Université Paris-Saclay
Faculté des Sciences d'Orsay Télécom ParisTech

Research Internship Report
(Master's Dissertation)

Efficiently Sampling from High-Dimensional Mixture of Gaussians Distributions and Applications to Inverse Problems

Université Paris-Saclay – Master Automatique Traitement du Signal et Images
Télécom ParisTech

Raphael Gasparian Chinchilla

Thursday September the 8th 2016

Laboratoire des Signaux et Systèmes
(Univ. Paris-Sud – CNRS – CentraleSupélec – Université Paris-Saclay)
3 rue Joliot-Curie, 91 192 Gif-sur-Yvette, France

This work is devoted to the intrinsic difficulty of sampling from high-dimensional distributions and to its applications in inverse problems. After a brief introduction on inverse problems, is presented the Rejection Perturbation Optimization algorithm, proposed in [1], that enables efficient sampling of high-dimensional Gaussian distributions. The algorithm is then applied to a unsupervised super-resolution microscopy imagery problem in a Bayesian framework. Its performance and results are compared to those obtained using a supervised optimization technique. Following those results, it is proposed a general method of constructing Mixtures of Gaussians that can efficiently be sampled from and used in unsupervised inverse problems. The constructed mixture is then used in order to obtain the edge preserving distribution proposed in [2]. The distribution is applied to the previous problem and the results are compared to those obtained earlier.

Acknowledgments

I think it is always pleasant to start an acknowledgment. Pleasant since the acknowledgment is for all of those that helped the person that is now typing this words to be able to type them. Acknowledgment are for all of those that I consider were essential to achieving what was achieved. Acknowledgment is a way of saying that without them it would not have been possible. It is for all that I knew I could count on if ever I needed.

That being said, I would like to start by thanking M. François Orieux, not only by being a really great adviser, but also by being a really great person. Advising is not only helping the student to achieve a work, it is also making sure that the human behind the student will achieve the work. Thank you for always taking a time to interact with me when it was needed and when you knew the work would progress faster. Thank you also for conducting my enthusiasm and will to explore new things towards directions that you were also not sure if there would be an answer to be found and trust me to find it; thank you for vibrating with me when I found something and by also being annoyed when we discover what we hopped to be possible was not. Thank you finally for reassuring me of the quality of my work all those time when I felt it was not progressing as fast as I wished and doubted its quality.

I would also like to thank my professors at ATSI and at Télécom for the quality of their classes and the importance they gave to them. A special thanks go to M.Yacine Chitour and to M.Georges Rodriguez that always received me when I wanted to talk with them, not only on the contents of their classes and of other classes, but also on what choices to make. I would also like to thank Mme. Odette Leroux, ATSI's secretary, for being so nice and pleasant to all the students; I feel that the master will lose a lot of its identity with your retirement.

I would also like to thank all my colleagues at L2S-GPI for the incredible sens of cooperation that we were able to develop together, always ready to help each other and also always ready to go lunch together and to speak of anything but work. A special thank to M.Ali Djafari, M.Camille Chapdelaine, Mme.Li Wang and M.Ceena Modarres. And even if he was not in the lab, I would like to thanks my friend M.Ivan Cadena that so many times accepted to interact with me about this work even if he had absolutely nothing to do with it.

My family and friends, and more generally all those that I love and that love me are to thank. Your contribution to me as a person is just so essential that this work would have never be written without you.

A final acknowledgment goes to Bayes and Gauss without whom this work could never be written. Thanks for your contribution in the understanding of this amazing science that is probability and that rules our daily life more than any other.

Contents

1 Inverse Problems and Bayesian Approaches

In this chapter are presented the main topics and issues related to this work. It allows readers that are not familiar with inverse problems to have a broad view of it. Here is also defined most of the notation that will be used in the whole work. In the last section are presented the specific issues of inverse problem that are addressed.

1.1 Inverse Problems

1.1.1 What are inverse problems

Inverse problems is the mathematical and engineering approach, ranging from astronomy images to audio processing, that deals with the problem of, having a set of observations, trying to determine what is the original signal that caused those observations. For this purpose, inverse problem is generally associated with two other problems, that together, compose the broad class of inverse problem.

- The forward problem: It is the work of determining a model which can explain how any set of input signal is changed into the output observations. Because of the inherited complexity of many problems, the model is normally a simplified version of reality that tries to mostly explain the process of data formation.

  In this work, models will be written as:

  $$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \epsilon \tag{1.1}$$

  where

  - $\boldsymbol{y} \in \mathcal{Y} \subseteq \mathbb{R}^m$ is the vector of output observation
  - $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the input signal
  - $\boldsymbol{H} : \mathcal{X} \to \mathcal{Y}$ is a linear operator defined by a matrix
  - ϵ is a multivariate real random variable of dimension m called noise, which represents everything that the model does not explain.

- The instrumentation problem: It deals with where and how should the observation $\boldsymbol{y}$ be taken in order to have the best information about the searched object.

- The inverse problem: By using the model $\boldsymbol{H}$ determined by the forward problem and the observation $\boldsymbol{y}$ determined by the instrumentation problem, retrieve the original input signal $\boldsymbol{x}$. Because of the simplification done in the construction of the model, represented by the noise ϵ, it is not possible retrieve the exact value of $\boldsymbol{x}$. Therefore the inverse problem focus in the creation of an estimator $\hat{\boldsymbol{x}}$.

1.1.2 Ill posed problems and estimation

Some reader might be thinking, as they read this introduction, that the problem can simply be solved by using the Ordinary Least Squares. Having some mathematical fluency, one could easily define a criterion $J(\boldsymbol{x})$ that needs to be minimized:

$$\begin{aligned}
\min_{x\in\mathbb{R}^n} J(\boldsymbol{x}) &= \|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2 \\
&= \boldsymbol{x}^\dagger\boldsymbol{H}^\dagger\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{x}^\dagger H^\dagger\boldsymbol{y} + \boldsymbol{y}^\dagger\boldsymbol{y} \qquad (1.2)\\
\text{First order conditions:}\quad \nabla J(\boldsymbol{x}) &= 2\boldsymbol{H}^\dagger\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{H}^\dagger\boldsymbol{y} = 0 \\
\iff \hat{\boldsymbol{x}} &= (\boldsymbol{H}^\dagger\boldsymbol{H})^{-1}\boldsymbol{H}^\dagger\boldsymbol{y} \\
\text{Second order condition:}\quad \nabla^2 J(\boldsymbol{x}) &= 2\boldsymbol{H}^\dagger\boldsymbol{H} \succeq 0 \\
2\boldsymbol{H}^\dagger\boldsymbol{H} &= 2\|\boldsymbol{H}\|^2 \text{ which is by definition } \succeq 0 \\
\text{therefore } \hat{\boldsymbol{x}} &\text{ is a minimum of } J(\boldsymbol{x})
\end{aligned}$$

where $\boldsymbol{H}^\dagger$[1] is the transpose conjugate and $\boldsymbol{H}^\dagger\boldsymbol{H}$ is a square matrix with a inverse matrix and $\hat{\boldsymbol{x}}$ is the estimation of the input signal.

However, as it is state in [3], many real applications of inverse problem are ill posed in the sens of Hadamard. This means that the problem does not satisfies at least one of Hadamard's condition for a well posed and well conditioned problem. This conditions are:

1. Existence: For any $\boldsymbol{y} \in \mathcal{Y}$ there exists at least one estimator $\hat{\boldsymbol{x}}$

2. Unicity: The solution is unique, which in our case is equivalent to say that $Ker(\boldsymbol{H}) = 0$

3. Stability: The behavior of $\hat{\boldsymbol{x}}$ changes continuously with $\boldsymbol{y}$ which means that a small variation $\partial\boldsymbol{y} \to 0$ creates a small $\partial\hat{\boldsymbol{x}} \to 0$

Inverse problem often do not at least one of those conditions. It is to note that in some cases, the time continuous problem is stable, however, due to the numerical instability, the time discrete problem is not; those are called ill conditioned problems. One can check in [3] for some classical examples of ill posed and ill conditioned problems.

[1] When dealing with real matrices, the notation $\boldsymbol{H}^T$ will be used instead of $\boldsymbol{H}^\dagger$

1.1.3 Bayesian Approach and Regularization

In order to achieve the Stability condition, the Bayesian approach is used. The Bayesian approach defines a probabilistic framework that allows one to easily incorporate information, known or supposed, into the inverse problem in order to stabilize it.

The first thing to do is to change the point of view and to consider $\boldsymbol{x}$ as a realization of a random variable X with probability density function $p(\boldsymbol{x})$ and to consider $\boldsymbol{y}$ as the realization of a random variable Y with probability density function $Y = p(\boldsymbol{y})$.

In this context, it is therefore possible to use the Bayes rule:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}$$

where:

- $p(\boldsymbol{x}|\boldsymbol{y})$ is called the posterior distribution
- $p(\boldsymbol{y}|\boldsymbol{x})$ is called the likelihood of the signal whereas the noise (normally just called likelihood)
- $p(\boldsymbol{x})$ is called the prior distribution of X (normally just called prior)

In the Bayesian interpretation, it is common to see $p(\boldsymbol{y})$ as a normalizing constant that is not related to X. Therefore, it is usual to simplify the notation by writing:

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}) \tag{1.3}$$

One can now define a probability density function to $p(\boldsymbol{y}|\boldsymbol{x})$ and to $p(\boldsymbol{x})$. The most common usual thing to do is to define:

$$\epsilon \sim N\Big(0, (\gamma_n)^{-1} I\Big) \Rightarrow p(\boldsymbol{y}|\boldsymbol{x}) \propto \exp\bigg(-\frac{1}{2}\Big((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})\Big)\bigg) \tag{1.4}$$

$$\boldsymbol{D}\boldsymbol{x} \sim N\Big(0, (\gamma_d)^{-1} I\Big) \Rightarrow p(\boldsymbol{x}) \propto \exp\bigg(-\frac{1}{2}((\boldsymbol{D}\boldsymbol{x})^T \gamma_d (\boldsymbol{D}\boldsymbol{x}))\bigg) \tag{1.5}$$

Both γ_n and γ_d are called hyper-parameters (or hidden variables). In this case they are the precision matrix (*i.e.,* the inverse of the variance) of two Gaussians distributions. In a more general way, hyper-parameters are all the variables that are used to characterize the probability density function of the random variables.

The noise is therefore modeled as a centered uncorrelated Gaussian of precision matrix $\gamma_d I$. Assigning a normal distribution to the noise is generally consider as a good hypothesis because of the central limit theorem.

$\boldsymbol{D}$ is called the regularization operator and it is a positive defined matrix of size $n \times n$. There are several different types of regularization operators and each one has a different usage depending on the problem, the one used in this work will be described ahead. For now, the important thing to understand is that the idea of the regularization operator is to set a prior not directly on $\boldsymbol{x}$ but on the behavior of $\boldsymbol{x}$

to its neighbors. Here the prior is that $\boldsymbol{Dx}$ follows a Gaussian distribution, centered, and of precision matrix $\gamma_d I$.

The best way to understand what is the regularization operator is to show an example of one. For instance $\boldsymbol{x}$ is an audio signal, a possible regularization operator could be the difference between the current and the previous samples. Stating that $\boldsymbol{Dx} \sim N(0, (\gamma_d)^{-1} I)$ is to make the prior that two neighbor samples are likely to be similar, and therefore their difference would be near 0. The matrix associated to this regularization operation is:

$$\boldsymbol{D} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots \\ \vdots & & \ddots & \cdots & \\ & & & & \\ 0 & \cdots & & 0 & 1 \end{bmatrix} \tag{1.6}$$

Independently of the regularization operator that is chosen, using the probability functions defined in 1.4 and in 1.5 and substituting them in 1.3 gives:

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto \exp\left(-\frac{1}{2}\left((\boldsymbol{y} - \boldsymbol{Hx})^T \gamma_n (\boldsymbol{y} - \boldsymbol{Hx}) + \boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{Dx}\right)\right) \tag{1.7}$$

There are two main estimators that can be used to this problem.

The first one is the Posterior Maximum (PM)[2] estimator. Using the optimization theory, maximizing $p(\boldsymbol{x}|\boldsymbol{y})$ *i.e.,* the most likely input signal knowing $\boldsymbol{y}$, is equivalent to minimize:

$$-\log(p(\boldsymbol{x}|\boldsymbol{y})) = J(\boldsymbol{x}) = \frac{1}{2}((\boldsymbol{y} - \boldsymbol{Hx})^T \gamma_n (\boldsymbol{y} - \boldsymbol{Hx}) + \boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{Dx}). \tag{1.8}$$

Therefore, by using the Ordinary Least Square in an equivalent way of the one used in 1.2 the estimator is:

$$\hat{\boldsymbol{x}} = (\gamma_n \boldsymbol{H}^T \boldsymbol{H} + \gamma_d \boldsymbol{D}^T \boldsymbol{D})^{-1} \gamma_n \boldsymbol{H}^T y. \tag{1.9}$$

The second estimator that can be used is the supervised Posterior Expectation (PE)[3]. This estimator consist in approximating the expectation of $p(\boldsymbol{x}|\boldsymbol{y})$ by the empirical mean of K samples $\boldsymbol{x}_i$ taken $p(\boldsymbol{x}|\boldsymbol{y})$:

$$\hat{\boldsymbol{x}} = \int_{\mathbb{R}^n} \boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} \approx \frac{1}{K} \sum_{i=1}^{K} \boldsymbol{x}_i. \tag{1.10}$$

The PE has the disadvantage that it needs a large K in order to approximate $p(\boldsymbol{x}|\boldsymbol{y})$ expectation. However, it has the advantage that the estimator variation can also be approximated:

[2]In french it is called Maximum A Posteriori (MAP)
[3]In french it is called Espérance A Posteriori (EAP)

$$\hat{\sigma}^2 = \int_{\mathbb{R}^n} (\boldsymbol{x} - \mu)^2 p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} \approx \frac{1}{K} \sum_{i=1}^{K} \boldsymbol{x}_i^2 - \hat{\boldsymbol{x}}^2 \tag{1.11}$$

In some applications, as astronomy, knowing the standard deviation in each pixel is an important information for the application. Therefore, being able to approximate it is an important advantage for PE.

As it can be seen, in both approaches $\hat{\boldsymbol{x}}$ depends on the values of γ_d and of γ_n. One needs to calculate several $\hat{\boldsymbol{x}}$ with different values of the ratio γ_d/γ_n in order to find the one that give the best result. This can be rather tricky if calculating $\hat{\boldsymbol{x}}$ takes time or in more complex models where there are more hyper-parameters. Besides that, both method rely on the paradigm of considering the hyper-parameters as known and then testing several different of them with human supervision. For this reason those are called supervised approaches. In the next section will be introduced the unsupervised approach where those hyper-parameters are estimated.

1.2 Unsupervised Approach

1.2.1 A Different Paradigm

As it was said in the previous section, the unsupervised approach is a change of paradigm as it aims to estimate the values of the hyper-parameters in a probabilistic way. In order to do so, the hyper-parameters will be seen as random variables; this is a complete Bayesian approach of the problem. In this new interpretation, the supervised approach is one where γ_n and γ_d are seen as known realization of respectively Γ_n and Γ_d. For instance, with this change of view, the equation 1.7: (written bellow)

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto \exp\left(-\frac{1}{2}\left((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{D}\boldsymbol{x}\right)\right)$$

becomes:

$$p(x|y, \gamma_n, \gamma_d) \propto \exp\left(-\frac{1}{2}\left((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{D}\boldsymbol{x}\right)\right).$$

For the rest of this work, this complete Bayesian approach will be the one used. As it will be seen, this way of treating the problem enables a great versatility of methods. Furthermore, it allows to know exactly what are the hypothesis that are been made and to use them in the best way to solve the problem.

So, being equipped with random variables to the hyper-parameters, they can also be considered as unknown and the probability distribution is $p(\boldsymbol{x}, \gamma_n, \gamma_d|\boldsymbol{y})$. Using the Bayes rule, the problem is:

$$p(\boldsymbol{x}, \gamma_n, \gamma_d|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x}, \gamma_n) p(\boldsymbol{x}|\gamma_d) p(\gamma_n) p(\gamma_d). \tag{1.12}$$

There are many possible choices for the prior distributions of $p(\gamma_n)$ and of $p(\gamma_d)$. The most common is to choose diffuse non informative priors and the most common

of them is Jeffreys prior. Jeffreys prior has the property of being scale invariant, non informative and diffuse. For a normal distribution, the Jeffreys prior is a gamma distribution of parameters $\gamma \sim \mathscr{G}(0, \infty)$ which is an improper prior. As it will be seen ahead, this will not be a problem if the likelihood brings enough information. The hyper-parameters have therefore as probability density function:

$$p(\gamma_n) \propto \gamma_n^{-1}$$
$$p(\gamma_d) \propto \gamma_d^{-1}$$

When applying this priors, 1.4 and 1.5 to 1.12, the posterior distribution becomes:

$$p(\boldsymbol{x}, \gamma_n, \gamma_d | \boldsymbol{y}) \propto \gamma_n^{-1} \gamma_d^{-1} \exp\left(-\frac{1}{2}\left((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{D}\boldsymbol{x}\right)\right) \quad (1.13)$$

To be able to explore this probability distribution in order to solve the inverse problem, a last mathematical tools needs to be introduced: the Gibbs Sampler.

1.2.2 Gibbs Sampler

The Gibbs Sampler allows to take samples from a multivariate distribution that cannot be directly sampled but from which the conditional posterior distributions of each variable can easily be sampled. Algorithm 1 describe how the Gibbs Sampler can be used to sample from 1.13.

Algorithm 1: Gibbs Sampler

while *number of samples not achieved* **do**

1. sample from:

$$p(\boldsymbol{x}|\boldsymbol{y}, \gamma_n, \gamma_d) \propto \exp\left(-\frac{1}{2}\left((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{D}\boldsymbol{x}\right)\right)$$
$$\propto \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m})(\gamma_n \boldsymbol{H}^T \boldsymbol{H} + \gamma_d \boldsymbol{D}^T \boldsymbol{D})(\boldsymbol{x} - \boldsymbol{m})\right) \quad (1.14)$$

where $\boldsymbol{m} = (\gamma_n \boldsymbol{H}^T \boldsymbol{H} + \gamma_d \boldsymbol{D}^T \boldsymbol{D})^{-1} \gamma_n \boldsymbol{H}^T \boldsymbol{y}$

2. sample from:

$$p(\gamma_n|y, x, \gamma_d) \propto \gamma_n^{M/2-1} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})\right)$$

3. sample from:

$$p(\gamma_d|y, x, \gamma_n) \propto \gamma_d^{N/2-1} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{D}\boldsymbol{x}\right)$$

As it can be seen, $p(\boldsymbol{x}|\boldsymbol{y}, \gamma_n, \gamma_d)$ is a multivariate normal distribution and $p_(\gamma_n|\boldsymbol{y}, \boldsymbol{x}, \gamma_d)$ and $p_(\gamma_d|\boldsymbol{y}, \boldsymbol{x}, \gamma_n)$ are gamma distributions. Furthermore, even if Jeffreys prior is improper the hyper-parameters likelihood is proper. This can still become a problem as the likelihood might not bring enough information to distribution.

The interesting thing about the Gibbs Sampler is that the values of γ_n and of γ_d are adjusted in a unsupervised way. There is therefore no need to test different values of hyper-parameters to discover those that maximize the probability, the algorithm does it by itself.

Now that it is possible to sample from the distribution, the Posterior Expectation (PE) estimator described in 1.10 can be applied to the samples $\boldsymbol{x}_i$ drawn from $p(\boldsymbol{x}|\boldsymbol{y}, \gamma_n, \gamma_d)$ during the Gibbs Sampling. Therefore, $\hat{\boldsymbol{x}}$ will be the estimation of the original input signal $\boldsymbol{x}$ that the inverse problem aims to determine.

This last point ends the explanation on the generalities about Inverse Problems. In the next section it will be described the specificities this work will be focusing on before entering on what was developed.

1.3 The Problem Treated in this Work

1.3.1 High-Dimensional Non-Stationary Problems

From the generalities about Inverse Problems that were presented previously, one understands the importance of sampling from distributions when using PE estimator for solving Inverse Problems. However, sampling from distributions can be rather complicated, especially when using high dimensional distributions [4].In order to better understand the problems, let us see an example.

Taking for instance a gray scale image of 256×256 pixels, which is rather a small image. This means that the size of the vector $\boldsymbol{x}$ is 65536. Therefore the precision matrix of $p(\boldsymbol{x}|\boldsymbol{y}, \gamma_n, \gamma_d)$ is $(\gamma_n \boldsymbol{H}^T \boldsymbol{H} + \gamma_d \boldsymbol{D}^T \boldsymbol{D})$ 1.14 and its size is $65536 \times 65536 = 4294967296$. The first problem is that this matrix has a size of 32Gb that would need to be stored in the computer's RAM, and there is no need to say that most computers today do not have this kind of capacity.

The second problem is that, even if one would have a computer that has enough memory for storing those matrices, sampling from the multivariate Gaussian distribution is not simple. According to [5] the classical way to sample $N(\boldsymbol{m}, (\gamma_n \boldsymbol{H}^T \boldsymbol{H} + \gamma_d \boldsymbol{D}^T \boldsymbol{D})^{-1})$ would be:

$$\boldsymbol{x}_i = \boldsymbol{m} + \boldsymbol{R}\sigma.$$

Where:

- $\boldsymbol{R}$ is the Cholesky decomposition of $\gamma_n \boldsymbol{H}^T \boldsymbol{H} + \gamma_d \boldsymbol{D}^T \boldsymbol{D}$
- $\sigma \sim N(0, I)$ of dimension n

However, finding the Cholesky decomposition generally requires $\mathbb{O}(n^3)$ operations [1] which makes impracticable to apply this method in high-dimensional problems.

A common solution to overcome this issue is to build the model of the forward problem using only convolution operators *i.e.,* circular matrices, that have the property of being diagonalizable in the Fourier domain[3]. If both $\boldsymbol{H}$ and $\boldsymbol{D}$ are circular, then the sampling can be achieved with:

$$\boldsymbol{x}_i = \boldsymbol{m} + \boldsymbol{F}^\dagger((\Lambda_H + \Lambda_D)^\dagger(\Lambda_H + \Lambda_D)\sqrt{(\gamma_d \Lambda_H + \gamma_d \Lambda_D)^{-1}}\boldsymbol{F}\sigma)$$

where:

- $\boldsymbol{F}$ is the matrix of the FFT operator
- Λ_H and Λ_D are the diagonalization in the Fourier domain of $\boldsymbol{H}$ and $\boldsymbol{D}$
- $\sigma \sim N(0, I)$ of dimension n

The solution of modeling the problem using only convolution operators is a good one in several cases, but it is generally a very restrictive one. This work aims to study algorithms to efficiently sample from high-dimensional Gaussian that are not diagonalizable in the Fourier domain. Those are called non-stationary problems. Those algorithm will be applied to solve an inverse problem related to microscopy images that will act as a case study. This inverse problem is the subject of the next paragraph.

1.3.2 Structured Illumination Microscopy

Structured Illumination Microscopy (SIM) is a type of optical microscopy that aims to achieve a better resolution by changing illumination patterns. This paragraph has as objective to give an overview SIM and how a Bayesian approach can be used to have better results; a better description of the problem is given in [6]. The issue that SIM tries to solve is that, because of diffraction phenomenon, the resolution of microscopes is limited. The main idea of SIM is that, by changing the sample illumination pattern in a way that the diffraction limitation can be overcome. The change of illumination causes an aliasing through modulation and exploring this aliasing allows to recover information beyond the cutoff frequency in order to reconstruct the high resolution image. In [6] it is proven that in order to accomplish this reconstruction, only four images are needed to be taken, one centered and three others with phase shifting. This overview explained, the forward problem can be studied.

The model aims to reproduce the image acquisition that is done by the microscope. The forward model is:

$$\boldsymbol{y} = \boldsymbol{OMRx} + \epsilon \tag{1.15}$$

where:

- $\boldsymbol{x}$ is a vectored version of the image
- $\mathbf{R}$ is a replication matrix that copies the $\boldsymbol{x}$ four times *i.e.,* $R = [IIII]^T$
- $\mathbf{M}$ is a bloc diagonal matrix where each bloc's diagonal element corresponds to the modulation pattern chosen for each image taken
- $\mathbf{O}$ is the optical transfer function of the microscope's lens, and therefore a convolution operator; $\mathbf{O}$ depends on the equipment used but this does not change the algorithm principle. The optical transfer function used in the work

has as convolution mask:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- ϵ is the noise

As it can be seen, the operator $\boldsymbol{H} = \boldsymbol{OMR}$ is not a convolution operator and therefore it is not diagonalizable in the Fourier domain. As it was seen in the previous paragraph, this is an issue as being diagonalizable in the Fourier domain is one of the main tools when dealing with high-dimensional distributions and therefore other methods to sample will be needed to treat this problem.

1.3.3 Quantifying the estimator quality

As this work is on the quality of algorithms, it will be tested on a known image. In this case, the algorithms will be applied on the popular image The Cameraman that can be seen in Figure 1.1. The use of this image is made primarily because of its borders that present a challenge to techniques of image restoration. In order to verify the quality of the estimation, the Euclidean norm will be used, *i.e.,* $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$.

Figure 1.1: The Cameraman, image where the test will be made

Another important image is the image simulated by the forward problem. In Figure 1.2 the four noised images from the SIM problem to which will be applied the inverse problem estimator. The precision of the applied noise is $\gamma_n = 0.1$ *i.e.,* the variance is $\sigma = 100$).

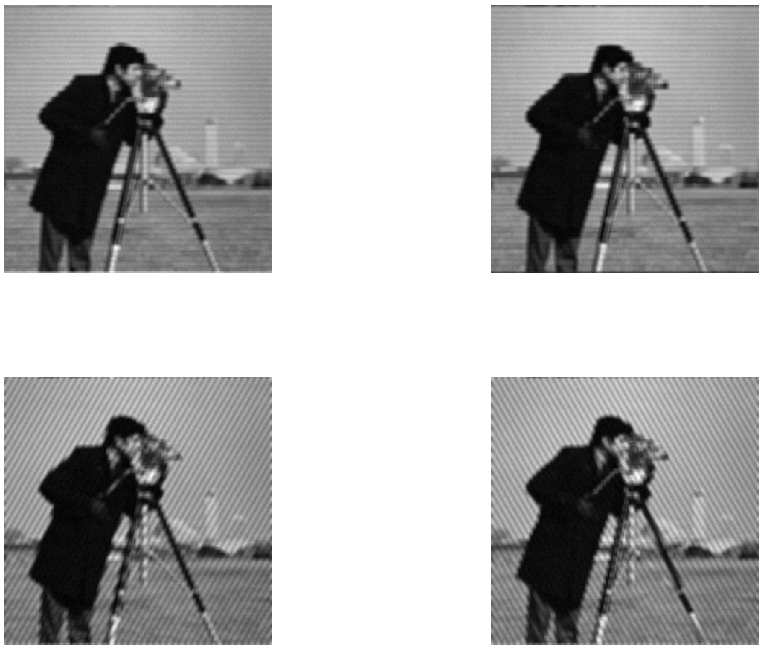


Figure 1.2: The four images created by the forward problem

The presentation of this forward model and of the image that is treated concludes this rather introductory chapter which lays down the basis of this work. The next chapter will describe an algorithm that can efficiently sample from high-dimensional Gaussian that was applied to the SIM problem in [6]. Next chapter will also explain the improvements that were made in this algorithm in order to have better and faster results.

2 Gaussian Prior

This chapter introduces the use of Gaussian priors to solve the SIM problem. The Rejection Perturbation Optimization algorithm that is used in the work to sample from high-dimensional non stationary Gaussian is detailed as well as its faster version. Different estimations are calculated using the Posterior Maximum, and the supervised and unsupervised Posterior Expectation. In the last section the results obtained are compared and analyzed.

2.1 Hypothesis made

2.1.1 Likelihood

The forward model of the SIM is, as described in 1.15:

$$\boldsymbol{y} = \boldsymbol{OMRx} + \epsilon = \boldsymbol{Hx} + \epsilon$$

The choice of noise that is made is the same one that in 1.4, *i.e.*, $\epsilon \sim N(0, (\gamma_n)^{-1} I)$, which is an independent identically distributed noise (*i.i.d*)

$$p(\boldsymbol{y}|\boldsymbol{x}, \gamma_n) \propto \gamma_n^{M/2} \exp\left(-\frac{1}{2}((\boldsymbol{y} - \boldsymbol{Hx})^T \gamma_n (\boldsymbol{y} - \boldsymbol{Hx}))\right) \tag{2.1}$$

2.1.2 Regularization Prior

There are two classical regularization operator, the Laplacian operator, and the Gradient operator. The two are known for having different behaviors and to represent different priors. Both of them will be tested in order to be able to compare their results.

The Laplacian has as operation mask:

$$\boldsymbol{h} = \frac{1}{8}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

The Gradient has two regularization mask, one for the lines and one for the colons:

$$\boldsymbol{h}_l = \frac{1}{2}\begin{bmatrix} 1 & -1 \end{bmatrix}$$

and:

$$\boldsymbol{h}_c = \frac{1}{2}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

The regularization matrices (that will not be explicitly written in here but that are easy to deduce) $\boldsymbol{D}$ for the Laplacian and $\boldsymbol{D}_l$ and $\boldsymbol{D}_c$ for the Gradient are by definition convolution matrices and therefore are diagonalizable in the Fourier domain. However, those matrices have the problem of not being positive defined. This happens since the mean level, represented in the Fourier domain by the first coefficient, is zero. This is obvious as both represent differences and therefore are unable to observe the mean level of the image. This become a problem if wanting to use those matrices as precision matrices of Gaussian distributions. Even if technically an improper prior could be use, the approach that was made was the one described in [2] which is equivalent to introduce a small positive value δ in the place of the zero of the diagonalized in the Fourier domain matrix, making it positive defined.

The choice of prior distribution for both regularization operation is of a Gaussian distribution.

For the Laplacian:

$$p(\boldsymbol{x}|\gamma_d) \propto \gamma_d^{N/2} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{D}^T\gamma_d\boldsymbol{D}\boldsymbol{x}\right)$$

For the Gradient:

$$p(\boldsymbol{x}|\gamma_l\gamma_c) \propto \det\left(\gamma_l\boldsymbol{D}_l^T\boldsymbol{D}_l + \gamma_c\boldsymbol{D}_c^T\boldsymbol{D}_c\right)^{1/2} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}^T\boldsymbol{D}_l^T\gamma_l\boldsymbol{D}_l\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{D}_c^T\gamma_c\boldsymbol{D}_c\boldsymbol{x}\right)\right)$$

2.2 Posterior Maximum

2.2.1 Conjugate Gradient

The Conjugate Gradient (CG) is an algorithm that is classified in the broad class of Gradient Descent. This class of algorithm uses the direction given by the gradient in a point in order to find a local minimum of a criterion. However, if the criterion is a quadratic one with a positive defined Hessian, then the minimum found by the criterion is the global minimum. The type of criterion can be written as:

$$J(\boldsymbol{x}) = (\boldsymbol{m} - \boldsymbol{x})^T\boldsymbol{Q}(\boldsymbol{m} - \boldsymbol{x})$$

with Q a positive defined matrix

CG has the property of converging very fast to an approximate solution and to converge to the exact solution in n iterations. This is done by, in each iteration, exploring one of the conjugated direction from the criterion's Hessian. Both CG demonstration and implementation are well known; however if the reader wants to know more, the explanation given in [7] is recommended. Even if the algorithm is well known, some points that will be important for the rest of the work need to be detailed.

The first one is that CG is a recursive algorithm that uses the estimation made in the previous iteration in order to calculate the next one. Because of this characteristic, CG needs a starting point that acts as the first estimation. In this work, the one that will be used is the average of the four images that SIM creates.

The other important point is that CG needs to calculate the forward problem several times. As this work is in large scale data, there is the need to make a fast implementation of it. How to efficiently calculate the forward problem is explained in algorithm 2 [1].

Algorithm 2: Forward Problem

1. Forward likelihood:

$$aux\_signal = \begin{bmatrix} \text{diag}(\Lambda_O) \\ \text{diag}(\Lambda_O) \\ \text{diag}(\Lambda_O) \\ \text{diag}(\Lambda_O) \end{bmatrix} \otimes \boldsymbol{F}\left(diag(\boldsymbol{M}) \otimes \begin{bmatrix} \hat{\boldsymbol{x}}_i \\ \hat{\boldsymbol{x}}_i \\ \hat{\boldsymbol{x}}_i \\ \hat{\boldsymbol{x}}_i \end{bmatrix} \right)$$

$$signal = sum\_layers\left(\text{diag}(\boldsymbol{M}) \otimes \boldsymbol{F}^{\dagger}\left(\begin{bmatrix} \text{diag}(\Lambda_O^{\dagger}) \\ \text{diag}(\Lambda_O^{\dagger}) \\ \text{diag}(\Lambda_O^{\dagger}) \\ \text{diag}(\Lambda_O^{\dagger}) \end{bmatrix} \otimes aux\_signal \right) \right)$$

2. Forward regularization:

$$reg = \boldsymbol{F}^{\dagger}(\text{diag}(\Lambda_D^{\dagger}) \otimes \text{diag}(\Lambda_D) \otimes \boldsymbol{F}(\hat{\boldsymbol{x}}_i))$$

3. Output:
out $= \gamma_n \, signal + \gamma_d \, reg$
where:
-$\boldsymbol{F}$ is the matrix of the FFT operator
-$\hat{\boldsymbol{x}}_i$ is the estimation did in the previous iteration
-$\otimes$ is the point wise product
-$diag(.)$ creates a vector with the diagonal elements of a matrix
-sum_layers is the sum of the each of the four correspondent pixels
-Λ_O is the diagonalization in the Fourier domain of the optical transfer function O
-Λ_D is the diagonalization in the Fourier domain of the regularization operator $\boldsymbol{D}$

The last important point is the preconditioner. As it is described in [5] the preconditioner has as goal to approximate the Hessian of the forward problem. By doing so, the convergence towards a good approximate solution is even faster than the traditional CG. The preconditioner used in this work is the approximation of the Hessian by the circular matrices. Since those matrices are easily diagonalizable, their inversion is easy. The preconditioner is described in Algorithm 3 [2].

[1] This algorithm describes the implementation using the Laplacian Operator, the Gradient Operator has a similar implementation

[2] This algorithm describes the implementation using the Laplacian Operator, the Gradient Operator has a similar implementation

Algorithm 3: Preconditioner

$$\bar{x} = \boldsymbol{F}^{\dagger}\left(\gamma_n|\boldsymbol{\Lambda_O}|^2+\gamma_d|\boldsymbol{\Lambda_D}|^2\right)^{-1}\boldsymbol{F}\boldsymbol{x}$$

2.2.2 Laplacian regularization

In the Laplacian regularization case, the criterion that needs to be minimized is defined by:

$$J(\boldsymbol{x}) = (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})^T\gamma_n(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T\boldsymbol{D}^T\gamma_d\boldsymbol{D}\boldsymbol{x}$$

Now that all the tools were presented, it is possible to apply the PM estimator to the SIM problem. By methodically testing several values of γ_d these are the optimal parameters found and norm of the error associated to them:

$\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|$	γ_n	γ_d
2.1719×10^3	0.1	0.0248

The restored image obtained using those parameters can be seen in Figure 2.1.

Figure 2.1: Restored image using Posterior Maximum Estimator with Gaussian priors and Laplacian regularization

2.2.3 Gradient regularization

The criterion that needs to be minimized for the Gradient regularization is:

$$J(\boldsymbol{x}) = (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})^T\gamma_n(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T\boldsymbol{D}_l^T\gamma_l\boldsymbol{D}_l^T\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{D}_c^T\gamma_c\boldsymbol{D}_c^T\boldsymbol{x}$$

Again by methodically testing several possibilities of hyper-parameters it is possible to find those that minimize the norm of the error. Those are shown below.

$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$	γ_n	γ_l	γ_c
2.1819×10^3	0.1	0.0034	0.0084

The restored image obtained using those parameters can be seen in Figure 2.1.

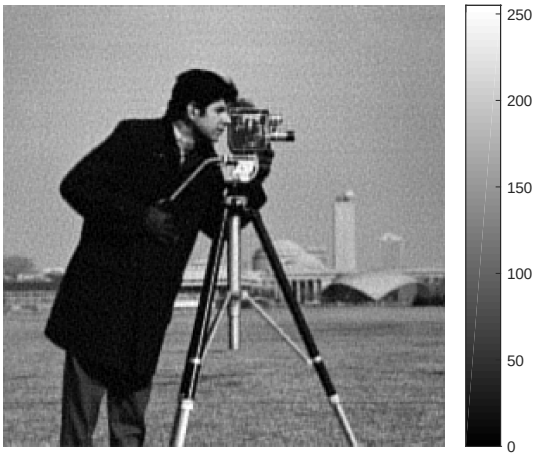


Figure 2.2: Restored image using Posterior Maximum estimator with Gaussian priors and Gradient regularization

As it can be seen, both regularization operators create good results. The main difference is that using the Laplacian operator gives the impression of small crosses distributed all over the image (those are specially visible in the sky), which is not the case in the Gradient regularization. Another difference is that corners are better in the gradient regularization, this can specially be seen in the cameraman's elbow.

By using the Conjugate Gradient it was possible to minimize the criterion $J(\boldsymbol{x})$. The values of hyper-parameters that obtain the minimum norm of error can therefore be used to compare with the results given by the Posterior Expectation estimator. However, before being able to do so, it is needed to be able to efficiently sample from high-dimensional Gaussian distributions. In next section is presented the algorithm that enables to do so and the improvements that were made to it.

2.3 Rejection Perturbation Optimization Algorithm

2.3.1 Perturbation Optimization Algorithm (PO)

PO was introduced in [4] and is an algorithm that efficiently samples from a specific type of high-dimensional Gaussian distribution. Suppose a Gaussian distribution from which one needs to be sample from and that can be written as:

$$g(\boldsymbol{x}) \propto \exp\Big(-\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{m}_k - \boldsymbol{M}_k\boldsymbol{x})^T\boldsymbol{R}_k^{-1}(\boldsymbol{m}_k - \boldsymbol{M}_k\boldsymbol{x})\Big) \tag{2.2}$$

A sample from $g(\boldsymbol{x})$ can be obtained with the following Algorithm:

Algorithm 4: Perturbation Optimization

1. Step P (Perturbation): Generate $k = 1...K$ independent vector

$$\eta_k \sim N(\boldsymbol{m}_k, \boldsymbol{R}_k)$$

2. Step O (Optimization): Compute $\hat{\boldsymbol{x}}_i$ as the minimizer of:

$$J(\boldsymbol{x}) = \sum_{k=1}^{K} (\eta_k - \boldsymbol{M}_k \boldsymbol{x})^T \boldsymbol{R}_k^{-1} (\eta_k - \boldsymbol{M}_k \boldsymbol{x})$$

$\hat{\boldsymbol{x}}_i$ is a sample from 2.2. The algorithm's demonstration is simple and can be looked in the original paper [4] where it is well explained.

In order for the Step P to be efficient, the sample from η_k needs to be fast and easy to do. This is the case if the $\boldsymbol{R}_k$ are for instance, diagonal, circular, or a mix of both.

The Step O needs an optimization to be performed. The natural choice is to use the Conjugate Gradient (CG) and explore every direction. However, being a high-dimensional problem, exploring all the directions would make the algorithm inefficient. The natural choice would be to only approximate the minimum by truncating the iterations. The problem with this approach is that there is no guarantee that the approximate minimum is a sample from $g(\boldsymbol{x})$. It is to solve this issue that the next algorithm was introduced.

2.3.2 PO improved: the Rejection Perturbation Optimization Algorithm (RJPO)

RJPO was introduced in [1]. As it was previously said, RJPO is way to certify that the truncated minimum obtained from the Step O is statistically guaranteed to be a sample from $g(\boldsymbol{x})$. The main idea of the paper is to introduce an acceptance-rejection step in order to guarantee it. The following algorithm is a modified version of the original RJPO algorithm which aims to reduce the number of iteration of the Optimization

step:

Algorithm 5: Rejection Perturbation Optimization

1. Step P (Perturbation): The same as PO
2. Step O (Optimization): Compute $\hat{\boldsymbol{x}}_i$ as the minimizer of $J(\boldsymbol{x})$
   a) Start a Conjugate Gradient initialized with $-\hat{\boldsymbol{x}}_{i-1}$ *i.e.,* the opposite of the previous sample
   b) **while** $\alpha \leq \alpha_{min}$ **do**
      i) Calculate $\hat{\boldsymbol{x}}_i^{(j)}$ the iteration's minimizer
      ii) Calculate

$$\alpha = \min(1, \exp(-\nabla J(\hat{\boldsymbol{x}}_i^{(j)})^T(\hat{\boldsymbol{x}}_i^{(j)} - \hat{\boldsymbol{x}}_{i-1})))$$

   c) Accept the sample $\hat{\boldsymbol{x}}_i^{(j)}$ with probability α

where:
- α_{min} is the minimum probability to test the sample, for instance $\alpha_{min} = 0.9$
- $\nabla J(\hat{\boldsymbol{x}}_i^{(j)})$ is $J(\cdot)$'s Gradient calculated in $\hat{\boldsymbol{x}}_i^{(j)}$

This sample is statistically guaranteed to be a sample from $g(\boldsymbol{x})$. The need of initializing CG in Step O with the opposite value of the previous sample is that, as it is proved in [1], the starting point and the final point need to be statistically independent and that the known way to achieve it is to use the opposite of the previous sample.

The step of testing for every iteration if the sample is good enough to go through the acceptance rejection step drastically reduces the number of iterations to have a sample of $g(\boldsymbol{x})$. This is positive since it reduces the calculation time of the sampling. However, this gain is not that substantial because of the need of initializing the RJPO's Conjugate Gradient with the opposite of the previous sample. In order to partially reduce this problem, it was implemented an RJPO using preconditioner.

2.3.3 A faster RJPO using a preconditioner

The possibility of using a preconditioner is mentioned in [1], the original paper that presents RJPO. However, there is no comparison on how more efficient is RJPO to PO neither on how using a preconditioner boosts up the speed of PO. In order to better understand RJPO behavior, both algorithm were applied to the SIM problem[3]. The preconditioner that was used is the same as described in Algorithm 3.

The algorithm was run 100 times. In Tables 2.1 and 2.2 are some of results of this comparison.

The use of PO initialized with $-\hat{\boldsymbol{x}}_{i-1}$ has as goal to compare how faster RJPO is then PO with the same initialization. As it can be seen, there is no doubt that RJPO

[3]The detailed explanation on how using RJPO to sample the Gaussian related to SIM is explained in section 2.4. This part aims only to compare the performances.

Table 2.1: Comparison of the time spent in each case (in seconds)

	PO with $\hat{\boldsymbol{x}}_{i-1}$	PO with $-\hat{\boldsymbol{x}}_{i-1}$	RJPO
normal	874	1100	687
preconditioned	772	1080	646

Table 2.2: Comparison of the average number of iterations in each case

	PO with $\hat{\boldsymbol{x}}_{i-1}$	PO with $-\hat{\boldsymbol{x}}_{i-1}$	RJPO
normal	67.9	81.9	51.8
preconditioned	52.5	74.7	45.2

is faster, both in time and in number of iterations, than PO, with the advantage that the $\hat{\boldsymbol{x}}_i$ is guaranteed to be a sample from $g(\boldsymbol{x})$.

The use of the preconditioner is less simple to analyze. It speeds up PO with the correct initialization in about 12% both in time and in number of iterations. However, in PO with a wrong initialization it has almost no effect. In RJPO the preconditioner speeds up the algorithm in about 6% in time and in about 10% in number o iterations. This difference between percentage of gain in time and in number of iterations is due to the fact that the operation of preconditioning itself takes some time, which reduces the gain. Nevertheless, the speed up brought by the preconditioner is important and its use will be maintain for the rest of the work.

Now, equipped with RJPO it is possible to efficiently sample from high-dimensional Gaussians and therefore to use the Posterior Expectation estimator and to apply it into the SIM problem that is being treated.

2.4 Supervised Posterior Expectation

The several samples $\hat{\boldsymbol{x}}_i$ from the distribution $p(\boldsymbol{x}|\boldsymbol{y}, \gamma_n, \gamma_l, \gamma_c)$ that need to be taken in order to apply PE are drawn using RJPO. As this is the supervised approach, the values of the hyper-parameters are set using the optimal ones determined in section 2.2. Here again will be tested both regularization operators

2.4.1 Laplacian Regularization

In the Laplacian regularization, the distribution that needs to be sampled is defined by:

$$p(\boldsymbol{x}|\boldsymbol{y}, \gamma_n, \gamma_l, \gamma_c) \propto \gamma_n^{M/2} \gamma_d^{N/2} \exp\left(-\frac{1}{2}\left((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{D}\boldsymbol{x}\right)\right) \tag{2.3}$$

Putting it as in 2.2:

$$g(\boldsymbol{x}) \propto \exp\Big(-\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{m}_k - \boldsymbol{M}_k\boldsymbol{x})^T\boldsymbol{R}_k^{-1}(\boldsymbol{m}_k - \boldsymbol{M}_k\boldsymbol{x})\Big)$$
$$K = 2$$
$$\boldsymbol{m}_1 = \boldsymbol{y}; \boldsymbol{M}_1 = \boldsymbol{H}; \boldsymbol{R}_k = (\gamma_n I)^{-1}$$
$$\boldsymbol{m}_2 = 0; \boldsymbol{M}_2 = \boldsymbol{D}; \boldsymbol{R}_k = (\gamma_d I)^{-1}$$

The PE was taken with 2000 samples. The result can be seen in figure 2.3. As it can be seen, the resultant estimation is really similar to the one obtained in the supervised approach. The norm of the error is of 2.1739×10^3 which is basically the same of the supervised case. In Figure 2.4 can be seen that the estimated standard deviation is small and regular all over the image. This regularity comes from the optical transfer function that forces some points to be really similar as for others to be more different but in a organized pattern. The Figure shows that RJPO can obtain good samples with not a lot of variation among them.

Figure 2.3: Restored image using the supervised Posterior Expectation estimator with Gaussian priors and Laplacian regularization

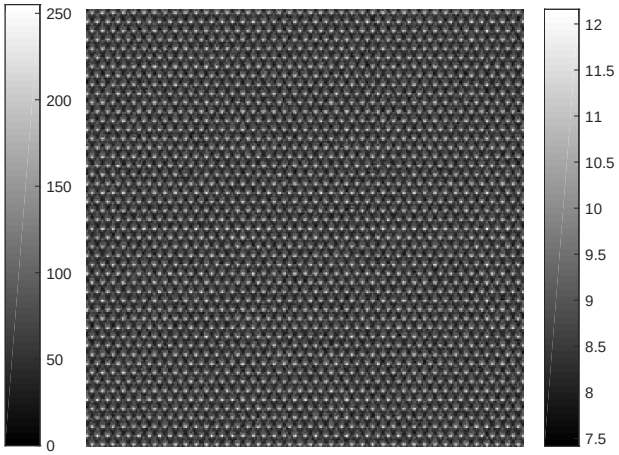


Figure 2.4: Standard deviation when using the supervised Posterior Expectation estimator with Gaussian priors and Laplacian regularization

2.4.2 Gradient Regularization

In the Gradient regularization, the distribution that needs to be sampled is defined by:

$$p(\boldsymbol{x}|\boldsymbol{y}, \gamma_n, \gamma_l, \gamma_c) \propto \gamma_n^{M/2} \det\Big(\gamma_l \boldsymbol{D}_l^T\boldsymbol{D}_l + \gamma_c\boldsymbol{D}_c^T\boldsymbol{D}_c\Big)^{1/2}$$
$$\exp\Big(-\frac{1}{2}\Big((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T\gamma_n(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T\boldsymbol{D}_l^T\gamma_l\boldsymbol{D}_l\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{D}_c^T\gamma_c\boldsymbol{D}_c\boldsymbol{x}\Big)\Big) \quad (2.4)$$

Putting it as in 2.2:

$$g(\boldsymbol{x}) \propto \exp\Big(-\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{m}_k - \boldsymbol{M}_k\boldsymbol{x})^T\boldsymbol{R}_k^{-1}(\boldsymbol{m}_k - \boldsymbol{M}_k\boldsymbol{x})\Big)$$

$$K = 3$$

$$\boldsymbol{m}_1 = \boldsymbol{y}; \boldsymbol{M}_1 = \boldsymbol{H}; \boldsymbol{R}_k = (\gamma_n I)^{-1}$$

$$\boldsymbol{m}_2 = 0; \boldsymbol{M}_2 = \boldsymbol{D}_l; \boldsymbol{R}_k = (\gamma_l I)^{-1}$$

$$\boldsymbol{m}_3 = 0; \boldsymbol{M}_2 = \boldsymbol{D}_c; \boldsymbol{R}_k = (\gamma_c I)^{-1}$$

The estimation was made again with 2000 samples. As it can be seen in Figure 2.5, the restored image is really similar to the one obtained with the supervised approach. The norm of the error is 2.1831×10^3 which is really close to the supervised case. In the Figure 2.6 it can be seen that the standard deviation when using the Gradient regularization is very similar to the one using the Laplacian regularization. It is interesting to note though that the regularity is less important and less rigid that in the Figure 2.4. This is a direct consequence of the use of an operator separable for the lines and the colons.

Figure 2.5: Restored image using the supervised Posterior Expectation estimator with Gaussian priors and Gradient regularization

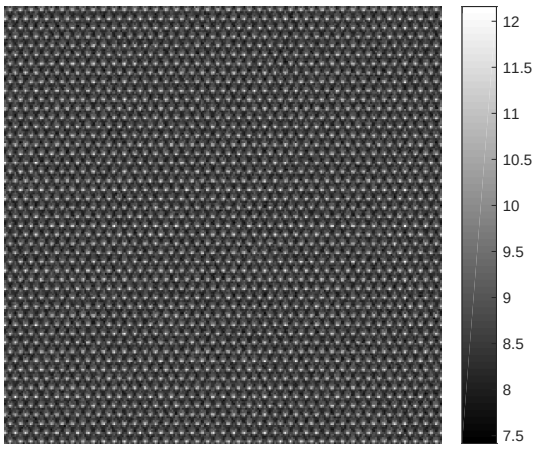


Figure 2.6: Standard deviation when using the supervised Posterior Expectation estimator with Gaussian priors and Gradient regularization

With those results, it is now possible to see how perform the unsupervised approach. Remember that the unsupervised approach is the desired one as the original image is not available so calculating the norm of the error is not possible.

2.5 Unsupervised Posterior Expectation

The difference of the supervised and the unsupervised approach is the sampling of the hyper-parameters. It can be done using the Gibbs sampler described in the Algorithm

1. The sample of the Gaussians is done using RJPO as described in the previous section.

2.5.1 Laplacian Regularization

The Laplacian case is exactly as it was described in the Algorithm 1. The distribution used to sample from the hyper-parameters are

$$\gamma_n \sim \mathscr{G}(M/2 - 1; 2/((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})))$$
$$\gamma_d \sim \mathscr{G}(N/2 - 1; 2/((\boldsymbol{D}\boldsymbol{x})^T(\boldsymbol{D}\boldsymbol{x})))$$

Where $\mathscr{G}$ is the gamma distribution.

The unsupervised PE estimator was used with 2000 samples. In Figure 2.7 it is possible to see the resultant estimation. As it can be seen, the resultant image is close to the one estimated in both the supervised case and in PM, with a norm of the error of being 2.1739×10^3. Figure 2.8 shows that even in the unsupervised PE the Laplacian regularization using RJPO produces samples close to each other *i.e.,* with a small standard deviation. This is obtained because the hyper-parameters converge fast enough and therefore the samples are Gaussians distributions.

Figure 2.7: Restored image using the unsupervised Posterior Expectation estimator with Gaussian priors and Laplacian regularization

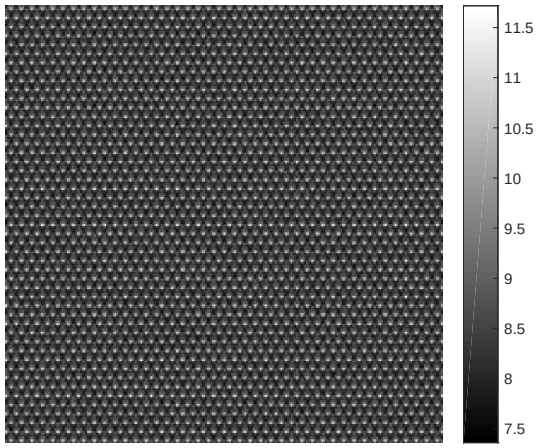


Figure 2.8: Standard deviation when using the unsupervised Posterior Expectation estimator with Gaussian priors and Laplacian regularization

As it can be seen in Figure 2.9 both hyper-parameters converge towards similar values of those determined in section 2.1. In Table 2.3 are compared the values determined by the PM estimator and the mean value and standard deviation of the hyper-parameters by the unsupervised case. As it can be seen, both mean are really close to the values determined for the PM estimator. Furthermore, the standard deviation is small in both cases. The property of convergence was tested for different

initialization and in all of them, convergence was achieved. However, the convergence to stable hyper-parameters can be slower for a bad initialization.

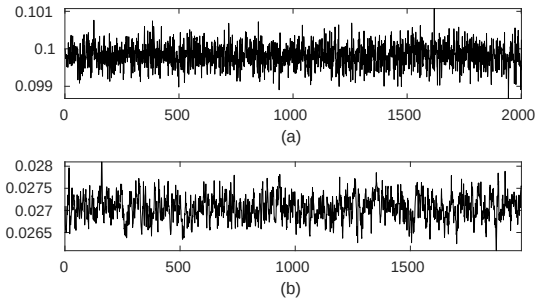


Figure 2.9: Hyper-parameters chain of values (a) γ_n (b) γ_d

	PM	PE	
	value	mean	std
γ_n	0.1	0.0998	0.0003
γ_d	0.0248	0.0270	0.0003

Table 2.3: Comparison of the values of hyper-parameters determined by PM and PE in the Laplacian Regularization

2.5.2 Gradient Regularization

The Gradient Regularization case has some differences for the sampling of the hyper-parameters. The noise's precision γ_n is still the same as in the previous case. However, the posterior distribution of the hyper-parameters γ_l and γ_c regularization operator is:

$$p(\gamma_l, \gamma_c \mid \boldsymbol{x}, \boldsymbol{y}, \gamma_n) \propto \gamma_l^{-1}\gamma_c^{-1} \det\left(\gamma_l \boldsymbol{D}_l^T \boldsymbol{D}_l + \gamma_c \boldsymbol{D}_c^T \boldsymbol{D}_c\right)^{1/2} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}^T \boldsymbol{D}_l^T \gamma_l \boldsymbol{D}_l \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{D}_c^T \gamma_c \boldsymbol{D}_c \boldsymbol{x}\right)\right)$$
$$\text{for } (\gamma_l, \gamma_c) > (0,0);\ 0 \text{ otherwise}$$

The problem with this probability density function is that there is no known way to directly sample from it. Therefore, two steps need to be taken. The first one is to take the conditional distribution of each which leads to the following probability density functions:

$$p(\gamma_l|\boldsymbol{x}, \boldsymbol{y}, \gamma_n, \gamma_c) \propto \gamma_l^{-1} \det\left(\gamma_l \Lambda_{Dl}^\dagger \Lambda_{Dl} + \gamma_c \Lambda_{Dc}^\dagger \Lambda_{Dc}\right)^{1/2} \quad \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{D}_l^T \gamma_l \boldsymbol{D}_l \boldsymbol{x}\right)$$
$$\propto \gamma_l^{-1}\left(\prod_{i=1}^{n}(\gamma_l|\lambda_{Dl;i}|^2 + \gamma_c|\lambda_{Dc;i}|^2)\right)^{1/2} \quad \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{D}_l^T \gamma_l \boldsymbol{D}_l \boldsymbol{x}\right)$$
$$\text{for } \gamma_l > 0;\ 0 \text{ otherwise}$$
$$p(\gamma_c|\boldsymbol{x}, \boldsymbol{y}, \gamma_n, \gamma_l) \propto \gamma_c^{-1} \det\left(\gamma_l \Lambda_{Dl}^\dagger \Lambda_{Dl} + \gamma_c \Lambda_{Dc}^\dagger \Lambda_{Dc}\right)^{1/2} \quad \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{D}_c^T \gamma_c \boldsymbol{D}_c \boldsymbol{x}\right)$$
$$\propto \gamma_c^{-1}\left(\prod_{i=1}^{n}(\gamma_l|\lambda_{Dl;i}|^2 + \gamma_c|\lambda_{Dc;i}|^2)\right)^{1/2} \quad \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{D}_c^T \gamma_c \boldsymbol{D}_c \boldsymbol{x}\right)$$
$$\text{for } \gamma_c > 0;\ 0 \text{ otherwise}$$

where $\lambda_{dl;i}$ is i^{th} element of the diagonalized (in the Fourier domain) regularization operator (and equivalently for $\lambda_{cl;i}$) .

There is also no way to directly sample from those conditional distributions. However, the second step is that, as they are single-dimensional, they can be sampled using a Metropolis-Hastings algorithm. If the reader is not familiar with Metropolis-Hastings, they can check in [8] for a very good and simple explanation. The instrumental distribution used was a Gaussian distribution with mean = 0 and variance = 0.001. The acceptance rate was calibrated to be around 66%

With all of this set, a first PE estimation was made using 2000 samples. The unsupervised PE estimator clearly presented some instability. As it can be seen in Figure 2.10 the hyper-parameters do not converge to any value and seem unstable. In Table 2.4 it can be seen that the mean of γ_n is close to the value determined by the PM estimator in section 2.2 and that its standard deviation is small. However both γ_l and γ_c have means very different from those determined by the PM estimator and both have very high standard deviation. The reason why this happens is not completely clear and is better discussed in the next section.

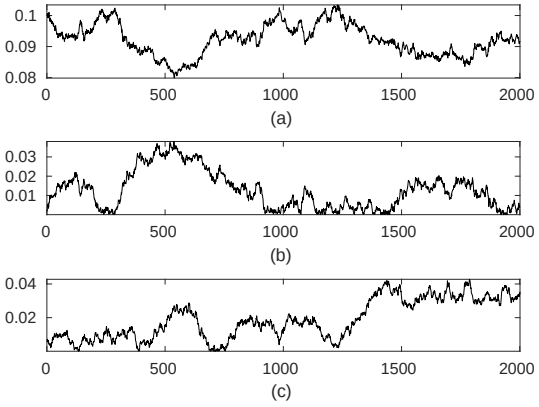


Figure 2.10: Hyper-parameters chain of values (a) γ_n (b) γ_l (c) γ_c

	PM	PE	
	value	mean	std
γ_n	0.1	0.0922	0.0050
γ_l	0.0034	0.0127	0.0097
γ_c	0.0084	0.0188	0.0116

Table 2.4: Comparison of the values of hyper-parameters determined by PM and PE in the Gradient Regularization

Furthermore, as it can be seen in Figure 2.11, the standard deviation of the image is extremely high, specially around the edges. This means that the algorithm has troubles to sample correctly from those parts of the image. This is an important indicator that the non convergence of the hyper parameters has an actual influence on the samples that are produced.

However, it can be seen in Figure 2.12, it seems that for the first 200 iterations, the values for γ_l and γ_c were close to those determined in section 2.1. This is confirmed by the results presented in Table 2.5: the mean value of the hyper-parameters are all closer to those determined by PM in section 2.2. This partial convergence is related to the initialization as it needs the hyper-parameters to be initialized with values not too different from those determined by the PM, however, if they are close enough, this property is always observed. The reason why this may be the case is discussed in the next section.

Figure 2.11: Standard deviation using the unsupervised Posterior Expectation estimator with Gaussian priors and Gradient regularization and 200 samples

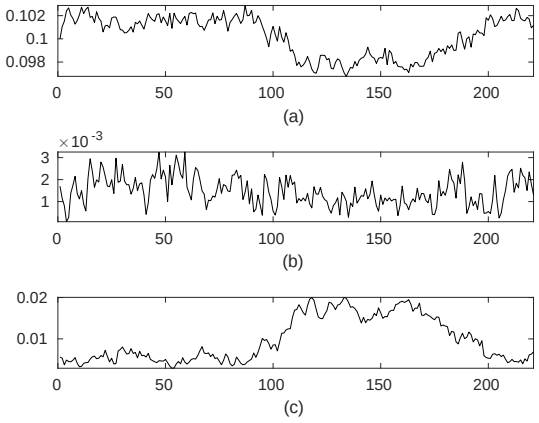


Figure 2.12: Hyper-parameters chain of values for 200 samples
(a) γ_n (b) γ_l (c) γ_c

	PM	PE	
	value	mean	std
γ_n	0.1	0.1002	0.0017
γ_l	0.0034	0.0019	0.0007
γ_c	0.0084	0.0097	0.0054

Table 2.5: Comparison of the values of hyper-parameters determined by PM and PE in the Gradient Regularization for 200 samples

It is therefore possible to use this property of initial convergence of the hyper-parameters to do a second unsupervised PE estimation using only 200 samples. The result can be seen in Figure 2.13. The estimated image is actually a good one with the norm of the error being 2.2248×10^3. This partially contradicts the theory that more samples create a better image, but it is the consequence of having hyper-parameters that are closer to the optimal hyper-parameters determined by the supervised approach. Of course normally those values are not known, and therefore this approach could not be directly done. In order to solve this issue an approach could be to use a supervised and not very precise PM in order to obtain hyper-parameters' values that can be used as initialization. As it can be seen in Figure 2.14, reducing the number of samples also has as consequence to create a smaller standard deviation. The pattern of the

optical transfer function starts to reappear which is a sign that the samples are closer to each other.

Figure 2.13: Restored image using the unsupervised Posterior Expectation estimator with Gaussian priors, Gradient regularization and 200 samples

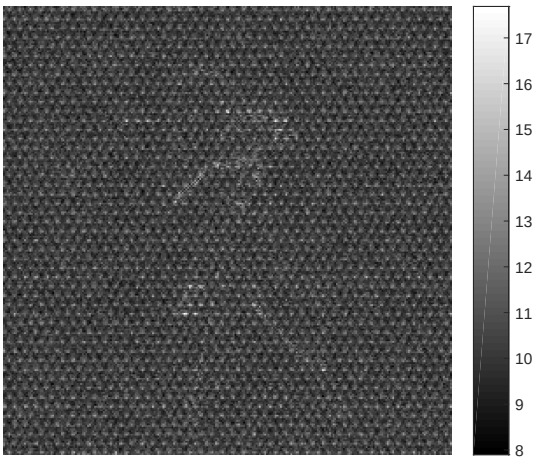


Figure 2.14: Standard deviation when using the unsupervised Posterior Expectation estimator with Gaussian priors, Gradient regularization and 200 samples

2.6 Results's Analyses

The conclusions presented in here, even if they are illustrated with the results obtained in the chapter, are quite general and can also be observed with other images as for different amount of noise.

The first and most important conclusion is that the results obtained using PM, supervised PE and unsupervised PE are very similar as it can be seen by the comparison of the norm of the error in Table 2.6. Even if this was expected, quantifying this result demonstrates the efficiency of the unsupervised Posterior Expectation as it is able to determine estimations as good as those obtained by manually testing the results and then comparing. It also shows that the method can be trusted to give good results when used with Laplacian regularization. It can also be trusted for the Gradient regularization if it is initialized with some caution and not to many samples are taken.

Table 2.6: Comparison of norm of the error obtained using the different methods and different regularization operators

$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$	PM	PE	
	supervised	supervised	unsupervised
Laplacian	2.1719×10^3	2.1739×10^3	2.1739×10^3
Gradient	2.1819×10^3	2.1831×10^3	2.2248×10^3

A second and also important conclusion is that using the Gradient regularization gives better results. Even if this does not appear numerically when calculating the norm of the error, it is visually noticeable as the resultant estimation seems as if it is more focused. It also does not produce the effect of small crosses that are visible when using the Laplacian regularization. This conclusion is the motivation for looking for potentials where the gradient regularization can be used as it will be done in next chapter.

A third conclusion is on the standard deviations. As it was seen in the standard deviations figures shown in the chapter, the standard deviation reveals a lot about the quality of the estimation. At least for the Gaussian case, if the estimation is good it means that the standard deviation will be small and that the image seen should be close to the pattern of the optical transfer function. However, if the estimation of the hyper parameters is not stable, this influences the standard deviation that will increase specially around the points were the model has more difficulty to sample, in this case, the edges.

The final conclusion, which is for now more of an empirical conclusion, is that using too many samples on the unsupervised case when using the Gradient Regularization produces bad results as the hyper-parameters start to diverge after a certain number of samples. It is not completely clear why this happens. The hypothesis that is made is the following. As it was already stated, the posterior distribution of γ_c and γ_l is:

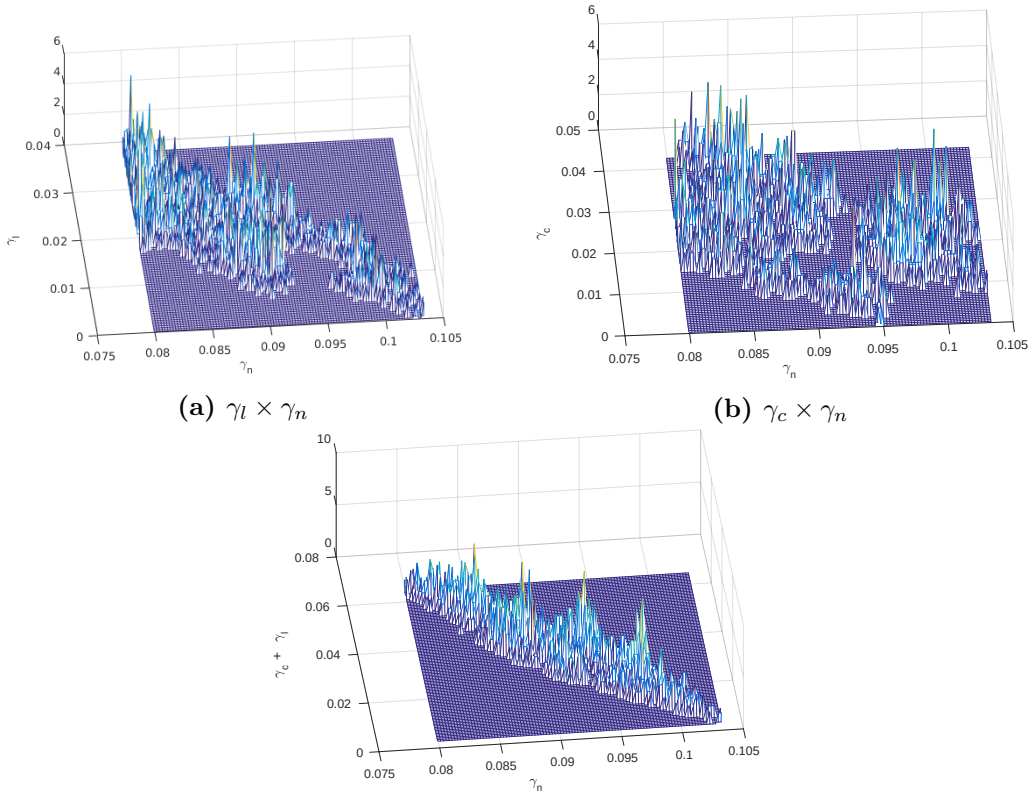
$$p(\gamma_l, \gamma_c | \boldsymbol{x}, \boldsymbol{y}, \gamma_n) \propto$$
$$\gamma_l^{-1}\gamma_c^{-1} \det\left(\gamma_l \boldsymbol{D}_l^T \boldsymbol{D}_l + \gamma_c \boldsymbol{D}_c^T \boldsymbol{D}_c\right)^{1/2} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}^T \boldsymbol{D}_l^T \gamma_l \boldsymbol{D}_l \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{D}_c^T \gamma_c \boldsymbol{D}_c \boldsymbol{x}\right)\right)$$

As it can be seen, the main term in the distribution is $\gamma_l \boldsymbol{D}_l^T \boldsymbol{D}_l + \gamma_c \boldsymbol{D}_c^T \boldsymbol{D}_c$. This means that the distribution is more influenced by the sum of the hyper-parameters than by the hyper-parameters themselves. Even if the addition only appears in a part of the conditional distribution, it might be the cause of the instability. What is probably happening is an indetermination of the joint distribution: when a sample of one of the hyper-parameters randomly goes to a less likely place, the sample of the next one will compensate it by also changing. Because of this, when taking few samples and with a good initialization there is less probability that one of the hyper-parameters might go to a less likely region and drives the other one there. This hypothesis is reinforced by the data on the histograms presented in Figure 2.15 [4] and in Table 2.7. The hyper-parameter γ_n has as strong correlation to $\gamma_c + \gamma_l$ and a smaller one to each one of them separately.

Table 2.7: Correlation between the hyper-parameters

$\gamma_l \times \gamma_n$	$\gamma_c \times \gamma_n$	$(\gamma_c + \gamma_l) \times \gamma_n$
-0.5825	-0.2502	-0.9065

[4] The angles of the histogram were deliberately chosen different for a better view



(a) $\gamma_l \times \gamma_n$ **(b)** $\gamma_c \times \gamma_n$

(c) $(\gamma_c + \gamma_l) \times \gamma_n$

Figure 2.15: Histogram of different relation between the hyper-parameters

It might be that different priors on γ_c and γ_l could solve this issue, nevertheless some other priors were tested without success. Therefore, for the rest of the work the strategy of maintaining a small number of samples and a good initialization will be used when using the Gradient Regularization.

With those results, a first step was achieved towards the image restoration. However, it was using only Gaussian distributions. In the next chapter is presented a way to efficiently sample from other types of distributions

3 Models Based on Mixture of Gaussian

The results obtained in the previous chapter show the quality of the restoration that can be achieved using Gaussian models. However, the framework was a rather restrictive one as it can only be applied to Gaussian priors and Gaussian noise. Nevertheless, many situations could possibly be better modeled with other types of distributions. In addition to that,typical one-dimensional distributions may have several multi-dimensional generalization, as it is the case of the exponential distribution [9], and those generalization are not necessarily easy to sample from.

In order to solve this issue, the approach that is taken is based on Mixtures of Gaussians [1]. This approach allows one to construct some distributions, called the target distribution, in terms of two separate distribution: a multivariate Gaussian distribution that contains the correlation information and an auxiliary distribution which samples are independent and therefore easy to sample from.

Two ways of creating mixture of Gaussians are presented, the Location Mixture of Gaussian (LMG) that acts on the Gaussian's mean and the Scale Mixture of Gaussian (SMG) that acts on the Gaussian's precision matrix.

The main difficulty that is addressed is that the auxiliary variables used in the mixture need to have a separable posterior distribution. In both cases this is overcome by a correct construction of the prior distribution of the auxiliary variable that leads to a separable partition function of the mixture model. Furthermore, by explicitly calculating the partition function, it is possible to construct a framework where hyper-parameters can be sampled and therefore this mixture models can be used in an unsupervised approach.

In the first two sections of this chapter the formal construction of LMG and SMG is presented. Both sections are divided in three distinct parts to ease up the understanding: First part is the formal construction of the mixture, second part is the calculation of the partition function and showing that it is separable and third part is how to determine the auxiliary distribution that creates the target one.

In the last section of the chapter it is explained how to efficiently sample from those distributions and how this formality can be applied in an inverse problem using unsupervised Posterior Estimation.

[1] Mixture models are also known in the continuous case as Compound Distributions

3.1 Location Mixture of Gaussian (LMG)

3.1.1 Constructing a Location Mixture of Gaussian

The Location Mixture of Gaussian is based on assigning random variables to a Gaussian's mean. The formality that is presented in here is a based in the one presented in [2]. The present work brings two main contributions. The first one is that the original paper only addresses cases where the precision matrices are diagonalizable in the Fourier domain, this work shows that the formality can be extended using RJPO. The second one is that it is claimed in the original paper that the regularization operation needs to have the same number of cliques that the images has of pixels. This would be equivalent to be restricted to a single regularization operator could be used. The contribution is to prove that several regularization operators can be used and that if those are well constructed, sampling hyper-parameters is possible.
Let us consider a target distributions $p(\boldsymbol{x})$. Suppose it is defined by:

$$p(\boldsymbol{x}) = C_p^{-1} \prod_{k=1}^{K} p_k(\boldsymbol{x}) \tag{3.1}$$

Where $p_k(\cdot)$ are probability density functions. In here it is important to note that it is not stated that the $p_k(\cdot)$ are independent. What is being said is that the probability density function is partially separable in a product of probability density functions and a common normalizer C_p. Each one of this distributions is associated with one regularization operator.
Now suppose that each $p_k(\boldsymbol{x})$ can be written as:

$$p_k(\boldsymbol{x}) \propto \int_{\mathbb{R}^n} \pi_k(\boldsymbol{x}, \boldsymbol{b}_k) d\boldsymbol{b}_k \tag{3.2}$$

where $\pi_k(\boldsymbol{x}, \boldsymbol{b}_k)$ is a multivariate probability density function of dimension n; this is the same as considering that $p_k(\boldsymbol{x})$ is the marginalization of $\pi_k(\boldsymbol{x}, \boldsymbol{b}_k)$. Therefore, $p(\boldsymbol{x})$ can be rewritten as:

$$p(\boldsymbol{x}) \propto \prod_{k=1}^{K} \int_{\mathbb{R}^n} \pi_k(\boldsymbol{x}, \boldsymbol{b}_k) d\boldsymbol{b}_k \tag{3.3}$$

Since $\pi_k(\boldsymbol{x}, \boldsymbol{b}_k)$ is a probability density function, Fubini's theorem can be applied:

$$p(\boldsymbol{x}) \propto \int_{\mathbb{R}^{nK}} \left(\prod_{k=1}^{K} \pi_k(\boldsymbol{x}, \boldsymbol{b}_k) \right) d\boldsymbol{b}_1 d\boldsymbol{b}_2 ... d\boldsymbol{b}_K \tag{3.4}$$

Consider now that $\pi_k(\boldsymbol{x}, \boldsymbol{b}_k)$ can be written as:

$$\pi_k(\boldsymbol{x}, \boldsymbol{b}_k) \propto g_k(\boldsymbol{x}|\boldsymbol{b}_k) F_k(\boldsymbol{b}_k) d\boldsymbol{b}_k \tag{3.5}$$

where $g_k(\boldsymbol{x})$ is multivariate Gaussian distribution of dimension n and $F_k(\boldsymbol{b}_k)$ is an auxiliary distribution of dimension n. In here it is important to clearly state that each

element $b_{k;j}$ of b_k is independent of each other, and therefore:

$$F_k(\boldsymbol{b}_k) = \prod_{j=1}^{n} f_k(b_{k;j}) \tag{3.6}$$

this will not be needed for the rest of the deduction, however it will be essential for the sampling and making it clear is fundamental.
Now reorganizing $p(\boldsymbol{x})$

$$p(\boldsymbol{x}) \propto \int_{\mathbb{R}^{nK}} \left(\prod_{k=1}^{K} g_k(\boldsymbol{x}|\boldsymbol{b}_k) F_k(\boldsymbol{b}_k) \right) d\boldsymbol{b}_1 d\boldsymbol{b}_2 ... d\boldsymbol{b}_K \tag{3.7}$$

$$p(\boldsymbol{x}) \propto \int_{\mathbb{R}^{nK}} \left(\prod_{k=1}^{K} g_k(\boldsymbol{x}|\boldsymbol{b}_k) \right) \left(\prod_{k=1}^{K} F_k(\boldsymbol{b}_k) \right) d\boldsymbol{b}_1 d\boldsymbol{b}_2 ... d\boldsymbol{b}_K \tag{3.8}$$

For the next steps the distributions need to be written detailed. By choosing $g_k(\boldsymbol{x}|\boldsymbol{b}_k)$ in the form:

$$g_k(\boldsymbol{x}|\boldsymbol{b}_k) \propto \exp\Big(-\frac{1}{2} \left(\boldsymbol{b}_k - \boldsymbol{D}_k \boldsymbol{x} \right)^T \gamma_k (\boldsymbol{b}_k - \boldsymbol{D}_k \boldsymbol{x}) \Big) \tag{3.9}$$

And by defining

$$G(\boldsymbol{x}|\boldsymbol{b}) := \exp\Big(-\frac{1}{2} \sum_{k=1}^{K} (\boldsymbol{b}_k - \boldsymbol{M}_k \boldsymbol{x})^T \gamma_k (\boldsymbol{b}_k - \boldsymbol{M}_k \boldsymbol{x}) \Big) \tag{3.10}$$

Then equation 3.8 can be written as:

$$\boxed{p(\boldsymbol{x}) = C_p^{-1} \int_{\mathbb{R}^{nK}} G(\boldsymbol{x}|\boldsymbol{b}) \left(\prod_{k=1}^{K} F_k(\boldsymbol{b}_k) \right) d\boldsymbol{b}_1 d\boldsymbol{b}_2 ... d\boldsymbol{b}_K} \tag{3.11}$$

where C_p is the normalizing coefficient.

The Equation 3.11 is an important result. As it can be seen, $p(\boldsymbol{x})$ has two distinct parts. The first one is a Gaussian distribution that contains all the information on the covariance. The second one is the product of separable density functions. Furthermore, the number K being the amount of regularization operator, it is clear that the constructed LMG is extendable to more than one regularization operator. In order to know if each $b_{k,j}$ has a posterior distribution which is conditionally independent, it is needed to show that the partition function is separable.

3.1.2 Calculating the partition function

The reason why it is needed to explicitly calculate the partition function is that both hyper-parameters and the auxiliary variables may be in it and they will be needed

when sampling their posterior distribution. The partition function can be explicitly calculated by:

$$C_p = \int_{\mathbb{R}^n} p(\boldsymbol{x}) d\boldsymbol{x} \tag{3.12}$$

$$C_p = \int_{\mathbb{R}^n} \int_{\mathbb{R}^{nK}} G(\boldsymbol{x}|\boldsymbol{b}) \left(\prod_{k=1}^{K} F_k(\boldsymbol{b}_k) \right) d\boldsymbol{b}_1 d\boldsymbol{b}_2 ... d\boldsymbol{b}_K d\boldsymbol{x} \tag{3.13}$$

$$C_p = \int_{\mathbb{R}^{nK}} \left(\int_{\mathbb{R}^n} G(\boldsymbol{x}|\boldsymbol{b}) d\boldsymbol{x} \right) \left(\prod_{k=1}^{K} F_k(\boldsymbol{b}_k) \right) d\boldsymbol{b}_1 d\boldsymbol{b}_2 ... d\boldsymbol{b}_K \tag{3.14}$$

$$C_p = \int_{\mathbb{R}^{nK}} C_G \left(\prod_{k=1}^{K} F_k(\boldsymbol{b}_k) \right) d\boldsymbol{b}_1 d\boldsymbol{b}_2 ... d\boldsymbol{b}_K \tag{3.15}$$

with:

$$C_G := \det \left(\sum_{k=1}^{K} \boldsymbol{D}_k^T \gamma_k \boldsymbol{D}_k \right)^{-1/2} (2\pi)^{n/2} \tag{3.16}$$

Since C_G does not depend on $\boldsymbol{b}_k$:

$$C_p = C_G \int_{\mathbb{R}^{nK}} \left(\prod_{k=1}^{K} F_k(\boldsymbol{b}_k) \right) d\boldsymbol{b}_1 d\boldsymbol{b}_2 ... d\boldsymbol{b}_K \tag{3.17}$$

$$C_p = C_G \prod_{k=1}^{K} \int_{\mathbb{R}^n} F_k(\boldsymbol{b}_k) d\boldsymbol{b}_k \tag{3.18}$$

Thus:

$$\boxed{C_p = C_G} \tag{3.19}$$

The partition function therefore depends only on the Gaussian's partition function which itself only depends on its precision matrix. The partition function is independent of the location of the Gaussian. This result means that each $b_{k,j}$ can be sampled independently of each other. Therefore any set of distributions h_k can be chosen and will not interfere on the sampling of the hyper-parameters. However, the problem is not completely solved as calculating the determinant in C_G would itself be a problem when working with high-dimensional Gaussians. There are therefore three cases that simplify this calculation:

1. If $K = 1$:

$$C_G^{-1} = \gamma^{N/2} \det \left(\boldsymbol{D}^T \boldsymbol{D} \right)^{1/2} \tag{3.20}$$

   In this case, the determinant does not need to be explicitly calculated as it enters the proportionality constant and therefore has no influence on the sampling of γ.

2. If all γ_k are the same, *i.e.*, $\gamma_1 = \gamma_2 ... = \gamma_K = \gamma$:

$$C_G^{-1} = \gamma^{N/2} \det \left(\sum_{k=1}^{K} \boldsymbol{D}_k^T \boldsymbol{D}_k \right)^{1/2} \tag{3.21}$$

Again in this case, the determinant does not need to be explicitly calculated in order to sample γ.

3. If the operators $\boldsymbol{D}_k$ are diagonalizable in the same base *i.e.,* they commute:

$$C_G^{-1} = \left(\prod_{i=1}^{n} \left(\sum_{k=1}^{K} \gamma_k |\lambda_{k;i}|^2 \right) \right)^{1/2} \tag{3.22}$$

where $\lambda_{k;i}$ is the i^{th} diagonal element of the diagonalized matrix $\boldsymbol{D}_k$. In this case, the determinant is just easier to calculate, however the sampling of the hyper-parameters is not as simple as in the two previous cases as it will be seen in section 3.3. Another point is that the diagonalization must be a known and easy one, which is not always the case. The two basic cases are evidently if the matrices $\boldsymbol{D}_k$ are already diagonal or if they are circular.

This concludes the construction of a separable Location Mixture of Gaussian. As it will be seen in the last section, because the distribution is composed of the distinct parts, it can be sampled easily associating RJPO and independent samples of the auxiliary variables $b_{k,j}$. Furthermore, the construction that was achieved is applicable when having several regularization operator, which is an important contribution as in the case treated in this work, sampling using line operators and colons operators was shown to produce better results.

3.1.3 Determining the auxiliary distribution based on the target

Before finishing this section it is necessary to mention which kind of distributions can be constructed using LMG. As this is not the main scope of this work and as non existing work was found, only a main idea on how to find the auxiliary distribution based on the target is given. By writing the equation 3.2

$$\begin{aligned} p_k(\boldsymbol{x}) &\propto \int_{\mathbb{R}^n} \pi_k(\boldsymbol{x}, \boldsymbol{b}_k) d\boldsymbol{b}_k \\ &\propto \int_{\mathbb{R}^n} \exp\left(-\frac{1}{2} (\boldsymbol{b}_k - \boldsymbol{D}_k \boldsymbol{x})^T \gamma_k (\boldsymbol{b}_k - \boldsymbol{D}_k \boldsymbol{x}) \right) F_k(\boldsymbol{b}_k) d\boldsymbol{b}_k \end{aligned}$$

It is clear that it is really similar to the convolution of the normal distribution and F_k. This particular view can be useful if the modeled process is the addition of a Gaussian random variable and another random variable. Another possible way of finding distributions would be through the characteristic function as it is a powerful tool to manage convolution operations.

This last topic closes the discussion on the Location Mixture of Gaussian. This method clearly cannot build any distribution and therefore it is useful to think in another type of mixture that would enable so. This is why another type of mixture is introduced, the Scale Mixture of Gaussian which can build other distributions.

3.2 Scale Mixture of Gaussian (SMG)

3.2.1 Constructing a Scale Mixture of Gaussian

Scale Mixture of Gaussian is based on the same idea as Location Mixture of Gaussian, however, instead of assigning a random variable to the mean, the random variable is assigned to the covariance matrix. The way this formulation is made is heavily influenced by what is done in the previous section. Even if Scale Mixtures of Gaussians are known for some time, to the knowledge of the author, what is presented in this section is an original contribution.

The change of the mixing parameters brings intrinsic differences to the equations that were developed in the previous section. One of those difference is that it was not find a way of constructing a separable SMG with more than one regularization operator. Details on the matter are given in section 5.2.4.
Let us consider a target distributions $p(\boldsymbol{x})$. Suppose it is defined by:

$$p(\boldsymbol{x}) \propto \int_{\mathbb{R}^n} g(\boldsymbol{x}|\boldsymbol{S})Q'(\boldsymbol{S})d\boldsymbol{S} \tag{3.23}$$

where $g(\boldsymbol{x}|\boldsymbol{S})$ is a multivariate Gaussian distribution of dimension n, $Q'(\boldsymbol{S})$ is an auxiliary measurable function of dimension n and $\boldsymbol{S}$ is a diagonal random matrix of dimension $n \times n$. In an analogous way of what is the case in LMG, each element s_j of $\boldsymbol{S}$ is independent and therefore:

$$Q'(\boldsymbol{S}) = \prod_{j=1}^{n} q'(s_j) \tag{3.24}$$

As it will be seen later, $q'(\cdot)$ is the derivation of a probability density function $q(\cdot)$ and this will ease up some calculations further in the calculations. However $Q'(\cdot)$ is not a derivation itself and in this case " $'$ " is just used as a reminder of the derivations.
Now explicitly writing $g(\boldsymbol{x}|\boldsymbol{S})$:

$$g(\boldsymbol{x}|\boldsymbol{S}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{D}^T\boldsymbol{S}^2\boldsymbol{D}\boldsymbol{x}\right) \tag{3.25}$$

and substituting it in 3.23

$$\boxed{p(\boldsymbol{x}) = C_p^{-1}\int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{D}^T\boldsymbol{S}^2\boldsymbol{D}\boldsymbol{x}\right)Q'(\boldsymbol{S})d\boldsymbol{S}} \tag{3.26}$$

As it was the case in LMG, $p(\boldsymbol{x})$ is composed of two distinct parts. One is an Gaussian distributions that can be sampled from using RJPO. The other part is a product of independent random variables s_j. As it was the case in LMG, in order to know if the elements of $\boldsymbol{S}$ are conditionally independent it is needed to calculate the partition function C_p.

3.2.2 Calculating the partition function

Having obtained a construction of the probability function $p(\boldsymbol{x})$ as a SMG, it is needed to calculate its partition function. As it was the case in LMG, the partition function needs to be separable in order for the sampling of S to be efficient.

$$C_p = \int_{\mathbb{R}^n} p(\boldsymbol{x}) d\boldsymbol{x} \tag{3.27}$$

$$C_p = \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} \exp\Big(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{D}^T \boldsymbol{S}^2 \boldsymbol{D}\boldsymbol{x}\Big) Q'(\boldsymbol{S}) d\boldsymbol{S} d\boldsymbol{x} \tag{3.28}$$

$$C_p = \int_{\mathbb{R}^n} \left(\int_{\mathbb{R}^n} \exp\Big(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{D}^T \boldsymbol{S}^2 \boldsymbol{D}\boldsymbol{x}\Big) d\boldsymbol{x} \right) Q'(\boldsymbol{S}) d\boldsymbol{S} \tag{3.29}$$

$$C_p = \int_{\mathbb{R}^n} (\det\left(\boldsymbol{D}^T \boldsymbol{S}^2 \boldsymbol{D}\right)^{1/2} (2\pi)^{-n/2}) Q'(\boldsymbol{S}) d\boldsymbol{S} \tag{3.30}$$

$$C_p = \det(D)(2\pi)^{-n/2} \int_{\mathbb{R}^n} \det(\boldsymbol{S}^2)^{1/2} Q'(\boldsymbol{S}) d\boldsymbol{S} \tag{3.31}$$

$$C_p = \det(D)(2\pi)^{-n/2} \int_{\mathbb{R}^n} \prod_{j=1}^{n} s_j \, q'(s_j) ds_1 ds_2 ... ds_n \tag{3.32}$$

$$\boxed{C_p = \det(D)(2\pi)^{-n/2} \prod_{j=1}^{n} \int_{\mathbb{R}} s_j \, q'(s_j) ds_j} \tag{3.33}$$

The equation 3.33 is the essential contribution of this section. As it can be seen, the partition function in the case of SMG does depend on the chosen distribution $q'(\cdot)$. However, the way that the construction was made, the auxiliary variable s_j only appears associated to its own distribution. This mean that each s_j can be dealt with independently and, in special, can be sampled from independently.

With this result, it is possible to study how to determine the auxiliary distribution that enables to sample using SMG.

3.2.3 Determining the auxiliary distribution based on the target

The last part, a way of determining how to construct a prior distribution is needed. The approach that will be used in here is the one described in [10], one can also look in [11] for another construction.

Since $Q'(\boldsymbol{S})$ is separable and that the partition function also is, equation 3.23 can be written as:

$$p(\boldsymbol{x}) \propto \int_{\mathbb{R}^n} g(\boldsymbol{x}|\boldsymbol{S}) Q'(\boldsymbol{S}) d\boldsymbol{S} \propto \prod_{j=1}^{n} \int_{\mathbb{R}} s_j \exp\Big(-\frac{1}{2} s_j^2 \bar{x}_j^2\Big) q'(s_j) ds_j \tag{3.34}$$

where $\bar{x}_j$ is the j^{th} element of $\boldsymbol{D}\boldsymbol{x}$ The main idea used in [10] is to see $p(\bar{x}_j)$ as a type of Laplace transform of $q'(s_j)$. It is proven that, if

$$\Big(-\frac{d}{d\bar{x}_j}\Big)^k p(\bar{x}_j^{1/2}) \geq 0 \text{ for } \bar{x}_j > 0 \tag{3.35}$$

then a representation of $p(\cdot)$ as a Scale Mixture of Gaussians exists.

Furthermore, a simple way of finding $q'(s_j)$ is proposed in the original paper. It is presented as Algorithm 6:

Algorithm 6: Finding SMG's auxiliary function using Laplace transform

1. Find $\xi(t)$ the inverse Laplace transform of $p(\bar{x}_j^{1/2})$
2. Make the change of variable $t = s_j^2/2$
3. Determine $q'(s_j) = \xi(s_j^2/2)$
4. Calculate the primitive function $q(s_j) = \int q'(s_j)ds_j$

Therefore, by sampling s_j determined by the distribution $q(\cdot)$, $p(\bar{x}_j)$ will follow the desired distribution.

This ends up the analyses on SMG. Next section will discuss on how to efficiently sample from the constructed distributions of LMG and SMG.

3.3 Efficiently Sampling from the Constructed Distributions

This section summarizes all the major contributions done by this work. In here, it is shown that the constructed distributions can not only be efficiently sampled but also can be integrated in a unsupervised Bayesian framework.

3.3.1 Constructing a Posterior Distribution

In the previous section, a model of multi-dimensional Gaussian mixtures was constructed. However, it was not presented how the construction can be used to efficiently sample from those distributions. This section has as goal to show how to do so and how to apply it to high-dimensional inverse problems.

In order to illustrate both LMG and SMG at the same time, a hypothetical example will be constructed. This example can then easily be extended to all possible combinations that are desired. The likelihood is described by a SMG:

$$p(\boldsymbol{y}|\boldsymbol{x}) \propto \int_{\mathbb{R}^m} \det(\boldsymbol{S})^{1/2} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})^T\boldsymbol{S}^2(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})\right) Q'(\boldsymbol{S})d\boldsymbol{S} \tag{3.36}$$

And the prior is described by an LMG with K=2

$$\begin{aligned} p(\boldsymbol{x}|\gamma_l\gamma_c) \propto \int_{\mathbb{R}^{n\times n}} &\exp\left(-\frac{1}{2}\left((\boldsymbol{b}_c-\boldsymbol{D}_c\boldsymbol{x})^T\gamma_c(\boldsymbol{b}_c-\boldsymbol{D}_c\boldsymbol{x}) + (\boldsymbol{b}_l-\boldsymbol{D}_l\boldsymbol{x})^T\gamma_l(\boldsymbol{b}_l-\boldsymbol{D}_l\boldsymbol{x})\right)\right) \\ &\det\left(\gamma_c\boldsymbol{D}_c^T\boldsymbol{D}_c + \gamma_l\boldsymbol{D}_l^T\boldsymbol{D}_l\right)^{1/2} F_c(\boldsymbol{b}_c)F_l(\boldsymbol{b}_l)d\boldsymbol{b}_c d\boldsymbol{b}_l \end{aligned} \tag{3.37}$$

And for the hyper-parameters Jeffrey priors are used *i.e.,*

$$p(\gamma) = \gamma^{-1} \tag{3.38}$$

The posterior distribution is therefore:

$$p(\boldsymbol{x}, \gamma_c, \gamma_l | y) \propto p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{x}|\gamma_l \gamma_c) p(\gamma_c) p(\gamma_l) \tag{3.39}$$

Substituting the terms:

$$p(\boldsymbol{x}, \gamma_c, \gamma_l | \boldsymbol{y}) \propto \gamma_c^{-1} \gamma_l^{-1} \det\left(\gamma_c \boldsymbol{D}_c^T \boldsymbol{D}_c + \gamma_l \boldsymbol{D}_l^T \boldsymbol{D}_l\right)^{1/2} \int_{\mathbb{R}^{m \times n^2}} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{b}_l, \boldsymbol{b}_c, \boldsymbol{S}, \gamma_l, \gamma_c) \det(\boldsymbol{S})^{1/2} Q'(\boldsymbol{S}) F_c(\boldsymbol{b}_c) F_l(\boldsymbol{b}_l) d\boldsymbol{S} d\boldsymbol{b}_c d\boldsymbol{b}_l \tag{3.40}$$

with:

$$p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{b}_l, \boldsymbol{b}_c, \boldsymbol{S}, \gamma_l, \gamma_c) \propto \exp\left(-\frac{1}{2}\left((\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \boldsymbol{S}^2 (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + (\boldsymbol{b}_c - \boldsymbol{D}_c\boldsymbol{x})^T \gamma_c (\boldsymbol{b}_c - \boldsymbol{D}_c\boldsymbol{x}) + (\boldsymbol{b}_l - \boldsymbol{D}_l\boldsymbol{x})^T \gamma_l (\boldsymbol{b}_l - \boldsymbol{D}_l\boldsymbol{x})\right)\right) \tag{3.41}$$

Now that a posterior distribution has been created, it is possible to sample from it using a Gibbs sampler.

3.3.2 Efficiently sampling

Since Gibbs sampler is a marginalization, the posterior distribution can be sampled easily using the conditional distributions. Each one of the sampling can be achieved in different ways as it is described in the following.

- $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{b}_l, \boldsymbol{b}_c, \boldsymbol{S}, \gamma_l, \gamma_c)$ detailed in 3.41 can be efficiently sampled using RJPO
- $p(\gamma_c|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}_l, \boldsymbol{b}_c, \boldsymbol{S}, \gamma_l)$ has three cases as it was presented in the first section of the chapter. If $\boldsymbol{D}_c$ and $\boldsymbol{D}_l$ are both diagonalizable in the same base, then:

  $$p(\gamma_c|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}_l, \boldsymbol{b}_c, \boldsymbol{S}, \gamma_l) \propto \gamma_c^{-1} \prod_{i=1}^{n} (\gamma_c |\lambda_{c;i}|^2 + \gamma_l |\lambda_{l;i}|^2)^{1/2} \exp\left(-\frac{1}{2}(\boldsymbol{b}_c - \boldsymbol{D}_c\boldsymbol{x})^T \gamma_c (\boldsymbol{b}_c - \boldsymbol{D}_c\boldsymbol{x})\right)$$

  which can be sampled using a Metropolis-Hastings algorithm.
- $p(\gamma_c|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}_l, \boldsymbol{b}_c, \boldsymbol{S}, \gamma_c)$ can be sampled in an analogous way
- $p(\boldsymbol{b}_c|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}_l, \boldsymbol{S}, \gamma_c, \gamma_l)$, as it was previously mentioned, each element $b_{c;i}$ from $\boldsymbol{b}_c$ is independent. Therefore, each $b_{c;i}$ has as distribution:

  $$p(b_{c;i}|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}_l, \boldsymbol{S}, \gamma_c, \gamma_l) \propto \exp\left(-\frac{1}{2}\gamma_c (b_{c;i} - \bar{x}_{c;i})^2\right) f_c(b_{c;i})$$

  where $\bar{x}_{c;i}$ is the i^{it} element of $\boldsymbol{D}_c\boldsymbol{x}$. The method to sample $b_{c;i}$ will depend on $f_c(\cdot)$, the worst possible scenario would be sampling using Metropolis-Hastings. However, the important thing is that, since each $\bar{x}_{c;i}$ is independent, the sampling can be done efficiently with no regard to the distribution's dimension.
- $p(\boldsymbol{b}_l|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}_c, \boldsymbol{S}, \gamma_c, \gamma_l)$, can be sampled in an analogous way

- $p(\boldsymbol{S}|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}_c, \boldsymbol{b}_l, \gamma_c, \gamma_l)$ in here, also each element s_i from $\boldsymbol{S}$ is independent. Therefore, each s_i has as distribution:

$$p(s_i|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}_c, \boldsymbol{b}_l, \gamma_c, \gamma_l) \propto \exp\left(-\frac{1}{2}s_i^2\bar{x}_{H;i}^2\right)q'(s_i)$$

  where $\bar{x}_{H;i}$ is the i^{it} element of $(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})$. Again in here, the method to sample s_i will depend on $q'(\cdot)$, the worst possible scenario would be sampling using Metropolis-Hastings. And also in this case, the important thing is because of the independence, the sampling can be done efficiently with no regard to the distribution's dimension.

- finally, $q'(\cdot)$, $f_c(\cdot)$ and $f_l(\cdot)$ may have hyper-parameters on their own that would need to be sampled from.

How to efficiently sample from the constructed distribution concludes this chapter. The previous example is a good illustration of the LMG and SMG as it allows one to have a broad overview of what was done in this chapter. The construction of LMG and SMG that were done are major contributions of this work as they increases the number of distributions that can be efficiently sampled in high-dimension problems.

In the next chapter is presented an application of LMG to construct a L2L1 potential and it is shown how the contributions in the work are important for its construction.

4 L2L1 Prior

The biggest problem of using Gaussian prior as regularization is that it is known for not being edge preserving. This can be a problem as most of the details in an image appear because of the edges. Therefore, the construction of a edge preserving potential is a well known research topic.

In this chapter are presented the results using L2L1 criteria, which have a quadratic behavior around the origin and a linear behavior at large values, allowing edge preserving [2]. In section 4.1 will be used the Huber Potential as an L2L1 potential. In the sections 4.2 and 4.3 will be used the potential developed in [2] that approximates a L2L1 potential. This potential is constructed using a LMG approach as it was described in section 3.1.

4.1 Posterior Maximum

4.1.1 Half-Quadratic Criterion

The criterion that was presented in section 2.1 was a quadratic criterion as it is based on two distinct quadratic parts

$$J(\boldsymbol{x}) = (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{D}^T \gamma_d \boldsymbol{D}\boldsymbol{x} \tag{4.1}$$

which can be rewritten as:

$$J(\boldsymbol{x}) = J_0(\boldsymbol{x}) + \sum_{i=1}^{n} \gamma_d \bar{x}_i^2 \tag{4.2}$$

where $\bar{x}_i^2$ is th i^{th} element of $\boldsymbol{D}\boldsymbol{x}$.

The L2L1 criterion that needs to be minimized is called a Half-Quadratic, as it is based on the same quadratic part $J_0(\boldsymbol{x})$ and in a non-quadratic one. The criterion can be written as:

$$J(\boldsymbol{x}) = J_0(\boldsymbol{x}) + \sum_{i=1}^{n} \phi(\bar{x}_i) \tag{4.3}$$

There are several choices of function $\phi(\cdot)$; the one that is chosen is the Huber potential:

$$\phi(\bar{x}_i) = \kappa \begin{cases} \bar{x}_i^2 & \text{if } \bar{x}_i \leq \tau; \\ 2\tau|\bar{x}_i| - \tau^2 & \text{if } \bar{x}_i \geq \tau. \end{cases} \tag{4.4}$$

Since the new criterion is not a quadratic one, it is not possible to use Conjugate Gradient in it. Therefore, another type of optimization tool is needed. The one chosen is the optimization of Geman Yang augmented criterion. Its implementation is explained in the next subsection.

4.1.2 Geman and Yang (GY) form of Augmented Criterion and LEGEND algorithm

The Geman and Yang augmented criterion is a way of transforming some half-quadratic criterion into two distinct parts. The LEGEND algorithm explores the augmented criterion in

an efficient way to minimize it. All the demonstration and condition for the criterion to exist and the LEGEND algorithm to work are detailed in [12]. All those demonstrations show that they can be applied to the Huber potential. The algorithm is detailed below:

Algorithm 7: LEGEND

while *number of iterations not achieved* **do**

1. optimize using Conjugate Gradient the criterion

$$J_Q(\boldsymbol{x}) = (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^T \gamma_n (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + (\boldsymbol{b} - \boldsymbol{D}\boldsymbol{x})^T \kappa (\boldsymbol{b} - \boldsymbol{D}\boldsymbol{x})$$

2. update the value of $\boldsymbol{b}$:

$$b_i = \bar{x}_i - 0.5 \frac{d\phi(x)}{dx}\bigg|_{x=\bar{x}_i}$$

The LEGEND algorithm makes it possible to test the parameters γ_n, κ and τ that minimize the norm of the error $|\boldsymbol{x} - \hat{\boldsymbol{x}}|$. However, this could only be done to the Laplacian regularization, for the Gradient regularization, the increased number of hyper-parameters (γ_n, γ_c, τ_c, γ_l and τ_l) would make it really slow to test enough different values to have a good precision on the parameters.

4.1.3 Laplacian Regularization

The criterion that needs to be minimize is exactly the one stated in the previous paragraph using as matrix $\boldsymbol{D}$ the Laplacian operator. It is therefore possible to test several combination of the hyper-parameters in order to find those that minimize the norm of the error. The results are presented bellow.

$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$	γ_n	κ	τ
2.0008×10^3	0.1	0.2154	1.3705

In Figure 4.1 it is possible to observe the restored image using those parameters. As it can be seen, the restored image has a better quality and the borders, specially around the cameraman's coat, are better defined. The general impression is that the details are more clear. This is not that much reflected in the norm of the error since the improvement in the quality is specially around the borders and those do not compose most of the image. In Figure 4.2 is shown the auxiliary variable $\boldsymbol{b}$ that is used as mean in the quadratic criterion. As it can be seen, the auxiliary variable detects the borders with an impressive precision. This happens because the auxiliary variable detects the places where the quadratic part is not able to regularize the most.

Figure 4.1: Restored image using the LEGEND Algorithm with Huber potential and Laplacian regularization

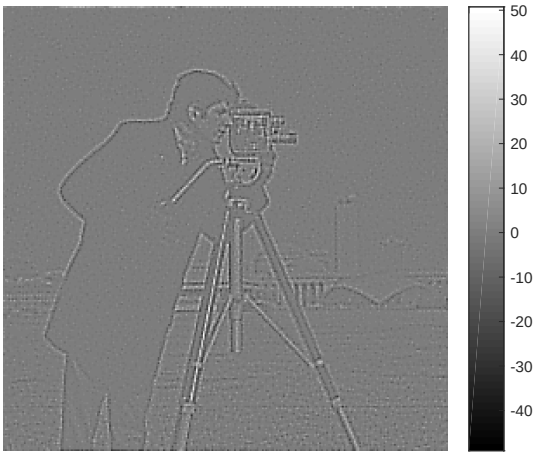


Figure 4.2: Auxiliary variable $\boldsymbol{b}$ using the LEGEND Algorithm with Huber potential and Laplacian regularization

In Figure 4.3 a specific line of the image was selected and compared with the original image, and the absolute difference between the original image and the estimation using the L2l1 norm and the absolute difference between the original image and the estimation using the L2L1 norm. As it can be seen, the estimation using the L2L1 norm is better as it can has less differences with the original image, specially in points of fast variation. This confirms what is known for the L2L1 potential which is that it has a better capacity to follow the edges.

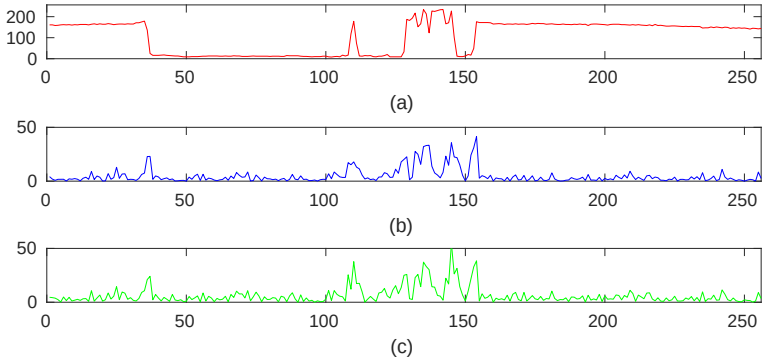


Figure 4.3: Comparison using the line 100
(a) Original image (b) $|\boldsymbol{x} - \hat{\boldsymbol{x}}_{l2l1}|$ (c) $|\boldsymbol{x} - \hat{\boldsymbol{x}}_{l2}|$

4.2 Supervised Posterior Expectation

4.2.1 Log-erf potential and distribution

In [2] is constructed a L2L1 potential similar to the Huber potential, called the log-erf potential. The potential is well studied in the original paper. As it will be seen, it can be sampled from

and the contributions made in 3.1 allow to use the potential when two regularization operator are used.
The log-erf potential is defined by:

$$\phi(\bar{x}_i) = -2\log\left(\chi(+\bar{x}_i) + \chi(-\bar{x}_i)\right) \tag{4.5}$$
$$\text{with: } \chi(\bar{x}_i) = \exp\left(\frac{\gamma_b \bar{x}_i}{2}\right)\text{erfc}\left((\frac{\gamma_b}{2\gamma_d} + \bar{x}_i)\sqrt{\gamma_d/2}\right)$$

where γ_b is a hyper-parameter of the distribution and $\bar{x}_i$ is the i^{th} element of $\boldsymbol{Dx}$. The associated distribution is:

$$p(\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\phi(\bar{x}_i)\right) \tag{4.6}$$

Which can be obtained through the following Gaussian mixture:

$$p(\boldsymbol{x}) \propto \int_{\mathbb{R}^n} \gamma_d^{n/2}\gamma_b^n \exp\left(-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{Dx})^T\gamma_d(\boldsymbol{b}-\boldsymbol{Dx})\right)F(\boldsymbol{b})d\boldsymbol{b} \tag{4.7}$$

with:

$$F(\boldsymbol{b}) = \prod_{i=1}^{n} f(b_i)$$
$$f(b_i) = \exp\left(-\frac{\gamma_b}{2}|b_i|\right)$$

Finally, it is also deduced a relation between the parameters of Huber potential and the log-erf, which is:

$$\tau = \frac{\gamma_b}{\phi''(0)}$$
$$\kappa = \frac{1}{2}\phi''(0)$$
$$\text{with: } \phi''(0) = \frac{\gamma_b^2}{2}((\eta\pi^{1/2}\,\text{erfcx}(\eta))^{-1} - 1) \text{ and } \eta = \frac{\gamma_b}{\sqrt{8\gamma_d}}$$

The relation between the parameters makes it possible to use the κ and τ calculated in the previous in a Posterior Expectation estimator.

γ_n	γ_d	γ_b
0.1	0.4740	0.7684

4.2.2 Sampling the mean

Once the construction of the distribution was explained, it is needed to sample $\boldsymbol{b}$. As it was presented in section 4.3, the sample from each b_i is independent. The posterior distribution is:

$$p(b_i|\boldsymbol{y},\boldsymbol{x},\gamma_n,\gamma_d,\gamma_b) \propto \exp\left(-\frac{1}{2}\left(\gamma_d(b_i-\bar{x}_i)^2 + \gamma_b|b_i|\right)\right) \tag{4.8}$$

where $\bar{x}_i$ is the i^{th} element of $\boldsymbol{Dx}$.

In the original paper it is presented a way of sampling $\boldsymbol{b}$ using the inversion of the cumulative distribution function. However, some numerical instability were encountered and some of the sampled b_i were equal to ∞. In order to solve this issue was implemented a step of verification

of the data to find if any sample had diverged. In the affirmative case, a Metropolis-Hastings algorithm was used to resample only the auxiliary variable b_i that had diverged. This small detail is very illustrative of the power that LMG and SMG has as only the sampled that had presented instability had to be resampled, not the all the auxiliary variable. For the Metropolis-Hastings used, the instrumental law used is a Gaussian distribution with standard deviation of 2 which allowed for acceptance rate of about 62%.

4.2.3 Application to Laplacian Regularization

With those results, it is possible to use the parameters obtained in section 4.1 and use them in the supervised PE estimator. In Figure 4.4 is shown the resultant estimation when using 2000 samples to calculate PE. The norm of the error obtained is 2.1105×10^3. As it can be seen, the result is really similar to the one obtained with the PM estimator. This again is important as it shows that PE can obtain similar results to PM. In Figure 4.5 it can be seen that the sampled auxiliary variable for the mean is really similar to the one that is determined with the PM estimator in the previous section. The edge detection is precise around the borders and show that created potential really has a similar behavior to the L2L1. Above all, this validates that the LMG can produce auxiliary variables that are similar enough to their counterparts in the Geman and Yang augmented criterion.

Figure 4.4: Restored image using the supervised Posterior Expectation estimator with log-erf priors and Laplacian regularization

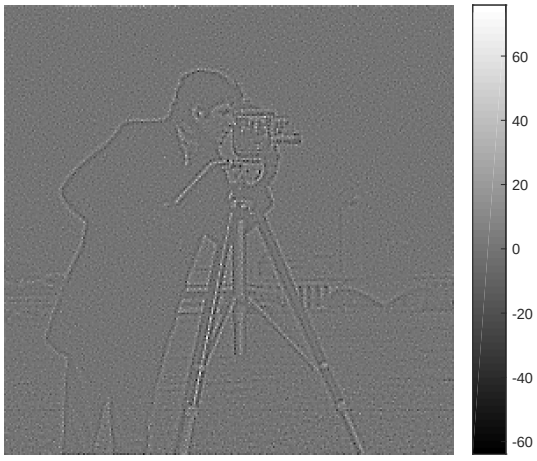


Figure 4.5: Auxiliary variable $\boldsymbol{b}$ when using the supervised Posterior Expectation estimator with log-erf priors and Laplacian regularization

However, as it can be seen in Figure 4.6, the image of the standard deviation looses some of its meaning because it is heavily influenced by the sampling of the auxiliary variables. This behavior was observed in all the cases of this work when using LMG and, as it will be seen ahead, the auxiliary variable brings more information than the standard deviation which is clearly biased. For this reason, in the case of LMG will only be shown the auxiliary variable even if, as it was said, in some cases as astronomy having the standard deviation is important.

In a last point, in the Figure 4.7 is compared the same row as in Figure 4.3 with the absolute difference between the original image and PM estimation using Huber potential and the absolute difference between the original image and supervised PE estimation. As it can be seen, both results are really similar which show that the log-erf potential not only gives as

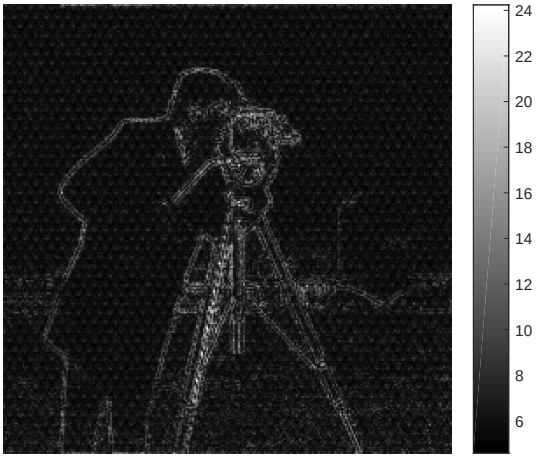


Figure 4.6: Standard deviation when using the supervised Posterior Expectation estimator with log-erf prior and Gradient regularization

good results as Huber potential, but also that the supervised PE estimation can be used to obtain good results.

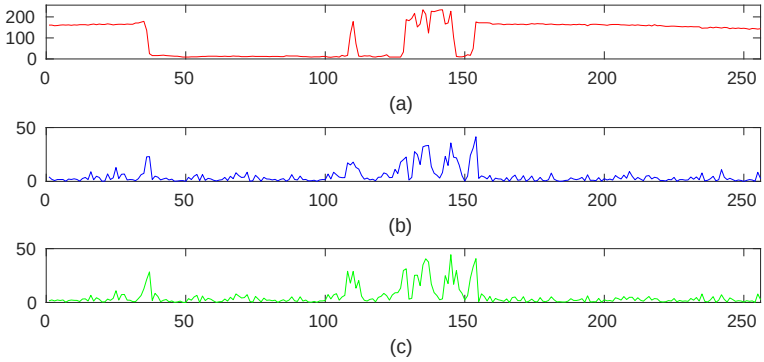


Figure 4.7: Comparison using the line 100
(a) Original image (b) $|\boldsymbol{x} - \hat{\boldsymbol{x}}_{PM}|$ (c) $|\boldsymbol{x} - \hat{\boldsymbol{x}}_{supervised\,PE}|$

4.3 Unsupervised Posterior Expectation

The unsupervised Posterior Expectation estimator needs to estimate all the hyper-parameters. Since this estimation is done in an automatic way, it should be possible to obtain good results both using the Laplacian and the Gradient Regularization. Most of the tools that are needed to the unsupervised PE estimator were described in the previous section. The only missing parameter is the one that is directly associated to the log-erf distribution, which is γ_b. Jeffreys prior for the Laplace distribution $h(\boldsymbol{b})$ is the same that for a Gaussian distribution, *i.e.*, $p(\gamma_b) = \gamma_b^{-1}$. The posterior distribution of γ_b is different in the Laplacian Regularization case

and in the Gradient Regularization case and will be detailed ahead.

4.3.1 Laplacian Regularization

The first question that needs to be addressed it therefore the sampling o γ_b. For the Laplacian regularization, the posterior distribution for γ_b is:

$$p(\gamma_b|\boldsymbol{y}, \boldsymbol{x}, \gamma_n, \gamma_d, \boldsymbol{b}) = \gamma_b^N \exp\left(-\frac{1}{2}\gamma_b \mathbf{N}_1(\boldsymbol{b})\right)$$

where $\mathbf{N}_1(\boldsymbol{b}) = \sum_{i=1}^{N} |b_i|$. This is a Gamma distribution and therefore can be easily sampled from.

Now it is possible to sample from the posterior distribution using a Gibbs sampler. A first PE estimation was made using 2000 samples. As it can be seen in Figure 4.8, γ_d do not converge to the value determined in section 4.1, and it would likely be needed many more samples to be sure that it is converging towards 8. However, the other parameters appear to converge to close values of the ones determined previously in the PM estimation in section 4.1. As it can be seen in Table 4.1 the mean values for γ_n and γ_b are close enough to those that were determined using PM. Furthermore, their standard deviation is small. As for γ_d the mean and the variance calculated do not mean a lot as clearly the hyper-parameter diverges. Finally ,he three hyper-parameter were also initialized with several different values and it was observed a similar behavior *i.e.,* the convergence of γ_n and γ_b and the divergence of γ_d.

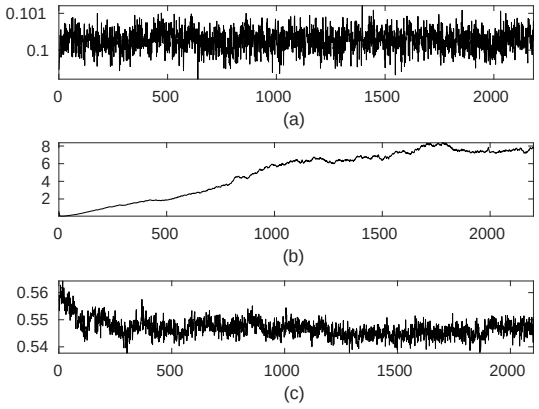


Figure 4.8: Hyper-parameters chain of values (a) γ_n (b) γ_d (c) γ_b

	PM	PE	
	value	mean	std
γ_n	0.1	0.1002	0.0007
γ_d	0.4740	4.9141	2.6186
γ_b	0.7684	0.5540	0.0657

Table 4.1: Comparison of the values of hyper-parameters determined by PM and PE in the Laplacian Regularization

The reason why γ_d does not stabilizes seems to be due to a property of the log-erf distribution in function of the variations of γ_d. In Figure 4.9 it can be seen that for γ_b constant and equal the value that the unsupervised approach determined($\gamma_b = 0.5$), the value of the potential changes very little from $\gamma_d = 0.4$(determined in the PM estimation) to $\gamma_d = 8$ (the maximum value attained by the unsupervised PE) for a same Δx_i. Because of this, the influence of γ_d is very little and in fact most of the determination of the log-erf value ends up by being done by γ_b. This does not explain however why γ_d rises instead of simply randomly varying. Finding this explanation goes beyond the scope of this work, and therefore was not studied.

Another point that reinforces the credibility of the hypothesis that γ_d does not have a lot of influence is the log-erf potential is observed in Figure 4.11. In there, it can be seen that even if γ_d diverged, the auxiliary variable $\boldsymbol{b}$ can still detect borders as well as it happens in the

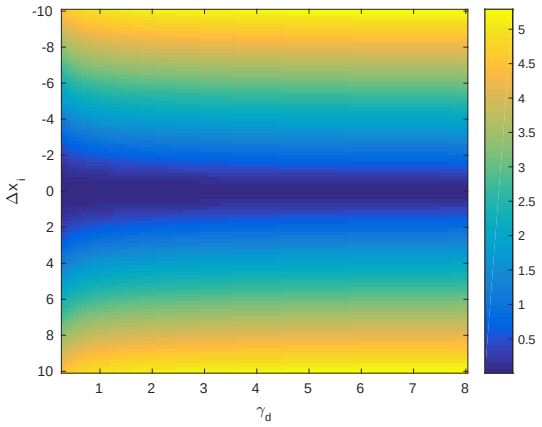


Figure 4.9: Values of log-erf potential for different values of γ_d and $\gamma_b = 0.5$

supervised case. It can be considered therefore with all this information that the variations of γ_d do not cause a major difference in the potential neither in the auxiliary variable and with the elements analyzed before, it is possible to not be give much importance to it.

However, even if the rise of γ_d seems to not influence a lot the result, it might seem more reliable to use the values of γ_d before they are to high. Therefore, the adopted strategy was to use less samples. In Figure 4.10 can be seen the resultant estimation using only 200 samples. The image is close to the one obtained using the PM estimator and the norm of the error was of 2.1355×10^3 which is close to the other results calculated using L2L1 potentials in section 4.1 and 4.2.

Figure 4.10: Restored image using the unsupervised Posterior Expectation estimator with log-erf priors and Laplacian regularization

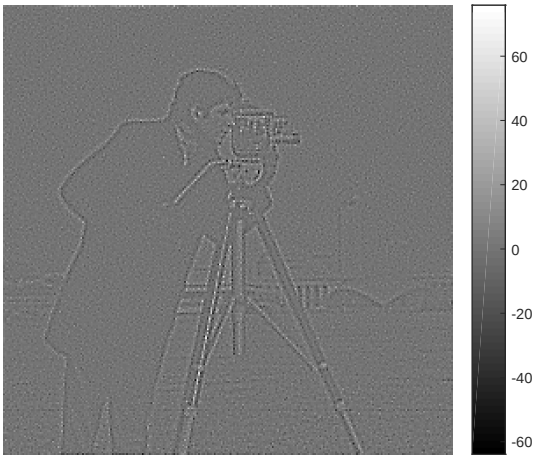


Figure 4.11: Auxiliary variable $\boldsymbol{b}$ when using the unsupervised Posterior Expectation estimator with log-erf priors and Laplacian regularization

In the Figure 4.12 is compared the same row as in Figure 4.3 with the absolute difference between the the original image and PM estimation using Huber potential and the absolute

difference between the original image and unsupervised PE estimation. As it can be seen, in here again both results are really similar. This means that the strategy of using fewer samples seams to be a good one as it enables for similar results even if the hyper-parameters do not converge to the value predicted in section 4.1.

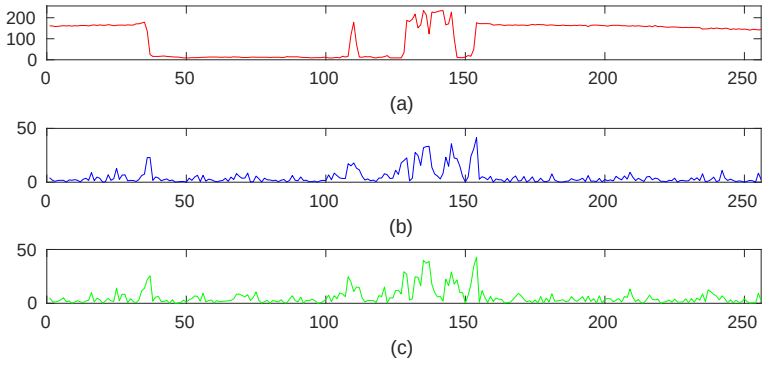


Figure 4.12: Comparison using the line 100
(a) Original image (b) $|\boldsymbol{x} - \hat{\boldsymbol{x}}_{PM}|$ (c) $|\boldsymbol{x} - \hat{\boldsymbol{x}}_{unsupervised\,PE}|$

4.3.2 Gradient Regularization

For the Gradient Regularization, there is the need to have two different hyper-parameters for the auxiliary distributions, one for the lines, γ_{bl} and one for the columns γ_{bc}. Their posterior distribution is:

$$p(\gamma_{bl}|\boldsymbol{y}, \boldsymbol{x}, \gamma_n, \gamma_l, \gamma_c, \boldsymbol{b}_l, \boldsymbol{b}_c, \gamma_{bc}) = \gamma_{bl}^N \exp\left(-\frac{1}{2}\gamma_{bl}\mathbf{N}_1(\boldsymbol{b}_l)\right)$$
$$p(\gamma_{bc}|\boldsymbol{y}, \boldsymbol{x}, \gamma_n, \gamma_l, \gamma_c, \boldsymbol{b}_l, \boldsymbol{b}_c, \gamma_{bl}) = \gamma_{bc}^N \exp\left(-\frac{1}{2}\gamma_{bc}\mathbf{N}_1(\boldsymbol{b}_c)\right)$$

both are Gamma distributions and therefore can be easily sampled.

With this distribution, it is possible to sample from the posterior distribution. Again in here a first estimation was done using 2000 samples. In Figure 4.13 it can be seen that the hyper-parameters seem to not converge and this behavior is observed for any set of initialization that is made. This result could have been predicted as the hyper-parameters are unstable in the case of the Laplacian regularization using the log-erf potential (section 4.3) and in the case of the Gradient regularization (section 2.5).

Furthermore, it can be observed in Figure 4.14 that the auxiliary variable for the row $\boldsymbol{b}_l$ was unable to detect any border and the obtained image seems like noise. This is the result of the parameter γ_l having decreased to values near 0. This caused the regularization term to not have any more information on the operator for the rows, making the auxiliary variable to become simple noise.

In order to try to obtain a result, the same idea of using only 200 samples that was used before is done. For the initialization, the following parameters were chosen: γ_l and γ_c were initialized with the values of γ_l and γ_c determined in section 2.2 for the quadratic case using

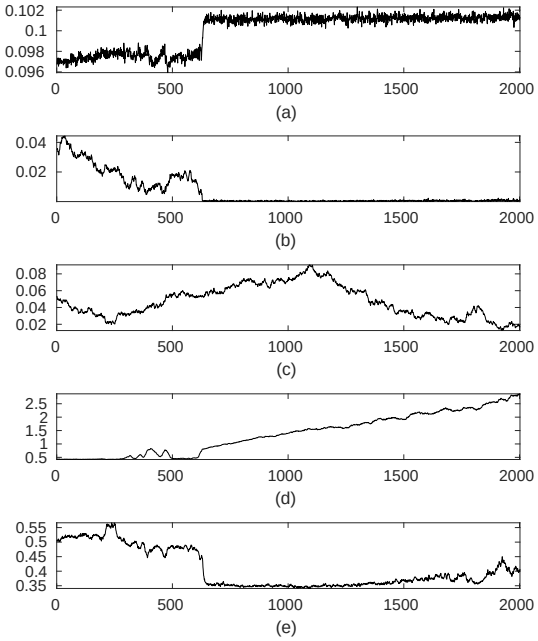
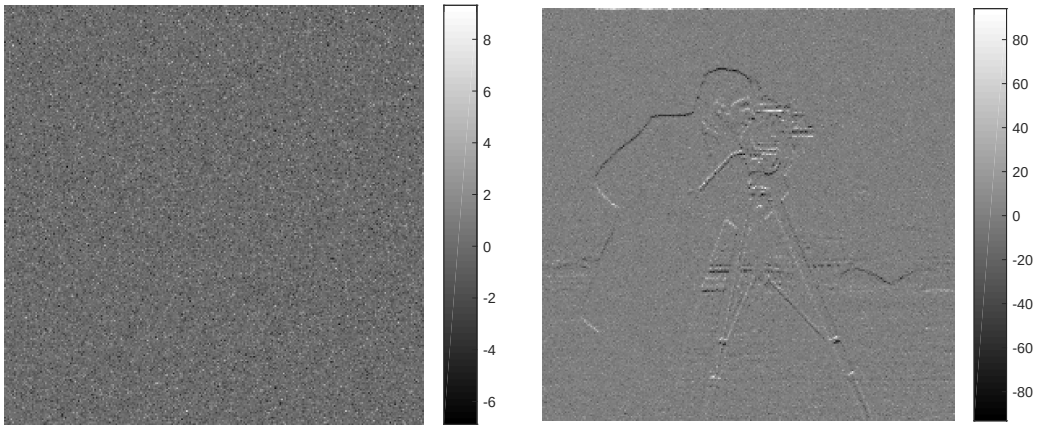


Figure 4.13: Hyper-parameters chain of values
(a) is γ_n (b) is γ_l (c) is γ_c (d) is γ_{bl} (e) is γ_{bc}



(a) $\boldsymbol{b}_l$ **(b)** $\boldsymbol{b}_c$

Figure 4.14: Auxiliary variables when using the unsupervised Posterior Expectation estimator with log-erf priors and Gradient regularization

the PM estimator; γ_{bl} and γ_{bc} were initialized with the values that the unsupervised approach determined for the Laplacian regularization case. The following results seem to be relatively sensible to the initialization of the hyper-parameters and those can not be the different otherwise the estimation does not converge.

In Figure 4.15 it can be seen that the hyper-parameters converged for some values. In Table 4.2 can be observed the mean value of the hyper-parameters as for their standard deviation. As it can be seen, the standard deviation is not very high. Clearly the estimation of the hyper-parameters was at least partially successful as it gave really good results.

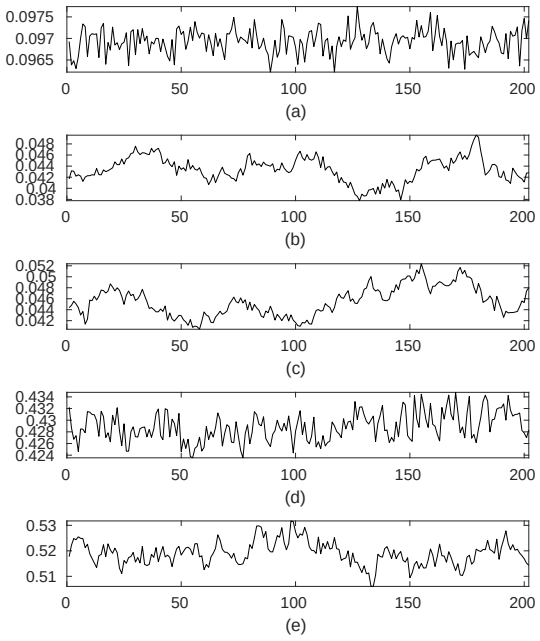


Figure 4.15: Hyper-parameters chain of values for 200 samples
(a) γ_n (b) γ_l (c) γ_c (d) γ_{bl} (e) γ_{bc}

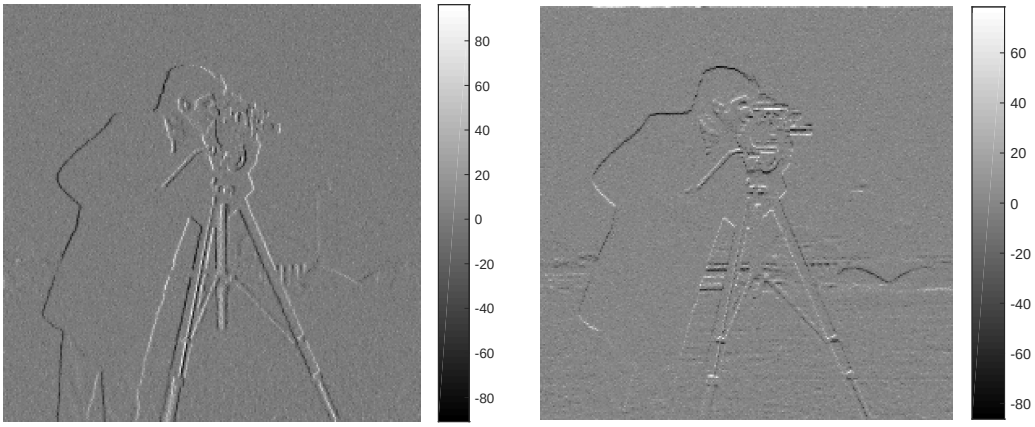
	PE	
	mean	std
γ_n	0.0966	0.0018
γ_l	0.0435	0.0023
γ_c	0.0453	0.0026
γ_{bl}	0.4351	0.0318
γ_{bc}	0.5263	0.0340

Table 4.2: Values of hyper-parameters determined by PE in the Gradient Regularization

In Figure 4.16 it can be seen that the auxiliary variable were able to detect with a high efficiency the borders of the image. They also have similar aspect with those that were presented when using the Laplacian regularization. In addition to it, this result is even more interesting because it estimates separately the vertical edges and the horizontal edges. This confirms the importance of having a mixture model that can accept different regularization operators as it can also be used to extract information from the image in an easy way.

As it can be seen in Figure 4.17 the resultant estimation is a really good one. The norm of the error is 2.0172×10^3 which is one of the lowest norm of the error obtained from all previous estimation. Furthermore, the result is aesthetically pleasant. The borders are well defined in the general aspect of a clear and neat result.

Those results concludes the analyses on the unsupervised case of using the L2L1. In the next section those results will be analyzed.



(a) $\boldsymbol{b}_l$ **(b)** $\boldsymbol{b}_c$

Figure 4.16: Auxiliary variables when using the unsupervised Posterior Expectation estimator with log-erf priors and Gradient regularization

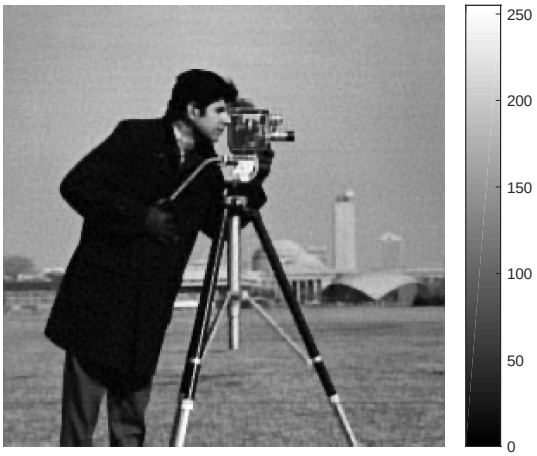


Figure 4.17: Restored image using the unsupervised Posterior Expectation estimator with log-erf priors and Gradient regularization

4.4 Results' Analyses

The conclusions presented in here, even if they are illustrated with the results obtained in the chapter, are quite general and can also be observed with other images as for different amount of noise.

The most important conclusion is the confirmation that using unsupervised PE estimator with LMG distributions enables for excellent results that are comparable to those obtained using the PM estimator. Furthermore, the edges were effectively better preserved using the L2L1 potential than when using the L2 potential and even if this is not quantified in the norm of the error (as the difference is specially visible only in the edges), the details in the image are more clear. This can be specially observed when comparing a single row and observing that when using the norm L2L1 the difference of the image to the estimation is smaller, specially around the edges. This is a known result in image processing literature, and therefore it was important to verify it for the inverse problem treated in this work. In Table 4.3 are compared the values of the norm of the error obtained in the different methods of this chapter.

Table 4.3: Comparison of norm of the error obtained using the different methods and different regularization operators

$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$	PM	PE	
	supervised	supervised	unsupervised
Laplacian	2.0008×10^3	2.1105×10^3	2.1355×10^3
Gradient	$--$	$--$	2.0172×10^3

The second important conclusion is that using the auxiliary variable brings important information about the estimation. It not only enables to detect the borders but it can also be used to check the quality of the estimator. An auxiliary variable that did not detect any edge means that there was a problem during the estimation and that the results must be treated with care because they can be wrong. This is another advantage of LMG over the normal Gaussian as it provides another tool that can be used in order to analyze the results.

The final and more important conclusion is that the use of the Gradient regularization effectively enables better results. Even if, as in the l2 potential case, this is not directly visible through the norm or the error, it is clearly visible in the estimation itself. The image seems clearer as it was better focused. Furthermore, the small crosses visible through the image do not appear in the Gradient regularization, which makes the quality of the image really better. In addition to that, having two separates auxiliary variables allow one to extract more information from the image that can be used for other purposes. This confirms the importance of the contribution that was made in section 4.1 that enables the use of LMG for more than only one regularization operator.

5 Conclusion and Further Development

5.1 Conclusion

The most important contribution of this work is the merge of the Rejection Perturbation Optimization algorithm and the Mixtures of Gaussian in order to propose a broad class of high-dimensional distributions that can be efficiently sampled and used in unsupervised inverse problems. This contribution expands, beyond the Gaussian, the distributions that high-dimensional inverse problem can efficiently use.

Applied to an image restoration problem, it was shown that being able to use other distributions effectively brings better results to the estimation as those can be more adequate to the problem that is treated. The restored images had a better quality over their counterparts when using Gaussian priors. Furthermore, it is also proposed that the auxiliary variables that are used in the construction of the Mixture of Gaussian can bring important information to the problem as those are revealing of the main structures that the constructed distribution aims to detect.

A final contribution is that, in image restoration, being able to construct priors that are separable for the rows and column enables better results than when using a single regularization operator. This is rather interesting as it motivates for the construction of distributions that could exploit both operators.

5.2 Further development

This section details some points of development that appeared to be interesting the work, but which considered during the work but were developed

5.2.1 Why hyper-parameters do not converge

Even if some elements of answer were given, those are much more empirical results than mathematical proves. As it seems that those are related to the built distribution, it is important to find why those distributions make that the hyper-parameters do not converge. Knowing this could orient the choice of priors that are used in order to find those that guarantee that hyper-parameters will not diverge.

5.2.2 Which distributions can be constructed using LMG and SMG

The framework proposed states that it is possible to efficiently sample from LMG and SMG. However, it was not find any list or compilation of all functions that can be constructed using those mixture. It would be an interesting thing to have this compilation or at least a way to easily discover if a distribution can be constructed from a LMG (as the SMG case is already treated).

5.2.3 Location and Scale Mixture of Gaussian

In [13] it is proposed the creation of distributions that are created from Location and Scale mixtures simultaneously. It is also shown that those distributions give some good results in some case. It is likely that those distributions can also be constructing using the same framework that was used for SMG and LMG and therefore efficiently sampled and used in inverse problems.

5.2.4 Scale Mixture of Gaussian with more distributions

As it was shown, using more than only one regularization operator should bring better results. However, in the framework developed to SMG, only one operator can be used. This is because the partition function of the Gaussian part would be of the form (with only two operators):

$$C_G^{-1} \propto \det\left(\boldsymbol{D}_1^T \boldsymbol{S}_1^2 \boldsymbol{D}_1 + \boldsymbol{D}_2^T \boldsymbol{S}_2^2 \boldsymbol{D}_2\right)^{1/2}$$

The problem is that, even if $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ were diagonalizable in the same base, $\boldsymbol{S}_1^2$ and $\boldsymbol{S}_2^2$ would not be and therefore the determinant could not be written as a product of separable $s_{1;j} + s2; j$.

However, it was considered during - the work but not further developed - that if $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are similar to each other in a permutation way (as it is the case for the Gradient operator used in this work) then it should be possible to write the mixture as a product of separable $s_{1;j} + s2; p(j)$ where $p(j)$ is a permutation of the line j.

Bibliography

[1] C. Gilavert, S. Moussaoui, and J. Idier. “Efficient Gaussian sampling for solving large-scale inverse problems using MCMC”. In: *IEEE Transactions on Signal Processing* 63.1 (2015), pp. 70–80.

[2] J.-F. Giovannelli. “Unsupervised Bayesian convex deconvolution based on a field with an explicit partition function”. In: *IEEE Transactions on Image Processing* 17.1 (2008), pp. 16–26.

[3] T. Rodet. *Lecture notes of the class "Inverse Problems and applications to electromagnetism"*. 2016.

[4] F. Orieux, O. Féron, and J.-F. Giovannelli. “Sampling high-dimensional Gaussian distributions for general linear inverse problems”. In: *IEEE Signal Processing Letters* 19.5 (2012), pp. 251–254.

[5] C. Fox and U. of Otago. Electronics Group. *A Conjugate Direction Sampler for Normal Distributions, with a Few Computed Examples*. Electronics technical report. Electronics Group, University of Otago, 2008.

[6] F. Orieux et al. “Bayesian estimation for optimized structured illumination microscopy”. In: *IEEE Transactions on image processing* 21.2 (2012), pp. 601–614.

[7] J. R. Shewchuk. *An introduction to the conjugate gradient method without the agonizing pain*. 1994.

[8] J. Niemi. *Metropolis-Hastings algorithm*. Youtube. 2013. URL: `https://www.youtube.com/watch?v=VGRVRjr0vyw`.

[9] A. W. Marshall and I. Olkin. “A multivariate exponential distribution”. In: *Journal of the American Statistical Association* 62.317 (1967), pp. 30–44.

[10] D. F. Andrews and C. L. Mallows. “Scale mixtures of normal distributions”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1974), pp. 99–102.

[11] M. West. “On scale mixtures of normal distributions”. In: *Biometrika* 74.3 (1987), pp. 646–648.

[12] J. Idier. “Convex half-quadratic criteria and interacting auxiliary variables for image restoration”. In: *IEEE Transactions on Image Processing* 10.7 (2001), pp. 1001–1009.

[13] D. Wraith and F. Forbes. “Location and scale mixtures of Gaussians with flexible tail behaviour: Properties, inference and application to multivariate clustering”. In: *Computational Statistics & Data Analysis* 90 (2015), pp. 61–73.