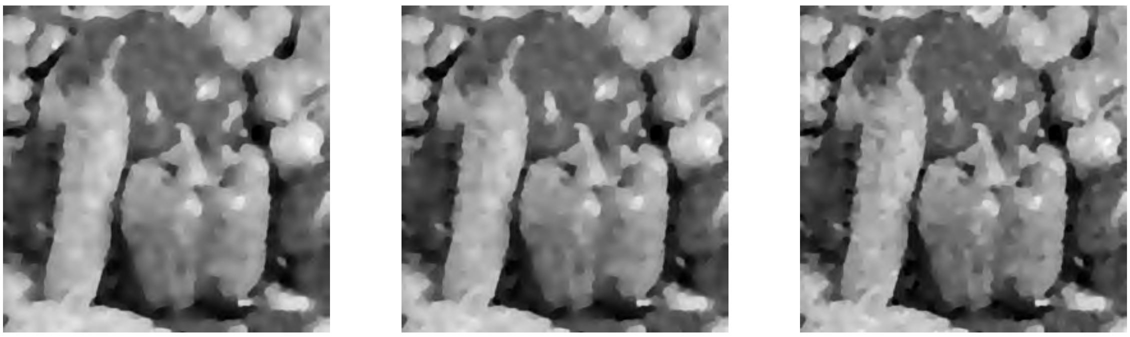
Figure A.9: Reconstructed images from a noisy greyscale Peppers with $\sigma = 0.05$ for additive Gaussian noise and a Gaussian 9×9 filter, using $Q\ell_0$-grad and different values of γ.

(a) $\sigma = 0.25$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$

(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.10: Reconstructed images from a noisy greyscale Peppers with $\sigma = 0.25$ for additive Gaussian noise and a Gaussian 9×9 filter, using $Q\ell_0$-grad and different values of γ.

(a) $\sigma = 0.5$ (b) $\gamma = 0.05$ (c) $\gamma = 0.025$



(d) $\gamma = 0.01$ (e) $\gamma = 0.0075$ (f) $\gamma = 0.005$

Figure A.11: Reconstructed images from a noisy greyscale Peppers with $\sigma = 0.5$ for additive Gaussian noise and a Gaussian 9×9 filter, using $Q\ell_0$-grad and different values of γ.

Bibliography

[1] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Transactions on Graphics*, vol. 30, no. 6, pp. 1–12, 2011.

[2] X. Cheng, M. Zeng, and X. Liu, "Feature-preserving filtering with L0 gradient minimization," *Computers and Graphics (Pergamon)*, vol. 38, no. 1, pp. 150–157, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cag.2013.10.025

[3] R. M. Nguyen and M. S. Brown, "Fast and effective L_0 gradient minimization by region fusion," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 208–216, 2015.

[4] M. Storath, A. Weinmann, and L. Demaret, "Jump-sparse and sparse recovery using potts functionals," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3654–3666, 2014.

[5] S. Ono, "L_0 Gradient Projection," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1554–1564, apr 2017.

[6] P. Rodriguez, "X-ray ct reconstruction via l0 gradient projection," in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2019, pp. 306–310.

[7] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014. [Online]. Available: http://dx.doi.org/10.1561/2400000003

[8] X.-D. Zhang, *Matrix analysis and applications*. Cambridge University Press, 2017.

[9] A. Beck, *First-order methods in optimization*. SIAM, 2017.

[10] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 2nd ed. Upper Saddle River, N.J: Prentice Hall, 2002.

[11] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[12] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers and Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[13] 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, no. 1, pp. 1–122, 2010.

[14] L. I. Rudin, "Images, numerical analysis of singularities and shock filters," Ph.D. dissertation, California Institute of Technology, 1987.

[15] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[16] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[17] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009.

[18] M. Carlsson, “On Convex Envelopes and Regularization of Non-convex Functionals Without Moving Global Minima,” *Journal of Optimization Theory and Applications*, vol. 183, no. 1, pp. 66–84, 2019.

[19] E. Soubies, L. Blanc-Féraud, and G. Aubert, “A continuous exact l0 penalty (cel0) for least squares regularized problem,” *SIAM Journal on Imaging Sciences*, vol. 8, no. 3, pp. 1607–1639, 2015.

[20] R. Hassin, “On maximizing functions by fibonacci search,” *Fibonacci Quart*, vol. 19, pp. 347–351, 1981.

[21] E. A. Vasquez-Ortiz and P. Rodriguez, “Fast gradient-based algorithm for a quadratic envelope relaxation of the ℓ_0 gradient regularization,” in *2021 XXIII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, 2021, pp. 1–6.

[22] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[23] P. Rodriguez and E. A. Vasquez-Ortiz, “$q\ell_0$ regularization,” 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5105410

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.