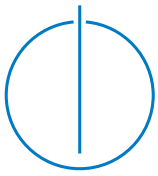
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Collaborative Total Variation Regularization of Hyperspectral Images

Gerard Martí Juan



DEPARTMENT OF INFORMATICS

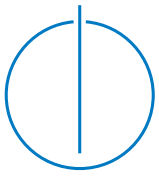
TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Collaborative Total Variation Regularization of Hyperspectral Images

Kollaborative Regularisierung von Hyperspektralbildern mittels Totaler Variation

Author:	Gerard Martí Juan
Supervisor:	Prof. Dr. Daniel Cremers
Advisor:	Dr. Michael Möller
Submission Date:	15/09/2015



I confirm that this bachelor's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15/09/2015 Gerard Martí Juan

Acknowledgments

I would like to express my sincere gratitude to Dr. Michael Möller not only for his continuous patience and assistance during the construction of the thesis but also for everything he has taught me during these past months.

I also want to thank my home university, Universitat Politècnica de Catalunya, for giving me the opportunity of doing my Bachelor Thesis abroad, and to all the staff in TUM who have helped me during these fantastic past months and have made my stay more enjoyable.

Finally, I would like to thank my parents, without whose support I would not have been able to have the fantastic opportunity to work in this thesis.

Abstract

Hyperspectral imaging (HSI) is a branch of image processing that deals with hyperspectral images, which are composed of a large number of channels covering the visible and near-infrared part of the electromagnetic spectrum. HSI is used in a lot of different fields and its use has greatly increased in the past few years. Image noise is a real concern in HSI, due to the way those kind of images are acquired. The aim of this project is to apply TV regularization for the denoising of hyperspectral images using Collaborative Total Variation (CTV), a new type of regularization which allows us to combine several norms to obtain different properties and behaviours from the denoising algorithm. A dual gradient projection algorithm is used to implement said variational method and the different norms are tested using a dataset of hyperspectral images. After testing, the conclusion reached is that the regularizer obtained by using the norm $\ell^{2,1,1}(col, der, pix)$ is the one which provides us with the best results. Results obtained open the door for possible further research in CTV applied to HSI for other processes like inpainting or unmixing.

Contents

1 Introduction

1.1 Hyperspectral imaging

Hyperspectral image (HSI) analysis has been, in the recent years, one of the most powerful and growing technology in the field of remote sensing. Hyperspectral images are images that represent information from across the visible and near-infrared part of the electromagnetic spectrum. While the human eye sees light as the superposition of three different colors, or bands, hyperspectral imaging divides the spectrum of light in many more bands, covering a wider range of wavelength spectrum. Capturing these images is done by special sensors and cameras. Unlike conventional color cameras, which capture light in the three spectral bands humans can process and see (referred to as RGB), hyperspectral cameras have the ability to capture a part of the electromagnetic spectrum at every pixel. One can imagine a hyperspectral image as a cube with dimensions (x,y,λ), where x and y are the two spatial dimensions of the scene and λ is the spectral dimension, each level corresponding to a different range of wavelengths in the electromagnetic spectrum. This description of an hyperspectral image can be seen more clearly in figure 1.1. The representation of the part of the electromagnetic spectrum captured in each pixel can be seen in the graphs next to the cube.

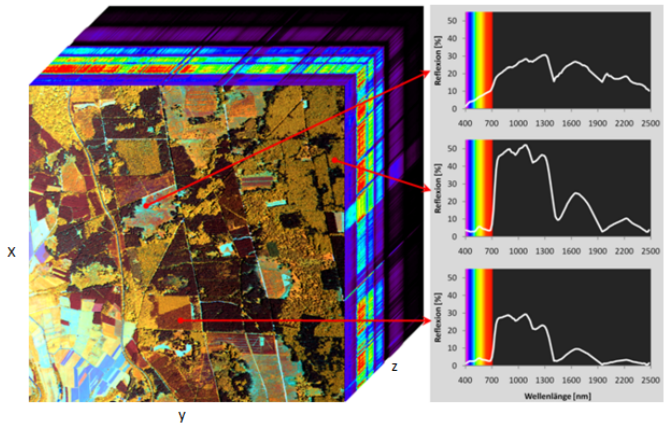


Figure 1.1: Data cube of a hyperspectral image. Source: Universität Trier

Processing and interpreting this type of images allows us to find materials or detect objects which could not be detected otherwise. Hyperspectral imaging is currently being used in fields like agriculture, astronomy, geology, biomedicine, physics, medicine, remote sensing or even government surveillance, among others, and it is still spreading into other areas because of its wider availability and the reducing cost of the technology needed.

A problem that arises when dealing with hyperspectral imaging is noise. HS images acquired by sensors are almost always disturbed by noise, because of the spatial distance between the aforementioned sensors and their targets. Hyperspectral image sensors are remote sensing systems, usually installed in a satellite. This means that the distance between the sensors and the source of the light we want to capture and process is usually quite large. This distance is filled with gases, vapor, particles, among other components, and all of them affect the light sensed by the systems. Also, we do not only get the light from the object we want to identify, but from other sources too: directly from the sun or reflected from other objects nearby, to cite some examples. For this reason, the light that reaches the sensors does not have the same composition as the light reflected by our objective: some wavelengths may have been absorbed by other objects, and others may have merged and reduced the detail from our desired band, and corrupt the information we want to gather.

1.2 Variational image processing

For the reason mentioned before, it is important to reduce noise in the raw images before processing their information. In the recent years several different methods for HSI denoising have been proposed. [14, 34, 19, 24, 3, 26]. The most popular classical regularization method in image processing is Total Variation (TV) [29]. This is a variational problem. In a variational problem in image processing, we design an energy functional E, mapping and image u to a real number $E(u)$ in a way that low energy corresponds to an image with the properties we desire, and viceversa. Once this functional is designed, we want to define an argument that minimizes $E(u)$:

$$\arg\min_{u \in V} E(u)$$

We will discuss the full energy functional in section 3.2.2. Part of that energy functional is the total variation of a signal or function, and can be defined as the integral over the absolute value of the gradient gradient. Let Ω be an open subset of $\mathbb{R}^n$, we define the TV of a function f as:

$$TV(f) = \int_{\Omega} |\nabla f(x)| \, \mathrm{d}x$$

Images with excessive noise are supposed to have high total variation [29], so by regularizing or reducing the TV of the signal we can get a closer match to the original signal or, in our case, image.

TV regularization has been used in a wide range of image processing fields. For example, color image denoising [32, 6], the process aforementioned in which the objective is to remove unwanted noise in the image, color image restoration, an inverse problem which aims to reconstruct an image from initial data [32, 21], or image inpainting, a process used to restore missing or heavily corrupted parts of the image and remove selected objets from the image, among others [1]. For HSI, TV has also been used for image decompression [17, 18], a procedure in which one tries to recover lost detail in a compressed hyperspectral image. This procedure is really important because hyperspectral images usually are huge in size, and compression is needed to deal with such amount of data. Unmixing [2, 25] is also an important process in HSI. In some hyperspectral images several different materials can be mixed in one pixel and this makes recognising their spectrum more difficult. Unmixing is the process to recover or unmix those materials. Those are only some examples of the processes in which TV is widely used.

The term Collaborative Total Variation (CTV) is a way to refer to a new class of regularization [16] that involves using different norms along the different dimensions (pixel, derivative and channel dimension, as defined in the paper) for the regularization term of the TV minimization. TV was defined originally on grayscale images, and efforts to extend it to multiple channels have been made [32, 6]. The aim of this project is to extend TV denoising from normal colored images, specifically the studies done in [16], to HSI, experimenting with different collaborative mixed matrix norms to achieve and find the best possible result.

Our objective is, then, to extend this regularization of the total variation to hyperspectral images, test it using various HSIs and regularization norms and compare the results. The main algorithm used will be a gradient projection algorithm (more information in chapter 4, using different regularization penalization norms to find out which one works best, comparing them by using image quality measures like peak signal-to-noise ratio and structural similarity, and finally extracting valid conclusions. This will be the main objective and focus of this project.

1.3 About the project

The algorithms and testing in this project will be done using the MATLAB environment. The advantages that this programming environment gives us, such as fast prototyping or several libraries and tools for image processing, makes it ideal for this kind of work. After talking about and explaining the mathematical background and concepts behind the project, we will present the algorithm used to implement those concepts and describe it. Then, we will talk about our testing methodology, present the results obtained on our data, compare and discuss said results and extract conclusions from them.

The organization of this thesis is as follows: in chapter 2 we will take an in-depth look at the state of the art to both TV regularization and hyperspectral image processing. we will compare and comment the available literature on the field and link it to the topic of this thesis. In chapter 3 the focus will be in detailing all the mathematical background in convex analysis and variational methods needed to gain a better understanding of the topic at hand. Next, in chapter 4, the focus will be on the algorithm we have chosen to implement the regularization. We will describe how it works and how it is implemented. Here we will also talk about the testing environment, the HSI images chosen for testing and the parameters of the algorithm that we will be using. Finally, in chapter 6 the results obtained are presented in proper plots and tables for a better understanding for the reader, said results are discussed and a proper conclusion is reached.

2 State of the Art

TV regularization as a method for denoising signals and images has received a fair deal of attention during the last two decades by researchers. In 1992, [29] Rudin et al. proposed a model for removing noise from images which involved a constrained minimization of the total variation of the image. The results presented were state of the art in image denoising at the time and served as a staring point for all kinds of research involving TV and multiple different applications and variations of the concept. This model was later improved by incorporating high-order derivatives [9, 10]. A logical next step for the method is the TV minimization for color images. The paper [6] describes how to apply TV to restore color images, with preserving edges and being rotationally invariant in the image space. Since then, TV has always been a competitive method for image denoising in color images.

In the recent years, a lot of different versions of TV minimization for denoising have appeared. In [23], a method which uses a Bregman iteration is presented, extending previous methods. [12] describes a TV method using box constraints which improves results with images that need projection after TV denoising to bring them back into their dynamic range. Even more recently, [22] presents a general total variation method that combines two different smoothing methods, with great results. In a different approach, [13] uses a proximity algorithm to solve the fractional-order TV image denoising model, a variation of the TV denoising model. In [21] a higher degree total variation model using weighted mixed norms of image derivatives is used for image restoration. We will use a similar concept in this project.

TV is not only used for denoising in the field of image processing: it has also been applied to image reconstruction or inpainting, methods we mentioned in chapter 1. An interesting study for color image restoration using TV regularization is the approach by Wen, Ng and Huang [32], where they use an alternating minimization algorithm with remarkable results. In their paper, [1] Afonso and Sanches announce a method for inpainting that uses total variation regularization to estimate the missing pixels of an image, and most interesting, that can be extended to denoising by changing the type of regularization. The data from experimentation show better results than other known methods. In a recent paper, [16], the concept of collaborative total variation was

defined and applied to color images, which showed that ℓ^∞ channel coupling obtained the best results.

Concerning hyperspectral imaging and looking at recent literature, a lot of different methods for HSI denoising have been proposed. For example, Chen and Qian [14] propose a denoising method for low noise HSIs that involves using principal component analysis (PCA) to detect and remove noise from the low energy PCA output channels. A different method has been using spectral-spatial kernel regularization [19, 34], whose most interesting feature is that it takes into account different noise distributions in different bands. The reliability of the proposed method in removing noise is demonstrated. Further strategies for HSI denoising have been proposed, based on wavelet transforms as presented in[24, 3, 26].

Total variation models have also been used in HSI. In a very recent work [20], a multidimensional nonlocal total variation model is proposed, taking into account the correlation of the spectral bands and the spatial structure similarity. Edge preservation and image denoising are both achieved with remarkably good results. A similar approach is taken in [33], taking also into account the different levels of noise intensity. Total variation has also been used in HSI for compressed image recovery and restoration. In [17, 18], Eason, et al. present convincing results using different methods on compressed image recovery with a method based on subgradients and total variation regularization. TV has also been used in hyperspectral images for feature extraction [26], which is an interesting use of the method.

Looking at previous work, our objective in this project is to extend the work done in [16] on CTV to hyperspectral images and experiment which norm works best. Rather than producing state of the art denoising results by incorporating nonlocal information, our goal is to investigate the optimal coupling of the different hyperspectral bands in the regularization. Such investigations have a potential to improve more sophisticated methods such as [20] or [33].

3 Background

The objective of this chapter is to talk about and explain the mathematical background which will be used and referenced throughout the project and that is needed to gain a better understanding of the algorithms and methods we will be using.

3.1 Convex Analysis

Convex analysis is a field in mathematics that studies convex functions and convex sets. The more usual application of convex analysis is on convex minimization. Most of the definitions and theorems described here are based on and extracted from the books *Convex Analysis* [28], and *Convex Optimization* [8]. A more in depth explanation and insight in convex analysis and optimization can be found there. In this project, we will only focus on the definitions and concepts that are useful and related to our work, presented in the most comprehensive way possible.

3.1.1 Convex functions

Convex functions are the main focus of study of convex analysis. Before defining a convex function, we will first state some basic useful concepts that will be needed for more complex definitions later:

Definition 3.1.1. For a function $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$, we call the domain of E the set:

$$dom(E) = \{u \in \mathbb{R}^n | E(u) < \infty\} \tag{3.1}$$

Definition 3.1.2. A function E is proper if its domain is not the empty set.

As we have said before, we will use these concepts later. Now, a convex function can be defined as follows:

Definition 3.1.3. [28, §2 p.10] A subset $S \subseteq \mathbb{R}^n$ is a convex set if for all $u, v \in S$ and all $\lambda \in [0, 1]$ it holds that:

$$(1 - \lambda)u + \lambda y \in S \tag{3.2}$$

Definition 3.1.4. [28, §4.1] A function $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$ is a convex function if:

1 dom(E) is a convex set. That is, the domain of the function E.

2 For all $u, v \in E$ and all $\lambda \in [0, 1]$ it holds that:

$$E(\lambda u - (1 - \lambda)uv) \leq \lambda E(u) + (1 - \lambda)E(v) \tag{3.3}$$

E is strictly convex if the inequality in 2 is strict, for all $u \neq v$ and $\lambda \in]0, 1[$.

An equivalent definition for convex functions can be described by using the concept of epigraph of a function.

Definition 3.1.5. [28, §4 p.23] Given a real-valued function f defined on a non-empty $M \subseteq \mathbb{R}^n \bigcup \{\infty\}$, the **epigraph** of f is:

$$Epi(f) = \{(t, x) \in \mathbb{R}^{n+1} \bigcup \{\infty\} | x \in M, t \geq f(x)\} \tag{3.4}$$

Knowing what the epigraph of a function is, one can understand the following more simple definition of convex function.

Theorem 3.1.1. *[28, §4 p.23] A function $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$ is a convex function if and only if its epigraph is convex as a subset of $\mathbb{R}^{n+1}$*

Definition 3.1.6. [28, §7.1] A convex function E is called **closed** if its epigraph is a closed set in $\mathbb{R}^{n+1}$.

In other words, a function defined on an interval is convex if, in a two dimensional setting, the segment between any two points on the graph of the function lies above the graph. Similar intuition also applies to higher dimensions. An example of a convex function is x^2. Regarding continuity of convex functions:

Theorem 3.1.2. *[28, §10.1] If the function $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$ is convex, then E is locally Lipschitz (and continuous) on $int(dom(E))$, where int is the interior of the set.*

From now on, for simplification purposes, every time we refer to a convex function, this will be proper.

3.1.2 Convex optimization

One of the main fields of study in optimization is convex optimization. Convex optimization, or convex minimization, studies the problem of minimizing convex functions over convex sets. Minimizing convex functions and not other type of functions offer some advantages that we will explain in this section. Most of the theorems and explanations found here are based on the book *Convex Optimization* [7] and *Convex Analysis*

[28], as mentioned earlier.

We will start by defining our problem. For a function $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$, which we call the energy functional, the minimization problem to optimize the function can be written as:

$$\hat{u} = \arg\min_{u \in \mathbb{R}^n} E(u) \tag{3.5}$$

For implementation, we will need to work with a discrete setting. More info on that can be found in chapter 4.

We now have to ask ourselves two questions in order to be able to find the minimum: First, if the minimum exists, and second, if said minimum is unique. For convex functions, we need to define the following concepts and properties first to properly define the existence of a minimizer:

Definition 3.1.7. [28, §7.1] A closed convex function $E : V \to\to \mathbb{R} \bigcup \{\infty\}$ is called lower semi-continuous if:

$$\liminf_{v \to u} E(v) \geq E(u) \tag{3.6}$$

holds, for any u.

Definition 3.1.8. [4, §2.31] We call $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$ coercive, if all sequences $(u_n)_n$ with $\|u_n\| \to \infty$ meet $E(u_n) \to \infty$.

With those new concepts, we can finally define the condition for a function to have a minimum:

Theorem 3.1.3. *[4, p. 2.32] If $E : \mathbb{R}^n \to \mathbb{R}$ is convex and coercive, then there exists:*

$$\hat{u} = \arg\min_{u \in \mathbb{R}^n} E(u) \tag{3.7}$$

This last theorem defines the existence of a minimizer for convex functions. About the uniqueness of convex minimization, we have the following one:

Theorem 3.1.4 (Uniqueness of convex minimization)**.** *[28, §27 p.264] If $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$ is convex, then any local minimum is a global minimum. If E is strictly convex, that global minimum is unique.*

Now the existence and uniqueness of the solution to the minimization of a convex function have been properly defined. We want also to describe one type of convex problems that are relevant to our work: saddle point problems. This will be needed during the development of the algorithm in chapter 4:

Theorem 3.1.5. *[28, §33 p.264] Let C and D be non-empty closed proper convex sets in $\mathbb{R}^n$ and $\mathbb{R}^m$, and let F be a continuous finite convex function on $C \times D$. If either C or D is bounded, then:*

$$\inf_{v \in C} \sup_{q \in C} F(v, q) = \sup_{q \in C} \inf_{v \in C} F(v, q) \tag{3.8}$$

The next step is to find a method on how to find said minimizer. To reach that point, we will first talk about the concept of subdifferential calculus.

3.1.3 Subdifferential calculus

In simple words, the subdifferential of a function is the generalization of the derivative to functions that are not differentiable. For example, the absolute value $f(x) = |x|$ is nondifferentiable when $x = 0$. Subdifferentials appear in convex analysis and are of utmost importance in solving minimization problems. Formally, the subdifferential of a function E can be defined as:

Definition 3.1.9. [28, §23 p.214]

$$\partial E(u) = \{\rho \in \mathbb{R}^n | E(v) - E(u) - \langle \rho, v - u \rangle \geq 0, \forall v \in \mathbb{R}^n\} \tag{3.9}$$

If $\partial E(u) \neq \emptyset$, then we say that E is subdifferentiable at u.

The theorem that interest us is:

Theorem 3.1.6 (Optimality Condition)**.** *[28, §27 p.264] Let $0 \in \partial E((\hat{u}))$ and E is convex. Then $\hat{u} \in \arg\min_u E(u)$.*

We now have an optimality condition that helps us to solve minimization problems. Expanding on the topic of subdifferential calculus, its important to note the relationship between subderivatives and derivatives. If a convex function $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$ is differentiable in $x \in dom(E)$, then:

$$\partial E(x) = \{\nabla f(x)\} \tag{3.10}$$

3.1.4 Duality

In mathematics, any vector space has a corresponding dual vector space. We will proceed by assuming that the reader already has some basic knowledge and understanding of dual space. Chapter 3 in [28] contains an in-depth explanation and insight on duality applied to convex analysis. Here we will only talk about the points and concepts used in this project.

We will start by defining the convex conjugate of a function:

Definition 3.1.10. [28, §12 p.104] We define the convex conjugate of the function $E : \mathbb{R}^n \to \mathbb{R} \bigcup \{\infty\}$ as

$$E^*(\rho) = \sup_{u \in \mathbb{R}^n} (\langle u, \rho \rangle - E(u)) \tag{3.11}$$

The conjugate of a convex function is also convex. Some other properties of the conjugates are:

Theorem 3.1.7 (Fenchel-Young Inequality)**.** *[8, §3.3.2] Let E be proper, convex and lower semi-continuous $u \in dom(E) \subset \mathbb{R}^n$, and $p \in \mathbb{R}^n$, then*

$$E(u) + E^*(\rho) \geq \langle u, \rho \rangle \tag{3.12}$$

Equality holds if and only if $\rho \in \partial E(u)$.

Theorem 3.1.8 (Biconjugate)**.** *[28, §12.2] Let E be proper, convex and lower semi-continuous, then $E^{**} = E$.*

Theorem 3.1.9 (Subgradient of convex conjugate)**.** *[28, §23.5] Let E be proper, convex and lower-semi continuous, then the following two conditions are equivalent:*

1. $\rho \in \partial E(u)$
2. $u \in \partial E^*(\rho)$

And most important:

Theorem 3.1.10 (Fenchel's Duality Theorem)**.** *[28, §31.1] Let f and g be proper convex functions on $\mathbb{R}^n$. One has:*

$$\inf_x \{f(x) + g(x)\} = \sup_x^* \{-f^*(y^*) - g^*(x^*)\} \tag{3.13}$$

if either of the following conditions is satisfied:

- $$ri(dom(f)) \cap ri(dom(g)) \neq 0$$
- *f and g are closed, and*

$$ri(dom(f^*)) \cap ri(dom(g^*)) \neq 0$$

This theorem, and duality as a whole, allows us to perform useful transformations of convex problems, making them easier to compute and translate to real algorithms. An application of this theorem will be explored in chapter 4.

3.2 Variational Methods

Variational calculus is the field of calculus that deals with functionals on vector spaces. One can think of a variational problem as the minimization of a integral functional E, usually called energy functional, over an argument $u \in V$ with several constraints and penalties that determine our problem:

$$\arg\min_{u \in V} E(u) \tag{3.14}$$

Variational calculus has gained popularity in the last years with the availability of powerful computational resources, that allows researchers to work on more complex problems. There exist a wide range of applications, problems, algorithms and open problems where variational methods are of utmost importance. Here we will only focus on the problem we are working on, which is TV regularization and on a discrete setting. For more context on the topic, as well as on analysis on the continuous setting, we refer the reader to the book *Variational Analysis* [27].

3.2.1 Total variation

Total Variation regularization (TV) is a variational problem in which the problem solution is the minimizer of an energy functional E, as explained before. In our case, the energy functional is the total variation of a signal or function f, which can be defined as the integral over Ω, which is an open subset of $\mathbb{R}^n$, of the gradient of the function:

$$TV(f) = \int_{\Omega} |\nabla f(x)| \, \mathrm{d}x \tag{3.15}$$

Reducing this energy functional is interesting for a reason: in [29], it was shown that images with a lot of noise are supposed to have high TV, so by trying to reduce the TV and staying close to the original image at the same time, we can reduce the noise.

Total variation has the properties of being lower-semi continuous, convex and homogeneous [11] [1]. This will prove to be useful when defining the minimization equation, down below. For a continuous setting analysis of TV, also refer to [11].

[1] Those properties will not be proven here. Refer to the paper [11] for proofs and more details on total variation.

3.2.2 Total variation minimization

Now, let us move onto our problem. For a noisy imput image f, we need to find an image u which has to be similar to f but with less total variation. The most well-known model for this variational problem, and the one which we will be using as base model, is the so called Rudin-Osher-Fatemi model [29], or ROF-model, for grayscale image denoising:

$$u(\alpha) = \arg\min_{u \in \mathbb{L}^2(\Omega)} \frac{1}{2} \int_\Omega (u - f)^2 dx + \lambda \int_\Omega |\nabla u| dx \tag{3.16}$$

This equation is divided in two terms. The first one is called the data or model term, and is the one that restricts the final result to be close to the original image. The second one is called the regularizer, and is the term that penalizes high total variation and force the minimum to have a low total variation. λ is the regularization parameter, which weights between the two terms. This parameter is of utmost importance, as it is usually the parameter that needs to be adjusted in order to obtain a better result.

3.2.3 Norms and projections

First of all, we will define what a norm is.

Definition 3.2.1 (Norm)**.** [8, §A1]
A function $f : \mathbb{R}^n \to \mathbb{R}$ with $dom(f) = \mathbb{R}^n$ is a norm if it fulfills the following properties:

- f is nonnegative: $f(x) \geq 0$ for all $x \in \mathbb{R}^n$.
- f is definite: $f(x) = 0$ only if $x = 0$.
- f is homogenuous: $f(tx) = |t|f(x)$ for all $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$.
- f is satisfies the triangle inequality: $f(x + y) \leq f(x) + f(y)$, for all $x, y \in \mathbb{R}^n$.

Examples of norms are the absolute value or the ℓ_2 norm. This last norm is a part of a family of norms called the ℓ_p-norm, defined by:

$$||x||_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p} \tag{3.17}$$

We are going to use norms for our different penalizations on TV. As seen in equation (3.16), the TV in the regularizer term is penalized by a ℓ_2 norm. By changing this norm, we can penalize the TV in different ways and obtain different results. As explained in chapter 1, this is one of the main aspects in the project and it will be mentioned and worked on several times in the next chapters.

3.2.4 Collaborative norms

Collaborative norms are a relatively new conc ept in the variational analysis field. Described in [16], they are a new class of regularization using mixed norms. We will not go into a lot of detail on the topic during this project, as most of the mathematical demonstration, properties and definition surpasses the scope of it, but a basic definition of the concept will be provided.

For a color image (note that is also true for a hyperspectral image) in a discrete setting, the gradient of the image can be described as a three dimensional matrix. One dimension corresponds to the space or pixels of the image or signal. Another one corresponds to the directional derivatives containing the differences between pixels. We can define the directional derivatives for values of the pixels x and channels y with the Jakobi matrix:

$$\mathbf{D}u = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \tag{3.18}$$

Note that, if we only have one channel, we obtain the following matrix:

$$\mathbf{D}u = \begin{bmatrix} \frac{\partial \mathbf{y}}{\partial x_1} & \cdots & \frac{\partial \mathbf{y}}{\partial x_n} \end{bmatrix} \tag{3.19}$$

Which is the gradient of the image, as seen in (3.16) for grayscale images (only one channel). It is clear to see now that we are generalizing the original equation for any number of channels.

The last dimension is the channel dimension (3 for color images, many more for hyperspectral images). Then, to compute the TV of such image, one can penalize each dimension in a different way depending on the norm we use. Combining different norms for the different dimensions allows us to obtain different results with the regularization.

To put an example, considering a three dimensional matrix $A \in \mathbb{R}^{N \times M \times C}$, where N is the number of pixels, M the number of derivatives, and C the number of channels, one can apply the ℓ^p norm to the channel dimensions, the ℓ^q norm to the derivative dimension and the ℓ^r norm to the final pixel dimension. The expression obtained will

be referred as the $\ell^{p,q,r}$ norm:

$$||A||_{p,q,r} = (\sum_{i=1}^{N}(\sum_{j=1}^{M}(\sum_{k=1}^{C}|A_{i,j,k}|^p)^{q/p})^{r/q})^{1/r} \tag{3.20}$$

In the next section we will explain and describe the collaborative mixed norms we will use during the project.

3.2.5 TV variations

As explained in the section before, one can use a wide variety of norms on the regularization to affect the algorithm. In the experiments of this thesis we will use different versions of the algorithm by modifying the type of TV regularization used. By changing the penalization norms used in the initial expression, a different projection is obtained in the final algorithm (see chapter 4 for more details) depending on the initial total variation term we use. For convenience, a discrete setting is used for the description of the norms. The dimensions of the term will correspond to the ones mentioned earlier. The notation of the different norms used will be based on the framework described in [16]: we refer to each norm as $\ell^{p,q,r}(dim1, dim2, dim3)$, with p, q, r being the norms applied to dimension $dim1, dim2, dim3$ respectively.

- $\ell^{1,1,1}(der, pix, col)$

  The expression for this collaborative norm in the discrete setting is;

$$TV(u) = \sum_{i,j}|D_x u_{ij}| + |D_y u_{ij}|. \tag{3.21}$$

- $\ell^{2,1,1}(col, der, pix)$

  The same as before, but now we perform a ℓ^2 norm alongside the channels or colors of the image. We use this total variation expression, with the dimensions ordered by channels, derivatives and pixels:

$$\mathrm{TV}(u) = \sum_{i}^{m}\sum_{j}^{n}\left(\sqrt{\sum_{k}^{c}(D_x u_{ij}^k)^2} + \sqrt{\sum_{k}^{c}(D_y u_{ij}^k)^2}\right) \tag{3.22}$$

- $\ell^{2,2,1}(col, der, pix)$

This is basically a $\ell^{2,2,1}$ norm, coupling the channel dimensions and the derivative dimension with the ℓ^2 norm.

$$\mathrm{TV}(u) = \sum_{i}^{m}\sum_{j}^{n}\sqrt{\sum_{k}^{c}(D_x u_{ij}^k)^2 + (D_y u_{ij}^k)^2} \tag{3.23}$$

- $\ell^{2,1,1}(der, col, pix)$

  This total variation is similar to the ones before, with the only change being that we do not couple the channel, only the derivative dimension:

$$\mathrm{TV}(u) = \sum_{i}^{m}\sum_{j}^{n}\sum_{k}^{c}\sqrt{(D_x u_{ij}^k)^2 + (D_y u_{ij}^k)^2} \tag{3.24}$$

- $\ell^{\infty,1,1}(col, der, pix)$

  Here the norm is $l^{\infty,1,1}$, alongside the channel or spectral dimension:

$$TV(u) = \sum_{i,j}\max_k |D_x u_{ij}^k| + \max_k |D_y u_{ij}^k|. \tag{3.25}$$

- $\ell^{\infty,1,1}(der, col, pix)$

  This norm is similar to the previous one, but we do ℓ^∞ across the derivative dimension, not the channel dimension:

$$TV(u) = \sum_{i,j}\sum_{k}\max(|D_x u_{ij}^k|, |D_y u_{ij}^k|). \tag{3.26}$$

- $\ell^{\infty,\infty,1}(col, der, pix)$

  Similar to the two norms mentioned before, the norm implemented in the algorithm this time is of the form $\ell^{\infty,\infty,1}$, where we apply the ℓ^∞ norm to both the derivative and the channel dimension.

$$TV(u) = \sum_{i,j}\max(\max_k |D_x u_{ij}^k|, \max_k |D_y u_{ij}^k|). \tag{3.27}$$

4 Total Variation for Hyperspectral Images

In this section we will describe the implementation of the TV regularization and the different norms mentioned in the previous chapter. We will describe the algorithm used and discuss its implementation in MATLAB.

4.1 Gradient projection algorithm

We will start by describing the problem: starting with a noisy image f, we need to find an image u which has to be similar to f but with less total variation. The model we used to define the problem is the one I talked about in the previous chapter, in section 3.2.2:

$$\hat{u}(\alpha) = \arg\min_{u \in \mathbb{R}^{mxn}} \frac{1}{2}\|u - f\|_2^2 + \alpha\|Ku\|_1 \tag{4.1}$$

Note the similarity with equation (3.16). The main difference is that we have moved from a continuous setting to a discrete setting. In (4.1), K is the matrix we use to discretize the gradient of u. Also, for the description of the algorithm, I will be using a ℓ^1 norm as penalizer the regularizer term. This will be done mainly for simplicity. Changing this norm for the collaborative norms described in the previous chapter will also change the behaviour of the regularization. We will talk about this in section 4.2.1.

This is a convex optimization problem: since both terms of the function are convex, we can use convex optimization to minimize it and find u, which will exists thanks to theorem 3.1.3. To find the expression that minimizes u, we approach the problem using the theorem of optimality condition of a function, defined in theorem 3.1.6. Finding the minimizer with this theorem is numerically hard to find, so another approach must be found. To do this, we will apply the biconjugate theorem (Theorem 3.1.8) to the regularization term of the initial expression, in order to transform it to a more manageable one:

$$\min_u \frac{1}{2}\|u-f\|_2^2 + \alpha\|Ku\|_1 \tag{4.2}$$

$$= \min_u \frac{1}{2}\|u-f\|_2^2 + \alpha \sup_{\|q\|_\infty \leq 1} \langle Ku, q\rangle \tag{4.3}$$

If I now apply theorem 3.1.5, the following expression can be obtained:

$$\sup_{\|q\|_\infty \leq 1} \min_u \frac{1}{2}\|u-f\|_2^2 + \alpha\langle Ku, q\rangle \tag{4.4}$$

Compared to equation (4.1), the expression we obtain after applying the optimality condition 3.1.6 for finding the optimum is much more simpler:

$$u_{opt} = f - \alpha K^T q \tag{4.5}$$

Now I want to solve the inner problem in (4.4) with the optimality condition from before. The following expression is obtained:

$$\sup_{\|q\|_\infty \leq 1} \frac{1}{2}\|f-\alpha K^T q - f\|_2^2 + \alpha\langle K(f-\alpha K^T q), q\rangle \tag{4.6}$$

And, by developing the equation:

$$\sup_{\|q\|_\infty \leq 1} \frac{1}{2}\|\alpha K^T q\|_2^2 + \alpha\langle Kf, q\rangle - \|\alpha K^T q\|_2^2 \tag{4.7}$$

We add $(-\frac{1}{2}\|f\|^2 + \frac{1}{2}\|f\|^2)$ to the expression before in order to reduce to the square of a binomial:

$$\sup_{\|q\|_\infty \leq 1} \frac{1}{2}\|\alpha K^T q\|_2^2 + \alpha\langle Kf, q\rangle - \|\alpha K^T q\|_2^2 - \frac{1}{2}\|f\|^2 + \frac{1}{2}\|f\|^2 \tag{4.8}$$

$$= \sup_{\|q\|_\infty \leq 1} -\frac{1}{2}\|\alpha K^T q - f\|_2^2 + \frac{1}{2}\|f\|^2 \tag{4.9}$$

The interest is in the argument $\hat{q}$ that solves the problem, since we know that the optimality 4.5 solves 4.4. For this reason, the last term of our last expression $(\frac{1}{2}\|f\|^2)$ can be left out, as it is not relevant for the argument that maximizes q. It is easier to work with minimizations than maximizations, so the following transformation is

performed on the equation:

$$\hat{q} = \arg\max_{\|q\|_{\inf}\leq 1} -\frac{1}{2}\|\alpha K^T q + f\|_2^2 \tag{4.10}$$

$$\hat{q} = \arg\min_{\|q\|_{\inf}\leq 1} \frac{1}{2}\|K^T q^k + \frac{f}{\alpha}\|_2^2 \tag{4.11}$$

To solve equation (4.1), and find the optimal, a wide range of methods can be used. As explained before, I will be using the gradient descent algorithm described in [8, §9.3]. A more in depth explanation about the algorithm can be found there. Basically, the idea behind the algorithm is that, to find the minimum of a differentiable function $F(x)$, one can move towards the direction of the negative gradient, such that:

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \gamma \nabla F(\mathbf{x_k}) \tag{4.12}$$

For a time step γ, a starting point $\mathbf{x_1}$ and a number of steps k. Applying this to equation , we only need to project each step onto a feasible set to keep the condition $\|q\|_{\inf} \leq 1$ and we have our final algorithm:

$$\hat{q}^{k+1} = \Pi_{\|\cdot\|_{\inf}\leq 1}(q^k - \gamma K(K^T q^k + \frac{f}{\alpha})), \text{with} \tag{4.13}$$

$$u^k = f - \alpha K^T q^k \tag{4.14}$$

As a remark, the time step size has to be $\gamma < 1/4$ for the TV minimization to converge.

4.2 Implementing and adapting the algorithm to hyperspectral images

It is not complicated to adequate the previous algorithm to work with spectral images, this is, images with more color channels. The two big things to take in account are the gradient matrix K and the projection. To implement the gradient, we can use a sparse matrix multiplication.

We need to create the matrix K and then vectorize the original image. We create K by defining two diagonal matrix with function $spdiags()$ of size $m \times m$ and $n \times n$ and then we use the function $kron()$ to combine them and create the sparse matrix. The resulting K will have a size of $[2mn, mn]$.

```
1
2  % We create the sparse differencing matrix
3  e = ones(m,1);
4  yDerMat = spdiags([-e e], 0:1, m, m);
5  yDerMat(end) = 0;
6  e = ones(n,1);
7  xDerMat = spdiags([-e e], 0:1, n, n);
8  xDerMat(end) = 0;
9  K = [kron(xDerMat',speye(n,n)); kron(speye(m,m), yDerMat)];
10
11 % We vectorize the image alongside each channel
12 F = reshape(F,m*n,c);
```

By reshaping the original image to $[mn, c]$ (putting each channel in one dimension, like a vector and not a matrix) will make our algorithm to work properly for hyperspectral images. Obviously, after running the algorithm we will have to reshape the result into a proper image.

To implement the projections, the fact that we will use CTV norms must be considered. In the following section we will talk about the the different norms commented in section 3.2.5 and how to implement them in MATLAB. Thanks to the advantages this environment provides relating images and matrices, this is not a difficult task.

4.2.1 TV variations

Depending on the norm chosen for the regularization term in equation (4.1), one will obtain a different projection in equation (4.13). More specifically, the projection obtained is the projection over the $\ell^{p,q,r}$ unit ball, with $\ell^{p,q,r}$ corresponding to the conjugate (definition 3.1.10) of the initial norm, as seen in the description of the algorithm. Here we will describe first the numerical and then the MATLAB implementation of the collaborative norms described in section 3.2.5.

- $\ell^{1,1,1}$ (der,pix,col)

  The projection resulting from the dual conjugate from the norm (3.21) is the form $||p||_\infty \leq 1$. To implement it, and knowing that the ∞-norm is defined as:

$$||p||_\infty = \max_i |p_i| \tag{4.15}$$

We have to limit every element of q to be 1 or less. This can easily be implemented in MATLAB:

```
q_out = sign(q).*min(abs(q),1);
```

This expression will work regardless of q being a vector, matrix, or high-dimensional matrix, thanks to the way MATLAB works with matrices.

- $\ell^{2,1,1}$ (col,der,pix)

  The resulting projection from the norm described in 3.22 will be $||p||_{2,\infty,\infty} \leq 1$. The 2-norm is defined as:

$$||p||_2 = \sqrt{\sum_{k=1}^{n} |x_k|^2} \tag{4.16}$$

  We need to group the dimensions on a ℓ^2 norm and then penalize the image vector by dividing the original image by that grouping to make sure that the condition ≤ 1 is met. This can be implemented in MATLAB with the following piece of code:

```
temp = sqrt(sum(q.^2,2));
q_out = q./repmat(max(temp, 1), [1,size(q,2)]);
```

- $\ell^{2,2,1}$ (col,der,pix)

  The projection obtained is $||p||_{2,2,\infty} \leq 1$. We can implement this in a similar way to the previous one:

```
[m,n,~] = size(q);
q = reshape(q, [m/2,n*2]);
temp = sqrt(sum(q.^2,2));
q = q./repmat(max(temp, 1), [1,size(q,2)]);
q_out = reshape(q,[m,n]);
```

  To understand this code, it is important to remember that the original image, which has a size $[m, n, c]$ has been resized to $[mn, c]$. So, by this point, what is being that is that the derivatives and the channels for each pixel are put alongside the second dimension to do the ℓ^2 norm on them. After the projection, of course, the image needs to be restored to its proper shape.

- $\ell^{2,1,1}$ (der,col,pix)

  The resulting projection from the collaborative norm described in equation 3.24 translates to a $||p||_{2,\infty,\infty} \leq 1$ projection and it is implemented in a similar way as before, reshaping the image:

```
[m,n,~] = size(q);
q = reshape(q, [m/2,2,n]);
temp = sqrt(sum(q.^2,2));
q = q./repmat(max(temp, 1), [1,size(q,2)]);
q_out = reshape(q,[m,n]);
```

  In this case, we put the channels alongside the third dimension so that we do not apply the ℓ_2 norm on them.

- $\ell^{\infty,1,1}$ (col,der,pix)

  When we translate this norm to the gradient projection algorithm, the resultant projection is $||p||_{1,\infty,\infty} \leq 1$. To implement the projection $||p||_1 \leq 1$, a more complex algorithm is needed. We have used the algorithm described in the paper [15] for projecting a vector onto the ℓ^1-ball. The algorithm is as follows:

Algorithm 1: Projection onto the l_1 Ball [15].

Input : A vector $v \in \mathbb{R}^n$ and a scalar $z > 0$

Sorting v into $\mu : \mu_1 \geq \mu_2 \geq \cdots \geq \mu_p$;

Find $\rho = \max\left\{j \in [n] : \mu_j - \frac{1}{j}\left(\sum_{r=1}^{j} \mu_r - z\right) > 0\right\}$;

Define $\theta = \frac{1}{\rho}\left(\sum_{i=1}^{\rho} \mu_i - z\right)$;

Output: β such that $\beta_i = max\{v_i - \theta, 0\}$

w such that $w_i = sign(v_i)\beta_i$

  For more information on the algorithm, the idea behind them, and extensions of it, refer to [15].

  We want to extend the algorithm for ℓ^1-projection to work for 2-dimensional vectors (matrices), in order to insert it into the code. To do this, we apply the algorithm to each column of the matrix: each column is sorted and then p and θ is computed for each column. For this, we transpose the image at the beginning of the algorithm to work with the columns, as it makes more sense for coding:

```
    q = q';
    [m,n,~] = size(q);
    % We sort each column
    v = sort(abs(q),1,'descend');
    w = cumsum(v,1);
    i = repmat((1:m)', [1,n]);
    P = ((v - (w - z)./i) > 0);
```

After this piece of code we have a logical matrix P with the indexes of the vector needed to compute θ. To get the values in an efficient way we proceed as follows: First, we switch the value of the logical matrix. After doing this, we use the MATLAB function $max()$ to look to the first 1 that appears, and then substract 1 to get the right index. Using this procedure is fast, but one extreme case appears that need to be dealt with: when all the values in a column in P are 1. In this case, when the values are changed, an index is obtained pointing to the first element of the column, while the wanted element is the last. To solve this, we just append an extra row of 1 at the end of the matrix:

```
    % We add an extra row of ones to take care of the extreme case
    P = cat(1,double(1-P),ones(1,n));
    [~,p] = max(P);
    p = max(1,p - 1);
    pl = p + (0:n-1)*m;
    theta = max(0,(w(pl) - z) ./ p);
```

Once the appropriate θ for each column is computed, we just need to proceed as with previous projections, and transpose again to obtain the original matrix:

```
    q_out = sign(q).*max(abs(q) - repmat(theta, [m,1]), 0);
    q_out = q_out';
```

- $\ell^{\infty,1,1}$ (der,col,pix)

  To compute the projection conjugate of the norm (3.26), defined as $||p||_{1,\infty,\infty} \leq 1$, we proceed as before but modifying the entry matrix such that the derivative are located on the columns, and then compute the projection for each different channel:

```
    [m,n,~] = size(q);
    q = reshape(q, [m/2,2,n]);
    for i = 1:n
        q_out(:,:,i) = ProjectionL1Ball(q(:,:,i), z);
    end
    q_out = reshape(q_out,[m,n]);
```

- $\ell^{\infty,\infty,1}$ (col,der,pix)

  The projection $||p||_{1,1,\infty} \leq 1$ resulting from the dual conjugate of the norm (3.27) is implemented by putting both the derivative and the color dimension alongside the columns of the matrix for our projection algorithm to work:

```
    % q will be [n*m*2,c]
    [m,n,~] = size(q);
    q = reshape(q, [m/2,2*n]);
    q_out = ProjectionL1Ball(q,z);
    q_out = reshape(q_out, [m,n]);
```

5 Testing Methodology

5.1 Introduction

After describing the algorithm and the different projections that we will be using, the next thing to do is to describe the sample hyperspectral images tested, and the measures of image quality we will use to compare the results.

5.2 Sample images

For the images, we have employed a set of images from the database of hyperspectral images from the University of Stanford [31]. The images cover wavelengths from 0.4 to 2.5 micrometers, spanning the visible, NIR and SWIR electromagnetic spectral ranges. The images are contained in .mat files, which makes them easier to process in MATLAB. The format of the data is a matrix of size n x m x w, where n and m are the rows and columns of the image, and w is the number of spectral bands. The database also provides us with extra information, like the scene radiance data, the scene illuminant and a vector listing the wavelenghts of the bands stored in w.

The criteria for choosing which hyperspectral image to use is simple: we could not simply use only one image, and the reasoning behind that is obvious, so we need a good enough number of hyperspectral images. Also, the images used should not be too similar, in order to avoid strange behaviour of our algorithms tied to a specific image. The images we will be using are *LoResFemale*2.*mat*, an image of a female face; *SanFrancisco.mat*, a picture from the city of San Francisco, and *StanfordMemorial.mat*, a picture from the facade of a Church. You can see a colored version of the pictures we will be using in figure 5.1 .

(a) Female

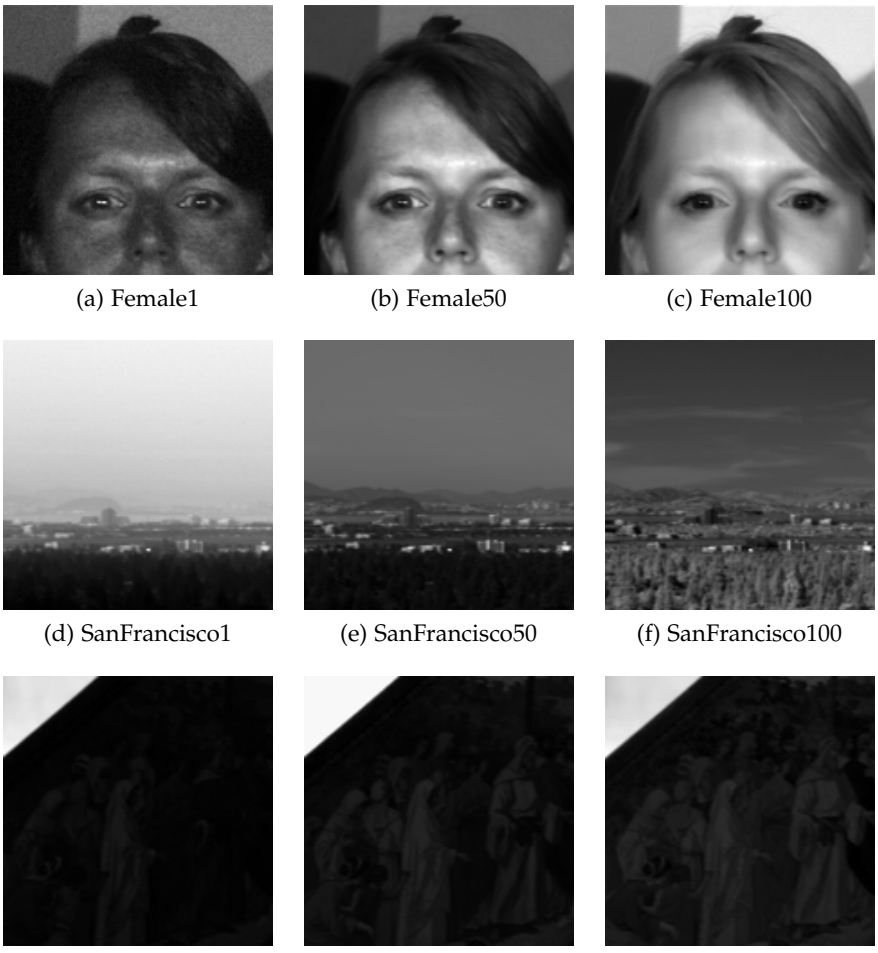
(b) SanFrancisco

(c) Stanford

Figure 5.1: Color renditions of the hyperspectral images used in the project

5.2.1 Preprocessing of the images

A problem arises when one has to work with hyperspectral images: their size is usually too big to process them in a reasonable amount of time. The solution is to reduce the original image to a manageable size. To do this, we first perform a *resize* of the image by 0.5 and then we crop a 150×150 section of the resized image. By doing those processes, we minimize the quality loss from the original image, while reducing its size to be more manageable. We also will scale the range of the values in the image matrix to [0,1] for convenience and easiness in working with them. You can see the cropped pictures that will be used in the project in figure 5.2. As it is difficult to accurately portrait an hyperspectral image on a paper, we have included pictures of several bands: for each image, bands number 1, 50 and 100 are displayed.



(a) Female1 (b) Female50 (c) Female100

(d) SanFrancisco1 (e) SanFrancisco50 (f) SanFrancisco100

(g) Stanford1 (h) Stanford50 (i) Stanford100

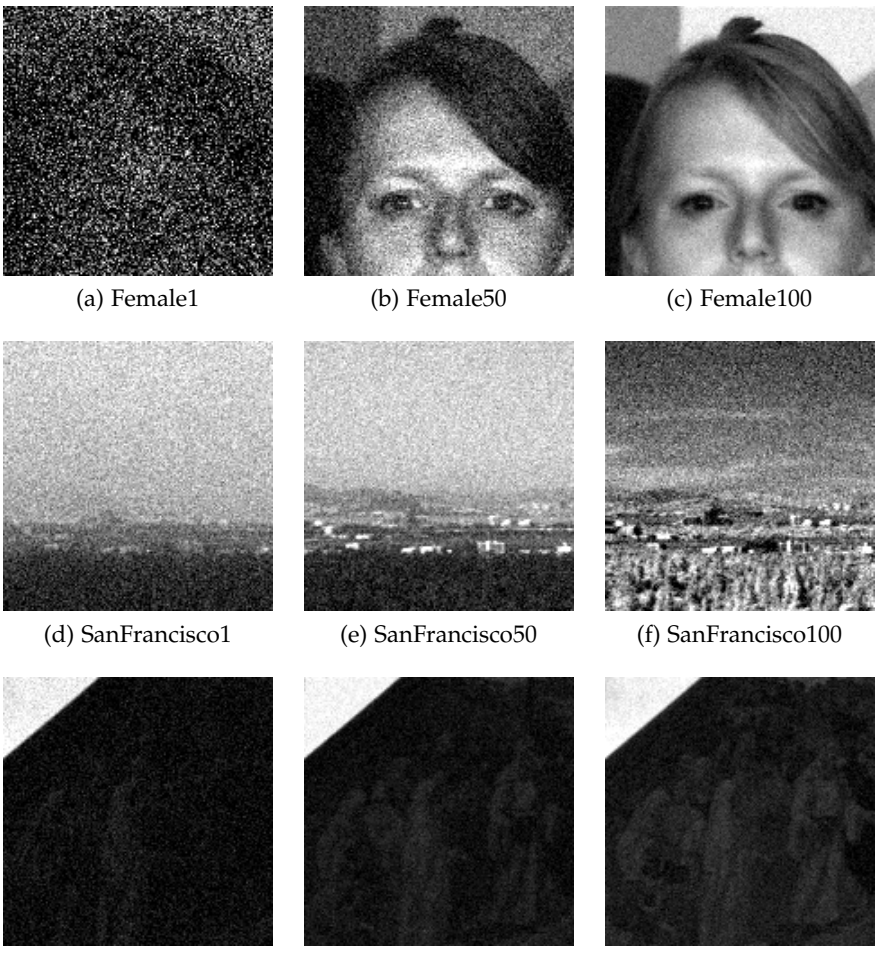
Figure 5.2: Bands 1, 50 and 100 of the three images used for testing

5.2.2 Noise model

To be able to evaluate the performance of the algorithm and the different regularizations, we need to create a synthetic data set by artificially adding noise to the images. The model used for the noise is a normal distribution of random numbers generated by the MATLAB function *randn*() and multiplied by an scalar *noise* that allows us to modify the final noise level:

```
1    noised = image + noise*randn(size(image));
```

We will be testing two sets of images with different noise values. This will allow me to test the algorithm in two different noise environments and see the robustness of the model for different noise levels. The values of the noise will be defined by the value of the scalar *noise*: 0.05 and 0.025 for high and low noise, respectively. In figures 5.3 and 5.4 you can see the previous bands of each image after adding the noise. Note that the images have been brightened using the MATLAB function *imadjust*() in order to obtain a better clarity.

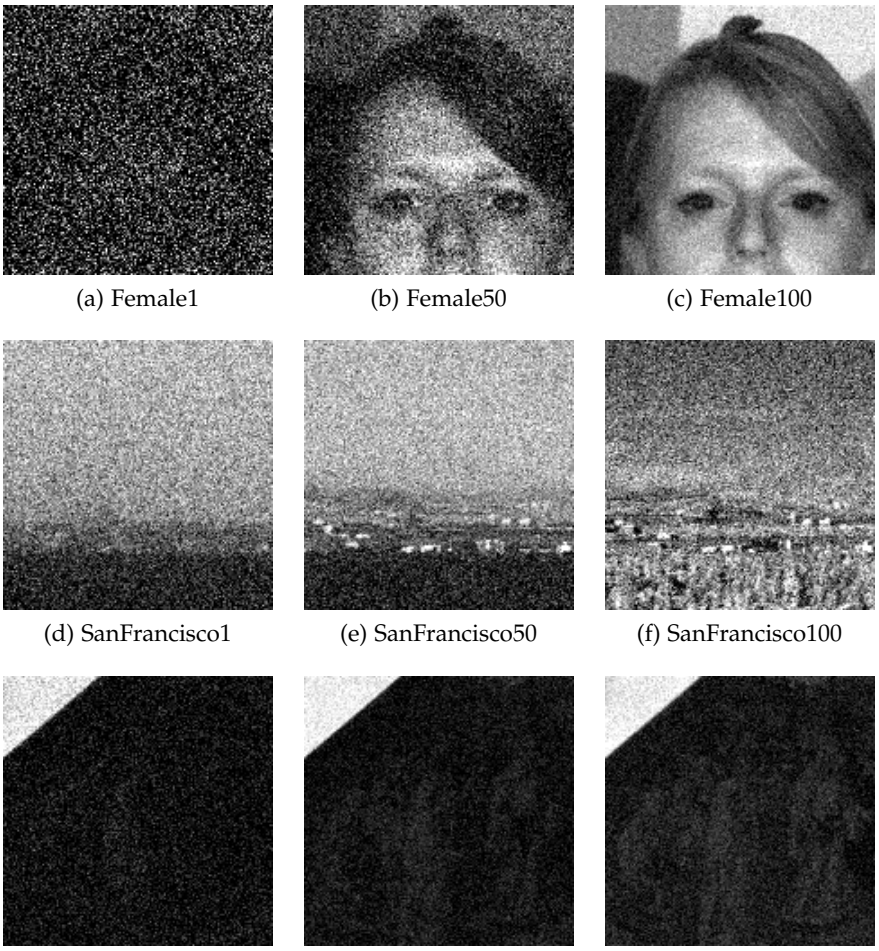


(a) Female1 (b) Female50 (c) Female100

(d) SanFrancisco1 (e) SanFrancisco50 (f) SanFrancisco100

(g) Stanford1 (h) Stanford50 (i) Stanford100

Figure 5.3: Bands 1, 50 and 100 of the three images used for testing, after adding low-value noise



(a) Female1 (b) Female50 (c) Female100

(d) SanFrancisco1 (e) SanFrancisco50 (f) SanFrancisco100

(g) Stanford1 (h) Stanford50 (i) Stanford100

Figure 5.4: Bands 1, 50 and 100 of the three images used for testing, after adding high-value noise

As one can see in the figure, the different bands have significantly different signal-to-noise ratios: some have more distortion than others. This can happen because in lower bands there is less difference between the values of the pixels (as one can see in the original images) and, exposed to random noise, this can lead to a higher level of distortion. In higher bands one can see that the noise, while not as pronounced, is still present.

5.2.3 Measures of image quality

To compare the results we obtain, just relying on our subjective direct view of the denoised images is not enough. For this reason, we will be using several measures and comparison methods to be able to make a more objective approach to comparing the different results. The methods we will be using are

PeakSNR

The first measure of image quality we will use to classify our projections is the Peak Signal-to-Noise ratio (PSNR) (cf. [30]). It is the ratio between the maximum possible value of a signal and the power of the noise that affects the quality of its representation. In simpler words, it measures the amount of noise the image has. It is expressed in dB.

Mathematically speaking, here is the expression that defines PSNR:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{5.1}$$

$$= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \tag{5.2}$$

$$= 20 \cdot \log_{10} (MAX_I) - 10 \cdot \log_{10} (MSE) \tag{5.3}$$

MAX_I is the maximum possible pixel value of the image, which equals 1 in our case. MSE is the mean squared error. Given a noise free image I and a noisy image J, of size $m \times n$, we define MSE as:

$$MSE = \frac{1}{m\,n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - J(i,j)]^2$$

In the case of multiple channels, like in color images or hyperspectral images, the definition is the same, and we just divide all the squared value differences between channels by the image size times the number of channels. A higher PSNR value means that the images are more similar. If both images were identical, the MSE would be 0, and the PSNR would be infinite (or division by zero)

We have used the command $psnr()$ in MATLAB to compare the different denoised imagesto the corresponding noise free images.

Structural similarity

The structural similarity index (SSIM) is a method to measure similarity between two images. Contrary to PSNR, SSIM values structural information: the method considers that pixels have inter-dependencies between them, that are stronger when they are spatially close.

The difference with respect to other techniques mentioned previously such as MSE or PSNR is they are often accused of not measuring the perceived visual quality. SSIM considers image degradation as perceived change in structural information. Structural information is the idea that the pixels have strong inter-dependencies, especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene.

The mathematical expression for this index, for two images or signals x and y of the same size is:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5.4}$$

where μ, σ^2 and σ_{xy} represent the average, variance and covariance of the images, and $c_1 = (0.01L)^2$ and $c_2 = (0.03L)^2$ are two variables that are used to stabilize the division. L is the dynamic range of the pixel values, and it changes depending on the format of the image. In our case, and for the function we are using, its value is 1. We have used the MATLAB function $ssim()$ to calculate this index.

Spectral Angle Mapper

Spectral Angle Mapper Classification (SAM) is a method that calculates the spectral angle between an image spectrum and a known spectrum (in our case, the denoised spectrum and the original spectrum) each spectrum is considered as a vector in a q-dimensional space, where q is the number of bands or channels. We then calculate the angle between those two spectra. One advantage of SAM is that it uses only the direction of the spectra, not the length. This means that this method is insensitive to different illumination between pixels, as it does not use the vector length.

The mathematical expression for this method is:

$$SAM = \cos^{-1}\left(\frac{\sum_{i=1}^{q} t_i r_i}{\sqrt{\sum_{i=1}^{q} t_i^2}\sqrt{\sum_{i=1}^{q} r_i^2}}\right) \tag{5.5}$$

Where q is the number of bands in the image, t is the spectrum of a pixel in the image and r is spectrum of a pixel on the original image. There is an angle for each pixel of the image, so the best way to represent those values is by an image. Despite this, direct visual comparison is difficult, so we have decided to compute the average and the variance of each image, and use the values obtained to compare between different regularizations.

5.2.4 Methodology

The simulations will be run with different values of α (the regluarization parameter) each time. This way, we can not only compare the projections against each other but also against different values of the regularization term, and find the optimal configuration to obtain the best possible results. This can be done in an iterative way, changing the values of the simulations after evaluating the results of the previous one.

There are some variables whose value will not be changed during testing. These are the number of iterations of the gradient projection algorithm, which will be at 200, and the time step z, with a value of 0.25, which is the largest possible value to still ensure the convergence of the algorithm.

6 Results

In this chapter we will present the results obtained during the experiments and discuss them. We will first present the results for images with a lower added noise value and after that we will show, with the same structure, the ones obtained for images with more noise. For information on all the parameters used and about the testing environment, please refer to chapter 5.

6.1 Results with lower noise model

The structure of the result presentation is as follows: for each image measure, the results obtained for each norm are plotted against the regularization parameter. We present four plots: in three of them, the measures are presented against their real scale, with the x axis representing the regularization parameter α and the y axis representing the value of each image quality measure we are plotting. Then, the last graph plots all the different regularizations without scale in the x axis, in order to compare them more easily. Finally, for each noise model, we present the results obtained for each image in four tables, one for each measure. The regularization parameter used for each regularization is the one that maximizes the PSNR.

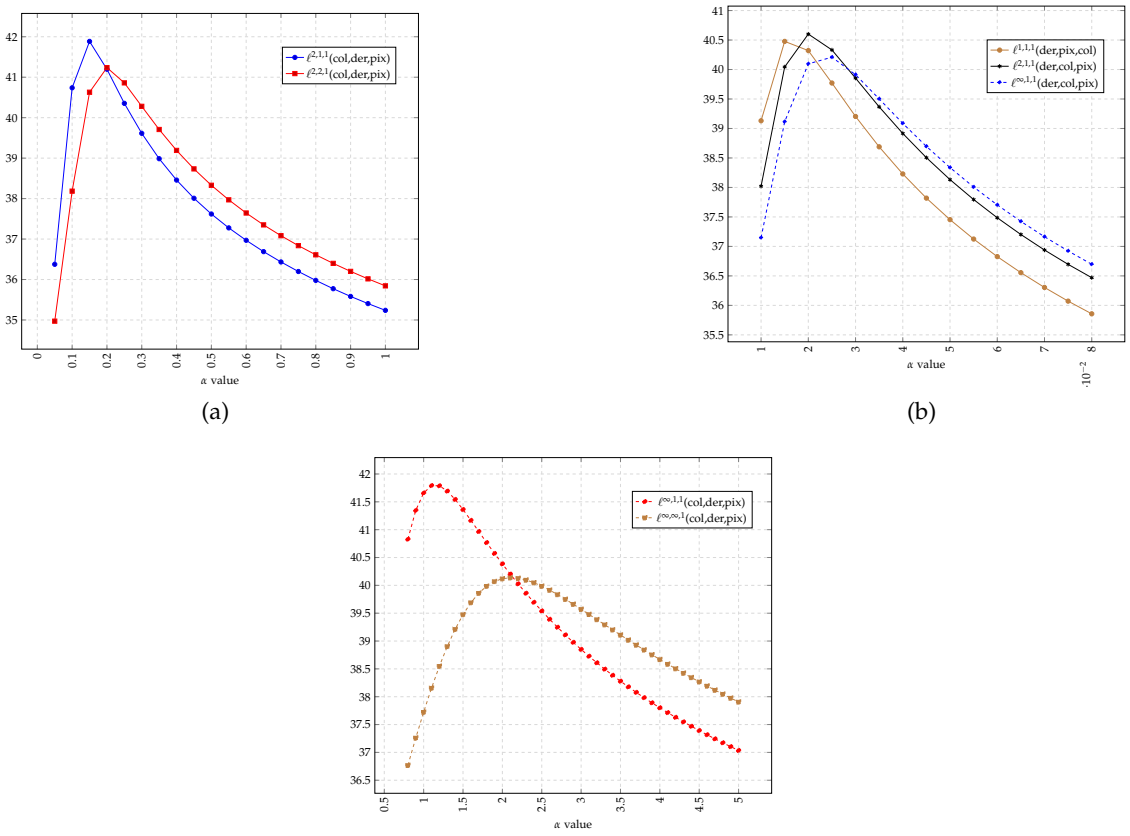


Figure 6.1: PSNR plot of the different norms relative to the regularization parameter.

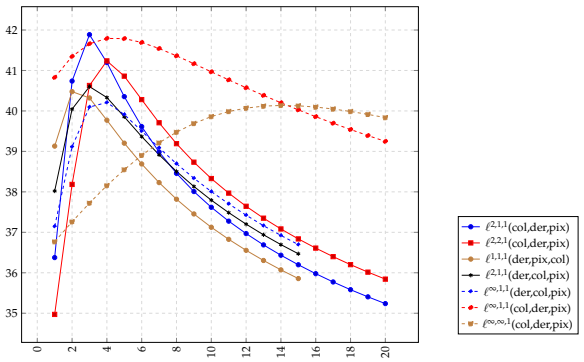
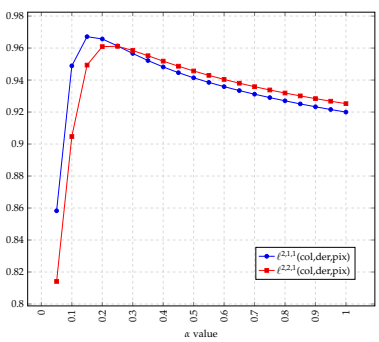
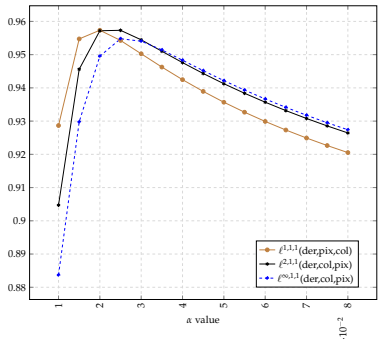


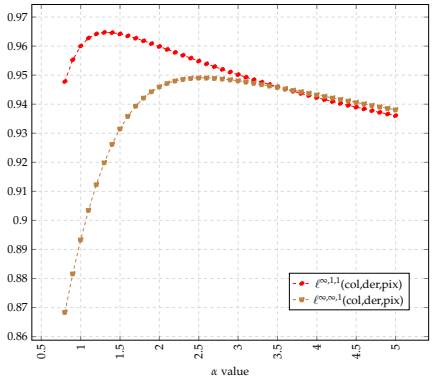
Figure 6.2: PSNR plot of the different norms.



(a)



(b)



(c)

Figure 6.3: SSIM curve of the different norms relative to the regularization parameter.

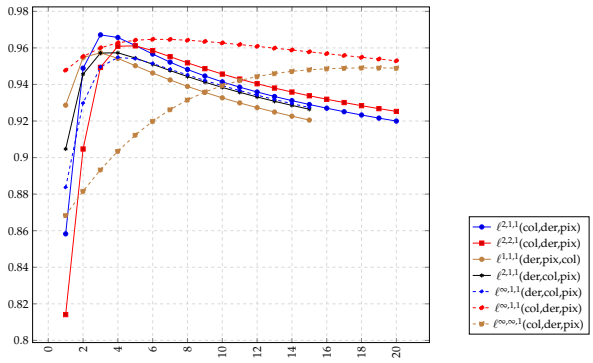


Figure 6.4: SSIM plot of the different norms.

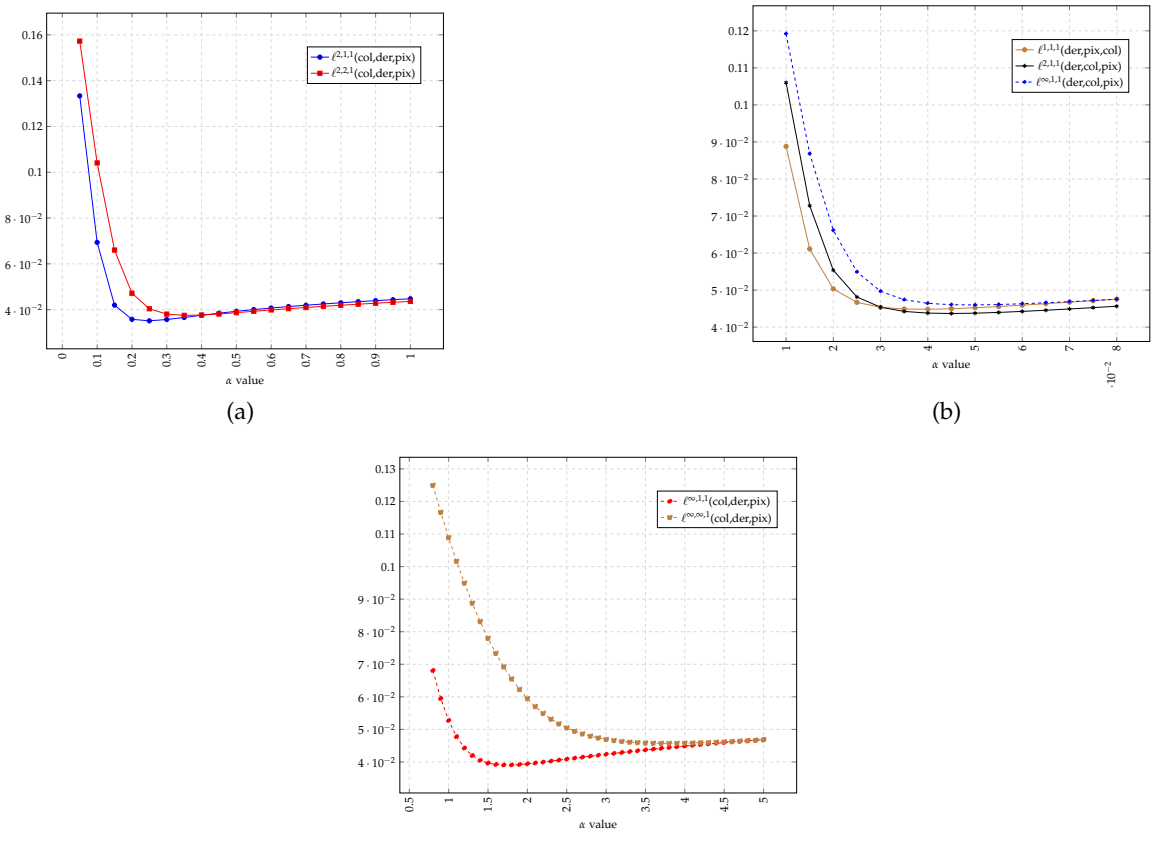


Figure 6.5: Average of the SAM plot of the different norms.

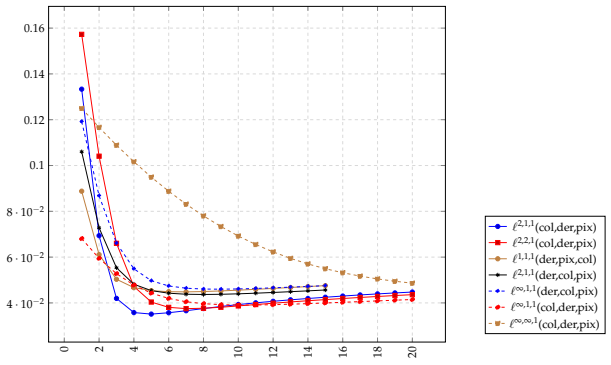


Figure 6.6: Average of the SAM plot of the different norms.

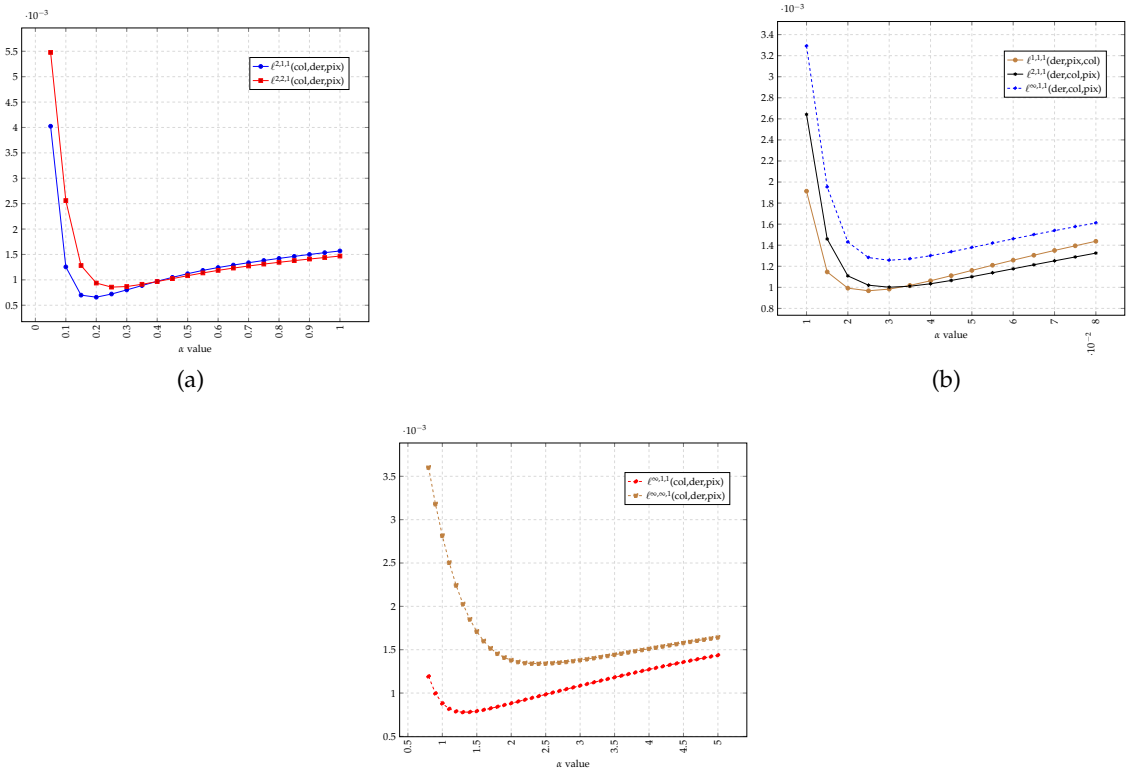


Figure 6.7: Variance of the SAM plot of the different norms.

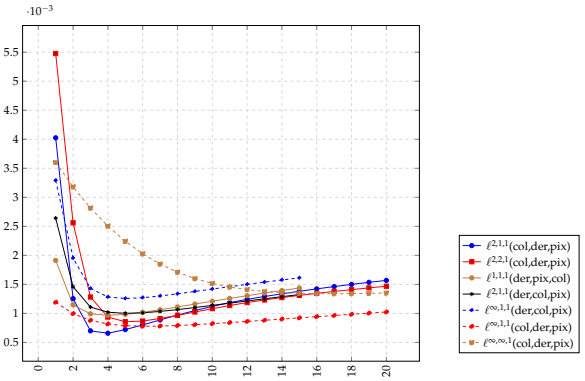


Figure 6.8: Variance of the SAM plot of the different norms.

Norm	$\ell^{2,1,1}$ (col,der,pix)	$\ell^{2,2,1}$ (col,der,pix)	$\ell^{1,1,1}$ (der,pix,col)	$\ell^{2,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (col,der,pix)	$\ell^{\infty,\infty,1}$ (col,der,pix)
SanFrancisco	**40.71**	40.08	38.58	39.68	39.16	40.56	38.9
Female	**41.45**	40.72	39.36	40.1	39.78	41.1	39.27
Stanford	43.48	42.9	41.36	42.01	41.68	**43.7**	42.22

Table 6.1: PSNR Table with ideal regularization parameter for each norm

Norm	$\ell^{2,1,1}$ (col,der,pix)	$\ell^{2,2,1}$ (col,der,pix)	$\ell^{1,1,1}$ (der,pix,col)	$\ell^{2,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (col,der,pix)	$\ell^{\infty,\infty,1}$ (col,der,pix)
SanFrancisco	**0.97**	0.96	0.96	0.96	0.96	0.97	0.95
Female	**0.96**	0.96	0.95	0.95	0.95	0.95	0.93
Stanford	0.97	0.96	0.96	0.96	0.95	**0.97**	0.96

Table 6.2: Structural Similarity Table with ideal regularization parameter for each norm

Norm	$\ell^{2,1,1}$ (col,der,pix)	$\ell^{2,2,1}$ (col,der,pix)	$\ell^{1,1,1}$ (der,pix,col)	$\ell^{2,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (col,der,pix)	$\ell^{\infty,\infty,1}$ (col,der,pix)
SanFrancisco	$\mathbf{2.90 \cdot 10^{-2}}$	$3.26 \cdot 10^{-2}$	$3.26 \cdot 10^{-2}$	$3.67 \cdot 10^{-2}$	$3.71 \cdot 10^{-2}$	$3.37 \cdot 10^{-2}$	$4.03 \cdot 10^{-2}$
Female	$\mathbf{2.80 \cdot 10^{-2}}$	$3.17 \cdot 10^{-2}$	$3.31 \cdot 10^{-2}$	$3.60 \cdot 10^{-2}$	$3.64 \cdot 10^{-2}$	$3.17 \cdot 10^{-2}$	$3.95 \cdot 10^{-2}$
Stanford	$\mathbf{6.91 \cdot 10^{-2}}$	$7.74 \cdot 10^{-2}$	$7.44 \cdot 10^{-2}$	$9.36 \cdot 10^{-2}$	$9.14 \cdot 10^{-2}$	$7.80 \cdot 10^{-2}$	$9.14 \cdot 10^{-2}$

Table 6.3: Average of the Spectral Angle values in each pixel with ideal regularization parameter for each projection

Norm	$\ell^{2,1,1}$ (col,der,pix)	$\ell^{2,2,1}$ (col,der,pix)	$\ell^{1,1,1}$ (der,pix,col)	$\ell^{2,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (col,der,pix)	$\ell^{\infty,\infty,1}$ (col,der,pix)
SanFrancisco	$\mathbf{5.49 \cdot 10^{-4}}$	$6.77 \cdot 10^{-4}$	$8.26 \cdot 10^{-4}$	$7.33 \cdot 10^{-4}$	$8.54 \cdot 10^{-4}$	$6.44 \cdot 10^{-4}$	$9.93 \cdot 10^{-4}$
Female	$\mathbf{2.62 \cdot 10^{-4}}$	$4.18 \cdot 10^{-4}$	$4.01 \cdot 10^{-4}$	$4.87 \cdot 10^{-4}$	$5.67 \cdot 10^{-4}$	$3.84 \cdot 10^{-4}$	$8.40 \cdot 10^{-4}$
Stanford	$\mathbf{1.30 \cdot 10^{-3}}$	$1.74 \cdot 10^{-3}$	$1.68 \cdot 10^{-3}$	$2.13 \cdot 10^{-3}$	$2.45 \cdot 10^{-3}$	$1.43 \cdot 10^{-3}$	$2.27 \cdot 10^{-3}$

Table 6.4: Variance of the Spectral Angle values in each pixel with ideal regularization parameter for each projection

6.2 Results with higher noise model

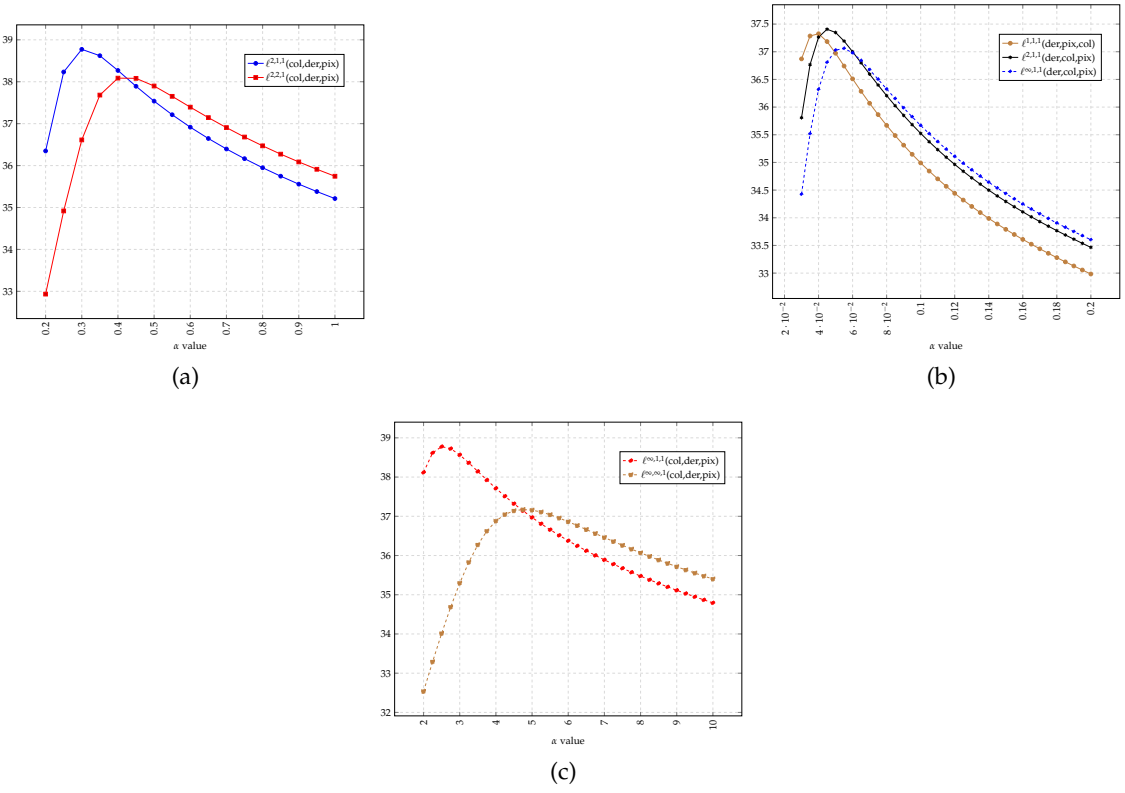


Figure 6.9: PSNR plot of the different norms relative to the regularization parameter.

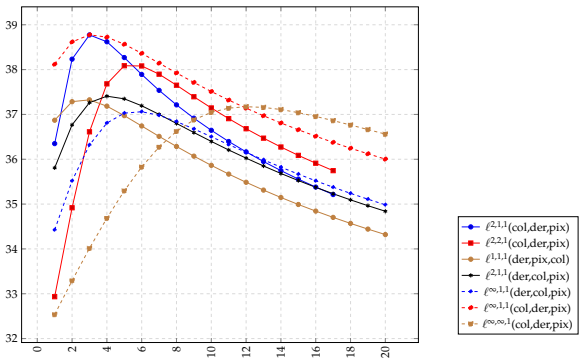


Figure 6.10: PSNR plot of the different norms.

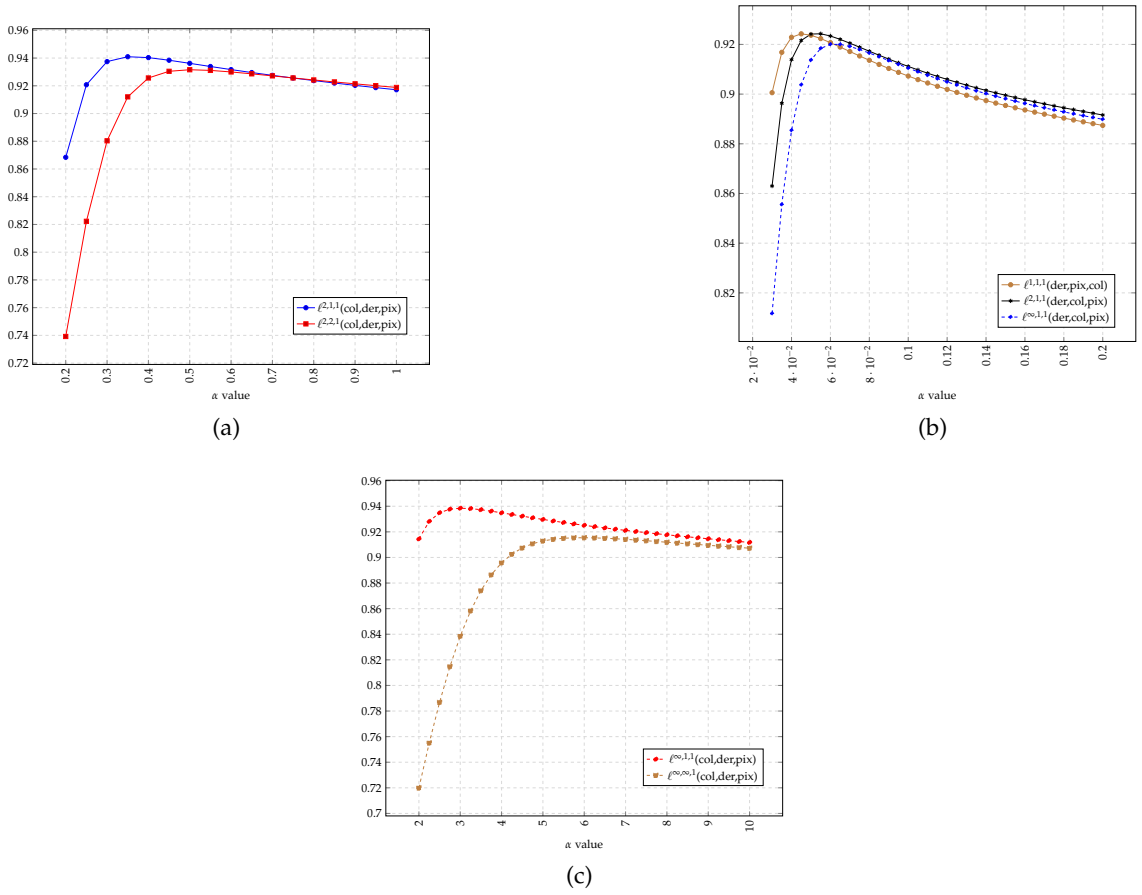


Figure 6.11: SSIM plot of the different norms relative to the regularization parameter.

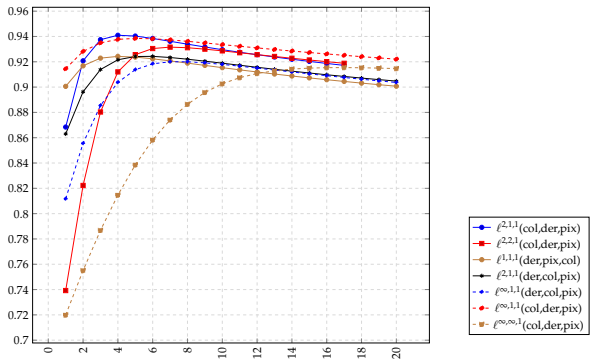


Figure 6.12: SSIM plot of the different norms.

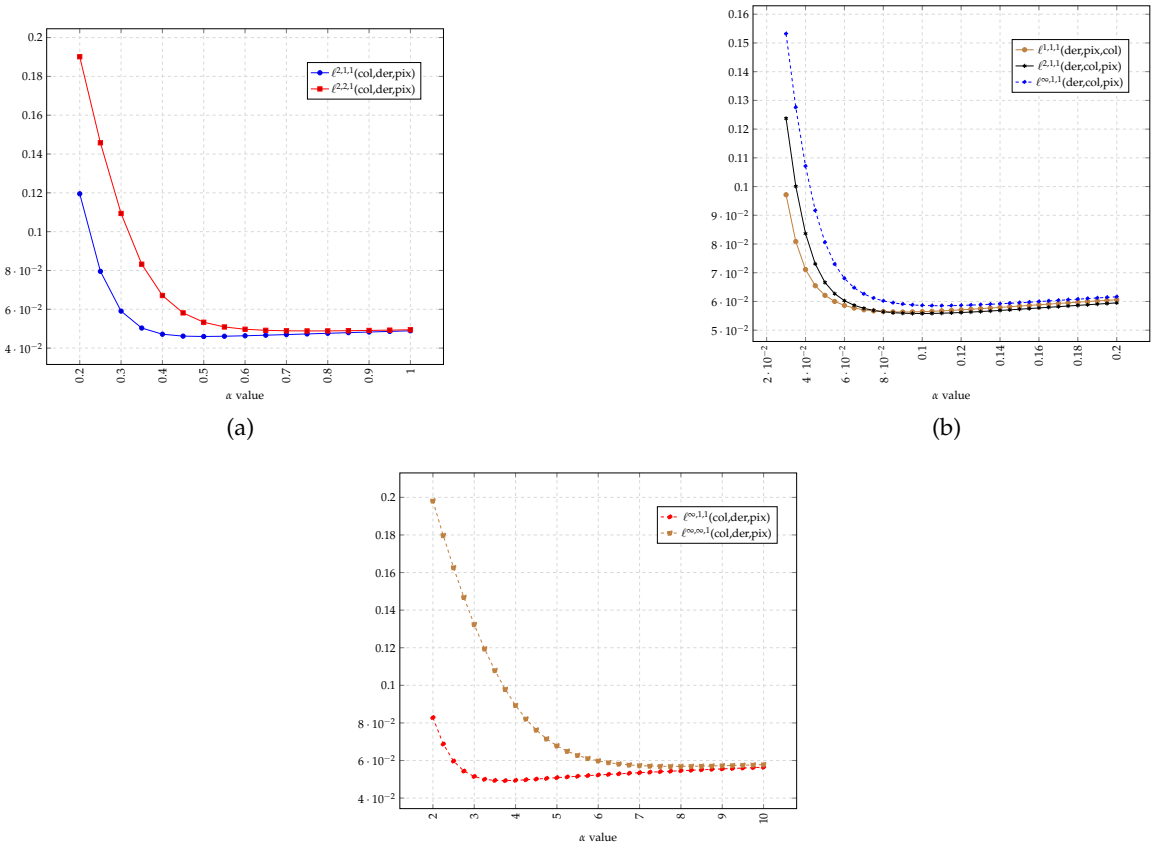


Figure 6.13: Average of the resulting SAM plot of the different norms.

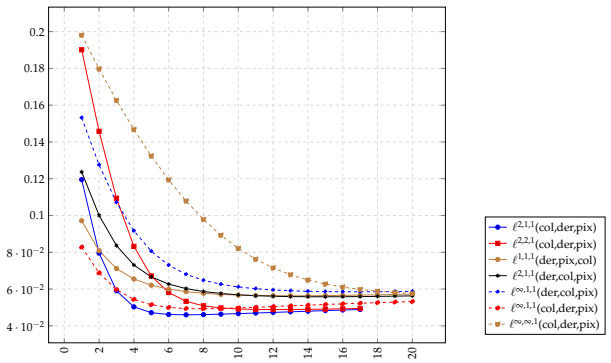
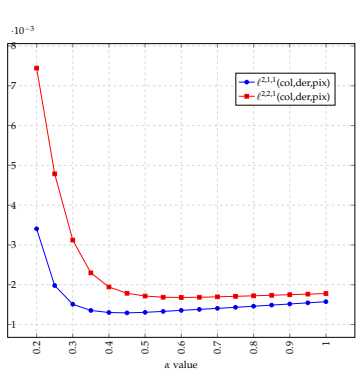
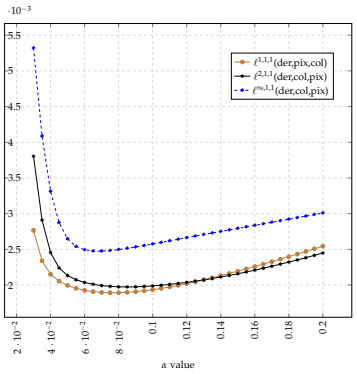


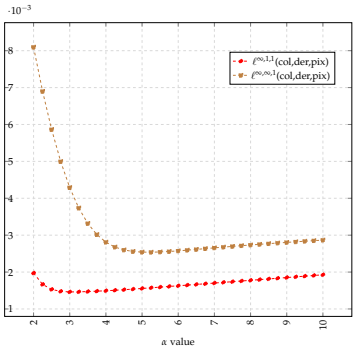
Figure 6.14: Average of the SAM of the different norms.



(a)



(b)



(c)

Figure 6.15: Variance of the SAM plot of the different norms.

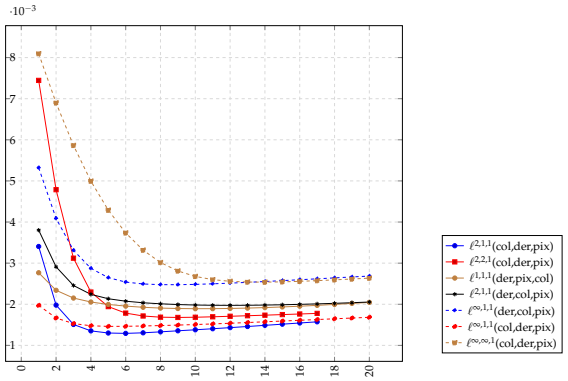


Figure 6.16: Variance of the SAM of the different norms.

Norm	$\ell^{2,1,1}$ (col,der,pix)	$\ell^{2,2,1}$ (col,der,pix)	$\ell^{1,1,1}$ (der,pix,col)	$\ell^{2,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (col,der,pix)	$\ell^{\infty,\infty,1}$ (col,der,pix)
SanFrancisco	**37.61**	36.96	30.94	36.38	35.92	37.47	35.97
Female	**38.19**	37.37	28.52	36.75	36.44	38.07	36.06
Stanford	40.52	39.92	32.95	39.07	38.8	**40.77**	39.48

Table 6.5: PSNR Table with ideal regularization parameter for each norm

Norm	$\ell^{2,1,1}$ (col,der,pix)	$\ell^{2,2,1}$ (col,der,pix)	$\ell^{1,1,1}$ (der,pix,col)	$\ell^{2,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (col,der,pix)	$\ell^{\infty,\infty,1}$ (col,der,pix)
SanFrancisco	**0.94**	0.93	0.87	0.92	0.92	0.94	0.92
Female	**0.93**	0.91	0.84	0.91	0.91	0.92	0.88
Stanford	0.94	0.94	0.89	0.93	0.92	**0.95**	0.93

Table 6.6: Structural Similarity Table with ideal regularization parameter for each projection

Norm	$\ell^{2,1,1}$ (col,der,pix)	$\ell^{2,2,1}$ (col,der,pix)	$\ell^{1,1,1}$ (der,pix,col)	$\ell^{2,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (col,der,pix)	$\ell^{\infty,\infty,1}$ (col,der,pix)
SanFrancisco	$\mathbf{4.06 \cdot 10^{-2}}$	$4.59 \cdot 10^{-2}$	$5.76 \cdot 10^{-2}$	$4.91 \cdot 10^{-2}$	$4.99 \cdot 10^{-2}$	$4.25 \cdot 10^{-2}$	$5.06 \cdot 10^{-2}$
Female	$\mathbf{3.99 \cdot 10^{-2}}$	$4.59 \cdot 10^{-2}$	$4.66 \cdot 10^{-2}$	$4.97 \cdot 10^{-2}$	$5.06 \cdot 10^{-2}$	$4.10 \cdot 10^{-2}$	$5.29 \cdot 10^{-2}$
Stanford	$9.68 \cdot 10^{-2}$	$1.10 \cdot 10^{-1}$	$9.69 \cdot 10^{-2}$	$1.21 \cdot 10^{-1}$	$1.19 \cdot 10^{-1}$	$\mathbf{9.58 \cdot 10^{-2}}$	$1.11 \cdot 10^{-1}$

Table 6.7: Average of the Spectral Angle values in each pixel with ideal regularization parameter for each projection

Norm	$\ell^{2,1,1}$ (col,der,pix)	$\ell^{2,2,1}$ (col,der,pix)	$\ell^{1,1,1}$ (der,pix,col)	$\ell^{2,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (der,col,pix)	$\ell^{\infty,1,1}$ (col,der,pix)	$\ell^{\infty,\infty,1}$ (col,der,pix)
SanFrancisco	$\mathbf{9.93 \cdot 10^{-4}}$	$1.25 \cdot 10^{-3}$	$3.58 \cdot 10^{-3}$	$1.37 \cdot 10^{-3}$	$1.63 \cdot 10^{-3}$	$1.08 \cdot 10^{-3}$	$1.75 \cdot 10^{-3}$
Female	$\mathbf{5.94 \cdot 10^{-4}}$	$9.93 \cdot 10^{-4}$	$1.14 \cdot 10^{-3}$	$1.03 \cdot 10^{-3}$	$1.28 \cdot 10^{-3}$	$7.30 \cdot 10^{-4}$	$2.16 \cdot 10^{-3}$
Stanford	$2.97 \cdot 10^{-3}$	$3.64 \cdot 10^{-3}$	$6.19 \cdot 10^{-3}$	$4.42 \cdot 10^{-3}$	$4.81 \cdot 10^{-3}$	$\mathbf{2.83 \cdot 10^{-3}}$	$3.84 \cdot 10^{-3}$

Table 6.8: Variance of the Spectral Angle values in each pixel with ideal regularization parameter for each projection

6.3 Discussions of results

After testing all the regularization on both sets of noised images, with higher and lower noise, the aim was to find the collaborative norm that provided us with the best results. As one can see in figures 6.18 and 6.19, it is particularly difficult to be able to discern the quality of the denoising for hyperspectral images with our bare eyes. One cannot just look at one of the bands and decide, so numbers and measures are needed to be able to compare with good criteria.

Let us first look at is the PSNR. As explained in chapter 4, section 2, the PSNR can be a measure of the noise an image has compared to its original. If one looks at the plots 6.2 and 6.10, it is seen that, for all regularizations, if the regularization parameter increases, the PSNR tends to increase until a peak is reached, and then it starts to decrease. Looking at the previous plots and at the tables 6.1 and 6.5 it can be seen that the projections that obtain the best results are the $\ell^{2,1,1}(col, der, pix)$ norm and the $\ell^{\infty,1,1}(col, der, pix)$ norm, both in low and in higher noise. As expected, the results with higher noise the results are worse. The regularizations with worse results are $\ell^{1,1,1}(der, pix, col)$ and $\ell^{\infty,\infty,1}(col, der, pix)$.

If we move to the structural similarity comparison (6.2 and 6.6), the results are similar: Both norms $\ell^{2,1,1}(col, der, pix)$ and $\ell^{\infty,1,1}(col, der, pix)$ obtain better results, although the differences among regularizations are small. For the Spectral Angle Mapper, it is difficult to compare them, but generally the seeked result is a low value of the SAM average. The SAM variance depends on the test picture, but in general we also aim for a low value. One can see here that the $\ell^{2,1,1}(col, der, pix)$ norm keeps presenting the best results, and $\ell^{2,2,1}(col, der, pix)$ and $\ell^{\infty,1,1}(col, der, pix)$ follow. There is a huge variation of results between images. This may hinder the testing: it is possible that some methods work well for some types of HSI but not so well for others. This is a very common behaviour in general for methods like this.

All the results have been gathered using the same regularization parameters that have been used in the PSNR plots 6.9 and 6.1. Thanks to this fact, we am able to compare how the other image measures evolve along the PSNR changes. In figures 6.4 and 6.12 we can see that the structural similarity peaks at around the same values as PSNR for each projection: this is an expected ocurrence. For SAM, one can see the corresponding plots in figures 6.13 and 6.5 and figures 6.7 and 6.15 that the same happens here.

There are two norms that stand out from the others: $\ell^{\infty,1,1}$ (col,der,pix) and the norm $\ell^{\infty,\infty,1}(col, der, pix)$. The regularization parameter needed to achieve their best results is

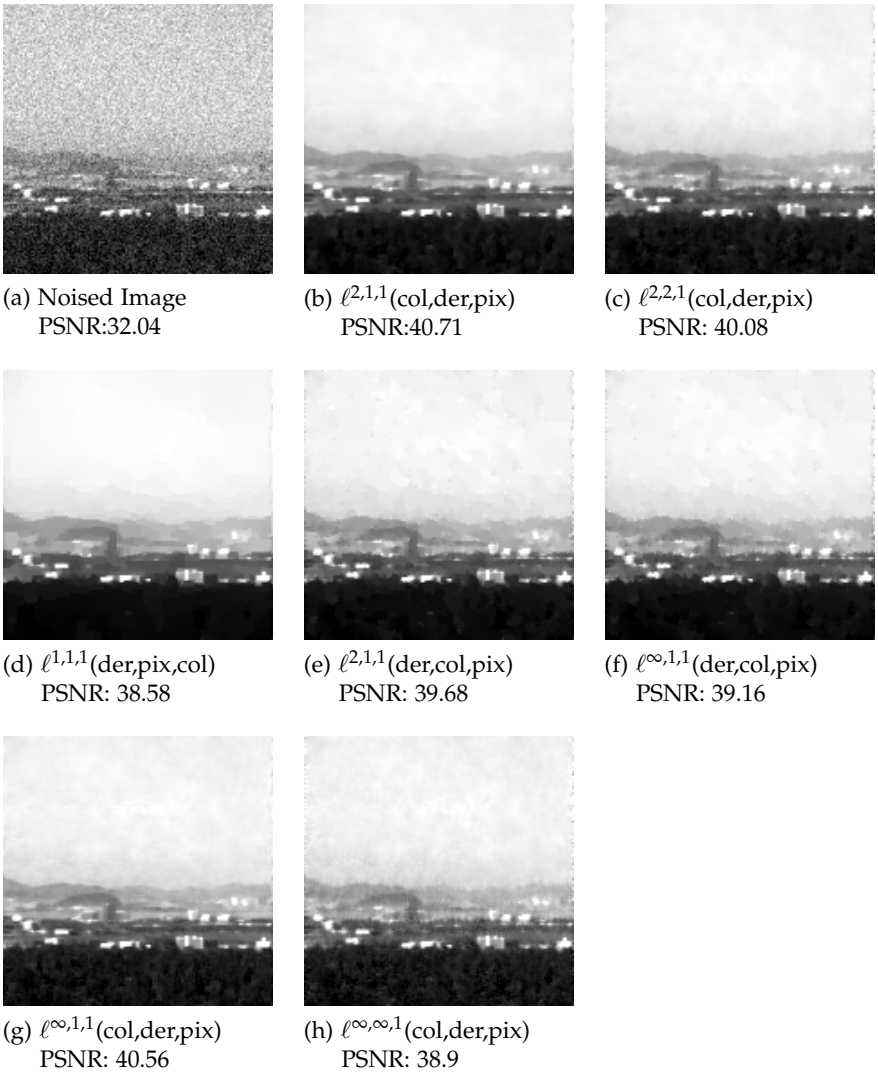
very high compared to the others, and their curve is much less pronounced that in the other norms. This different behavior makes them interesting for further studies.

Finally, we have to mention one strange occurrence: with high noise, the norm $\ell^{1,1,1}(der, pix, col)$ seemed to obtain a really poor denoised image, discernible even by bare eye (as seen in figure 6.19). This issue happens also with the other images, but only with that projection. We have not found a specific reason for this issue to occur, and it seems to be tied to the projection. Our guess is tat it could be the gradient projection not having fully converged, or the original band having almost no information from other channels, giving this result. As this norm also obtained really poor results, it is probably not adequate for the algorithm.

In figures 6.19 and 6.18, the images have been brightened using the MATLAB function $imadjust()$ in order to obtain a higher quality for the reader.

(a) Original Image (b) Noised Image (High noise) (c) Noised Image (Low noise)

Figure 6.17: Original San Francisco image (band 50) and the result after adding the two different noise values described.



(a) Noised Image
PSNR:32.04

(b) $\ell^{2,1,1}$(col,der,pix)
PSNR:40.71

(c) $\ell^{2,2,1}$(col,der,pix)
PSNR: 40.08

(d) $\ell^{1,1,1}$(der,pix,col)
PSNR: 38.58

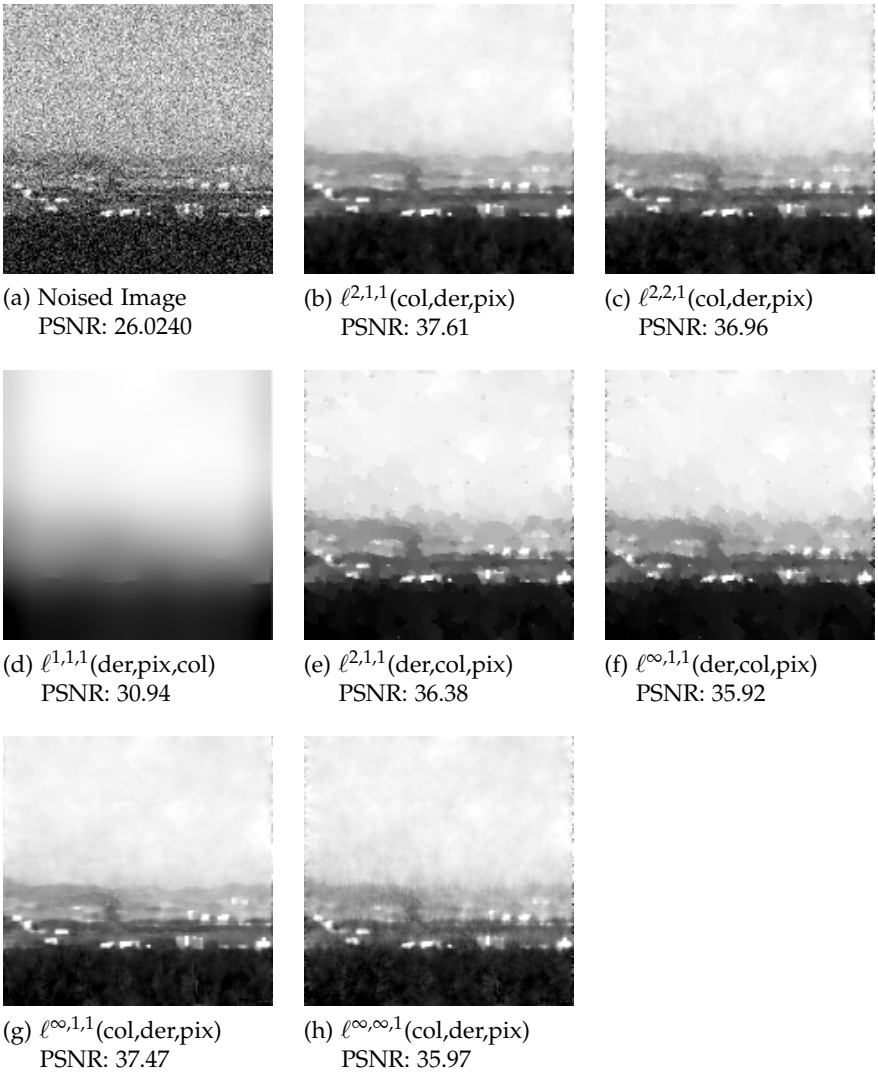
(e) $\ell^{2,1,1}$(der,col,pix)
PSNR: 39.68

(f) $\ell^{\infty,1,1}$(der,col,pix)
PSNR: 39.16

(g) $\ell^{\infty,1,1}$(col,der,pix)
PSNR: 40.56

(h) $\ell^{\infty,\infty,1}$(col,der,pix)
PSNR: 38.9

Figure 6.18: Denoised bands for each projection, from an original image with low value noise added.



(a) Noised Image
PSNR: 26.0240

(b) $\ell^{2,1,1}$(col,der,pix)
PSNR: 37.61

(c) $\ell^{2,2,1}$(col,der,pix)
PSNR: 36.96

(d) $\ell^{1,1,1}$(der,pix,col)
PSNR: 30.94

(e) $\ell^{2,1,1}$(der,col,pix)
PSNR: 36.38

(f) $\ell^{\infty,1,1}$(der,col,pix)
PSNR: 35.92

(g) $\ell^{\infty,1,1}$(col,der,pix)
PSNR: 37.47

(h) $\ell^{\infty,\infty,1}$(col,der,pix)
PSNR: 35.97

Figure 6.19: Denoised bands for each projection, from an original image with high value noise added.

7 Conclusions

Different TV regularizations for hyperspectral images using collaborative norms have been tested. TV and gradient projection algorithm had already been used for color images, as well as collaborative norms. Using them for hyperspectral images, and being a simpler method compared to those already in literature for TV HSI denoising [20, 34], makes this an interesting field of research.

After performing testing on three different sample hyperspectral images, seven different types of Collaborative norms and two levels of noise, we obtained a dataset of results, which we proceeded to analyse.

Analyzing the results, we found that the $\ell^{2,1,1}(col, der, pix)$ and the $\ell^{\infty,1,1}(col, der, pix)$ norms are the regularizers that obtain better results across the board. For its interesting properties, the $\ell^{\infty,1,1}(col, der, pix)$ and $\ell^{\infty,\infty,1}(col, der, pix)$ norms could be useful in other contexts, using a much higher optimal regularization parameter than the rest.

Comparing our results to the ones obtained for normal color images in [16], the norm with better results on color images, $\ell^{\infty,1,1}(col, der, pix)$, translates well to hyperspectral images, maintaining a good performance. But with both norms $\ell^{2,1,1}(col, der, pix)$ and $\ell^{\infty,1,1}(col, der, pix)$ being at least equally good, it is better to use $\ell^{2,1,1}(col, der, pix)$, as it is faster and simpler. However, one could try to implement a local $\ell^{\infty,1,1}(col, der, pix)$ norm in order to adapt to the color image case and even improve the results, as it makes little sense to couple all the channels at once, and a local coupling could perform better:

$$||Du_{bands1-10}||_{\infty,1,1} + ||Du_{bands11-20}||_{\infty,1,1} + \cdots \tag{7.1}$$

This could be a topic for future research.

Future research could be extended at using other types of regularization with the same algorithm and compare to the results obtained or extend the results with a bigger pool of hyperspectral images to further test the best norms. Also, an extension to other types of variational problems in hyperspectral imaging, like unmixing or

uncompressing, using TV and the norms obtained here could be an interesting field of research. One could also try to implement a nonlocal TV model [20, 5] with the norms described in this project.

List of Figures

List of Tables

Bibliography

[1] M. V. Afonso and J. M. R. Sanches. "Blind Inpainting Using l0 and Total Variation Regularization." English. In: *IEEE Transactions on Image Processing* 24.7 (2015), pp. 2239–2253. ISSN: 10577149 (ISSN). DOI: 10.1109/TIP.2015.2417505.

[2] R. Ammanouil, A. Ferrari, C. Richard, and D. Mary. "Blind and fully constrained unmixing of hyperspectral images." In: *IEEE Transactions on Image Processing* 23.12 (2014), pp. 5510–5518. DOI: 10.1109/TIP.2014.2362056.

[3] I. Atkinson, F. Kamalabadi, and D. L. Jones. "Wavelet-Based Hyperspectral Image Estimation." English. In: *2003 IGARSS: Learning From Earth's Shapes and Colours*. Vol. 2. Department of Electrical Engineering, Coordinated Science Laboratory, Univ. Illinois at Urbana-Champaign, Urbana-Champaign, IL, United States, 2003, pp. 743–745.

[4] A. Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. SIAM, 2014, p. 282. ISBN: 1611973643.

[5] M. Benning, C. Brune, M. Burger, and J. Müller. "Higher-order TV methods - Enhancement via Bregman iteration." English. In: *Journal of Scientific Computing* 54.2-3 (2013), pp. 269–310. ISSN: 08857474 (ISSN). DOI: 10.1007/s10915-012-9650-3.

[6] P. Blomgren and T. F. Chan. "Color TV: Total variation methods for restoration of vector-valued images." English. In: *IEEE Transactions on Image Processing* 7.3 (1998), pp. 304–309. ISSN: 10577149 (ISSN). DOI: 10.1109/83.661180.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Mar. 2004. ISBN: 0521833787.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Vol. 25. 3. 2010, pp. 487–487. ISBN: 9780521833783. DOI: 10.1080/10556781003625177. arXiv: 1111.6189v1.

[9] K. Bredies, K. Kunisch, and T. Pock. "Total generalized variation." English. In: *SIAM Journal on Imaging Sciences* 3.3 (2010), pp. 492–526. ISSN: 19364954 (ISSN). DOI: 10.1137/090769521.

[10] A. Chambolle and P. L. Lions. "Image recovery via total variation minimization and related problems." English. In: *Numerische Mathematik* 76.2 (1997), pp. 167–188. ISSN: 0029599X (ISSN).

[11] T. Chambolle, A; Caselles, V; Novaga, M; Cremers, D; Pock. "An introduction to Total Variation for Image Analysis." In: <hal-00437 (2009).

[12] R. H. Chan, M. Tao, and X. Yuan. "Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers." English. In: *SIAM Journal on Imaging Sciences* 6.1 (2013), pp. 680–697. ISSN: 19364954 (ISSN). DOI: 10.1137/110860185.

[13] D. Chen, Y. Chen, and D. Xue. "Fractional-order total variation image denoising based on proximity algorithm." English. In: *Applied Mathematics and Computation* 257 (2015), pp. 537–545. ISSN: 00963003 (ISSN). DOI: 10.1016/j.amc.2015.01.012.

[14] G. Chen and S.-E. Qian. "Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage." English. In: *IEEE Transactions on Geoscience and Remote Sensing* 49.3 (2011), pp. 973–980. ISSN: 01962892 (ISSN). DOI: 10.1109/TGRS.2010.2075937.

[15] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. "Efficient projections onto the L1 -ball for learning in high dimensions." In: *Proceedings of the 25th international conference on Machine learning - ICML* (2008), pp. 272–279. DOI: 10.1145/1390156.1390191.

[16] J. Duran, M. Moeller, C. Sbert, and D. Cremers. "Collaborative Total Variation: A General Framework for Vectorial TV Models." In: (Aug. 2015). arXiv: 1508.01308.

[17] D. Eason and M. Andrews. "Compressed hyperspectral image recovery via total variation regularization assuming linear mixing." English. In: *2014 IEEE International Conference on Image Processing, ICIP 2014*. Department of Electrical and Electronic Engineering, University of Auckland, New Zealand: Institute of Electrical and Electronics Engineers Inc., 2014, pp. 620–624. ISBN: 9781479957514 (ISBN). DOI: 10.1109/ICIP.2014.7025124.

[18] D. T. Eason and M. Andrews. "Total variation regularization via continuation to recover compressed hyperspectral images." English. In: *IEEE Transactions on Image Processing* 24.1 (2015), pp. 284–293. ISSN: 10577149 (ISSN). DOI: 10.1109/TIP.2014.2376273.

[19] A. Karami, M. Yazdi, and A. Zolghadre Asli. "Noise reduction of hyperspectral images using kernel non-negative tucker decomposition." English. In: *IEEE Journal on Selected Topics in Signal Processing* 5.3 (2011), pp. 487–493. ISSN: 19324553 (ISSN). DOI: 10.1109/JSTSP.2011.2132692.

[20] J. Li, Q. Yuan, H. Shen, and L. Zhang. "Hyperspectral image recovery employing a multidimensional nonlocal total variation model." In: *Signal Processing* 111 (2015), pp. 230–248. DOI: 10.1016/j.sigpro.2014.12.023.

[21] P. Liu, L. Xiao, and J. Zhang. *A fast higher degree total variation minimization method for image restoration*. English. School of Computer Science, Engineering, Nanjing University of Science, and Technology, Nanjing, Jiangsu, China, 2015. DOI: 10.1080/00207160.2015.1046848.

[22] Q. Liu, B. Xiong, D. Yang, and M. Zhang. *A generalized relative total variation method for image smoothing*. English. Department of Electronic Information Engineering, Nanchang University, Nanchang, China, 2015. DOI: 10.1007/s11042-015-2709-z.

[23] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. "An iterative regularization method for total variation-based image restoration." English. In: *Multiscale Modeling and Simulation* 4.2 (2005), pp. 460–489. ISSN: 15403459 (ISSN). DOI: 10.1137/040605412.

[24] H. Othman and S. Qian. "Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage." English. In: *IEEE Transactions on Geoscience and Remote Sensing* 44.2 (2006), pp. 397–408. ISSN: 01962892 (ISSN). DOI: 10.1109/TGRS.2005.860982.

[25] H. Pu, W. Xia, B. Wang, and G.-M. Jiang. "A fully constrained linear spectral unmixing algorithm based on distance geometry." English. In: *IEEE Transactions on Geoscience and Remote Sensing* 52.2 (2014), pp. 1157–1176. ISSN: 01962892 (ISSN). DOI: 10.1109/TGRS.2013.2248013.

[26] B. Rasti, J. R. Sveinsson, and M. O. Ulfarsson. "Wavelet-based sparse reduced-rank regression for hyperspectral image restoration." English. In: *IEEE Transactions on Geoscience and Remote Sensing* 52.10 (2014), pp. 6688–6698. ISSN: 01962892 (ISSN). DOI: 10.1109/TGRS.2014.2301415.

[27] R. T. Rockafellar and R. J. B. Wets. *Variational Analysis*. Vol. 317. Grundlehren der mathematischen Wissenschaften. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. ISBN: 978-3-540-62772-2. DOI: 10.1007/978-3-642-02431-3.

[28] R. Rockafellar. *Convex analysis*. Princeton university press, 2015.

[29] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms." In: *Physica D: Nonlinear Phenomena* 60.1-4 (Nov. 1992), pp. 259–268. ISSN: 01672789. DOI: 10.1016/0167-2789(92)90242-F.

[30] D. Salomon. *Data Compression: The Complete Reference*. Springer Science & Business Media, 2007, p. 1092. ISBN: 1846286034.

[31] T. r. Skauli and J. Farrell. "A collection of hyperspectral images for imaging systems research." In: *IS&T/SPIE Electronic Imaging*. Ed. by N. Sampat and S. Battiato. International Society for Optics and Photonics, Feb. 2013, p. 86600C. DOI: 10.1117/12.2007097.

[32] Y.-W. Wen, M. K. Ng, and Y.-M. Huang. "Efficient total variation minimization methods for color image restoration." English. In: *IEEE Transactions on Image Processing* 17.11 (2008), pp. 2081–2088. ISSN: 10577149 (ISSN). DOI: 10.1109/TIP.2008.2003406.

[33] Q. Yuan, L. Zhang, and H. Shen. "Hyperspectral image denoising employing a spectral-spatial adaptive total variation model." English. In: *IEEE Transactions on Geoscience and Remote Sensing* 50.10 PART1 (2012), pp. 3660–3677. ISSN: 01962892 (ISSN). DOI: 10.1109/TGRS.2012.2185054.

[34] Y. Yuan, X. Zheng, and X. Lu. "Spectral-Spatial Kernel Regularized for Hyperspectral Image Denoising." English. In: *IEEE Transactions on Geoscience and Remote Sensing* 53.7 (2015), pp. 3815–3832. ISSN: 01962892 (ISSN). DOI: 10.1109/TGRS.2014.2385082.