

Universidad de Granada

Departamento de Ciencias de la Computación
e Inteligencia Artificial



Imágenes milimétricas pasivas: procesamiento y mejora para la detección de amenazas

Tesis Doctoral

Juan Gabriel Serra Pérez

Directores

Rafael Molina Soriano

Aggelos K. Katsaggelos

Programa de Doctorado en
Tecnologías de la Información y la Comunicación

Granada, April 2019

Editor: Universidad de Granada. Tesis Doctorales
Autor: Juan Gabriel Serra Pérez
ISBN: 978-84-1306-245-7
URI: <http://hdl.handle.net/10481/56431>

Agradecimientos

A todo el que ha contribuido a la realización de esta tesis.

Índice general

I. Introducción	9
1. Motivación	9
2. Formación de las imágenes milimétricas	10
3. Detección de amenazas en PMMWI	12
4. Degradaciones en imágenes milimétricas	14
5. Objetivos y estructura de la tesis	15
II. <i>Denoising</i> e <i>Inpainting</i> mediante aprendizaje de diccionarios	19
1. Introducción	20
2. Revisión bibliográfica	23
3. K-SVD	26
4. BKSVD	28
4.1. Modelado bayesiano	28
4.2. Inferencia variacional	30
4.2.1. Estimación de los vectores <i>sparse</i> y sus hiperparámetros	32
4.2.2. Estimación del diccionario	34
4.2.3. Estimación del parámetro de ruido	35
4.2.4. Algoritmo variacional completo	35
4.3. Inferencia Rápida <i>Sparse</i>	36
4.3.1. Inferencia subóptima	41
4.4. Influencia del tamaño del diccionario	42
4.5. Experimentos	43
4.5.1. <i>Denoising</i>	44
4.5.2. <i>Inpainting</i>	47
4.5.3. Comparación de tiempos de ejecución	49
5. Aprendizaje de diccionarios doblemente <i>sparse</i>	51
5.1. Modelado bayesiano	51
5.2. Inferencia variacional bayesiana	53
5.3. Experimentos	55
6. Representaciones <i>sparse</i> con <i>a posteriori</i> laplaciana	57
6.1. Modelado bayesiano e inferencia variacional EM	58

ÍNDICE GENERAL

6.2.	Algoritmo EM con <i>a posteriori</i> laplaciana	61
6.3.	Funcionamiento experimental del algoritmo	61
III	Deconvolución ciega de imágenes	65
1.	Introducción	65
2.	Revisión bibliográfica	67
3.	Modelado bayesiano	71
4.	Inferencia variacional EM	72
4.1.	Paso E	74
4.2.	Paso M	77
4.3.	Algoritmo BID	78
5.	Experimentos	79
5.1.	Comportamiento de la distribución <i>Spike-and-Slab</i> . .	80
5.2.	Análisis de resultados	84
IV	Aplicación a imágenes milimétricas pasivas	95
1.	Base de datos	95
2.	Preprocesamiento de las imágenes	96
3.	Detección basada en extracción de rasgos	99
3.1.	Filtros Haar	102
3.2.	<i>Local Binary Patterns</i>	102
3.3.	Modelo de detección	103
4.	Detección basada en red neuronal convolucional	104
5.	Resultados de detección	106
V	Conclusiones y trabajo futuro	109
1.	Conclusiones	109
1.1.	<i>Denoising</i> e <i>Inpainting</i> mediante aprendizaje de diccionarios	110
1.2.	Deconvolución ciega de imágenes	111
1.3.	Aplicación a imágenes milimétricas pasivas	112
2.	Trabajo futuro	112
	Bibliografía	115

Resumen

La detección y localización de amenazas resulta una tarea de vital importancia en la seguridad de instalaciones públicas o privadas tales como, por ejemplo, aeropuertos, instalaciones gubernamentales, estadios o almacenes de grandes empresas de distribución. Las ondas milimétricas son emitidas de forma natural por todos los cuerpos debido a su temperatura; esta radiación es capaz de atravesar tejidos como la ropa mostrando objetos ocultos bajo esta de distintos materiales: metales, líquidos, plásticos, etc. Las cámaras milimétricas pasivas son dispositivos que captan estas ondas formando una imagen sin la emisión artificial de ningún tipo de radiación, por lo que resultan inocuas. No obstante, la calidad de las imágenes generadas es típicamente muy limitada debido a su baja resolución, escasa relación señal a ruido y otras posibles degradaciones inherentes a los sistemas de captación.

Esta tesis doctoral está dedicada al procesamiento y mejora de las imágenes captadas con el objetivo final de hacer más precisos los sistemas de detección. Se proponen soluciones a los problemas de *inpainting*, eliminación de ruido y deconvolución ciega desde una misma perspectiva probabilística en un marco bayesiano haciendo uso de la inferencia variacional. La aproximación utilizada permite la estimación automática de los parámetros del modelo conjuntamente con la de las propias imágenes subyacentes. Los dos primeros problemas se resuelven mediante el aprendizaje basado en diccionarios, desarrollando un nuevo método para la estimación de los diccionarios y los vectores de coordenadas que representan a las señales en estas «bases». La deconvolución ciega elimina el emborronamiento desconocido presente en las imágenes; se propone un método de estimación de este emborronamiento para la estimación de la imagen nítida latente. Ambos trabajos presentan un denominador común, las distribuciones *a priori sparse* que se utiliza para modelizar nuestro conocimiento sobre la señal subyacente. La presente tesis aborda la inferencia variacional con el uso de estas distribuciones, garantizando soluciones analíticas que aproximan las distribuciones *a posteriori* de las imágenes latentes, lo que constituye una novedosa contribución a la literatura.

Finalmente, se aplican los métodos de mejora de imágenes

ÍNDICE GENERAL

desarrollados sobre una base de datos de imágenes propia y se estudia el efecto sobre dos métodos de detección con filosofías totalmente distintas: un primer método clásico basado en extracción de características y detección por bloque; y un segundo basado en una red neuronal convolucional con estructura de *autoencoder*. En ambos casos se observarán mejoras significativas en la detección.

Abstract

Threat detection and localization is a vital task in security and surveillance applications in private, public or government facilities such as airports, stadiums, warehouses of large logistics companies or embassies. All bodies at a temperature higher than absolute zero naturally emit millimeter wave radiation which is capable of travelling through layers of clothing revealing hidden objects of different materials: metals, liquids, pastics, etc. Passive millimeter wave cameras/scanners receive the radiation in this band of the spectrum generating an image without the need of artificial emission of waves, thus not posing a health hazard. However, passive millimeter wave images display very limited quality due to their low signal-to-noise ratio and resolution, as well as other degradations caused by the acquisition systems.

This doctoral thesis is devoted to the processing and enhancement of the acquired images with the ultimate goal of improving the accuracy of the detection systems. We propose comprehensive solutions to the inpainting, denoising and blind deconvolution problems from a probabilistic perspective within a Bayesian framework using variational inference. This approach allows for the automatic estimation of all model parameters along with the latent images simultaneously. The first two problems are solved with dictionary learning: we develop an efficient technique for the estimation of the dictionaries and sparse coordinate vectors which represent the observed signals in these “bases”. Blind deconvolution supresses the unknown blur in images; a method for the estimation of this blur is proposed to estimate the underlying sharp images. Both works share a common denominator: the use of sparse priors to model the a priori knowledge on the latent signals. This thesis tackles variational inference with the use of these priors, ensuring tractable solutions that closely approximate the posterior distribution of the latent images, constituting a novel contribution to the literature.

Finally, we apply the developed image enhancement techniques on an image database and study the impact on two detection methods with radically different philosophies: a classical approach based on feature extraction and per-block classification and a second one which uses an autoencoder-like convolutional neural network. In both cases substantial

ÍNDICE GENERAL

improvement in detection performance will be observed.

Conclusions

This doctoral thesis has tackled the threat detection problem in passive millimeter wave images focusing on the preprocessing of the acquired images due to its limited quality. Concretely, we have dealt the inpainting, denoising and blind deconvolution problems, utilizing Bayesian modelling and obtaining robust solutions thanks to variational inference. Both inpainting and denoising have been approached developing a Bayesian dictionary learning method. Additionally, a second dictionary learning technique where dictionaries are also sparse has been presented. Secondly, we have proposed a blind image deconvolution method focusing on blur estimation using the Spike-and-Slab sparse prior together with variational EM inference, which constitutes a novel contribution in the field. Last, we applied the developed processing techniques to an extensive database of passive millimeter wave images in order to assess the impact on detection performance. For this purpose, two different detection philosophies have been considered: a classical approach using feature extraction and per-block detection, and a second one using deep learning. We can draw further specific conclusions on the three main themes in which the thesis has been divided.

Dictionary learning: Denoising and Inpainting

- We first presented the BKSVD algorithm to solve the ℓ_1 dictionary learning problem. The prior imposed on the coefficient vectors enforces sparsity while rendering Bayesian inference tractable. We use a Bayesian framework for inference that allowed us to take into account the uncertainties of the model, improving the solutions. All model parameters were estimated automatically within the model.
- The model was applied to image inpainting and denoising obtaining state-of-the-art results. The popular K-SVD has two major drawbacks: it requires the knowledge of the true noise level and the number of non-zeros in the sparse signals. The proposed method solves both problem, although at the expense of a higher computational cost. BKSVD and BFPA are the best-performing dictionary learning methods, but the

latter introduces more visible artifacts in the images.

- inpainting has been tackled in two different manners: learning a dictionary from a set of clean images and from the corrupt image itself.
- BKSVD dictionaries have a better mutual coherence, resulting in higher quality reconstructions with more optimal sparse solutions.
- dictionary size has an impact on the quality of the sparse representations. The estimation of this parameter is not embedded in the inference procedure.
- A double sparsity dictionary learning model has also been presented using an analogous hierarchical modelling for the dictionary representation vectors. The main advantage over deterministic doubly sparse techniques lies in the automatic estimation of the noise and the inclusion of the uncertainty of the estimates to refine the solutions. This technique produces competitive PSNR and SSIM values for the image denoising problem. Additionally, a general formulation was used which is also valid for inpainting and compressive sensing.
- Finally, we developed a different approach to the sparse representation problem, using Laplace posteriors and showed its tractability.

Blind Image Deconvolution

- We presented a new blind image deconvolution method in the filter space. The main novelty of the method is the introduction of the spike-and-slab prior on the high-pass filtered images generated from the observed blurry image.
- This prior shrinks irrelevant information, discarding pixels which can compromise blur estimation.
- A simple, yet powerful, reparametrization of the prior allows for the use of variational inference. Additionally, a sensible factorization of the approximation of the image posterior better suits the multimodal nature of the true posterior.
- We developed an efficient EM algorithm for the estimation of the PSF, which automatically estimates all model parameters.
- We provided a deep study of the spike-and-slab prior behavior which casts light on the pixel selection capabilities of the blur estimation method.

- Experimental results show the model competitiveness, producing more precise blur estimations which lead to a higher mean PSNR score with lower variance. This indicates the model robustness.
- The proposed BID method is also robust against noise in the images.

Application to PMWWI

- Two different methodologies to approach the threat detection problem have been used. First, a classical feature extraction and per-block classification scheme has been explored. Haar and LBP features were considered and an ensemble of Random Forest was used as classifier. Secondly, a deep learning approach has been used for the binary segmentation of the image.
- The presence of missing data in the acquired signals made image inpainting necessary, for which the BKSVD method in chapter II was used. This is a crucial step, especially for the feature extraction approach.
- The improvements due to BID highly depends on the previous denoising of the images. However, due to the presence of high-intensity non-stationary noise in the image, which is not completely removed by the denoising method, a number of artifacts appear in the deconvolved images, limiting the improvement in detection.
- The experiments show a clear improvement in detection rate after denoising with BKSVD, but it is still important to correctly select the signal and dictionary size. However, the high level of noise in the images limit the gains of this preprocessing technique when using the per-block classifier.
- The convolutional neural network approach produces remarkably better detection results, which are suitable for practical implementation of the system. The classical methodology needs a deeper study of feature extraction strategies to obtain competitive results.

Capítulo I

Introducción

Las ondas milimétricas son capaces de atravesar los tejidos, permitiendo la localización de objetos ocultos bajo la ropa. Los sistemas de detección basados en tecnología milimétrica resultan de gran utilidad en aplicaciones de seguridad y han sido objeto de investigación reciente. Este capítulo se centra en el estudio de las imágenes milimétricas pasivas, la naturaleza de su formación y las degradaciones presentes en ellas. A la vista de estas degradaciones, principalmente, datos perdidos en la adquisición, alto nivel de ruido y desenfoque de las imágenes y, previo estudio de los trabajos de la literatura sobre detección de amenazas en este tipo de imágenes, se propone el procesamiento de las imágenes con el fin de mejorar los resultados de detección.

1. Motivación

La radiación milimétrica, es decir, la debida a las ondas en el rango de frecuencias 30-300 GHz (1-10 mm de longitud de onda), se ha convertido en una prometedora herramienta para sistemas de seguridad en aeropuertos y otras instalaciones con trasiego continuo de usuarios como estaciones de tren, metro o instalaciones gubernamentales [1], motivando una ingente cantidad de trabajos de investigación e ingeniería.

La utilidad de estas ondas para la detección de amenazas radica en su capacidad de atravesar tejidos como la ropa, revelando así diversos objetos potencialmente peligrosos que pudieran haber sido ocultados bajo esta. La radiación milimétrica se encuentra de forma natural en el ambiente y es debida a la temperatura de los cuerpos. Las cámaras milimétricas que captan esta radiación natural procedente de los objetos, ya sea emitida por ellos, o simplemente reflejada, reciben el calificativo de «pasivas». En contraposición, los sistemas de captación «activos» emiten pulsos de radiación milimétrica que serán reflejados por los objetos dando lugar a imágenes de mayor relación señal a ruido, llegando a suponer una invasión a la privacidad de los sujetos.

La radiación milimétrica no es ionizante, por lo que no supone un riesgo para la salud de los sujetos escaneados en ninguno de los casos. Podemos encontrar también sistemas de detección basados en rayos X. Al contrario que los equipos médicos basados en esta tecnología, la interacción con la piel es mínima, con niveles de radiación por debajo de los umbrales establecidos por las organizaciones de salud pertinentes. No obstante, al igual que los sistemas activos de ondas milimétricas, estos escáneres son controvertidos por su invasión de privacidad. Véase el interesante estudio sobre privacidad y efectos en la salud de ambos sistemas en [2].

2. Formación de las imágenes milimétricas

La radiación milimétrica procedente de los cuerpos transporta información intrínseca de los objetos que la emiten, posibilitando su identificación mediante el uso de sensores adecuados. Según la teoría del cuerpo negro, todo cuerpo con un temperatura superior a 0K emite una radiación característica natural que transporta información sobre sí, siendo más intensa cuanto mayor sea la temperatura de este. Esta relación viene dada por la fórmula de radiación de Planck [3], según la cual, la densidad de energía de radiación por unidad de volumen y frecuencia de un cuerpo negro/cavidad es

$$S_\nu = \frac{8\pi\nu^2}{c^3} \frac{h\nu}{e^{h\nu/kT} - 1}, \quad (\text{I.1})$$

donde ν es la frecuencia de la radiación, T la temperatura absoluta del objeto y h y k son las constantes de Planck y Boltzmann respectivamente, ver fig. 1. Vemos que la densidad de radiación depende de dos factores: el número de modos resonantes presentes en la cavidad, que es directamente proporcional al cuadrado de la frecuencia, y un segundo factor que expresa la energía media por modo. En el espectro de las ondas milimétricas, donde $h\nu \ll kT$, podemos utilizar la ley de Rayleigh-Jeans, según la cual

$$S_\nu \approx \frac{8\pi\nu^2}{c^3} kT, \quad (\text{I.2})$$

en donde se asume una energía media por modo no dependiente de la frecuencia. La radiación total captada por una cámara resultará de la integración en el ancho de banda de funcionamiento $\Delta\nu$ de la densidad de radiación y, dado que los objetos reales no son cuerpos negros ideales, esta energía vendrá multiplicada por un factor de emisividad ϵ

$$E(\nu_0, T) = \epsilon \int_{\nu_0 - \frac{\Delta\nu}{2}}^{\nu_0 + \frac{\Delta\nu}{2}} S_\nu = \epsilon \frac{8\pi\nu_0^2 \Delta\nu}{c^3} kT, \quad (\text{I.3})$$

siendo proporcional a la temperatura del cuerpo, lo que nos permitirá representar la radiación total procedente de un cuerpo a través de una temperatura efectiva.

2. FORMACIÓN DE LAS IMÁGENES MILIMÉTRICAS

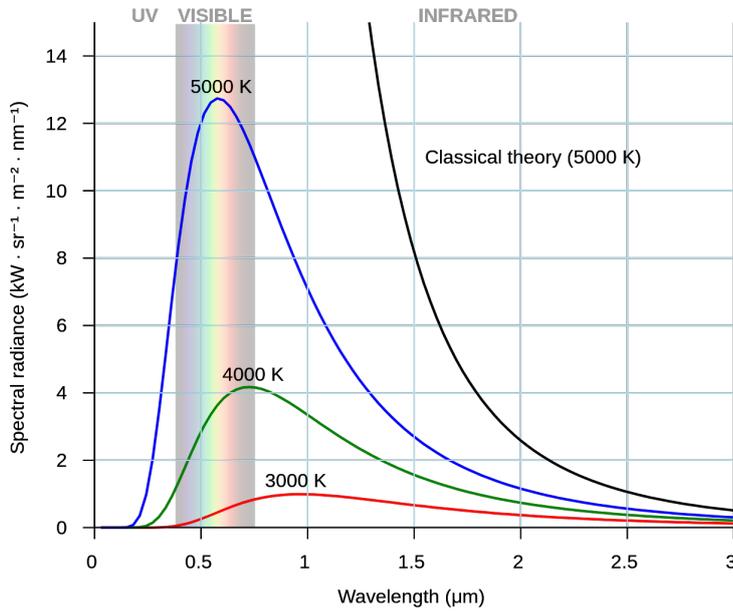


Figura 1: Radiación del cuerpo negro.

Aparte de la radiación emitida por el propio cuerpo que será recogida por la cámara milimétrica, hay que considerar otras contribuciones. Por un lado, todo cuerpo reflejará una fracción de la radiación procedente de objetos situados frente a él y, por otro, parte de la radiación milimétrica originada (o reflejada) por objetos situados detrás atravesará a este incidiendo también en los sensores, ver fig. 2. Podemos, consecuentemente, representar la radiación total procedente de un cuerpo teniendo en cuenta estas tres contribuciones

$$T_{ef} = \epsilon T_o + \rho T_f + \tau T_t, \quad (I.4)$$

donde T_o , T_f y T_t denotan las temperaturas del objeto y los cuerpos frente y tras él respectivamente; ρ y τ la reflectividad y transmitancia del cuerpo.

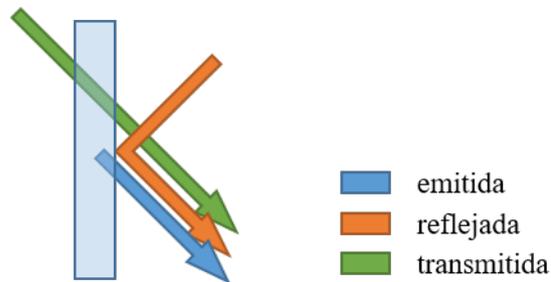


Figura 2: Contribuciones de radiación recibidas por un sensor en una dirección del espacio.

Gracias al concepto de temperatura efectiva de radiación podemos analizar una situación más compleja, como por ejemplo, la presencia de un objeto oculto bajo la ropa, ver fig. 3. Un estudio de las diferentes contribuciones

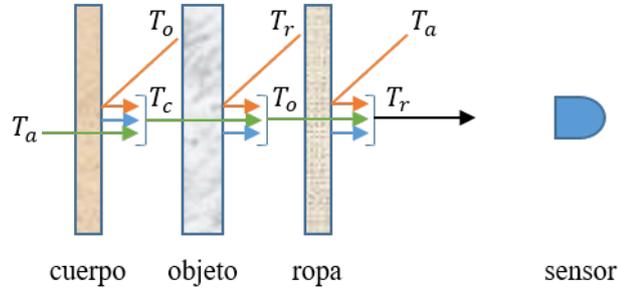


Figura 3: Contribuciones de la radiación recibida por un escáner de ondas milimétricas en presencia de un objeto oculto. T_a denota la temperatura ambiente.

con objetos de distintos materiales [4] apunta, entre otras conclusiones, que para que el objeto sea distinguible del cuerpo la temperatura ambiente ha de ser preferentemente baja (fría). Además, la diferencia de temperatura de radiación efectiva entre cuerpo y objetos metálicos es la más pronunciada (en comparación con plásticos y materiales cerámicos), haciendo que estos sean más fáciles de detectar.

3. Detección de amenazas en PMMWI

Las condiciones no óptimas de captación de las señales, junto a la baja relación señal a ruido inherente a los sistemas pasivos provocan una baja calidad en las imágenes formadas, complicando sobremanera las tareas de detección de amenazas. Además de estos problemas, las imágenes milimétricas pasivas presentan una baja resolución, alta variabilidad espacial de intensidad de señal. La diversidad de las potenciales amenazas en cuanto a tamaño, forma, posición y propiedades de transmisión (ϵ , ρ y τ) complican aún más el problema. En consecuencia, los sistemas de detección presentan una alta tasa de errores, ya sea por incapacidad de detectar ciertas amenazas, verdaderos positivos, o por identificar como tales a regiones de la imagen que realmente no lo son, falsos positivos, ocasionando que dichos sistemas presenten una utilidad reducida. Idealmente, los sistemas de detección de amenazas han de ser capaces de detectar todas las amenazas presentes en las imágenes, presentando, a su vez, un muy bajo, o nulo, número de falsos positivos por imagen y, además, han de funcionar en tiempo real.

En la literatura podemos encontrar diferentes aproximaciones a la detección de amenazas en imágenes milimétricas pasivas como se explica

en [5, 6]. Por ejemplo, en [7] se utiliza K -medias junto con *Active Shape Models* para la segmentación de las imágenes en tres regiones correspondientes al fondo, cuerpo del sujeto escaneado y objeto. Sin embargo, la segmentación no siempre es correcta. En un trabajo posterior [8], se utilizan modelos de mezclas de gaussianas para caracterizar las mismas tres regiones mejorando sus resultados previos. Martínez *et al.* [9] utilizan métodos de eliminación de ruido previos a la segmentación mediante *Local Binary Fitting*. A pesar de obtener buenos resultados de detección, al menos para imágenes de calidad media/alta, el algoritmo no es apto para aplicaciones en tiempo real. En [10] se propone una detección basada en una segmentación global y otra local mediante mezcla de gaussianas en imágenes milimétricas tomadas en exteriores. No obstante, el método fue testeado con una base de datos muy limitada en cuanto a número de imágenes y tipos de amenazas. Posteriormente, en [11] se utiliza la misma segmentación pero inicializando los parámetros de la mezcla con un K -medias. Gómez *et al.* proponen un algoritmo eficiente basado en eliminación de ruido y morfología matemática. Encontramos también trabajos dedicados a la identificación de amenazas concretas, como pistolas o armas de filo, eso sí, por supuesto, esto solo es posible utilizando imágenes milimétricas activas. Por ejemplo, en [12] los autores utilizan una combinación de técnicas con este propósito: SVD para la reducción de artefactos/ruido, transformada Wavelet para mejorar la resolución, umbralización para la localización de objetivos y, por último, una red neuronal para una identificación invariante a rotaciones. Otro trabajo en esta línea [13] utiliza PCA para eliminar la varianza de escala y rotación y un clasificador con dos capas para distinguir entre amenazas y objetos corrientes. Finalmente, los trabajos de S. López-Tapia *et al.* [5, 6] abordan el problema de detección de amenazas como un problema de clasificación binaria. El primero de ellos aborda el problema mediante la eliminación de ruido y posterior extracción de características. Para la clasificación se estudian múltiples clasificadores, siendo *Random Forest* el que mejores resultados obtiene. No obstante, los resultados obtenidos presentan un número elevado de falsos positivos en algunos de los casos. El segundo de los trabajos [6] propone utilizar una red convolucional con estructura de *autoencoder* para la detección. En este caso, se alcanza una clasificación perfecta con una cantidad insignificante de falsos positivos pero concluyen que el uso de preprocesamiento, el mismo que se utiliza en [5], no mejora en ningún caso los resultados de detección. Serán estos dos métodos los que utilicemos en el capítulo IV para confirmar una mejora sustancial en la detección tras aplicar los métodos de preprocesamiento propuestos.

En esta tesis estudiaremos métodos de procesamiento y mejora de las imágenes para ayudar a la detección en imágenes milimétricas pasivas, no buscando una segmentación, sino, directamente, la localización de amenazas.

4. Degradaciones en imágenes milimétricas

En este trabajo nos centraremos en las imágenes milimétricas pasivas. Estas imágenes presentan degradaciones intrínsecas debidas al proceso de captación. Son habituales las cámaras milimétricas que realizan un escaneo por barrido; estas cámaras generan una señal intermedia, a partir de la cual, mediante una transformación adecuada, se formará la propia imagen milimétrica. Las imágenes intermedias adquiridas presentan datos perdidos debidos al propio sistema de captación por fallos electrónicos, mecánicos y/o físicos. Resultará vital para la formación de la imagen final la recomposición de los datos perdidos. Este problema, habitualmente conocido como *inpainting*, puede ser abordado también mediante el uso de diccionarios. Por otro lado, debido a la poca intensidad de radiación natural presente en las escenas y condiciones de temperatura no ideales, las imágenes captadas presentarán altos niveles de ruido. En segundo lugar, fallos en el sistema de captación pueden provocar la pérdida de datos en las imágenes y, por último, el movimiento del sujeto durante la captación supone un riesgo de emborronamiento en la imagen captada.

El ruido en imágenes se debe principalmente a los sistemas de captación o cámaras que toman las imágenes y, en menor medida, al proceso de conversión analógico-digital (ruido de cuantización). El ruido inherente a la cámara tiene un carácter aleatorio sin correlación espacial y puede modelarse como un proceso aditivo. Es decir, siendo \mathbf{y} la imagen ruidosa captada, \mathbf{x} la imagen libre de ruido y \mathbf{n} la «imagen de ruido»,

$$\mathbf{y} = \mathbf{x} + \mathbf{n}. \quad (\text{I.5})$$

Esta degradación suele deberse a los sensores y demás componentes electrónicos del sistema de captación. Adicionalmente, la correcta iluminación¹ de la escena tiene también un papel importante en la presencia de ruido. El proceso de eliminación del ruido de las imágenes para obtener la imagen «limpia» subyacente suele referirse en la literatura mediante su denominación en inglés, *denoising*. Existen multitud de técnicas de *denoising* en la literatura. Estas dependen en gran medida del modelo de ruido considerado o, en otras palabras, la distribución de probabilidad asumida para modelarlo. En este trabajo utilizaremos el aprendizaje basado en diccionarios como técnica de eliminación de ruido y asumiremos una distribución gaussiana de media nula sobre el mismo. Nótese, no obstante, que el ruido presente en estas imágenes es no estacionario (ver fig. 4), utilizándose el modelo gaussiano como una simplificación.

Las imágenes presentan adicionalmente una degradación exclusiva inherente debida al tiempo de adquisición de las muestras. El movimiento

¹En el caso de las imágenes milimétricas una correcta iluminación no consiste en la presencia de luz de suficiente intensidad, sino a unas condiciones adecuadas de temperatura ambiente.

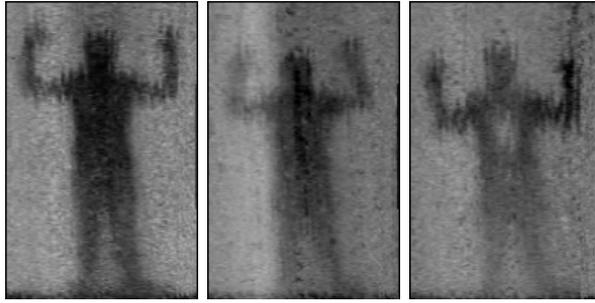


Figura 4: Ejemplos de imágenes milimétricas pasivas.

no controlado del sensor durante la adquisición o el del objeto captado provocan distorsiones en las imágenes, haciendo que el valor en un píxel sea una suma de contribuciones correspondiente no a un única dirección en el espacio, sino a un rango continuo de direcciones, provocando emborronamiento en la imagen. Formalmente, es posible modelar esta degradación mediante la convolución

$$\mathbf{y} = \mathbf{h} * \mathbf{x} \quad (\text{I.6})$$

de la imagen nítida \mathbf{x} con un filtro \mathbf{h} que modelará el movimiento de la cámara y/o objeto. El problema de la estimación de la imagen nítida sin conocimiento previo del filtro de emborronamiento recibe el nombre de deconvolución «ciega» de imágenes. El trabajo propuesto en esta tesis presenta un modelado bayesiano del problema que permite imponer conocimiento previo sobre las características de la solución.

Nótese que es habitual encontrar más de una de estas degradaciones actuando simultáneamente sobre las imágenes. Concretamente consideraremos los problemas de *inpainting* y deconvolución en presencia de ruido.

5. Objetivos y estructura de la tesis

El objetivo principal de la tesis será, pues, el desarrollo de técnicas de preprocesamiento de imágenes con el fin de mejorar la precisión de los clasificadores y, con ello, la tasa de acierto en la detección de amenazas en imágenes milimétricas pasivas. La consecución de este objetivo se aborda mediante el cumplimiento de los siguientes objetivos parciales más específicos:

- Estudio de las técnicas de preprocesamiento de imágenes susceptibles de resultar en una potencial mejora en la calidad de las imágenes milimétricas. Concretamente, *denoising*, *inpainting* y deconvolución ciega.

CAPÍTULO I. INTRODUCCIÓN

- Utilización del paradigma bayesiano para la modelización de estos problemas.
- Estudio y aplicación de distribuciones *a priori sparse*. Principalmente, las distribuciones normal, laplaciana, mezcla infinita de gaussianas y *Spike-and-Slab*.
- Estudio y desarrollo de técnicas bayesianas de aprendizaje de diccionarios con *a priori sparse*.
- Estudio de diccionarios *sparse* para el problema de aprendizaje de diccionarios bayesiano.
- Estudio del efecto del uso de distribuciones *a posteriori* laplacianas en el problema de codificación *sparse*.
- Aplicación de las técnicas basadas en diccionarios desarrolladas para los problemas de *denoising* e *inpainting*.
- Desarrollo de una técnica para la estimación del emborronamiento en imágenes mediante inferencia variacional y *a priori Spike-and-Slab*.
- Diseño de un método completo de deconvolución ciega haciendo uso de la estimación del emborronamiento desarrollada.
- Aplicación de las técnicas de preprocesamiento diseñadas sobre imágenes milimétricas para cuantificar su influencia sobre la tasa de detección.

Los objetivos marcados conducen, de manera natural, a la siguiente estructura de la memoria. En el presente capítulo se ha presentado el problema de detección de objetos/amenazas en imágenes, exponiendo las diversas degradaciones de las imágenes que afectan directamente a su rendimiento, motivando así la necesidad de preprocesamiento de estas.

El capítulo II está dedicado a los problemas de *denoising* e *inpainting* dado que ambos pueden abordarse mediante la representación *sparse* de las imágenes en base a diccionarios. Veremos la modelización bayesiana del problema de aprendizaje de diccionarios clásico, y una modificación en la que usamos diccionarios que serán, a su vez, *sparse*. Además, se describirá la inferencia variacional bayesiana utilizada para la estimación de las diferentes variables latentes y parámetros del problema. Ambos problemas comparten una misma modelización e inferencia de las señales *sparse*; finalizaremos el capítulo con la propuesta preliminar de una nueva modelización para estas señales, imponiendo una distribución *a posteriori* laplaciana sobre ellas.

A continuación, el capítulo III se centra en el problema de deconvolución ciega imágenes. En él, se propone una técnica de estimación del emborronamiento mediante el uso de la *a priori Spike-and-Slab* para

5. OBJETIVOS Y ESTRUCTURA DE LA TESIS

modelar el conocimiento previo sobre las imágenes nítidas. En este caso, desarrollamos un método de inferencia variacional EM.

El último capítulo teórico de la memoria, cap. IV, está enfocado a la aplicación práctica de los métodos desarrollados a las imágenes milimétricas. Se utilizarán dos metodologías de detección diferentes sobre las imágenes preprocesadas utilizando los métodos de *inpainting*, eliminación de ruido y deconvolución vistos en los dos capítulos previos y analizando las ganancias producidas por este preprocesamiento.

El capítulo V enumera las conclusiones finales de la presente tesis y expone una serie de posibles vías futuras de investigación y, finalmente, la tesis concluye con la bibliografía consultada para la elaboración de este trabajo.

Capítulo II

Denoising e Inpainting mediante aprendizaje de diccionarios

Como ya hemos indicado en el capítulo anterior, la eliminación de ruido en imágenes milimétricas debe permitir la mejora en la clasificación de las mismas. En este capítulo, proponemos un método de mejora de imágenes a imágenes milimétricas. La metodología propuesta es, como veremos, no solo capaz de eliminar ruido en imágenes, sino que también permite abordar el problema de completado de imágenes o *inpainting*, que, como se hizo intuir en el capítulo I, es necesario para la formación de la imagen milimétrica final. Más adelante, en el capítulo IV, quedará justificada la aplicación de esta técnica.

Tanto la eliminación de ruido como el *inpainting* son problemas que pueden ser abordados desde la teoría del aprendizaje de diccionarios. Desarrollamos un método bayesiano para la estimación de estos, utilizando la inferencia variacional para la estimación de las imágenes subyacentes. Según la teoría de diccionarios, las señales admiten una representación a partir de un pequeño número de los elementos de un diccionario, recibiendo el nombre de *sparse*. Para modelar este tipo de representaciones utilizaremos distribuciones *a priori* adecuadas y métodos de representación *sparse* que serán los encargados de estimar los coeficientes de esta representación.

En primer lugar se expondrá el método bayesiano de aprendizaje de diccionarios con *a priori* laplaciana. Este trabajo ha sido publicado en revista [14]. Como extensión de este trabajo, expondremos a continuación la adaptación al modelo doblemente *sparse* (publicado en [15]), en el cual se consideran diccionarios a su vez *sparse*. Por último, proponemos una alternativa a la estimación de las señales *sparse* pendiente de publicación basada en el uso de distribuciones *a posteriori* laplacianas en el apartado 6.

1. Introducción

Un diccionario no es más que un conjunto de K vectores $\mathbf{d}_k \in \mathbb{R}^P$ que permiten la representación de señales a partir de combinaciones lineales de los mismos

$$\mathbf{y} = \sum_k \mathbf{d}_k x_k, \quad (\text{II.1})$$

donde $\mathbf{y} \in \mathbb{R}^P$ es la señal representada y x_k los coeficientes de la combinación lineal. Esta relación puede expresarse de forma matricial como

$$\mathbf{y} = \mathbf{D}\mathbf{x}, \quad (\text{II.2})$$

donde $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_K]$ y $\mathbf{x} = [x_1, \dots, x_K]^T$. Un primer problema reside en la resolución de esta ecuación, i. e., encontrar el vector \mathbf{x} que representa a la señal \mathbf{y} en base al diccionario \mathbf{D} . Si este fuera una base de \mathbb{R}^P , la solución resultaría sencilla, pues \mathbf{D} sería una matriz invertible y podríamos obtener \mathbf{x} simplemente como $\mathbf{D}^{-1}\mathbf{y}$. No obstante, esto correspondería a un simple cambio de base sin ninguna ganancia clara.

La situación real es algo más compleja. Por un lado, queremos tomar en consideración el ruido presente en la adquisición de señales no incluido en el modelo simplista visto en (II.2). Por otro, queremos exigir que la solución obtenida solo haga uso de un pequeño subconjunto de los elementos \mathbf{d}_k del diccionario, conocidos como «átomos». Este tipo de soluciones son conocidas como representaciones *sparse*. Podemos expresar matemáticamente el problema de representación *sparse* como

$$\begin{aligned} \hat{\mathbf{x}} &= \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2, \\ &s.a. \|\mathbf{x}\|_0 \leq T, \end{aligned} \quad (\text{II.3})$$

donde la pseudonorma ℓ_0 cuenta el número de componentes distintos de cero en un vector. Este es un problema NP-completo y mal definido, en el sentido de Hadamard, que requiere técnicas específicas para su resolución, véase el apartado 2.

El número de átomos necesarios para representar una señal depende en gran medida del diccionario elegido. Una base, pese a ser un diccionario válido que garantiza la posibilidad de encontrar una representación de cualquier señal, resulta una elección de diccionario muy poco eficiente en cuanto a su capacidad de conseguir soluciones *sparse*. Es de esta necesidad de facilitar las soluciones *sparse*, dada la poca expresividad de las bases, la que condujo al desarrollo de los diccionarios sobrecompletos. Esto es, diccionarios que cuentan con un número de átomos bastante mayor que la dimensionalidad de las señales a representar ($K > P$). La transición paulatina hacia los diccionarios sobrecompletos comenzó con el uso de diccionarios analíticos ortonormales basados en diferentes transformadas

como DCT (*Discrete Cosine Transform*), Wavelet o la transformada Gabor. Sus limitaciones [16] condujeron a sucesivas mejoras, para solucionar deficiencias como su sensibilidad a la translación o rotación. Unas primeras aproximaciones al diseño de diccionarios sobrecompletos trataron de preservar las favorables propiedades de ortogonalidad de estas bases, pero pronto se demostró insuficiente. Los trabajos posteriores en [17] y [18] tuvieron una gran influencia en los avances recientes en el campo del aprendizaje de diccionarios para la representación *sparse* de señales. En ellos se proponía el uso de colecciones de datos que describieran mejor las señales en contraposición al uso de funciones matemáticas.

La tendencia actual consiste en el diseño de diccionarios específicos que permitan representaciones lo más *sparse* posible para un tipo de señales. Parece obvio que un diccionario que proporcione una buena representación para imágenes de rostros no será muy adecuado para la representación de imágenes de texto, y viceversa y, obviamente tampoco lo será para la representación de imágenes milimétricas. Aparece aquí el aprendizaje conjunto del diccionario y de los vectores de representación. El problema se transforma en encontrar el diccionario que mejor represente, i. e., que produzca soluciones más *sparse*, un conjunto de Q señales $\{\mathbf{y}_q\}_{q=1}^Q$ tal que

$$\mathbf{y}_q = \mathbf{D}\mathbf{x}_q + \mathbf{n}_q, \quad (\text{II.4})$$

donde \mathbf{n}_q representa el ruido presente en la señal. Podemos finalmente expresar el problema de aprendizaje de diccionarios como un problema de optimización similar a (II.3)

$$\begin{aligned} \hat{\mathbf{D}}, \hat{\mathbf{X}} &= \arg \min_{\mathbf{D}, \{\mathbf{x}_q\}} \sum_q \|\mathbf{y}_q - \mathbf{D}\mathbf{x}_q\|^2 \\ &= \arg \min_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\text{F}}^2, \\ &s.a. \|\mathbf{x}_q\|_0 \leq T, \forall q, \end{aligned} \quad (\text{II.5})$$

con $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_Q]$ e $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_Q]$. En las próximas secciones veremos como se ha abordado la resolución del problema de aprendizaje de diccionarios (II.5), íntimamente relacionado con el problema de representación *sparse* (II.3), así como la solución que proponemos en esta tesis.

Los diccionarios se han aplicado en multitud de tareas de procesamiento de imagen y aprendizaje automático. Entre muchas otras, podemos encontrar aplicaciones en eliminación de ruido, inpainting y deconvolución [19, 20], superresolución de imágenes [21], restauración [22], reconocimiento facial [23], Compressive Sensing en MRI [24] y clasificación [25, 26]. No obstante, esta técnica nunca ha sido aplicada a las imágenes milimétricas. Veremos en el capítulo IV la aplicación del aprendizaje basado en diccionarios para la mejora de estas imágenes. La

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

clasificación suele basarse en el uso diferente de los átomos del diccionario que hacen diferentes clases.

Comenzaremos por mostrar cómo se aplican estos conceptos a la eliminación de ruido en imágenes e *inpainting*, problemas que tratamos en esta tesis. Dada una imagen, extraeremos pequeños bloques de tamaño $h \times w$ mediante un proceso de ventana deslizante. Estos bloques serán dispuestos en forma de vectores columna conformando cada una de las señales \mathbf{y}_q .

Si nos fijamos en el modelo de representación *sparse* (II.4), resulta obvio el proceso de *denoising*: habrá que estimar los coeficientes *sparse* de cada señal ruidosa \mathbf{y}_q y obtendremos su versión «limpia» utilizando el diccionario aprendido y los coeficientes estimados $\mathbf{y}_q^{den} = \hat{\mathbf{D}}\hat{\mathbf{x}}_q$. Nótese que, como veremos más adelante, la calidad del diccionario aprendido, así como la precisión de las estimaciones obtenidas para \mathbf{x}_q dependerán, en gran medida, de la precisión con la que se ha estimado el ruido presente en las señales y, por ende, en la imagen. Tras la reconstrucción de las señales será necesario revertir el proceso de extracción de bloques. Si estos se habían extraído con solapamiento, tendremos múltiples estimaciones para cada píxel. Todas las estimaciones de cada píxel serán promediadas dando lugar a la imagen final.

Por lo que respecta al completado de imágenes (*inpainting*), la formulación presentada necesita una pequeña modificación para incluir información sobre cuáles son los píxeles no observados. Esta modificación consiste en eliminar los píxeles del vector \mathbf{y}_q que no han sido observados y, además, repetir este proceso sobre cada uno de los átomos del diccionario, resultando en la eliminación de filas completas en la matriz \mathbf{D} . Alternativamente, podemos emular este proceso con la introducción de una matriz diagonal \mathbf{M}_q , cuya la entrada (k, k) -ésima será igual a 1 si el píxel había sido observado, 0 en caso contrario. Con esto, el modelo del problema de representación *sparse* queda

$$\mathbf{M}_q \mathbf{y}_q = \mathbf{M}_q \mathbf{D} \mathbf{x}_q + \mathbf{M}_q \mathbf{n}_q. \quad (\text{II.6})$$

Este modelo nos sirve tanto para el aprendizaje del diccionario como para obtener la representación *sparse* de una señal dada. Por ejemplo, si únicamente disponemos de una imagen con valores perdidos y queremos aprender un diccionario a partir de esta misma imagen, tendremos que introducir la matriz \mathbf{M}_q (que suponemos conocida o previamente calculada) en el proceso de aprendizaje (II.5). Otra posibilidad consiste en preaprender un diccionario a partir de un conjunto de imágenes limpias de características similares. En ambos casos es necesario, sin embargo, la inclusión de esta matriz en el cálculo del vector *sparse* de coeficientes (II.3). La representación obtenida por esta proceso se utilizará posteriormente con el diccionario completo para recomponer el bloque

original sin datos perdidos, es decir, $\mathbf{y}_q^{inp} = \hat{\mathbf{D}}\hat{\mathbf{x}}_q$. Una vez más, queda recomponer la imagen completa a partir de los bloques de forma idéntica a le explicada más arriba.

2. Revisión bibliográfica

Veremos ahora los trabajos más relevantes de la literatura acerca del aprendizaje de diccionarios y obtención de representaciones *sparse*.

Dado que la función objetivo $\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2$ del problema de aprendizaje no es convexa globalmente, pero sí en \mathbf{X} y \mathbf{D} por separado, su optimización puede abordarse alternando la minimización sobre cada una de las variables. Sin embargo, como ya avanzábamos anteriormente, la optimización de cada uno de los \mathbf{x}_q de \mathbf{X} dado un \mathbf{D} fijo y sujeto a la restricción de la pseudonorma ℓ_0 es un problema NP-completo. La solución óptima del problema (II.5) se obtendría tras probar todas las posibles combinaciones $C(t, K)$ con $1 \leq t \leq T$ y elegir cuál de ellas, para cada q , produce un menor valor de $\|\mathbf{y}_q - \mathbf{D}\mathbf{x}_q\|^2$. Esto hace un total de

$$N_{comb} = \sum_{t=1}^T C(t, K) = \sum_{t=1}^T \frac{K!}{(K-t)!t!} \quad (\text{II.7})$$

combinaciones. Obviamente, este número resulta impracticable para cualquier algoritmo. Por ello, habitualmente se han propuesto métodos constructivos «*greedy*» para aproximar la solución de este problema, entre los cuales encontramos el popular K-SVD [27], ver apartado 3. Estos métodos van añadiendo componentes iterativamente a la solución, minimizando lo máximo posible la función objetivo en cada iteración.

Alternativamente, otros trabajos optan por la relajación de la restricción impuesta sobre los vectores de coeficientes, utilizando la norma-1 en su lugar. Encontramos un amplio abanico de técnicas de optimización convexa aplicables a este problema. Véanse, p. ej., [27–30].

El problema de aprendizaje de diccionarios admite una formulación probabilística. Encontramos múltiples trabajos en esta vertiente. En los trabajos de Olshausen y Field [18] y Lewicki y Sejnowski [31] se introdujo un modelo generativo para los datos que permitía la estimación por máxima verosimilitud (ML, del inglés, *Maximum Likelihood*) del diccionario y las representaciones *sparse*. Según este modelo, cuando la *a priori* sobre los vectores de representación es una laplaciana de media nula y el residuo $(\mathbf{y} - \mathbf{D}\mathbf{x})$ se aproxima por una distribución normal, también de media cero, este es equivalente a utilizar la norma-1 en el problema de aprendizaje original (II.5). Tras este trabajo, apreció una sucesión de propuestas en las que se modificaba la actualización del diccionario, el proceso de estimación de los vectores *sparse* o ambos. En [32], utilizando el modelo generativo de [18], los autores proponen el uso de *Orthogonal*

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

Matching Pursuit (OMP) para resolver la fase de clasificación *sparse* y una solución analítica para la actualización del diccionario. Posteriormente, la investigación se centró en el uso de estimación MAP (Máximo *a Posteriori*), permitiendo introducir restricciones también sobre el diccionario y no solo sobre los vectores de representación. El trabajo en [33], por ejemplo, se impone una *a priori* de norma de Frobenius unidad sobre el diccionario, con la misma norma para cada átomo, y el algoritmo FOCUSS [34] para obtener los vectores *sparse*. Elecciones alternativas de *a priori* incluyen distribuciones menos *sparse* basadas en la divergencia de Kullback-Leibler para la regularización ℓ_1 [35]. Las principales ventajas de esta aproximación residen en su mayor estabilidad de soluciones *sparse* y su eficiente inferencia convexa.

Todas las referencias comentadas utilizan estimadores ML o MAP para la resolución del problema de aprendizaje de diccionarios. Desafortunadamente, pese a su relativa simplicidad, su mayor inconveniente es que no tienen en cuenta la incertidumbre relativa a los coeficientes *sparse*, que, como veremos, afecta a la precisión de las estimaciones. Adicionalmente, puesto que la varianza del ruido tampoco se tiene en cuenta directamente, estos algoritmos dependen de otras técnicas para la estimación del ruido. El trabajo en [29] muestra que resulta crucial la correcta estimación de este parámetro, mostrando como el algoritmo K-SVD usado en problemas de denoising [19] se ve gravemente afectado por la precisión en la estimación de la varianza. Surgen naturalmente diferentes técnicas para superar esta limitación, incluyendo la varianza de ruido como parámetro del modelo que puede ser estimado utilizando la incertidumbre relativa a las variables estimadas. En [36] se propone un algoritmo de maximización de la esperanza (EM) para estimar diccionario, sin *a priori*, y señales *sparse*, imponiendo una laplaciana sobre las últimas. Sin embargo, dada la intratabilidad de la inferencia con esta *a priori* los autores proponen una aproximación variacional de esta que resulta en una *a posteriori* gaussiana.

En los trabajos de Zhou *et al.*, [29] and [37] se utiliza una *a priori* Bernoulli-gaussiana sobre los vectores de coeficientes y se modela BFPA (*Beta Process Factor Analysis*) introducido en [38], siendo capaz de estimar el tamaño del diccionario, de manera similar a [39]. El método de inferencia utilizado es el costoso MCMC (*Markov Chain Monte Carlo*). El trabajo en [24] sigue una metodología similar utilizando BFPA pero añadiendo una regularización TV sobre la señal (imagen de resonancia magnética). Este utiliza el método de los ADMM para la minimización de la TV y MCMC para la inferencia.

El modelado BFPA pertenece a la categoría de distribuciones denominadas *Spike and Slab*, véase el capítulo III para una explicación detallada. La idea principal de esta distribución consiste en la introducción de una variable discreta (binaria) que permita la selección de variables,

haciendo que algunas sean no cercanas a, sino exactamente cero. Véase, p. ej., el trabajo [40], donde se utiliza esta *a priori* sobre los coeficientes de la regresión lineal. Por otro lado, [41] propone una aproximación EM en la que no se asume independencia entre átomos del diccionario. En [42] se adaptó el trabajo en [37] para poder trabajar con grandes conjuntos de datos. La idea básica consiste en la estimación de los parámetros mediante una suma ponderada de estimaciones locales sobre subconjuntos de menor tamaño, resultando en un método más robusto frente a mínimos locales con menores requisitos computacionales. Hansen *et al.* [43] diseñan un modelo jerárquico de esta distribución que permite la incorporación de información previa sobre el soporte de los vectores de representación mediante procesos gaussianos, estimando los parámetros relacionados en un trabajo posterior [44]. Los GP (*Gaussian Processes*) modelan correlaciones entre las probabilidades SnS de diferentes vectores de representación. Los autores de [45] modifican esta *a priori* mediante una función indicador basándose en la técnica de estimación MAP de [46] y permitiendo la inferencia de los coeficientes óptimos mediante la solución de un problema de optimización sin relajación. Por último, el trabajo de Zhang *et al.* [47], además de utilizar esta *a priori*, impone un GP multiplicativo sobre las coordenadas de cada bloque en los que subdivide la imagen para imponer similitud sobre el soporte de bloques vecinos. Nótese que la inferencia utilizando distribuciones *Spike-and-Slab*, ver [37] y [48] por ejemplo, puede no conducir a una solución exactamente *sparse* dado que el cálculo de la esperanza sobre una distribución binaria no tiene por qué producir un valor binario. No obstante, esto puede resolverse fácilmente mediante umbralización [42]. El método que proponemos y presentaremos a continuación también conduce a soluciones exactamente *sparse* puesto que permite añadir y eliminar átomos del modelo.

Algunas propuestas recientes se alejan de formulaciones probabilísticas en pro de mejorar el tiempo de ejecución del proceso completo, aún siendo métodos aproximados. Los trabajos de LeCun *et al.* entrenan una red neuronal [49] o un árbol de decisión binario [50] que posibilitan su empleo en aplicaciones de visión por computador en tiempo real como, por ejemplo, el reconocimiento de objetos. Por otro lado, [51] propone una actualización parcial del diccionario para acelerar la convergencia, donde cada átomo se actualiza conjuntamente con la fila correspondiente de la matriz de coeficientes. Al contrario que en K-SVD, este algoritmo permite la modificación del soporte de dicha fila. Además, incorporan el algoritmo FISTA [52] para la fase de obtención de los códigos *sparse*. En otra línea, el trabajo de Skretting y Engan [53] realiza una actualización continua del diccionario tras el procesamiento de cada una de las señales observadas, en lugar de procesar todo el *batch* de señales para su actualización, como se hace generalmente en otras técnicas de aprendizaje de diccionarios como el

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

predecesor de este trabajo [54] o en el K-SVD. Sin embargo, su convergencia no está demostrada. Liu *et al.* [55] proponen un método de estimación para el diccionario y las representaciones *sparse* para la reconstrucción de imágenes en MRI donde la estimación de los átomos consiste en un refinamiento progresivo, válido en un trabajo posterior [56] con aplicación en deconvolución de imágenes.

Por último, resulta interesante el trabajo teórico [57], donde los autores analizan los mínimos locales inherentes al problema no convexo del aprendizaje de diccionarios, garantizando, con alta probabilidad, la existencia de un mínimo local entorno al diccionario que generó las señales *sparse*.

En este capítulo explicaremos brevemente el algoritmo K-SVD para permitir una comparación con el método bayesiano propuesto para el problema de aprendizaje de diccionarios ℓ_1 . Este algoritmo, al que llamamos BKSVD, estima las distribuciones *a posteriori* completas de los vectores \mathbf{x}_q y toma en consideración la incertidumbre relativa a los coeficientes estimados, integrando, además, la estimación de los parámetros del modelo en el mismo marco de inferencia. Después, veremos una extensión de este trabajo que considera que el diccionario admite, a su vez, una representación *sparse*. Por último, estudiaremos una aproximación diferente para el cálculo de las representaciones *sparse*, utilizando distribuciones *a posteriori* laplacianas.

3. K-SVD

Este simple y popular algoritmo de aprendizaje de diccionarios resuelve aproximadamente el problema expuesto en (II.5). La optimización se aborda alternadamente entre los átomos \mathbf{d}_k del diccionario \mathbf{D} y las filas $\mathbf{x}_{k,:}$ de la matriz de coeficientes \mathbf{X} .

Dado el estado actual del diccionario $\mathbf{D}^{(t)}$, que denotaremos simplemente por \mathbf{D} por claridad, el método K-SVD emplea el algoritmo *Orthogonal Matching Pursuit* (OMP) para determinar el soporte de \mathbf{X} , es decir, las posiciones no nulas de cada vector de representación \mathbf{x}_q . Sin embargo, los valores obtenidos en estas posiciones por dicho algoritmo se descartan. Nótese aquí un gran inconveniente de este algoritmo: es necesario prefijar el número de componentes distintas de cero.

Una vez determinado el soporte de los vectores \mathbf{x}_q con OMP, el algoritmo K-SVD actualiza los átomos del diccionario y los elementos no nulos de la matriz de coeficientes. Para entender este paso, resulta conveniente observar que el término \mathbf{DX} se puede descomponer de la siguiente forma

$$\mathbf{DX} = \sum_{k=1}^K \mathbf{d}_k \mathbf{x}_{k,:}, \quad (\text{II.8})$$

donde $\mathbf{x}_{k,:}$ representa la k -ésima fila de \mathbf{X} .

Esta descomposición en matrices de rango uno, a lo sumo, constituye la base de un proceso de actualización cíclica en la que cada par $\{(\mathbf{d}_k, \mathbf{x}_{k,:})\}_{k=1}^K$ se actualiza individualmente manteniendo constantes el resto. Específicamente, para el k -ésimo par, podemos desarrollar la función objetivo como

$$\begin{aligned} \min_{\mathbf{d}_k, \mathbf{x}_{k,:}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 &= \min_{\mathbf{d}_k, \mathbf{x}_{k,:}} \|\mathbf{Y} - \sum_{i \neq k} \mathbf{d}_i \mathbf{x}_{i,:} - \mathbf{d}_k \mathbf{x}_{k,:}\|_F^2 \\ &= \min_{\mathbf{d}_k, \mathbf{x}_{k,:}} \|\mathbf{R}_k - \mathbf{d}_k \mathbf{x}_{k,:}\|_F^2, \end{aligned} \quad (\text{II.9})$$

donde claramente se aprecia que el término residual

$$\mathbf{R}_k = \mathbf{Y} - \sum_{i \neq k} \mathbf{d}_i \mathbf{x}_{i,:} \quad (\text{II.10})$$

no depende de \mathbf{d}_k ni $\mathbf{x}_{k,:}$.

Dado que el rango de $\mathbf{d}_k \mathbf{x}_{k,:}$ es, como máximo, uno, el problema en (II.9) se puede resolver mediante la descomposición en valores singulares (SVD, de sus siglas en inglés). La SVD de \mathbf{R}_k nos permite factorizar esta matriz de la forma $\mathbf{R}_k = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$, donde $\mathbf{U} \in \mathbb{R}^{P \times P}$ y $\mathbf{V} \in \mathbb{R}^{K \times K}$ son dos matrices ortogonales y $\mathbf{\Sigma} \in \mathbb{R}^{P \times K}$ una matriz diagonal formada con los valores singulares σ_i de \mathbf{R}_k ordenados de mayor a menor. Podemos ahora comprobar fácilmente que

$$\mathbf{R}_k = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (\text{II.11})$$

y que obtendremos la mejor aproximación de rango uno (en términos de la norma de Frobenius) quedándonos únicamente con el primer sumando de la serie, es decir, $\mathbf{R}_k \approx \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ y, por tanto, las actualizaciones correspondientes serían $\mathbf{d}_k^{(t+1)} = \mathbf{u}_1$ y $\mathbf{x}_{k,:}^{(t+1)} = \sigma_1 \mathbf{v}_1^T$. No obstante, este proceso destruiría la estructura *sparse* de $\mathbf{x}_{k,:}$. Para solucionar este problema se considera la matriz $\tilde{\mathbf{R}}_k$ formada solamente por las columnas de \mathbf{R}_k correspondientes a las entradas no nulas de $\mathbf{x}_{k,:}$ previamente determinadas por el algoritmo OMP. Así tenemos ahora $\tilde{\mathbf{R}}_k \approx \tilde{\sigma}_1 \tilde{\mathbf{u}}_1 \tilde{\mathbf{v}}_1^T$, con lo que

$$\mathbf{d}_k^{(t+1)} = \tilde{\mathbf{u}}_1 \quad (\text{II.12})$$

$$\tilde{\mathbf{x}}_{k,:}^{(t+1)} = \tilde{\sigma}_1 \tilde{\mathbf{v}}_1^T. \quad (\text{II.13})$$

Nótese que esta actualización a partir de la matriz truncada $\tilde{\mathbf{R}}_k$ no tiene una justificación matemática y reducirá la calidad del ajuste producido por la SVD. Sin embargo, y como veremos más adelante, este procesado truncado

Algoritmo 1 K-SVD.

ENTRADA: $\mathbf{D}^{(0)}$, \mathbf{Y} , T .

```

1:  $t = 0$ 
2: while  $\|\mathbf{Y} - \mathbf{D}^{(t)}\mathbf{X}^{(t)}\|_{\text{F}}^2 > \text{tol}$  do
3:   Calcular el soporte de  $\mathbf{X}^{(t)}$  a partir de  $\mathbf{D}^{(t)}$  con OMP.
4:   for  $k = 1, \dots, K$  do
5:     Calcular  $\mathbf{R}_k$  según (II.10).
6:     Obtener  $\tilde{\mathbf{R}}_k$  descartando las columnas indicadas por el soporte de
        $\mathbf{x}_{k,:}^{(t)}$ .
7:     Calcular la SVD de  $\tilde{\mathbf{R}}_k$  y actualizar  $\mathbf{d}_k^{(t+1)}$  y  $\tilde{\mathbf{x}}_{k,:}^{(t+1)}$  según (II.12) y
       (II.13) respectivamente.
8:   end for
9:    $t = t + 1$ 
10: end while
SALIDA:  $\mathbf{D}$ ,  $\mathbf{X}$ .
```

tiene una explicación clara en un contexto bayesiano. Podemos encontrar un pseudocódigo de este proceso en el Algoritmo 1.

La principal ventaja del K-SVD es su simplicidad dada su naturaleza constructiva. Sin embargo, el algoritmo no tiene en cuenta la incertidumbre en las estimaciones de \mathbf{D} y \mathbf{X} , lo que representa un fuerte inconveniente. Si bien es cierto que no considerar la incertidumbre relativa a los átomos del diccionario podría no ser un grave problema, dado el usual elevado número de columnas en \mathbf{X} , cada columna \mathbf{x}_q de \mathbf{X} tiene un número reducido de componentes distintas de cero, por lo que se debería considerar su incertidumbre. Además, el K-SVD requiere el conocimiento previo del número de entradas no nulas T de cada columna. Esta información puede no estar disponible o, incluso, cambiar para cada columna.

El método bayesiano que proponemos en el siguiente apartado [14] tiene en cuenta la incertidumbre relativa a las estimaciones y produce representaciones *sparse* sin la necesidad de conocer previamente el número de componentes distintas de cero de cada una de las señales. Este método utiliza una *a priori* jerárquica de tres niveles sobre los vectores de coeficientes que garantiza la obtención de resultados *sparse*, y proponemos un método de inferencia rápida que reduce la complejidad computacional.

4. BKSVD

4.1. Modelado bayesiano

El problema de aprendizaje de diccionarios expuesto en (II.5), utilizando su relajación con la norma-1, acepta un modelado bayesiano.

Podemos modelar el proceso de observación mediante una distribución normal

$$p(\mathbf{Y}|\mathbf{D}, \mathbf{X}, \beta) \propto \beta^{\frac{PQ}{2}} \exp \left\{ -\frac{\beta}{2} \|\mathbf{Y} - \mathbf{DX}\|_{\mathbb{F}}^2 \right\}, \quad (\text{II.14})$$

donde β representa la precisión (inversa de la varianza) del ruido. Asumimos que la precisión β sigue una distribución Gamma de la forma

$$p(\beta|a, b) = \Gamma(\beta|a, b) \propto \beta^{a-1} \exp(-b\beta), \quad (\text{II.15})$$

con $a > 0$ y $b > 0$ siendo los parámetros de forma y escala inverso respectivamente.

Debemos modelar ahora la información *a priori* que tenemos sobre el diccionario y la matriz de coeficientes. En cuanto a las columnas \mathbf{d}_k de \mathbf{D} , como esperamos que sean vectores unitarios, utilizamos la siguiente distribución *a priori* sobre \mathbf{D}

$$p(\mathbf{D}) = \prod_{k=1}^K p(\mathbf{d}_k), \quad (\text{II.16})$$

donde

$$p(\mathbf{d}_k) = \begin{cases} \text{const} & \|\mathbf{d}_k\| = 1 \\ 0 & \|\mathbf{d}_k\| \neq 1. \end{cases} \quad (\text{II.17})$$

En cuanto a la matriz de coeficientes, aunque se podrían considerar múltiples distribuciones *sparse*, véase [58], nos centramos en este trabajo en la distribución *a priori* de Laplace. El inconveniente de esta distribución reside en que esta y la verosimilitud (II.14) no son conjugadas, por lo que la inferencia resultará analíticamente intratable. Para solucionar este problema, utilizaremos un modelo jerárquico de distribución *a priori* para cada una de las columnas de \mathbf{X} . Este modelo conserva las propiedades *sparse* que la distribución de Laplace pero posibilitando la inferencia. Concretamente, para cada una de las columnas $\{\mathbf{x}_q\}_{q=1}^Q$ imponemos una gaussiana de media cero

$$p(\mathbf{x}_q|\boldsymbol{\gamma}_q) = \prod_{k=1}^K \mathcal{N}(x_{kq}|0, \gamma_{kq}) = \mathcal{N}(\mathbf{x}_q|\mathbf{0}, \boldsymbol{\gamma}_q), \quad (\text{II.18})$$

donde $\boldsymbol{\gamma}_q \in \mathbb{R}^{K \times 1}$ es un vector columna cuyos elementos $\{\gamma_{kq}\}_{k=1}^K \in \mathbb{R}^+$ representan la inversa de la varianza asociada a cada una de las entradas x_{kq} y $\boldsymbol{\Gamma}_q = \text{diag}(\boldsymbol{\gamma}_q)$. Imponemos, además, una distribución Gamma sobre los hiperparámetros

$$p(\boldsymbol{\gamma}_q|\lambda_q) = \prod_{k=1}^K \Gamma(\gamma_{kq}|1, \lambda_q/2) \quad (\text{II.19})$$

y

$$p(\lambda_q|\nu_q) = \Gamma(\lambda_q|\nu_q/2, \nu_q/2), \quad (\text{II.20})$$

donde asumimos una distribución plana para ν_q .

Podemos comprobar que, marginalizando, este modelo jerárquico produce una distribución de Laplace sobre los vectores \mathbf{x}_q

$$p(\mathbf{x}_q|\lambda_q) = \int p(\mathbf{x}_q|\gamma_q)p(\gamma_q|\lambda_q)d\gamma_q = \frac{\lambda_q^{K/2}}{2^K} \exp\{-\sqrt{\lambda_q}\|\mathbf{x}_q\|_1\}. \quad (\text{II.21})$$

Una vez expuestos todos los ingredientes de nuestro modelo, pasamos a denotar al conjunto de todas las incógnitas por

$$\Theta = \left\{ \{\mathbf{d}_k\}_{k=1}^K, \{\mathbf{x}_q\}_{q=1}^Q, \mathbf{\Gamma}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \beta \right\}, \quad (\text{II.22})$$

donde $\mathbf{\Gamma}$, $\boldsymbol{\lambda}$ y $\boldsymbol{\nu}$ denotan a los conjuntos $\{\gamma_q\}_{q=1}^Q$, $\{\lambda_q\}_{q=1}^Q$ y $\{\nu_q\}_{q=1}^Q$ respectivamente.

Finalmente, podemos condensar toda nuestra información acerca del problema mediante la distribución conjunta

$$p(\mathbf{Y}, \Theta) = p(\mathbf{Y}|\mathbf{D}, \mathbf{X}, \beta)p(\beta)p(\mathbf{D})p(\mathbf{X}|\mathbf{\Gamma})p(\mathbf{\Gamma}|\boldsymbol{\lambda})p(\boldsymbol{\lambda}|\boldsymbol{\nu})p(\boldsymbol{\nu}). \quad (\text{II.23})$$

4.2. Inferencia variacional

El objetivo ahora es encontrar la distribución *a posteriori* de todas las incógnitas de nuestro problema, es decir, $p(\Theta|\mathbf{Y})$. Esto es, vamos a calcular la distribución completa, no solo su moda (MAP). Sin embargo, su cálculo, dado por

$$p(\Theta|\mathbf{Y}) = \frac{p(\mathbf{Y}, \Theta)}{p(\mathbf{Y})} \quad (\text{II.24})$$

donde

$$p(\mathbf{Y}) = \int p(\mathbf{Y}, \Theta)d\Theta \quad (\text{II.25})$$

resulta intratable analíticamente. Para solucionar este problema, los métodos variacionales introducen una distribución variacional sobre las variables latentes $q(\Theta)$ para aproximar la distribución *a posteriori*. Típicamente se utiliza la divergencia de Kullback-Leibler para medir la disimilitud entre estas distribuciones

$$\text{KL}(q(\Theta) || p(\Theta|\mathbf{Y})) = \int q(\Theta) \log \frac{q(\Theta)}{p(\Theta|\mathbf{Y})} d\Theta \quad (\text{II.26})$$

y se trata de minimizar esta medida. Podemos observar fácilmente que aplicando (II.24), tenemos

$$\text{KL}(q(\Theta) || p(\Theta|\mathbf{Y})) = \int q(\Theta) \log \frac{q(\Theta)}{p(\mathbf{Y}, \Theta)} d\Theta + \log p(\mathbf{Y}), \quad (\text{II.27})$$

comúnmente expresado como

$$\log p(\mathbf{Y}) = \mathcal{L}(q(\Theta)) + \text{KL}(q(\Theta) \parallel p(\Theta|\mathbf{Y})), \quad (\text{II.28})$$

donde el funcional $\mathcal{L}(q(\Theta))$ recibe el nombre de «cota inferior de la evidencia», en adelante, denominado ELBO por sus más conocidas siglas en inglés (*Evidence Lower Bound*). Al ser el término de la izquierda, el logaritmo de la verosimilitud marginal $p(\mathbf{Y})$, una cantidad fija, resulta evidente que podemos minimizar la divergencia $\text{KL}(q \parallel p)$ maximizando la ELBO. Si no aplicamos ninguna restricción a $q(\Theta)$, podemos observar que el máximo de la ELBO se produce cuando se anula la divergencia, que, a su vez, ocurre cuando $q(\Theta)$ coincide con la distribución *a posteriori*. No obstante, como ya comentábamos, el cálculo directo de esta es intratable. Cosecuentemente, habrá que considerar una familia restringida de distribuciones de probabilidad con la suficiente riqueza y flexibilidad que permita aproximar convenientemente la verdadera distribución *a posteriori* pero que, a su vez, garanticen la tratabilidad de la optimización. Una aproximación frecuente está basada en la teoría del campo medio (*Mean Field Theory*) [59] y consiste en particionar Θ en conjuntos disjuntos de variables latentes θ_m , donde $m = 1, \dots, M$ y asumir que la distribución $q(\Theta)$ factoriza como

$$q(\Theta) = \prod_{m=1}^M q_m(\theta_m). \quad (\text{II.29})$$

Nótese que no se realiza ninguna suposición sobre la forma concreta de cada uno de las distribuciones q_m . Es bien conocido [59] que podemos obtener las distribuciones q_m que maximizan la ELBO como

$$\log q(\theta_m) = \langle \log p(\mathbf{Y}, \Theta) \rangle_{\Theta \setminus \theta_m} + \text{const}, \quad (\text{II.30})$$

donde $\langle \cdot \rangle_{\Theta \setminus \theta_m}$ denota la esperanza tomada respecto a todas las variables latentes excepto θ_m y la constante es independiente de la variable de interés. En el caso de asumir una distribución *a posteriori* degenerada, podemos calcular el valor de la variable como

$$\hat{\theta}_m = \arg \max_{\theta_m} \log q(\theta_m) = \arg \max_{\theta_m} \log \langle \log p(\mathbf{Y}, \Theta) \rangle_{\Theta \setminus \theta_m}. \quad (\text{II.31})$$

Vamos ahora a aplicar esta teoría utilizando la siguiente factorización de nuestra distribución *a posteriori*

$$q(\Theta) = \left[\prod_{k=1}^K q(\mathbf{d}_k) \right] q(\mathbf{X})q(\Gamma), q(\lambda)q(\nu)q(\beta) \quad (\text{II.32})$$

donde, en nuestro caso, asumiremos que las distribuciones $q(\Gamma)$, $q(\lambda)$ y $q(\nu)$ son degeneradas. Además, asumimos que cada una de las distribuciones $q(\mathbf{d}_k)_{k=1}^K$ es degenerada con $\|\mathbf{d}_k\| = 1$.

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE
APRENDIZAJE DE DICCIONARIOS

A continuación se expone de forma detallada la derivación de estas distribuciones en tres grandes bloques. En primer lugar nos centraremos en obtener las distribuciones relacionadas con la matriz de coeficientes *sparse* dado que necesitaremos tanto su media como su matriz de covarianza para la posterior obtención de las estimaciones de las columnas del diccionario. En último lugar obtendremos la aproximación de la distribución *a posteriori* del ruido.

4.2.1. Estimación de los vectores *sparse* y sus hiperparámetros

Para encontrar una aproximación de la distribución *a posteriori* de \mathbf{X} , aplicamos (II.30) y obtenemos

$$\begin{aligned} \log q(\mathbf{X}) &= \langle \log p(\mathbf{Y}|\mathbf{D}, \mathbf{X}, \beta) + \log p(\mathbf{X}|\mathbf{\Gamma}) \rangle_{\Theta \setminus \mathbf{X}} + \text{const} \\ &= \sum_{q=1}^Q \langle -\frac{\beta}{2} \|\mathbf{y}_q - \mathbf{D}\mathbf{x}_q\|^2 - \frac{1}{2} \mathbf{x}_q^T \mathbf{\Gamma}_q^{-1} \mathbf{x}_q \rangle_{\Theta \setminus \mathbf{X}} + \text{const} \\ &= \sum_{q=1}^Q -\frac{\beta}{2} \|\mathbf{y}_q - \hat{\mathbf{D}}\mathbf{x}_q\|^2 - \frac{1}{2} \mathbf{x}_q^T \hat{\mathbf{\Gamma}}_q^{-1} \mathbf{x}_q + \text{const}, \end{aligned} \quad (\text{II.33})$$

de lo que claramente se desprende que las columnas \mathbf{x}_q de la distribución *a posteriori* aproximada son independientes con

$$\log q(\mathbf{x}_q) = -\frac{\hat{\beta}}{2} \|\mathbf{y}_q - \hat{\mathbf{D}}\mathbf{x}_q\|^2 - \frac{1}{2} \mathbf{x}_q^T \hat{\mathbf{\Gamma}}_q^{-1} \mathbf{x}_q + \text{const}. \quad (\text{II.34})$$

Observamos, por tanto, que $q(\mathbf{x}_q)$ son distribuciones gaussianas

$$q(\mathbf{x}_q) = \mathcal{N}(\mathbf{x}_q | \hat{\mathbf{x}}_q, \mathbf{\Sigma}_{\mathbf{x}_q}) \quad (\text{II.35})$$

con media y matriz de covarianza dadas por

$$\hat{\mathbf{x}}_q = \hat{\beta} \mathbf{\Sigma}_{\mathbf{x}_q} \hat{\mathbf{D}}^T \mathbf{y}_q \quad (\text{II.36})$$

$$\mathbf{\Sigma}_{\mathbf{x}_q} = (\hat{\beta} \hat{\mathbf{D}}^T \hat{\mathbf{D}} + \hat{\mathbf{\Gamma}}_q^{-1})^{-1}. \quad (\text{II.37})$$

Queda ahora obtener estimaciones para los hiperparámetros asociados a \mathbf{X} . Para ello, utilizaremos (II.31) con la esperanza apropiada en cada caso.

Para γ_q tenemos

$$\begin{aligned} \hat{\gamma}_q &= \arg \max_{\gamma_q} \langle \log p(\mathbf{Y}, \Theta) \rangle_{\Theta \setminus \gamma_q} \\ &= \arg \max_{\gamma_q} \langle \log p(\mathbf{x}_q | \gamma_q) + \log p(\gamma_q | \lambda_q) \rangle_{\mathbf{x}_q, \lambda_q} + \text{const}, \end{aligned} \quad (\text{II.38})$$

donde la constante engloba todos los términos que no dependen de γ_q . Utilizando las expresiones dadas en (II.18) y (II.19) obtenemos

$$\begin{aligned} \langle \log p(\mathbf{Y}, \Theta) \rangle_{\Theta \setminus \gamma_q} &= -\frac{1}{2} \sum_{k=1}^K \log \gamma_{kq} - \frac{1}{2} \langle \mathbf{x}_q^T \Gamma_q^{-1} \mathbf{x}_q \rangle_{\mathbf{x}_q} + K \log \hat{\lambda}_q \\ &\quad - \frac{\hat{\lambda}_q}{2} \sum_{k=1}^K \gamma_{kq} + \text{const}, \end{aligned} \quad (\text{II.39})$$

con $\langle \mathbf{x}_q^T \Gamma_q^{-1} \mathbf{x}_q \rangle_{\mathbf{x}_q} = \hat{\mathbf{x}}_q^T \Gamma_q^{-1} \hat{\mathbf{x}}_q + \text{tr}(\Gamma_q^{-1} \Sigma_{\mathbf{x}_q})$. Podemos encontrar el valor óptimo para γ_{kq} derivando e igualando a cero, lo que resulta en

$$\hat{\gamma}_{kq} = -\frac{1}{2\hat{\lambda}_q} + \sqrt{\frac{1}{4\hat{\lambda}_q^2} + \frac{\hat{x}_{kq}^2 + s_{kkq}}{\hat{\lambda}_q}}, \quad (\text{II.40})$$

donde s_{ijq} denota la entrada (i, j) de la matriz $\Sigma_{\mathbf{x}_q}$.

De forma análoga para λ_q tenemos

$$\begin{aligned} \hat{\lambda}_q &= \arg \max_{\lambda_q} \langle \log p(\mathbf{Y}, \Theta) \rangle_{\Theta \setminus \lambda_q} \\ &= \arg \max_{\lambda_q} \langle \log p(\gamma_q | \lambda_q) + \log p(\lambda_q | \nu_q) \rangle_{\gamma_q, \nu_q} + \text{const}. \end{aligned} \quad (\text{II.41})$$

Sustituyendo ahora (II.19) y (II.20) nos queda

$$\begin{aligned} \langle \log p(\mathbf{Y}, \Theta) \rangle_{\Theta \setminus \lambda_q} &= K \log \lambda_q - \frac{\lambda_q}{2} \sum_{k=1}^K \hat{\gamma}_{kq} + \left(\frac{\hat{\nu}_q}{2} - 1 \right) \log \lambda_q \\ &\quad - \frac{\hat{\nu}_q}{2} \lambda_q + \text{const}, \end{aligned} \quad (\text{II.42})$$

cuyo valor óptimo $\hat{\lambda}_q$ viene dado por

$$\hat{\lambda}_q = \frac{\hat{\nu}_q + 2K - 2}{\hat{\nu}_q + \sum_{k=1}^K \hat{\gamma}_{kq}}. \quad (\text{II.43})$$

Por último, para ν_q tenemos

$$\begin{aligned} \hat{\nu}_q &= \arg \max_{\nu_q} \langle \ln p(\mathbf{Y}, \Theta) \rangle_{\Theta \setminus \nu_q} \\ &= \arg \max_{\nu_q} \langle \log p(\lambda_q | \nu_q) + \log p(\nu_q) \rangle_{\lambda_q} + \text{const}, \end{aligned} \quad (\text{II.44})$$

que, tras usar (II.20) produce

$$\frac{\nu_q}{2} \log \frac{\nu_q}{2} - \log \Gamma\left(\frac{\nu_q}{2}\right) + \frac{\nu_q}{2} (\log \hat{\lambda}_q - \hat{\lambda}_q). \quad (\text{II.45})$$

La maximización de esta expresión no admite solución analítica, por lo que resulta necesario recurrir a métodos de optimización numérica para hallar su valor óptimo $\hat{\nu}_q$.

4.2.2. Estimación del diccionario

Para empezar, asumimos que las columnas \mathbf{d}_k son independientes en nuestra aproximación de la distribución *a posteriori*, esto es,

$$q(\mathbf{D}) = \prod_{k=1}^K q(\mathbf{d}_k), \quad (\text{II.46})$$

donde $q(\mathbf{d}_k)$ son distribuciones degeneradas sobre la esfera unidad $\|\mathbf{d}_k\| = 1$.

Centrándonos en un solo \mathbf{d}_k , y aplicando (II.31), tenemos

$$\begin{aligned} \hat{\mathbf{d}}_k &= \arg \min_{\mathbf{d}_k} \langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathbb{F}}^2 \rangle_{\Theta \setminus \mathbf{d}_k} \\ \text{s.a. } &\|\mathbf{d}_k\| = 1. \end{aligned} \quad (\text{II.47})$$

Para aplicar la esperanza indicada en (II.47) resulta conveniente desarrollar el producto $\mathbf{D}\mathbf{X}$ para aislar el término \mathbf{d}_k

$$\begin{aligned} \langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathbb{F}}^2 \rangle_{\Theta \setminus \mathbf{d}_k} &= \langle \|\mathbf{Y} - \sum_{i \neq k} \mathbf{d}_i \mathbf{x}_{i,:} - \mathbf{d}_k \mathbf{x}_{k,:}\|_{\mathbb{F}}^2 \rangle_{\Theta \setminus \mathbf{d}_k} \\ &= \langle \|\mathbf{Y} - \sum_{i \neq k} \mathbf{d}_i \mathbf{x}_{i,:}\|_{\mathbb{F}}^2 \rangle_{\Theta \setminus \mathbf{d}_k} + \langle \text{tr}(\mathbf{x}_{k,:}^T \mathbf{d}_k^T \mathbf{d}_k \mathbf{x}_{k,:}) \rangle_{\Theta \setminus \mathbf{d}_k} \\ &\quad - 2 \langle \text{tr}([\mathbf{Y} - \sum_{i \neq k} \mathbf{d}_i \mathbf{x}_{i,:}]^T \mathbf{d}_k \mathbf{x}_{k,:}) \rangle_{\Theta \setminus \mathbf{d}_k}. \end{aligned} \quad (\text{II.48})$$

Observamos que el primer término no depende de \mathbf{d}_k con lo que no es relevante su cálculo. En cuanto al segundo,

$$\begin{aligned} \langle \text{tr}(\mathbf{x}_{k,:}^T \mathbf{d}_k^T \mathbf{d}_k \mathbf{x}_{k,:}) \rangle_{\Theta \setminus \mathbf{d}_k} &= \langle \text{tr}(\mathbf{x}_{k,:} \mathbf{x}_{k,:}^T \mathbf{d}_k^T \mathbf{d}_k) \rangle_{\Theta \setminus \mathbf{d}_k} = \text{tr}(\langle \mathbf{x}_{k,:} \mathbf{x}_{k,:}^T \rangle_{\mathbf{x}_{k,:}} \mathbf{d}_k^T \mathbf{d}_k) \\ &= (\|\hat{\mathbf{x}}_{k,:}\|^2 + \sum_q s_{kkq}) \mathbf{d}_k^T \mathbf{d}_k, \end{aligned} \quad (\text{II.49})$$

y el término cruzado

$$\begin{aligned} \langle \text{tr}([\mathbf{Y} - \sum_{i \neq k} \mathbf{d}_i \mathbf{x}_{i,:}]^T \mathbf{d}_k \mathbf{x}_{k,:}) \rangle_{\Theta \setminus \mathbf{d}_k} &= \langle \mathbf{x}_{k,:} [\mathbf{Y} - \sum_{i \neq k} \mathbf{d}_i \mathbf{x}_{i,:}]^T \mathbf{d}_k \rangle_{\Theta \setminus \mathbf{d}_k} \\ &= [\hat{\mathbf{x}}_{k,:} \mathbf{Y}^T - \sum_{i \neq k} \langle \mathbf{x}_{k,:} \mathbf{x}_{i,:}^T \rangle_{\mathbf{x}_{i,:}, \mathbf{x}_{k,:}} \hat{\mathbf{d}}_i^T] \mathbf{d}_k \\ &= [\hat{\mathbf{x}}_{k,:} \mathbf{Y}^T - \sum_{i \neq k} (\hat{\mathbf{x}}_{k,:} \hat{\mathbf{x}}_{i,:}^T + \sum_q s_{ikq}) \hat{\mathbf{d}}_i^T] \mathbf{d}_k \quad (\text{II.50}) \\ &= [\hat{\mathbf{x}}_{k,:} \mathbf{R}_k^T - \sum_{i \neq k} \sum_q s_{ikq} \hat{\mathbf{d}}_i^T] \mathbf{d}_k. \end{aligned}$$

Sustituyendo los resultados (II.49) y (II.50) en (II.47) y resolviendo, obtenemos

$$\hat{\mathbf{d}}_k = \frac{\mathbf{R}_k \hat{\mathbf{x}}_{k,:}^T - \sum_{i \neq k} \sum_q s_{ikq} \hat{\mathbf{d}}_i}{\|\mathbf{R}_k \hat{\mathbf{x}}_{k,:}^T - \sum_{i \neq k} \sum_q s_{ikq} \hat{\mathbf{d}}_i\|}. \quad (\text{II.51})$$

4.2.3. Estimación del parámetro de ruido

Quedándonos únicamente con los términos que dependen de β en la distribución conjunta y aplicando, una vez más, (II.30) tenemos

$$\log q(\beta) = \frac{PQ}{2} \log \beta - \frac{\beta}{2} \langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathbb{F}}^2 \rangle_{\Theta \setminus \beta} + (a-1) \log \beta - b\beta + \text{const}, \quad (\text{II.52})$$

donde el término de la esperanza puede desarrollarse como

$$\begin{aligned} \langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathbb{F}}^2 \rangle_{\Theta \setminus \beta} &= \sum_{q=1}^Q \langle \|\mathbf{y}_q - \hat{\mathbf{D}}\mathbf{x}_q\|^2 \rangle_{\mathbf{x}_q} \\ &= \sum_{q=1}^Q [\|\mathbf{y}_q - \hat{\mathbf{D}}\hat{\mathbf{x}}_q\|^2 + \text{tr}(\hat{\mathbf{D}}^T \hat{\mathbf{D}} \Sigma_{\mathbf{x}_q})] \\ &= \|\mathbf{Y} - \hat{\mathbf{D}}\hat{\mathbf{X}}\|_{\mathbb{F}}^2 + \sum_{q=1}^Q \text{tr}(\hat{\mathbf{D}}^T \hat{\mathbf{D}} \Sigma_{\mathbf{x}_q}). \end{aligned} \quad (\text{II.53})$$

De la expresión dada en (II.52) y aplicando el resultado anterior podemos deducir que β sigue una distribución Gamma de media

$$\hat{\beta} = \frac{PQ + 2a}{\|\mathbf{Y} - \hat{\mathbf{D}}\hat{\mathbf{X}}\|_{\mathbb{F}}^2 + \sum_{q=1}^Q \text{tr}(\hat{\mathbf{D}}^T \hat{\mathbf{D}} \Sigma_{\mathbf{x}_q}) + 2b}, \quad (\text{II.54})$$

donde vemos claramente la influencia de la metodología variacional: el ruido no solo depende del término de fidelidad $\|\mathbf{Y} - \hat{\mathbf{D}}\hat{\mathbf{X}}\|_{\mathbb{F}}^2$ como pasaría en una estimación MAP, sino que presenta un término corrector que depende de la incertidumbre relativa a la estimación de \mathbf{X} . Una mayor incertidumbre provocará una mayor varianza de ruido estimado.

Nótese que para obtener el resultado anterior para la estimación de β (II.54) basta con obtener de (II.52) los parámetros de forma y escala inversa de la distribución $q(\beta)$ y, a partir de estos, su media.

4.2.4. Algoritmo variacional completo

El Algoritmo 2 nos presenta un resumen esquemático del proceso de inferencia descrito. Observamos dos fases bien diferenciadas, una primera en el que calculamos los vectores de representación para cada una de nuestras Q señales a partir del estado actual del diccionario y una segunda fase de actualización de este último, sin olvidar la actualización final del parámetro de ruido. Apuntamos aquí que el algoritmo completo tendría que repetir una última vez la fase de representación *sparse* tras la convergencia, lo que se ha decidido no incluir en el algoritmo por motivos de simplicidad.

Vemos ahora una diferencia clara con K-SVD: mientras que en este, el k -ésimo átomo del diccionario se actualiza paralelamente a la fila

Algoritmo 2 Algoritmo BKSVD variacional.

ENTRADA: $\mathbf{D}^{(0)}$, \mathbf{Y} .

```

1:  $t = 0$ 
2: while  $\|\mathbf{Y} - \hat{\mathbf{D}}^{(t)} \hat{\mathbf{X}}^{(t)}\|_{\text{F}}^2 > \text{tol}$  or  $\mathcal{L}(\mathbf{q}(\Theta))$  no converja do
3:   Calcular el soporte de  $\mathbf{X}^{(t+1)}$  a partir de  $\mathbf{D}^{(t)}$ :
4:   for  $q = 1, \dots, Q$  do
5:     Obtener  $\hat{\mathbf{x}}_q^{(t+1)}$  y  $\Sigma_{\mathbf{x}_q}^{(t+1)}$  según (II.36) y (II.37).
6:     Actualizar los hiperparámetros de  $\mathbf{X}$ :  $\gamma_q^{(t+1)}$  según (II.40),  $\lambda_q^{(t+1)}$ 
       con (II.43) y maximizar numéricamente (II.45) para obtener  $\nu_q^{(t+1)}$ .
7:   end for
8:   for  $k = 1, \dots, K$  do
9:     Obtener  $\hat{\mathbf{d}}_k^{(t+1)}$  a partir de (II.51).
10:  end for
11:  Actualizar  $\beta$  según (II.54).
12:   $t = t + 1$ 
13: end while
SALIDA:  $\mathbf{D}$ ,  $\mathbf{X}$ .
```

correspondiente de \mathbf{X} , en el método propuesto ambas actualizaciones son independientes y se realizan, consecuentemente, por separado. Sin embargo, observamos aquí, como ya habíamos adelantado anteriormente, dos claras ventajas de nuestro trabajo frente al K-SVD: no se requiere el conocimiento del número de elementos no nulos de cada una de las columnas de la matriz de coeficientes y además, tampoco se requiere la varianza de la señal, es más, se estima en el mismo marco de trabajo.

Este proceso de inferencia, pese a su validez matemática, presenta dos inconvenientes. Por un lado, la estimación de las columnas \mathbf{x}_q no produce soluciones exactamente *sparse*, sino que simplemente fuerza a que algunos de sus elementos sean muy pequeños, pero no cero. Esto podría no ser un grave problema. No obstante, observamos que el cálculo de la matriz de covarianza $\Sigma_{\mathbf{x}_q}$ expuesto en (II.37) requiere de la inversión de una matriz $K \times K$ para cada una de las Q columnas de \mathbf{X} en cada una de las iteraciones. Esto representa un desafío computacional y de memoria. Vemos a continuación como solucionar estos problemas.

4.3. Inferencia Rápida *Sparse*

Se propone un esquema de inferencia rápida basado en los trabajos [60–62]. Las bases de esta aproximación fueron presentadas originalmente en [60] en el contexto del aprendizaje bayesiano *sparse* y adaptadas posteriormente a la recuperación de señales *sparse* en [61] y [62]. La inferencia de los vectores de representación se convierte en un proceso constructivo para identificar su

soporte, obteniendo los valores de los hiperparámetros en estas posiciones no nulas mediante estimación MAP. Gracias a este método de estimación se logra reducir drásticamente la dimensión efectiva del problema debido a la naturaleza *sparse* de las señales. Mostramos aquí la adaptación de estos conceptos al problema de aprendizaje de diccionarios.

Para comenzar vamos a sustituir la estimación variacional del hiperparámetro γ_q por la maximización directa de

$$\mathcal{L}(\gamma_q) = \log \left[p(\gamma_q | \hat{\lambda}_q) \int p(\mathbf{y}_q | \mathbf{x}_q, \hat{\beta}) p(\mathbf{x}_q | \gamma_q) d\mathbf{x}_q \right], \quad (\text{II.55})$$

donde se han utilizado los valores de \mathbf{D} , β y λ_q estimados según lo expuesto previamente y $p(\mathbf{y}_q | \mathbf{x}_q, \hat{\beta}) = \mathcal{N}(\mathbf{y}_q | \hat{\mathbf{D}}\mathbf{x}_q, \hat{\beta}^{-1}\mathbf{I})$, lo que puede deducirse fácilmente del modelo de observación completo (II.14). Teniendo en cuenta que

$$\int p(\mathbf{y}_q | \mathbf{x}_q, \hat{\beta}) p(\mathbf{x}_q | \gamma_q) d\mathbf{x}_q = \mathcal{N}(\mathbf{y}_q | \mathbf{0}_P, \mathbf{C}_q), \quad (\text{II.56})$$

donde

$$\mathbf{C}_q = \hat{\beta}^{-1}\mathbf{I} + \hat{\mathbf{D}}\mathbf{\Gamma}_q\hat{\mathbf{D}}^T, \quad (\text{II.57})$$

podemos desarrollar (II.55) como

$$\begin{aligned} \mathcal{L}(\gamma_q) &= \log \left[\prod_{k=1}^K \Gamma(\gamma_{kq} | 1, \hat{\lambda}_q/2) \mathcal{N}(\mathbf{y}_q | \mathbf{0}_P, \mathbf{C}_q) \right] \\ &= -\frac{1}{2} \left[\log |\mathbf{C}_q| + \mathbf{y}_q^T \mathbf{C}_q^{-1} \mathbf{y}_q + \hat{\lambda}_q \sum_k \gamma_{kq} \right] + \text{const.} \end{aligned} \quad (\text{II.58})$$

Nótese que una vez calculada $\hat{\gamma}_q$ podemos encontrar fácilmente la distribución *a posteriori* de \mathbf{x}_q . Además, si $\gamma_{kq} = 0$, la *a posteriori* de x_{kq} será degenerada en cero.

Resulta ahora fundamental observar que podemos descomponer la matriz \mathbf{C}_q como

$$\mathbf{C}_q = \hat{\beta}^{-1}\mathbf{I}_P + \sum_{i \neq k} \gamma_{iq} \hat{\mathbf{d}}_i \hat{\mathbf{d}}_i^T + \gamma_{kq} \hat{\mathbf{d}}_k \hat{\mathbf{d}}_k^T =: \tilde{\mathbf{C}}_q + \gamma_{kq} \hat{\mathbf{d}}_k \hat{\mathbf{d}}_k^T, \quad (\text{II.59})$$

donde, claramente, la recién definida matriz $\tilde{\mathbf{C}}_q$ incluye todos los términos que no dependen de γ_{kq} . Utilizando esta descomposición de \mathbf{C}_q y gracias al lema de inversión de Woodbury y la siguiente propiedad del determinante obtenemos

$$\mathbf{C}_q^{-1} = \tilde{\mathbf{C}}_q^{-1} - \frac{\tilde{\mathbf{C}}_q^{-1} \hat{\mathbf{d}}_k \hat{\mathbf{d}}_k^T \tilde{\mathbf{C}}_q^{-1}}{\gamma_{kq}^{-1} + \hat{\mathbf{d}}_k^T \tilde{\mathbf{C}}_q^{-1} \hat{\mathbf{d}}_k}, \quad (\text{II.60})$$

$$|\mathbf{C}_q| = |\tilde{\mathbf{C}}_q| \left(1 + \gamma_{kq} \hat{\mathbf{d}}_k^T \tilde{\mathbf{C}}_q^{-1} \hat{\mathbf{d}}_k \right), \quad (\text{II.61})$$

CAPÍTULO II. *DENOISING E INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

lo que nos permite reescribir (II.58) como

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma}_q) &= -\frac{1}{2} \left[\log |\tilde{\mathbf{C}}_q| + \mathbf{y}_q^T \tilde{\mathbf{C}}_q^{-1} \mathbf{y}_q + \hat{\lambda}_q \sum_{i \neq k} \gamma_{iq} \right] \\ &\quad + \frac{1}{2} \left[\log \frac{1}{1 + \gamma_{kq} s_{kq}} + \frac{h_{kq}^2 \gamma_{kq}}{1 + \gamma_{kq} s_{kq}} - \hat{\lambda}_q \gamma_{kq} \right] \\ &= \mathcal{L}(\tilde{\boldsymbol{\gamma}}_q) + l(\gamma_{kq}),\end{aligned}\tag{II.62}$$

donde s_{kq} y h_{kq} se han definido como

$$s_{kq} = \hat{\mathbf{d}}_k^T \tilde{\mathbf{C}}_q^{-1} \hat{\mathbf{d}}_k \tag{II.63}$$

$$h_{kq} = \hat{\mathbf{d}}_k^T \tilde{\mathbf{C}}_q^{-1} \mathbf{y}_q. \tag{II.64}$$

Nótese que estas cantidades no dependen de γ_{kq} . La función objetivo $\mathcal{L}(\boldsymbol{\gamma}_q)$ queda ahora descompuesta en dos términos, habiendo convenientemente aislado en $l(\gamma_{kq})$ los términos que dependen de γ_{kq} .

Podemos obtener una solución analítica de la maximización de $\mathcal{L}(\boldsymbol{\gamma}_q)$, cuando solo cambia el k -ésimo elemento, fijando el resto de hiperparámetros e igualando su derivada, respecto a γ_{kq} , a cero, esto es,

$$\frac{\partial \mathcal{L}(\boldsymbol{\gamma}_q)}{\partial \gamma_{kq}} = \frac{\partial l(\gamma_{kq})}{\partial \gamma_{kq}} = 0, \tag{II.65}$$

de lo que obtenemos un único máximo en

$$\hat{\gamma}_{kq} = \begin{cases} \frac{s_{kq} + 2\hat{\lambda}_q + \sqrt{(s_{kq} + 2\hat{\lambda}_q)^2 - 4\hat{\lambda}_q(s_{kq} - h_{kq}^2 + \hat{\lambda}_q)}}{2\hat{\lambda}_q s_{kq}}, & h_{kq}^2 - s_{kq} \geq \hat{\lambda}_q \\ 0, & \text{en otro caso.} \end{cases} \tag{II.66}$$

Resulta crucial en este punto realizar todos los cálculos eficientemente, específicamente, el cálculo de las cantidades s_{kq} y h_{kq} para $k = 1, \dots, K$, así como de $\hat{\mathbf{x}}_q$ y $\boldsymbol{\Sigma}_{\mathbf{x}_q}$. Para la actualización del primer par de variables resulta más sencillo definir y actualizar

$$S_k := \hat{\mathbf{d}}_k^T \mathbf{C}_q^{-1} \hat{\mathbf{d}}_k \tag{II.67}$$

$$H_k := \hat{\mathbf{d}}_k^T \mathbf{C}_q^{-1} \mathbf{y}_q, \tag{II.68}$$

de donde, utilizando (II.60) obtenemos

$$s_{kq} = \frac{S_{kq}}{1 - \gamma_{kq} S_{kq}} \tag{II.69}$$

$$h_{kq} = \frac{H_{kq}}{1 - \gamma_{kq} S_{kq}}. \tag{II.70}$$

Para ilustrar el cálculo eficiente de S_{kq} y H_{kq} recargaremos levemente la notación marcando con una tilde aquellas variables que incluyan solo

las columnas ($\tilde{\mathbf{D}}$), filas ($\tilde{\mathbf{x}}_q$) o ambas ($\tilde{\Sigma}_{\mathbf{x}_q}, \tilde{\Gamma}_q$) correspondientes al soporte actual ($\mathcal{S} = \{k/\gamma_{kq} > 0\}$) de la solución. Nótese que podemos reescribir la matriz \mathbf{C}_q como

$$\mathbf{C}_q = \hat{\beta}^{-1} \mathbf{I} + \sum_{k \in \mathcal{S}} \gamma_{kq} \hat{\mathbf{d}}_k \hat{\mathbf{d}}_k^T + \sum_{k \notin \mathcal{S}} \gamma_{kq}^0 \hat{\mathbf{d}}_k \hat{\mathbf{d}}_k^T = \hat{\beta}^{-1} \mathbf{I} + \tilde{\mathbf{D}} \tilde{\Gamma}_q \tilde{\mathbf{D}}^T, \quad (\text{II.71})$$

y utilizar ahora la identidad de Woodbury para obtener de (II.67) y (II.68)

$$S_k = \hat{\beta} \hat{\mathbf{d}}_k^T \hat{\mathbf{d}}_k - \hat{\beta}^2 \hat{\mathbf{d}}_k^T \tilde{\mathbf{D}} \tilde{\Sigma}_{\mathbf{x}_q} \tilde{\mathbf{D}}^T \hat{\mathbf{d}}_k \quad (\text{II.72})$$

$$H_k = \hat{\beta} \hat{\mathbf{d}}_k^T \mathbf{y}_q - \hat{\beta}^2 \hat{\mathbf{d}}_k^T \tilde{\mathbf{D}} \tilde{\Sigma}_{\mathbf{x}_q} \tilde{\mathbf{D}}^T \mathbf{y}_q. \quad (\text{II.73})$$

Tenemos ahora un método eficiente para decidir cómo actualizamos nuestro modelo: buscamos el γ_{kq} que produzca un mayor incremento de la verosimilitud marginal (II.62) y

- lo añadimos al modelo si el γ_{kq} previo era 0,
- lo eliminamos si γ_{kq} ya estaba en el modelo pero $h_{kq}^2 - s_{kq} < \hat{\lambda}_q$,
- lo reestimamos si ya pertenecía al modelo y $h_{kq}^2 - s_{kq} \geq \hat{\lambda}_q$.

Finalmente, cabe destacar que se pueden actualizar incremental y eficientemente las variables $\tilde{\mathbf{x}}_q$ y $\tilde{\Sigma}_q$ cuando únicamente cambia un γ_{kq} , así como S_{kq} y H_{kq} , véase [60].

Una vez descrita la actualización eficiente de los parámetros podemos establecer el algoritmo completo, expuesto en el Algoritmo 3, al cual nos referiremos como BKSVD (*Bayesian* K-SVD) a partir de este punto.

Obsérvese que en el paso 5 del algoritmo debemos seleccionar un γ_{kq} candidato para actualizar el modelo. Podríamos escoger aleatoriamente un átomo del diccionario actual pero esta estrategia tan poco elaborada nos llevaría a una convergencia subóptima. Consecuentemente, elegiremos el γ_{kq} que produzca un mayor incremento en verosimilitud, acelerando así la convergencia.

Una contribución importante de este algoritmo es la estimación de la precisión del ruido β , la cual se derivó en la sección anterior. En el algoritmo propuesto en [63], la estimación de este parámetro no es válida hasta pasadas un elevado número de iteraciones, dependiendo directamente del número de entradas distintas de cero del vector de representación. Una estimación imprecisa de este parámetro puede entorpecer gravemente el comportamiento global del algoritmo, por ello, el citado método fija este valor. No obstante, en el método que proponemos, β se estima utilizando un conjunto de señales, de las cuales se asume que comparten la varianza de ruido, lo que da lugar a estimaciones precisas de este parámetro, incluso en iteraciones tempranas.

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE
APRENDIZAJE DE DICCIONARIOS

Algoritmo 3 Pseudocódigo del algoritmo BKSVD

ENTRADA: $\mathbf{D}^{(0)}$, \mathbf{Y} .

```

1: Inicializar  $\mathbf{\Gamma}$  y  $\boldsymbol{\lambda}$  a cero.
2: while no converja do
3:   for  $q = 1, \dots, Q$  do
4:     while  $\mathcal{L}(\boldsymbol{\gamma}_q)$  no converja do
5:       Elegir  $k \in \{1, \dots, K\}$  (y, por tanto,  $\gamma_{kq}$ ) que maximize  $\mathcal{L}(\boldsymbol{\gamma}_q)$ 
        usando (II.66).
6:       Añadir, eliminar o corregir  $\gamma_{kq}$  según su valor previo y el signo
        de  $h_{kq}^2 - s_{kq} - \hat{\lambda}_q$ .
7:       Actualizar  $\boldsymbol{\Sigma}_{\mathbf{x}_q}$  y  $\hat{\mathbf{x}}_q$  en base a  $\hat{\gamma}_{kq}$ .
8:       Actualizar  $s_{kq}$  y  $h_{kq}$  según (II.69)-(II.73).
9:       Obtener  $\hat{\lambda}_j$  con (II.43) y  $\hat{v}_j$  a partir de (II.45).
10:    end while
11:  end for
12:  for  $k = 1, \dots, K$  do
13:    Actualizar  $\mathbf{d}_k$  según (II.51).
14:  end for
15:  Actualizar  $\hat{\boldsymbol{\beta}}$ 
16: end while

```

SALIDA: $\hat{\mathbf{D}}$, $\hat{\mathbf{\Gamma}}$, las distribuciones *a posteriori* aproximadas $q(\mathbf{x}_q), \forall q$.

Compararemos ahora este algoritmo con el K-SVD. Indicar que podemos encontrar una relación del método rápido aquí expuesto con *Relevance Vector Machine* en [61]. En primer lugar, recordemos que en el K-SVD, ver Sección 3, se ha de fijar previamente el número T de componentes no nulas. En el BKSVD podríamos, bien actualizar $\boldsymbol{\gamma}_q$ hasta converger y seleccionar solo los T valores mayores, bien proceder de forma constructiva hasta haber incorporado los T elementos. No obstante, y como se ha descrito más arriba, nuestro algoritmo decide automáticamente el número de elementos a incorporar.

Por otro lado, existe una diferencia significativa en las iteraciones de los algoritmos. En K-SVD, para actualizar el k -ésimo átomo del diccionario, se selecciona, mediante el algoritmo OMP, las componen diferentes a cero de la fila $\mathbf{x}_{k,\cdot}$. Estas componentes se corresponden con las señales que utilizarán este átomo en su representación. Equivalentemente, si la q -ésima entrada es seleccionada, su correspondiente γ_{kq} será mayor que cero en nuestra formulación rápida. Una vez seleccionado este soporte, K-SVD utiliza la descomposición en valores singulares de la matriz residual truncada $\tilde{\mathbf{R}}_k$, esto es, la matriz formada por las columnas de \mathbf{R}_k correspondientes al soporte determinado por OMP o, análogamente, aquellas con γ_{kq} mayor que cero. A partir del valor singular máximo y sus

correspondientes vectores singulares, derecho e izquierdo, obtenemos las actualizaciones para $\hat{\mathbf{d}}_k$ y las entradas no nulas de $\hat{\mathbf{x}}_{k,:}$. En cambio, BKSVD no solo tiene en cuenta el valor de \mathbf{R}_k para actualizar $\hat{\mathbf{d}}_k$ (II.51), sino que también hace partícipe a la incertidumbre en las estimaciones de \mathbf{X} (s_{ikq}). Por último, K-SVD también actualiza el soporte de $\mathbf{x}_{k,:}$ a partir de la SVD de $\tilde{\mathbf{R}}_k$.

4.3.1. Inferencia subóptima

El algoritmo presentado puede resultar muy costoso, especialmente en el caso de tener un elevado número Q de señales para procesar. El cuello de botella del algoritmo está en la fase de representación *sparse* tanto en la fase reconstrucción como, sobre todo, en la de aprendizaje. En esta, debemos añadir, reestimar o eliminar qué elementos conforman el soporte, dado el diccionario actual, de cada una de las señales a representar.

Para acelerar la convergencia global de nuestro algoritmo de aprendizaje proponemos un algoritmo subóptimo (*greedy*) el cual solo permite la adición de elementos al soporte. Concretamente, se va incrementando el soporte del modelo hasta que la verosimilitud se ve maximizada por la reestimación o eliminación de alguno de los elementos previamente incluidos. Ante esta eventualidad el algoritmo de obtención de la representación *sparse* finaliza.

Comparemos este procedimiento subóptimo alternativo con el algoritmo matemáticamente derivado en la sección anterior para su validación experimental. Para ello diseñamos un ejemplo sintético en el que generamos un diccionario $\mathbf{D} \in \mathbb{R}^{64 \times 150}$ y $Q = 1500$ vectores *sparse* $\mathbf{x}_q \in \mathbb{R}^{150}$ con diferente número de componentes distintas a cero. Las señales correspondientes $\mathbf{y}_q \in \mathbb{R}^{64}$ se calculan siguiendo el modelo de representación sin añadir ruido $\mathbf{y}_q = \mathbf{D}\mathbf{x}_q$. La tabla 1 muestra para diferente número de componentes diferentes a cero s (*sparsity*) el porcentaje de señales para las que el soporte coincide al menos en el 80% de los átomos seleccionados. Como podemos ver, ambos algoritmos tienen un comportamiento similar para valores pequeños de s , es decir, cuando las señales son muy *sparse*.

El segundo experimento, realizado en un Intel Xeon[®] CPU E5-4640 @ 2.40 GHz, tiene que ver con el tiempo de ejecución. La fig. 1 muestra los tiempos de cómputo para diferentes tamaños del diccionario, concretamente $K = kP$ con $k = \{2, 3, 4\}$ y $P = 64$. El número Q de señales corresponde al número total de bloques 8×8 extraídos, con solapamiento máximo, de imágenes de tamaños 128×128 , 256×256 y 512×512 . De la figura se desprende que el algoritmo subóptimo presenta un ahorro significativo, sobre todo cuando el número de señales es elevado, caso habitual cuando trabajamos con imágenes. Por otro lado, también se observa que el número de átomos K del diccionario tiene un efecto mínimo en el tiempo de cómputo.

CAPÍTULO II. *DENOISING E INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

Cuadro 1: Comparativa de acierto de soporte del algoritmo BKSVD y su versión rápida subóptima para diferentes porcentajes de componentes no nulas

s (%)	Óptima	<i>Greedy</i>
3	100.00	100.00
6	100.00	99.87
10	99.67	98.50
13	92.00	70.00

Todos los experimentos presentados en el apartado 4.5 han sido realizados utilizando esta revisión más rápida del algoritmo de inferencia.

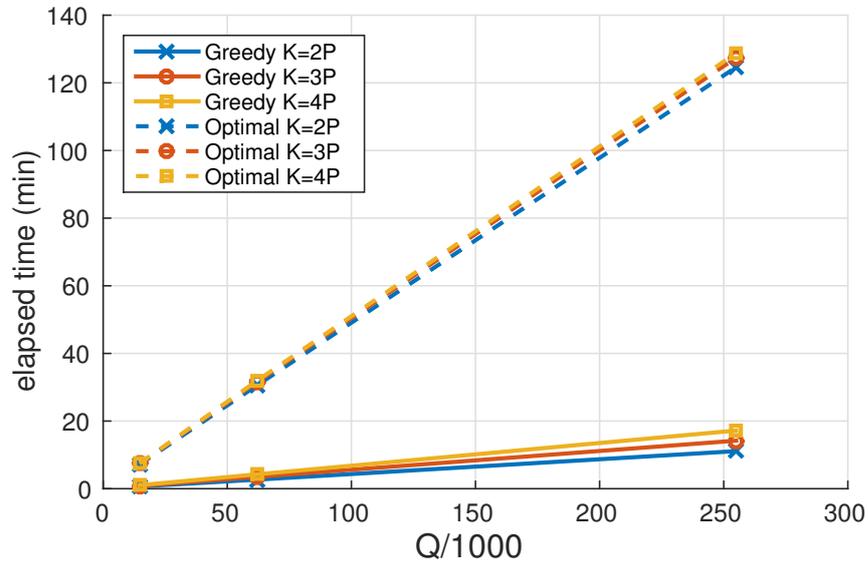


Figura 1: Tiempo de cómputo para la representación *sparse* para el algoritmo óptimo y su versión *greedy*.

4.4. Influencia del tamaño del diccionario

Acabamos de comprobar que el tamaño del diccionario tiene un pequeño impacto sobre el tiempo de ejecución del algoritmo. La fig. 2 muestra este efecto de manera más detallada, así como la influencia del mismo parámetro K sobre el error de reconstrucción MSE (*Mean Squared Error*) y el número medio de átomos usados en las representaciones. Para este conjunto de experimentos se ha usado la imagen *Lena* de tamaño 256×256 extrayendo bloques 8×8 y se han entrenado diferentes diccionarios con $K = kP$ y $k = 1, 1.5, \dots, 8$. Nótese que para el caso $k = 1$

el diccionario no es sobrecompleto, sino una base. El error de reconstrucción se ha calculado como $\|\mathbf{Y} - \mathbf{DX}\|_F/PQ$.

En primer lugar podemos apreciar una tendencia lineal en el tiempo de ejecución, lo que implica que el incremento del tamaño del diccionario no tiene un impacto drástico en el tiempo de cómputo. Por otro lado, y como era de esperar, cuanto mayor es el número de átomos del diccionario, menor es el error de reconstrucción para un mismo número de componentes no nulas. Esto es debido a la mayor diversidad de átomos. Por último, el subpanel inferior muestra el número medio de átomos utilizados en las representaciones. Se observa que al incrementar el tamaño del diccionario se obtienen representaciones más *sparse*. Sin embargo, encontramos una zona más o menos plana para $k \in [2.5, 4]$ en la que apenas decrece el uso de átomos. Para valores de k mayores que 4.5 las representaciones son cada vez más *sparse* pero a expensas de un elevado tiempo de cómputo debido a la presencia innecesaria de átomos que estarán altamente correlados.

A la vista de estos resultados y dado que trabajaremos con el mismo tamaño de bloque en el siguiente apartado, podemos concluir que el valor $k = 4$ proporciona un buen compromiso.

Finalmente, nótese que los valores de uso de átomos son superiores aquí que los que veremos posteriormente en los experimentos de *denoising*. Esto se debe a que la presencia de ruido favorece representaciones más *sparse* ya que incrementa el grado de libertad de la solución. Por lo contrario, una pequeña varianza del ruido implicará un mayor uso de átomos.

4.5. Experimentos

Veamos ahora la aplicación del algoritmo presentado a la eliminación de ruido en imágenes e *inpainting*. Compararemos los resultados obtenidos por BKSVD con métodos del estado del arte: K-SVD, BPFA [37] y SnS [48]. BKSVD, al igual que BPFA y SnS son capaces de estimar la varianza de ruido a partir de los datos corruptos. El valor de esta variable para el K-SVD se especificará en cada uno de los subapartados siguientes. Para garantizar una justa comparación se utilizarán únicamente operaciones básicas del aprendizaje de diccionarios sin pre/postprocesamiento adicional.

Los experimentos presentados se han realizado con imágenes en blanco y negro, concretamente *Barbara*, *Boat*, *Lena* y *Peppers*, ya que con ellas es posible realizar una comparación cuantitativa de los resultados. Su uso sobre imágenes milimétricas como estrategia para la mejora en la clasificación se realizará, como ya hemos comentado, con posterioridad. Tanto para *inpainting* como para *denoising* se han aprendido 256 átomos a partir de bloques 8×8 utilizando un solapamiento máximo. Tras el aprendizaje del diccionario y la fase de representación final, los bloques «restaurados» se recomponen en la imagen final promediando

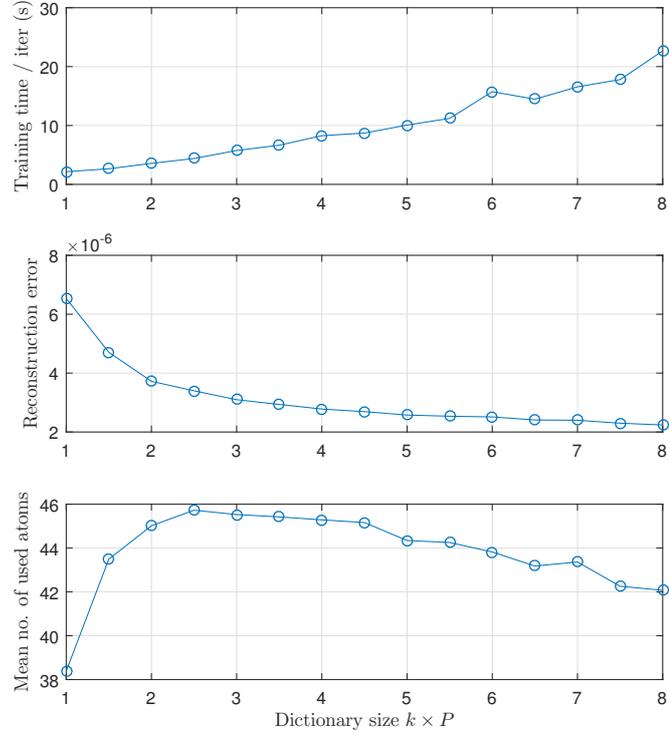


Figura 2: Influencia de K sobre el tiempo de cómputo, MSE y uso medio de átomos con $P = 64$.

convenientemente en las zonas de solapamiento. Nótese que \mathbf{D} se ha inicializado con diccionarios DCT sobrecompletos y que se ha utilizado $a = b = 1$ para los hiperparámetros relativos a la *a priori* sobre la precisión del ruido.

4.5.1. *Denoising*

Para este experimento contaminamos las imágenes 256×256 con ruido aditivo gaussiano de media nula con desviaciones típicas $\sqrt{1/\beta} \in \{5, 10, 15, 20, 25, 50\}$ y extraemos los bloques 8×8 ruidosos a partir de los cuales aprenderemos los diccionarios correspondientes.

La tabla 5 muestra una comparativa cuantitativa de los métodos utilizados. Para el algoritmo K-SVD mostramos dos resultados: uno que se obtiene proporcionándole la varianza real de ruido de la imagen y otro con la estimada por nuestro algoritmo para una comparación justa, dado que este parámetro no siempre está disponible. Ambos resultados son muy

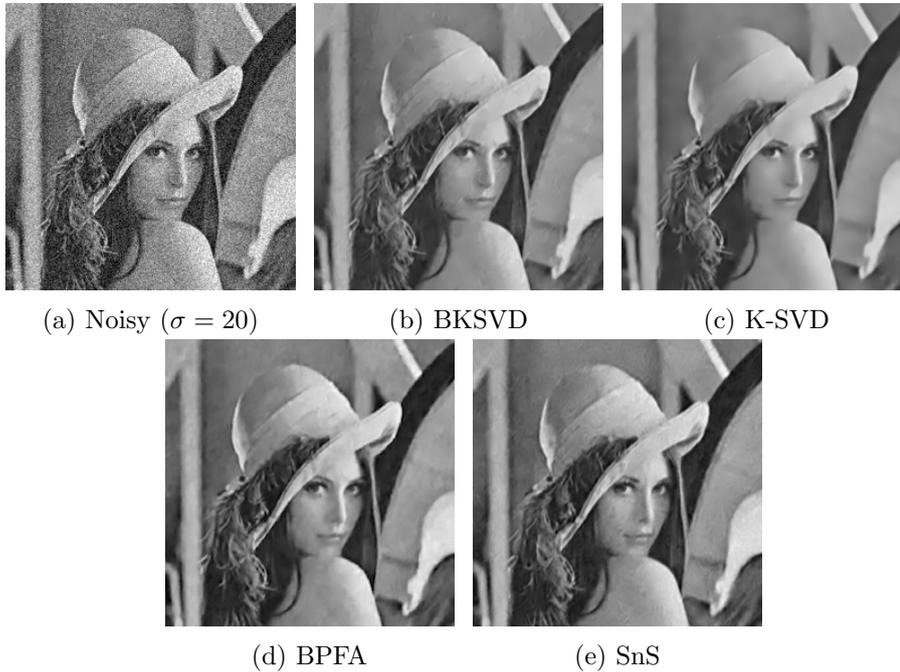


Figura 3: Comparación de resultados para *denoising* de of BKSVD, K-SVD, BPFA [37] y Spike and Slab (SnS) model [48].

similares dado que la estimación de ruido proporcionada por nuestro algoritmo es muy precisa (ver tabla 2).

Cuadro 2: σ estimada para la imagen *Lena*.

σ	5	10	15	20	25	50
$\hat{\sigma}$	5.54	10.39	15.01	19.55	24.46	46.80

El método propuesto produce resultados similares o mejores que el K-SVD en 20 de los 24 experimentos, siendo capaz, como decíamos, de estimar la varianza de ruido. Nótese también el K-SVD es muy sensible a imprecisiones en este parámetro. Al contrario, el método propuesto estima automáticamente este parámetro y es más robusto a niveles altos de ruido puesto que tiene en cuenta las incertidumbres relativas a los valores estimados. La tabla 5 muestra, además, el porcentaje promedio de componentes no nulas en $\hat{\mathbf{X}}$. Ambas técnicas obtienen valores de PSNR y SSIM similares, no obstante, el método propuesto siempre obtiene representaciones más *sparse*, lo que indica que el diccionario aprendido tiene una mayor capacidad de representación. El método SnS en [48] es comparable directamente a nuestro método por su capacidad de estimar el

CAPÍTULO II. *DENOISING E INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS



Figura 4: *Inpainting* con diccionario aprendido a partir de una base de datos de imágenes sin ruido.



Figura 5: Comparación de resultados de *inpainting* para $r = 75\%$ de BKSVD, K-SVD [64], BPFA [37] y Spike and Slab (SnS) [48]. El diccionario ha sido aprendido de la propia imagen corrupta.

nivel de ruido obteniendo resultados competitivos. Sin embargo, nuestro método es superior en términos de PSNR y SSIM. En cambio, los resultados obtenidos por BPFA [37] son muy similares. Las diferencias de PSNR y SSIM son, en la mayoría de los casos, ínfimas. Veamos, pues, como métrica de comparación adicional sobre los diccionarios aprendidos, la coherencia mutua del diccionario, definida como

$$\mu(\mathbf{D}) = \max_{i \neq j} \frac{|\mathbf{d}_i^T \mathbf{d}_j|}{\|\mathbf{d}_i\| \|\mathbf{d}_j\|}.$$

La coherencia mutua proporciona una medida de cuan diferentes, incorrelados, son los átomos de un diccionario, tomando valores entre 0 y 1, siendo deseable que los átomos sean lo más incorrelados posibles, lo que equivale a un valor de coherencia pequeño. Dos átomos con alta correlación puede confundir a los algoritmos de representación *sparse* [65]. La tabla 3 muestra la coherencia mutua de los diccionarios aprendidos por nuestra técnica y [37] a partir de 5000 bloques 8×8 extraídos de un conjunto de imágenes naturales. Puede apreciarse que el BKSVD es capaz de obtener

consistentemente valores de coherencia mutua menores y, por tanto, mejores, para diferentes tamaños de diccionario. Diseñamos un pequeño experimento para ahondar en este aspecto. Concretamente, aprendemos sendos diccionarios (de 256 átomos) a partir de un conjunto de 94000 bloques sin ruido. Posteriormente, para garantizar una comparación imparcial, utilizamos el algoritmo OMP para obtener las representaciones *sparse* correspondientes con el mismo número de componentes no nulas para ambos casos. Como puede comprobarse en la tabla 4, el diccionario aprendido por el BKSVD no solo tiene menor coherencia mutua, sino que además presenta una mayor capacidad de representación *sparse* de las señales, traducida en PSNR superiores para distintos niveles de ruido.

Cuadro 3: Coherencia mutua para diferentes tamaños de diccionario.

K	64	96	128	160	192	224	256
BKSVD	0.86	0.87	0.91	0.94	0.94	0.94	0.95
BPFA [37]	0.96	0.97	0.97	0.97	0.98	0.98	0.98

Cuadro 4: Comparativa de PSNR en *denoising* para diccionarios aprendidos con el algoritmo propuesto y [37].

σ	5	10	15	20	25	50
BKSVD	33.03	32.20	31.15	29.83	28.57	23.88
BPFA [37]	28.48	28.29	28.00	27.52	26.93	23.70

La fig. 3 muestra un ejemplo visual de los resultados obtenidos para la imagen *Lena*. Puede apreciarse que BKSVD conserva mejor las altas frecuencias que K-SVD, el cual produce una imagen más plana y borrosa. En cuanto a BFPA y SnS, puede observarse que estos generan artefactos de alta frecuencia más visibles.

4.5.2. *Inpainting*

Las representaciones *sparse* también son capaces de tratar información con datos perdidos. Las adaptaciones necesarias pueden realizarse a dos niveles: solo al estimar las representaciones *sparse* o también en la fase misma del aprendizaje del diccionario si no poseemos datos completos a partir de los cuales aprenderlo, como vimos en la introducción del capítulo. Veremos dos experimentos mostrando estas adaptaciones.

En primer lugar, aprendemos un diccionario de 256 átomos a partir de un conjunto de 23 imágenes limpias (sin ruido ni datos perdidos) siguiendo la aproximación en [27]. Para ello, de cada imagen seleccionamos los 4096 bloques con mayor varianza. Para la fase de test, eliminamos el 25 %, 50 %

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

Cuadro 5: Comparación del BKSVD (σ real y estimada por BKSVD) con K-SVD, BPPA (*batch* y *online*) y SnS. PSNR (dB), SSIM y número medio de componentes no nulas (nz). Para SnS no mostramos este último valor dado que el diccionario aprendido no es sobrecompleto.

σ	Barbara			Boats			Lena			Peppers			
	PSNR	SSIM	nz										
5	38.02	0.97	0.05	36.71	0.96	0.03	38.90	0.97	0.04	38.98	0.97	0.04	BKSVD
	38.00	0.97	0.08	36.70	0.96	0.08	38.92	0.97	0.08	38.91	0.97	0.07	K-SVD $\hat{\sigma}$
	38.10	0.97	0.10	36.40	0.96	0.12	38.93	0.97	0.08	38.89	0.97	0.08	K-SVD σ
	35.47	0.94	0.03	36.11	0.96	0.03	37.77	0.97	0.03	38.02	0.97	0.03	BPPA [37]
	35.99	0.95	-	36.44	0.96	-	38.21	0.97	-	38.32	0.97	-	SnS [48]
10	33.97	0.93	0.02	32.80	0.91	0.02	34.96	0.94	0.02	35.27	0.94	0.01	BKSVD
	33.95	0.93	0.04	32.70	0.90	0.04	34.94	0.94	0.04	35.08	0.94	0.03	K-SVD $\hat{\sigma}$
	33.97	0.93	0.06	33.00	0.92	0.04	34.95	0.94	0.04	35.09	0.94	0.02	K-SVD σ
	32.97	0.91	0.02	33.00	0.92	0.02	34.75	0.94	0.02	35.15	0.94	0.02	BPPA [37]
	32.98	0.92	-	32.88	0.92	-	34.59	0.94	-	34.81	0.94	-	SnS [48]
15	31.85	0.90	0.01	31.01	0.87	0.01	32.78	0.91	0.01	33.12	0.92	0.01	BKSVD
	31.82	0.90	0.02	30.90	0.87	0.03	32.73	0.91	0.03	33.00	0.92	0.03	K-SVD $\hat{\sigma}$
	31.84	0.90	0.02	31.00	0.87	0.02	32.70	0.91	0.03	33.04	0.92	0.02	K-SVD σ
	31.43	0.88	0.01	31.06	0.88	0.02	32.71	0.91	0.01	33.10	0.92	0.01	BPPA [37]
	30.87	0.87	-	30.73	0.87	-	32.33	0.90	-	32.63	0.91	-	SnS [48]
20	30.30	0.87	0.01	29.44	0.83	0.01	31.32	0.88	0.01	31.36	0.89	0.01	BKSVD
	30.24	0.87	0.02	29.42	0.83	0.02	31.27	0.88	0.02	31.35	0.89	0.02	K-SVD $\hat{\sigma}$
	30.28	0.87	0.02	29.44	0.83	0.02	31.30	0.88	0.02	31.36	0.89	0.02	K-SVD σ
	30.01	0.85	0.01	29.56	0.84	0.01	31.19	0.88	0.01	31.56	0.90	0.01	BPPA [37]
	29.52	0.84	-	29.30	0.83	-	30.76	0.87	-	31.01	0.89	-	SnS [48]
25	29.10	0.84	0.01	28.38	0.80	0.01	29.99	0.86	0.01	30.20	0.88	0.01	BKSVD
	29.10	0.83	0.02	28.38	0.80	0.02	29.87	0.86	0.02	30.12	0.87	0.02	K-SVD $\hat{\sigma}$
	29.05	0.83	0.01	28.36	0.80	0.01	30.00	0.86	0.01	30.15	0.87	0.01	K-SVD σ
	29.01	0.83	0.01	28.63	0.81	0.01	30.12	0.86	0.01	30.27	0.88	0.01	BPPA [37]
	28.49	0.81	-	28.13	0.79	-	29.56	0.84	-	29.82	0.86	-	SnS [48]
50	25.60	0.72	0.01	25.10	0.67	0.01	26.31	0.73	0.01	26.41	0.77	0.01	BKSVD
	25.51	0.71	0.01	25.01	0.66	0.01	26.16	0.72	0.01	26.19	0.76	0.01	K-SVD $\hat{\sigma}$
	25.43	0.71	0.01	24.92	0.67	0.01	26.23	0.72	0.01	26.32	0.77	0.01	K-SVD σ
	25.78	0.72	0.01	25.26	0.68	0.01	26.43	0.76	0.01	26.71	0.80	0.01	BPPA [37]
	25.33	0.70	-	24.90	0.66	-	26.13	0.73	-	26.16	0.77	-	SnS [48]

y 75 % de los píxeles de imágenes 512×512 que no estaban en el conjunto de entrenamiento sin añadir ruido.

Como podemos ver en la tabla 6, los resultados obtenidos por el método propuesto son mejores que los del K-SVD, lo que sugiere una mejor capacidad de representación del diccionario aprendido por nuestro método. Para porcentajes r elevados de datos perdidos ambos algoritmos producen resultados similares, a pesar de que BKSVD sigue funcionando ligeramente mejor. La figura 4 muestra, para $r = 75\%$ un ejemplo de las restauraciones. Se aprecia una mejora notable en la calidad visual de la imagen recuperada por BKSVD en contraste a la imagen más suavizada producida por el K-SVD.

En segundo lugar, y para realizar una comparativa con los demás métodos utilizados en el anterior experimento, adaptamos el problema para que sea capaz de aprender el diccionario a partir de la propia imagen corrupta (con datos perdidos). Para ello introducimos la matriz \mathbf{M}_q en el

Cuadro 6: Comparación de PSNR y SSIM en *inpainting* con diccionario aprendido de datos limpios para diferentes porcentajes r de datos perdidos.

r (%)	Barbara		Boat		Lena		Peppers		
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
25	39.18	0.99	38.54	0.97	41.54	0.98	39.49	0.97	BKSVD
	38.80	0.99	36.79	0.96	40.58	0.97	37.17	0.97	K-SVD [27]
50	33.15	0.96	32.65	0.92	36.14	0.95	34.73	0.93	BKSVD
	32.59	0.95	32.25	0.90	36.02	0.94	33.82	0.89	K-SVD [27]
75	27.36	0.86	27.39	0.80	30.37	0.88	29.29	0.85	BKSVD
	25.03	0.82	26.93	0.76	29.23	0.85	28.70	0.83	K-SVD [27]

modelo de observación, como se indica al principio del capítulo. Esto tiene un mayor efecto en la estimación de los átomos del diccionario, pero no tanto en la fase de representación *sparse*, en la que simplemente bastará con utilizar, para cada señal, las filas de \mathbf{D} correspondientes a los datos observados de \mathbf{y}_q . La tabla 7 muestra una comparativa del BKSVD con los métodos de *inpainting* en [64], [37] and [48]. Estos métodos también aprenden el diccionario de la propia imagen corrupta. Por supuesto, se han utilizado los mismos patrones de píxeles perdidos para todos los métodos. BKSVD obtiene unos resultados similares a [64], [37] y [48] en términos de PSNR y SSIM. Sin embargo, la calidad visual es algo superior, con menor número de artefactos apreciables alrededor de las fronteras, como puede observarse en la fig. 5. En esta, BFPA y SnS tienen mayor PSNR pero una peor apariencia visual. Véanse también los artefactos de bloque producidos por la reconstrucción del K-SVD.

4.5.3. Comparación de tiempos de ejecución

Para concluir mostramos los tiempos de ejecución de los diferentes algoritmos, todos ellos ejecutados en un procesador AMD Opteron™ CPU @ 2.30 GHz. La tabla 8 muestra los tiempos de cómputo para la eliminación de ruido en la imagen «Lena» con $\sigma = 20$ para diferentes tamaños de imagen y diccionario.

Observamos que, claramente, K-SVD es el algoritmo más rápido, pero no incluye el tiempo de estimación del número de componentes distintas de cero para cada una de las representaciones *sparse*. De entre los métodos bayesianos BFPA es el más rápido, seguido del método propuesto. Sin embargo, nótese que, según lo expuesto previamente, BKSVD estima un parámetro λ_{kq} para cada componente de la matriz \mathbf{X} , mientras que BFPA únicamente estima un vector (K parámetros) que comparten todos los \mathbf{x}_q .

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

Cuadro 7: Comparación de PSNR y SSIM en *inpainting* con diccionario aprendido de datos corruptos para diferentes porcentajes r de datos perdidos.

r (%)	Barbara		Boat		Lena		Peppers		
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
25	38.53	0.98	36.28	0.97	38.85	0.97	39.04	0.97	BKSVD
	38.85	0.98	36.83	0.96	39.14	0.97	40.14	0.97	K-SVD [64]
	34.77	0.94	35.19	0.96	37.87	0.97	38.04	0.97	BPFA [37]
	36.34	0.96	35.24	0.95	38.52	0.98	39.03	0.98	SnS [48]
50	33.67	0.95	30.97	0.92	33.73	0.95	33.92	0.96	BKSVD
	32.58	0.91	30.73	0.88	33.09	0.90	33.48	0.91	K-SVD [64]
	33.33	0.94	32.24	0.93	34.43	0.96	35.10	0.96	BPFA [37]
	32.65	0.93	31.21	0.91	33.89	0.95	34.69	0.96	SnS [48]
75	27.99	0.86	25.88	0.79	28.18	0.87	27.84	0.88	BKSVD
	23.48	0.70	22.84	0.62	24.13	0.69	23.16	0.70	K-SVD [64]
	28.24	0.85	26.91	0.83	29.46	0.91	29.36	0.92	BPFA [37]
	27.57	0.82	26.45	0.79	28.93	0.89	28.79	0.90	SnS [48]

Cuadro 8: Tiempo de ejecución (en segundos) de los distintos algoritmos para diferentes tamaños de imagen u diccionario.

Tam. imagen	K	BKSVD	BPFA	K-SVD	SnS
256x256	256	2.843	1.054	165	9.960
	320	3.603	1.468	171	15.355
128x128	256	774	281	62	2.451
	320	894	407	71	3.893

5. Aprendizaje de diccionarios doblemente *sparse*

Los diccionarios aprendidos por las técnicas vistas previamente resultan altamente estructurados [66], lo que sugiere que los mismos diccionarios podrían tener una representación *sparse* sobre un diccionario primario básico $\Psi \in \mathbb{R}^{P \times M}$, como podría ser un diccionario DCT sobrecompleto. Esta observación da lugar a los métodos de aprendizaje de diccionarios *sparse*, llamados también «doblemente» *sparse*, dado que no solo proporcionan una representación *sparse* de las señales, sino que lo hacen mediante un diccionario que también tiene una representación *sparse*, esto es, $\mathbf{D} = \Psi \mathbf{A}$, donde $\mathbf{A} \in \mathbb{R}^{M \times K}$ y cada una de sus columnas \mathbf{a}_k será *sparse*.

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{X}} \|\mathbf{Y} - \Psi \mathbf{A} \mathbf{X}\|_F^2 & \quad (\text{II.74}) \\ \text{s.t.} \quad \begin{cases} \|\mathbf{x}_q\|_0 \leq T, \forall q, \\ \|\mathbf{a}_k\|_0 \leq S, \forall k. \end{cases} \end{aligned}$$

La fortaleza de este modelo sobre el tradicional radica en que la restricción sobre la representación del diccionario actúa como regularizador, permitiendo la selección de características de Ψ , ayudando a reducir el sobreaprendizaje [67].

Entre las aplicaciones de este método podemos encontrar la eliminación de ruido [68], superresolución [69], compresión en imágenes de teledetección [70], detección de vóxeles activos en fMRI [71] y reconocimiento facial [72].

Una vez más, el problema de aprendizaje es NP-completo por lo que se han propuesto técnicas basadas en métodos *greedy* o la común relajación ℓ_1 para salvar este obstáculo. K-SVDS [66] es una técnica basada en el K-SVD estándar para resolver el problema en (II.74). Este método alterna la minimización sobre \mathbf{A} y \mathbf{X} . En primer lugar, se utiliza OMP para determinar el soporte de las señales de la matriz de coeficientes, teniendo una expresión analítica cerrada para su actualización. Después, la minimización sobre \mathbf{A} resulta equivalente a un problema de codificación *sparse*. No obstante, como sucedía con K-SVD, un potencial inconveniente es la necesidad de conocer el verdadero nivel de ruido, el cual rara vez es conocido en experimentos reales.

5.1. Modelado bayesiano

La relajación mediante norma ℓ_1 del problema de aprendizaje doblemente *sparse* (II.74) admite, de manera natural, un modelado probabilístico jerárquico análogo al visto en la sección anterior (BKSVD) utilizando una distribución normal para el modelo de observación y distribuciones laplacianas tanto para las columnas de \mathbf{X} con las de \mathbf{A} que modelan la norma-1.

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

\mathbf{M}_q	Φ_q	Application
$\mathbf{I}_{P \times P}$	$\mathbf{I}_{P \times P}$	denoising
\mathbf{M}_q	$\mathbf{I}_{P \times P}$	inpainting
$\mathbf{I}_{P \times P}$	Φ_q	c. sensing

Cuadro 9: Matrices \mathbf{M}_q y Φ_q para diferentes aplicaciones.

Diseñamos un modelo de observación versátil que permite, con un única formulación, representar tres problemas clásicos del procesado de imagen: eliminación de ruido, *inpainting* y *compressive sensing*

$$p(\mathbf{Y}|\beta, \mathbf{A}, \mathbf{X}) \propto \beta^{\frac{\sum_q N_q}{2}} e^{-\frac{\beta}{2} \sum_{q=1}^Q \|\mathbf{M}_q(\mathbf{y}_q - \Psi_q \mathbf{A} \mathbf{x}_q)\|^2}, \quad (\text{II.75})$$

donde \mathbf{M}_q es una matriz diagonal cuyo elemento (n, n) -ésimo valdrá 1 si el valor correspondiente y_{nq} ha sido observado y 0 en caso contrario; $N_q = \|\text{diag}(\mathbf{M}_q)\|_0$ y $\Psi_q = \Phi_q \Psi$, donde Ψ denota un diccionario sobrecompleto predefinido. Nótese, en primer lugar, que las matrices de selección \mathbf{M}_q solo intervendrán ante la presencia de datos perdidos, como en el caso del *inpainting*. Para otras aplicaciones en las que no hay datos no observados, todas las entradas de las diagonales de estas matrices serán iguales a la unidad, luego $\mathbf{M}_q = \mathbf{I}, \forall q$. En segundo lugar, la formulación permite el uso de una matriz de muestreo diferente Φ_q para cada señal en caso del problema de CS. Esta matriz no se utilizará para *denoising* o *inpainting*. La tabla 9 muestra las posibles elecciones de estas matrices para las diferentes aplicaciones.

La relajación mediante norma ℓ_1 de las restricciones sobre las columnas de \mathbf{A} y \mathbf{X} en (II.74) pueden ser modeladas mediante distribuciones laplacianas. Como ya hemos visto, dado que esta distribución no es conjugada con el modelo de observación gaussiano, hace la inferencia intratable desde un punto de vista analítico por lo que, una vez más, emplearemos un modelo jerárquico basado en la mezcla infinita de gaussianas, equivalente a esta distribución.

Comenzamos imponiendo una distribución normal de media nula con varianza desconocida sobre las columnas de \mathbf{A}

$$p(\mathbf{a}_k | \omega_k) = \prod_{m=1}^M \mathcal{N}(a_{mk} | 0, \omega_{mk}) = \mathcal{N}(\mathbf{a}_k | \mathbf{0}_M, \mathbf{\Omega}_k), \quad (\text{II.76})$$

donde $\mathbf{\Omega}_k = \text{diag}(\omega_k)$, siendo ω_{mk} la varianza asociada al elemento m -ésimo de \mathbf{a}_k . Estas varianzas comparten una misma hiperdistribución *a priori*

$$p(\omega_k | \rho_k) = \prod_{m=1}^M \Gamma(\omega_{mk} | 1, \rho_k/2). \quad (\text{II.77})$$

Nótese que la distribución de \mathbf{a}_k dado ρ_k puede ser obtenida por marginalización

$$\begin{aligned} p(\mathbf{a}_k|\rho_k) &= \int p(\mathbf{a}_k|\boldsymbol{\omega}_k)p(\boldsymbol{\omega}_k|\rho_k)d\boldsymbol{\omega}_k \\ &= \frac{\rho_k^{M/2}}{2^M} \exp(-\sqrt{\rho_k}\|\mathbf{a}_k\|_1). \end{aligned} \quad (\text{II.78})$$

El modelo puede ser completado introduciendo una distribución *a priori* sobre el hiperparámetro ρ_k , como vimos en 4. Omitimos aquí esta extensión puesto que concluimos que la adición de esta distribución no introducía información significativa al proceso de inferencia.

Imponemos una jerarquía análoga a la propuesta en (II.76), (II.77) sobre las columnas de \mathbf{X}

$$p(\mathbf{x}_q|\boldsymbol{\gamma}_q) = \prod_{k=1}^K \mathcal{N}(x_{kq}|0, \gamma_{kq}) = \mathcal{N}(\mathbf{x}_q|\mathbf{0}_K, \boldsymbol{\Gamma}_q) \quad (\text{II.79})$$

$$p(\boldsymbol{\gamma}_q|\lambda_q) = \prod_{k=1}^K \Gamma(\gamma_{kq}|1, \lambda_q/2) \quad (\text{II.80})$$

con $\boldsymbol{\Gamma}_q = \text{diag}(\boldsymbol{\gamma}_q)$.

Finalmente, para completar el modelo, asumimos una distribución gamma sobre la precisión de ruido

$$p(\beta) = \Gamma(\beta|a_\beta, b_\beta) \propto \beta^{a_\beta-1} \exp(-b_\beta\beta), \quad (\text{II.81})$$

con a_β y b_β , dos escalares positivos que representan el parámetro de forma y escala inversa respectivamente.

Con todos estos elementos podemos expresar la distribución conjunta, que condensa todo nuestro conocimiento acerca del problema, como

$$p(\mathbf{Y}, \boldsymbol{\Theta}) = p(\mathbf{Y}|\beta, \mathbf{A}, \mathbf{X})p(\beta) \left[p(\mathbf{A}|\boldsymbol{\Omega})p(\boldsymbol{\Omega}|\boldsymbol{\rho}) \right] \left[p(\mathbf{X}|\boldsymbol{\Gamma})p(\boldsymbol{\Gamma}|\boldsymbol{\lambda}) \right], \quad (\text{II.82})$$

donde $\boldsymbol{\Theta} = \{\mathbf{A}, \boldsymbol{\Omega}, \boldsymbol{\rho}, \mathbf{X}, \boldsymbol{\Gamma}, \boldsymbol{\lambda}\}$ denota el conjunto de incógnitas, $\boldsymbol{\Omega} = [\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_K]$, $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_Q]$.

5.2. Inferencia variacional bayesiana

Nuestro objetivo será la estimación de la distribución *a posteriori* $p(\boldsymbol{\Theta}|\mathbf{Y})$, la cual no es analíticamente viable dada la intratabilidad de la marginal de \mathbf{Y} . En consecuencia, utilizamos un proceso de inferencia variacional análogo al visto en el apartado 4 considerando una aproximación de la *a posteriori* que factoriza como

$$q(\boldsymbol{\Theta}) = q(\beta) \left[\prod_k q(\mathbf{a}_k) \right] q(\boldsymbol{\Omega})q(\boldsymbol{\rho}) \left[\prod_q q(\mathbf{x}_q) \right] q(\boldsymbol{\Gamma})q(\boldsymbol{\lambda}), \quad (\text{II.83})$$

CAPÍTULO II. *DENOISING E INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

donde las distribuciones *a posteriori* sobre $\mathbf{\Omega}$, $\boldsymbol{\rho}$, $\mathbf{\Gamma}$ y $\boldsymbol{\lambda}$ se asumen degeneradas. Nótese también que se ha asumido independencia entre las columnas \mathbf{a}_k y \mathbf{x}_q de \mathbf{A} y \mathbf{X} .

La *a posteriori* sobre \mathbf{A} puede obtenerse mediante (II.30). Centrándonos en una única columna \mathbf{a}_k , siendo A el conjunto de índices q para los cuales la \mathbf{x}_q satisface $\langle x_{kq}^2 \rangle > 0$, tenemos

$$\begin{aligned} \log q(\mathbf{a}_k) &= -\frac{\beta}{2} \sum_q \langle \|\mathbf{M}_q(\mathbf{y}_q - \mathbf{\Psi}_q \mathbf{A} \mathbf{x}_q)\|^2 \rangle_{\Theta \setminus \mathbf{a}_k} - \frac{1}{2} \mathbf{a}_k^T \mathbf{\Omega}_k^{-1} \mathbf{a}_k + \text{const} \\ &= -\frac{\hat{\beta}}{2} \|\mathbf{t}_k - \mathbf{U}_k \mathbf{a}_k\|^2 - \frac{1}{2} \mathbf{a}_k^T \hat{\mathbf{\Omega}}_k^{-1} \mathbf{a}_k + \text{const}, \end{aligned} \quad (\text{II.84})$$

donde \mathbf{t}_k y \mathbf{U}_k son matrices por bloques cuyos bloques tienen la forma $1/\langle x_{kq}^2 \rangle^{1/2} \mathbf{M}_q(\mathbf{y}_q \langle x_{kq} \rangle - \mathbf{\Psi}_q \sum_{k' \neq k} \hat{\mathbf{a}}_{k'} \langle x_{k'q} x_{kq} \rangle)$ y $\langle x_{kq}^2 \rangle^{1/2} \mathbf{M}_q \mathbf{\Psi}_q$, $q \in A$ respectivamente. lo que resulta en un distribución gaussiana para $q(\mathbf{a}_k)$, con media y matriz de covarianzas dadas por

$$\hat{\mathbf{a}}_k = \hat{\beta} \mathbf{\Sigma}_{\mathbf{a}_k} \mathbf{U}_k^T \mathbf{t}_k \quad (\text{II.85})$$

$$\mathbf{\Sigma}_{\mathbf{a}_k} = (\hat{\beta} \mathbf{U}_k^T \mathbf{U}_k + \mathbf{\Omega}_k^{-1})^{-1}, \quad (\text{II.86})$$

A continuación, podemos obtener las actualizaciones correspondientes a los hiperparámetros relacionados con \mathbf{A} resolviendo la ec. (II.31). Específicamente, para ω_{mk} tenemos que maximizar

$$-\frac{1}{2} \log \omega_{mk} - \frac{1}{2} \frac{\langle a_{mk}^2 \rangle}{\omega_{mk}} - \frac{1}{2} \langle \rho_k \rangle \omega_{mk} + \text{const}. \quad (\text{II.87})$$

Igualando a cero la derivada de esta expresión con respecto a ω_{mk} obtenemos

$$\hat{\omega}_{mk} = \frac{-1 + \sqrt{1 + 4\hat{\rho}_k(\hat{a}_{mk}^2 + \mathbf{\Sigma}_{\mathbf{a}_k}(m, m))}}{2\hat{\rho}_k}. \quad (\text{II.88})$$

Siguiendo un procedimiento similar, el valor óptimo estimado para ρ_k será

$$\hat{\rho}_k = \frac{2M}{\sum_{m=1}^M \hat{\omega}_{mk}}. \quad (\text{II.89})$$

La inferencia sobre cada una de las columnas de \mathbf{X} conduce, una vez más, a una distribución gaussiana

$$\begin{aligned} \log q(\mathbf{x}_q) &= -\frac{\hat{\beta}}{2} \sum_q \langle \|\mathbf{M}_q(\mathbf{y}_q - \mathbf{\Psi}_q \mathbf{A} \mathbf{x}_q)\|^2 \rangle_{\Theta \setminus \mathbf{x}_q} - \frac{1}{2} \mathbf{x}_q^T \hat{\mathbf{\Gamma}}_q^{-1} \mathbf{x}_q + C \\ &= -\frac{\hat{\beta}}{2} \|\mathbf{M}_q \mathbf{y}_q - \mathbf{V}_q \mathbf{x}_q\|^2 - \frac{1}{2} \mathbf{x}_q^T (\mathbf{Z}_q + \hat{\mathbf{\Gamma}}_q^{-1}) \mathbf{x}_q + \text{const}, \end{aligned} \quad (\text{II.90})$$

cuya media $\hat{\mathbf{x}}_q$ y matriz de covarianza $\Sigma_{\mathbf{x}_q}$ vienen dadas por

$$\hat{\mathbf{x}}_q = \hat{\beta} \Sigma_{\mathbf{x}_q} \mathbf{V}_q \mathbf{y}_q \quad (\text{II.91})$$

$$\Sigma_{\mathbf{x}_q} = (\hat{\beta} \mathbf{V}_q^T \mathbf{V}_q + \mathbf{Z}_q + \Gamma_q^{-1})^{-1}, \quad (\text{II.92})$$

donde $\mathbf{V}_q = \mathbf{M}_q \Psi_q \hat{\mathbf{A}}$ y \mathbf{Z}_q es una matriz diagonal con $\mathbf{Z}_q(k, k) = \hat{\beta} \text{tr}(\Psi_q^T \mathbf{M}_q \Psi_q \Sigma_{\mathbf{a}_k})$. Las actualizaciones correspondientes a los hiperparámetros asociados a \mathbf{x}_q tienen expresiones similares a (II.88) y (II.89).

La estimación de los vectores *sparse* \mathbf{a}_k y \mathbf{x}_q siguen un procedimiento análogo al visto en el apartado 4.3.

Finalmente, para estimar la precisión de ruido volvemos a aplicar (II.30) considerando únicamente los términos de (II.82) que dependen de β . Tenemos

$$\begin{aligned} \log q(\beta) &= \frac{\sum_q N_q}{2} \log \beta + (a_\beta - 1) \log \beta - b_\beta \beta \\ &\quad - \frac{\beta}{2} \sum_q \langle \|\mathbf{M}_q(\mathbf{y}_q - \Psi_q \mathbf{A} \mathbf{x}_q)\|^2 \rangle_{\Theta \setminus \beta} + \text{const}, \end{aligned} \quad (\text{II.93})$$

lo cual corresponde a una distribución Gamma sobre β de media

$$\hat{\beta} = \langle \beta \rangle = \frac{\sum_q N_q + 2a_\beta}{\sum_q \langle \|\mathbf{M}_q(\mathbf{y}_q - \Psi_q \mathbf{A} \mathbf{x}_q)\|^2 \rangle_{\Theta \setminus \beta} + 2b_\beta}, \quad (\text{II.94})$$

con

$$\langle \|\mathbf{M}_q(\mathbf{y}_q - \Psi_q \mathbf{A} \mathbf{x}_q)\|^2 \rangle_{\Theta \setminus \beta} = \text{tr} \left((\hat{\mathbf{A}}^T \Psi_q \mathbf{M}_q^T \mathbf{M}_q \Psi_q \hat{\mathbf{A}} + \frac{\mathbf{Z}_q}{\hat{\beta}}) (\hat{\mathbf{x}}_q^T \hat{\mathbf{x}}_q + \Sigma_{\mathbf{x}_q}) \right). \quad (\text{II.95})$$

5.3. Experimentos

Veamos ahora el funcionamiento del modelo doblemente *sparse* en el caso de la eliminación de ruido. Para ello utilizaremos tres imágenes estándar: *Barbara*, *Boat* y *Peppers*, añadiendo ruido gaussiano a cada una de ellas con desviaciones típicas $\sigma = 10, 20, 30, 40, 50$. Para el entrenamiento de los diccionarios emplearemos solamente $Q = 2000$ bloques de la imagen de tamaño 8×8 . Como diccionario utilizaremos un diccionario DCT sobrecompleto $\Psi \in \mathbb{R}^{64 \times 100}$.

Utilizaremos dos versiones de nuestro algoritmo de aprendizaje de diccionarios *sparse*. Por un lado, la versión completa en la que se permite, además de la adición, la eliminación y reestimación de componentes (BDS) y, por otro, una versión más rápida análoga a la vista en el apartado del algoritmo BKSVD, en la que solo se puede añadir elementos a la representación (BDSA). Comparamos estos métodos con otras técnicas

CAPÍTULO II. *DENOISING E INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

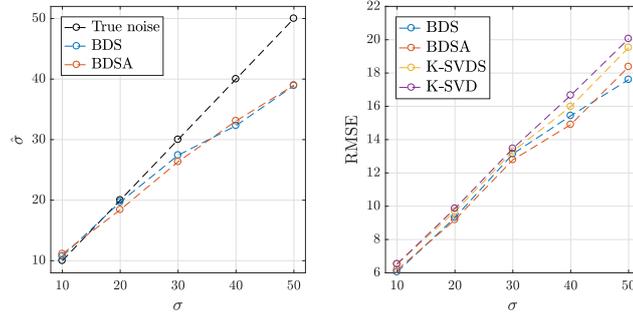


Figura 6: Estimated σ (left) at different noise levels and RMSE reconstruction errors (right).

populares del estado del arte. En primer lugar, veremos como comparan con la versión doblemente *sparse* del K-SVD (K-SVDS) y, en segundo lugar, también realizaremos los experimentos con la versión original. Estos dos métodos requieren el valor de la varianza del ruido y del número de componentes no nulas. Para una mejor comparación de los métodos, estos valores se corresponden a los obtenidos por nuestro algoritmo.

La fig. 6 (izqda) muestra que el valor estimado de σ , la desviación típica del ruido, es muy preciso para valores pequeños o medios de ruido. Para valores más altos se producen errores en la estimación debidos a la gran degradación de la imagen. La estimación de este parámetro podría ser mejorada utilizando un mayor número de señales de entrenamiento o bloques de mayor tamaño. En cuanto al error de reconstrucción, el subpanel derecho de la fig. 6 muestra que todos los métodos funcionan de manera similar para los valores más bajos de σ . No obstante, los dos métodos propuestos son ligeramente superiores (menor RMSE) a K-SVD y K-SVDS para mayor desviación de ruido.

Table 10 shows the superiority of the proposed method in PSNR and SSIM. BDS presents the best results, being better than K-SVDS in 10 out of 15 test cases in terms of PSNR. The faster BDSA achieves competitive results, while accelerating the original algorithm. Notice also the better general behaviour of the double sparsity algorithms, compared to the standard single-sparse technique. ¿adrpor qué están estos textos en inglés?

Para finalizar, mostramos un pequeño ejemplo gráfico del funcionamiento de los algoritmos en la figura 7. De izquierda a derecha podemos ver la imagen ruidosa y las imágenes procesadas utilizando nuestro algoritmo y el K-SVDS respectivamente.

6. REPRESENTACIONES *SPARSE* CON *A POSTERIORI*
LAPLACIANA

Cuadro 10: PSNR y SSIM medios.

	Barbara		Boat		Peppers		σ
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
BDS	29.09	0.91	32.19	0.86	23.14	0.85	10
BDSA	29.05	0.91	31.53	0.85	23.69	0.86	
K-SVDS	31.12	0.92	31.66	0.85	23.57	0.86	
K-SVD	29.07	0.88	29.27	0.82	24.53	0.84	
BDS	27.23	0.86	29.13	0.79	23.93	0.81	20
BDSA	26.81	0.86	28.17	0.79	22.31	0.80	
K-SVDS	28.08	0.86	28.37	0.79	21.44	0.80	
K-SVD	26.18	0.80	26.12	0.72	21.91	0.77	
BDS	26.61	0.76	26.51	0.73	24.26	0.70	30
BDSA	25.97	0.76	25.80	0.72	21.77	0.66	
K-SVDS	25.17	0.76	26.02	0.72	21.86	0.66	
K-SVD	23.33	0.69	23.17	0.64	20.64	0.63	
BDS	23.66	0.63	23.95	0.61	22.27	0.64	40
BDSA	22.92	0.68	23.24	0.63	21.60	0.55	
K-SVDS	23.74	0.69	23.31	0.62	21.60	0.56	
K-SVD	22.09	0.63	21.91	0.56	20.54	0.53	
BDS	21.56	0.55	21.81	0.51	19.99	0.48	50
BDSA	20.43	0.51	21.88	0.52	19.81	0.41	
K-SVDS	20.41	0.50	21.91	0.51	19.35	0.41	
K-SVD	19.77	0.46	20.96	0.46	19.25	0.39	

6. Representaciones *sparse* con *a posteriori* laplaciana

En los dos métodos de aprendizaje basado en diccionarios vistos hasta ahora utilizábamos una modelización idéntica de los vectores de representación *sparse*, tanto en los vectores \mathbf{x}_q relativos a las señales *sparse* vistas en el K-SVD bayesiano y la posterior versión doblemente *sparse* como en la modelización de las columnas \mathbf{a}_k del diccionario *sparse* del apartado anterior. En estos casos asignábamos una *a priori* jerárquica de tres niveles que, como veíamos, era equivalente a una distribución laplaciana. La inferencia variacional resultaba en una distribución *a posteriori* gaussiana que embebíamos en un proceso de inferencia rápido y eficiente que producía, finalmente, las representaciones *sparse*. Veremos ahora, sin embargo, una modelización alternativa utilizando directamente la distribución laplaciana como *a priori*. A pesar de la presencia de la función valor absoluto en la definición de esta distribución, función no diferenciable en cero, comprobaremos que podemos utilizar distribuciones

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

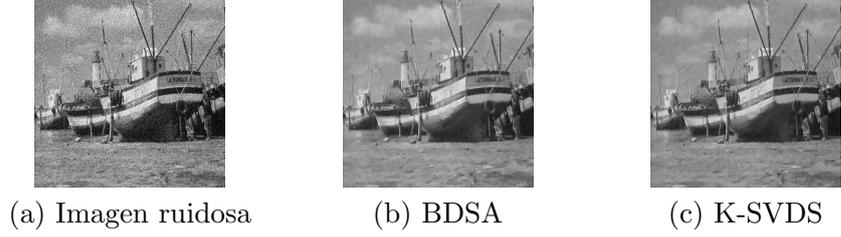


Figura 7: *Denoising* de la imagen *Boat*, $\sigma = 30$.

a posteriori laplacianas permitiendo soluciones analíticas.

Nótese que solo consideramos el problema de la representación *sparse*, por lo que, al contrario de lo visto en los apartados anteriores, el diccionario no estará involucrado en el proceso de optimización, considerándolo fijo. Por tanto, podemos formular el problema de representación *sparse* como

$$\begin{aligned} \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2 \\ \text{s.a.} \|\mathbf{x}\|_0 \leq T, \end{aligned} \quad (\text{II.96})$$

y su relajación con norma-1

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ \|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2 + \frac{1}{\gamma} \|\mathbf{x}\|_1 \right\}. \quad (\text{II.97})$$

6.1. Modelado bayesiano e inferencia variacional EM

El modelado bayesiano del problema de representación *sparse* (II.97) requiere la definición de la distribución conjunta $p(\mathbf{y}, \mathbf{x} | \boldsymbol{\Omega})$, donde $\boldsymbol{\Omega} = \{\gamma, \beta\}$. Podemos utilizar, como es habitual, la factorización

$$p(\mathbf{y}, \mathbf{x} | \boldsymbol{\Omega}) = p(\mathbf{y} | \mathbf{x}, \beta) p(\mathbf{x} | \gamma). \quad (\text{II.98})$$

con el modelo de observación

$$p(\mathbf{y} | \mathbf{x}, \beta) = \mathcal{N}(\mathbf{y} | \mathbf{D}\mathbf{x}, \beta^{-1} \mathbf{I}) \quad (\text{II.99})$$

y, de (II.97), la *a priori*

$$p(\mathbf{x} | \gamma) = \prod_{i=1}^N \mathcal{L}(x_i | 0, \gamma), \quad (\text{II.100})$$

donde

$$\mathcal{L}(x | \mu, \delta) = \frac{1}{2\delta} e^{-\frac{|x-\mu|}{\delta}}. \quad (\text{II.101})$$

6. REPRESENTACIONES *SPARSE* CON *A POSTERIORI*
LAPLACIANA

Nótese que, para esta definición de la distribución laplaciana, tenemos

$$\langle x \rangle = \mu \quad (\text{II.102})$$

$$\text{var}(x) = 2\delta^2. \quad (\text{II.103})$$

Emplearemos el método EM para la inferencia de los parámetros del modelo así como de la señal subyacente \mathbf{x} , el cual está basado en la maximización de la verosimilitud

$$\hat{\Omega} = \arg \max_{\Omega} p(\mathbf{y}|\Omega) \quad (\text{II.104})$$

con respecto a los parámetros Ω , donde

$$p(\mathbf{y}|\Omega) = \int p(\mathbf{y}, \mathbf{x}|\Omega) d\mathbf{x}. \quad (\text{II.105})$$

Al no poder realizar el cálculo analítico de la verosimilitud, el método EM trata de maximizar la ELBO en su lugar. Veremos una explicación más clara del funcionamiento de este método en el siguiente capítulo, donde se hace un uso más general de este método. Podemos expresar esta cota inferior de la verosimilitud como

$$\mathcal{L}(q(\mathbf{x}), \Omega) = \int q(\mathbf{x}) \log \frac{p(\mathbf{y}|\mathbf{x}, \beta)p(\mathbf{x}|\gamma)}{q(\mathbf{x})} d\mathbf{x}, \quad (\text{II.106})$$

y alternaremos la maximización con respecto a $q(\mathbf{x})$ y Ω de esta expresión.

Supodremos en este caso que la distribución *a posteriori* sobre \mathbf{x} es un producto de laplacianas independientes sobre cada una de las entradas x_i del vector de representación *sparse*, esto es,

$$q(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\delta}) = \prod_{i=1}^N \mathcal{L}(x_i|\mu_i, \delta_i) \quad (\text{II.107})$$

donde $\boldsymbol{\mu} = [\mu_1, \dots, \mu_N]^T$ y $\boldsymbol{\delta} = [\delta_1, \dots, \delta_N]^T$. Ya habíamos visto en las secciones anteriores la posibilidad de imponer una mezcla infinita de gaussianas como *a priori*, que resultaba en una aproximación gaussiana de la *a posteriori*. No obstante, la distribución que aquí obtenemos es más *sparse* por lo que debería producir una mejor solución.

En primer lugar debemos obtener la distribución $q(\mathbf{x})$ que maximiza la ELBO. Para ello fijamos ahora los parámetros a $\Omega^{(t)}$ y $\{\mu_j, \delta_j, \forall j \neq i\}$ y optimizaremos $\mathcal{L}(q(\mathbf{x}), \Omega^{(t)})$ respecto a μ_i y δ_i . Nótese que, utilizando las propiedades del logaritmo, podemos descomponer la ELBO (II.106). El primero relacionado con el modelo de observación

$$\begin{aligned} \langle \log p(\mathbf{y}|\mathbf{x}, \beta) \rangle_{\mathbf{x}} &= \left\langle -\frac{\beta}{2} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2 \right\rangle_{\mathbf{x}} + \text{const} \\ &= -\frac{\beta}{2} (\|\mathbf{d}_i\|^2 (\mu_i^2 + 2\delta_i^2) - 2\mathbf{z}_i^T \mathbf{d}_i \mu_i) + \text{const}, \end{aligned} \quad (\text{II.108})$$

CAPÍTULO II. *DENOISING E INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

con $\mathbf{z}_i = \mathbf{y} - \sum_{j \neq i} \mathbf{d}_j \mu_j$. Un segundo término de la *a priori*

$$\langle \log p(\mathbf{x}|\gamma) \rangle_{\mathbf{x}} = -\frac{1}{\gamma} \langle |x_i| \rangle + \text{const}, \quad (\text{II.109})$$

y un tercero que depende de la entropía de la *a posteriori*

$$-\langle \log q(\mathbf{x}) \rangle_{\mathbf{x}} = \sum_i \mathbb{H}[q(x_i)] = \log \delta_i + \text{const}. \quad (\text{II.110})$$

Nótese que las constantes incluyen todos los términos que no dependen de μ_i o δ_i , irrelevantos, por tanto, para la optimización. Puede demostrarse que el término $\langle |x_i| \rangle$ en (II.109) resulta

$$\langle |x_i| \rangle = \delta_i e^{-\frac{|\mu_i|}{\delta_i}} + |\mu_i|, \quad (\text{II.111})$$

que, interesantemente, es diferenciable en $\mu_i = 0$. Es más, sus derivadas parciales con respecto a μ_i y δ_i son también diferenciables en todo el dominio de definición de estos parámetros, $\mathbf{D} = \{\boldsymbol{\mu} \in \mathbb{R}^N, \boldsymbol{\delta} \in \mathbb{R}_{>0}^N\}$. Eso permitirá el uso de algoritmos de optimización de segundo orden.

Para acabar, podemos encontrar la solución óptima que maximiza la ELBO sumando estos tres términos e igualando a cero la derivada de la expresión resultante con respecto a μ_i y δ_i , con lo que tenemos

$$\begin{cases} \beta(\|\mathbf{d}_i\|^2 \mu_i - \mathbf{z}_i^T \mathbf{d}_i) + \frac{1}{\gamma} \text{sgn}(\mu_i) (1 - e^{-\frac{|\mu_i|}{\delta_i}}) = 0 \\ 2\beta\|\mathbf{d}_i\|^2 \delta_i + \frac{1}{\gamma} e^{-\frac{|\mu_i|}{\delta_i}} (1 + \frac{|\mu_i|}{\delta_i}) - \frac{1}{\delta_i} = 0 \end{cases} \quad (\text{II.112})$$

Este sistema de ecuaciones no lineales no puede resolverse analíticamente, con lo que requerirá métodos numéricos. Dado que ambas ecuaciones son diferenciables utilizaremos el método de Newton-Raphson dada su mayor velocidad de convergencia.

Una vez actualizados los parámetros relativos a la distribución *a priori* pasamos al cálculo de los parámetros $\boldsymbol{\Omega}$. Para ello, fijando $\boldsymbol{\mu}$ y $\boldsymbol{\delta}$, actualizamos $\boldsymbol{\Omega}$ maximizando la ELBO

$$\hat{\boldsymbol{\Omega}} = \arg \max_{\boldsymbol{\Omega}} \mathcal{L}(q(\mathbf{x}), \boldsymbol{\Omega}). \quad (\text{II.113})$$

Para el cálculo de $\hat{\gamma}$, tomando solo los término de la ELBO que contienen este parámetro, tenemos

$$\hat{\gamma} = \arg \max_{\gamma} \langle \log p(\mathbf{x}|\gamma) \rangle_{\hat{\mathbf{x}}}, \quad (\text{II.114})$$

lo que resulta en

$$\gamma = \frac{\sum_{i=1}^N \langle |x_i| \rangle}{N}, \quad (\text{II.115})$$

6. REPRESENTACIONES *SPARSE* CON *A POSTERIORI* LAPLACIANA

y, para la precisión del ruido

$$\hat{\beta} = \arg \max_{\beta} \langle \log p(\mathbf{y}|\mathbf{x}, \beta) \rangle_{\hat{\mathbf{x}}}, \quad (\text{II.116})$$

cuya solución es

$$\hat{\beta}^{-1} = \frac{\|\mathbf{y} - \mathbf{D}\boldsymbol{\mu}\|^2 + 4 \sum_i \|\mathbf{d}_i\|^2 \delta_i^2}{M}. \quad (\text{II.117})$$

Sin embargo, dada la naturaleza constructiva del algoritmo, el parámetro de ruido no puede calcularse de esta manera. El término de fidelidad será altamente impreciso en iteraciones tempranas del algoritmo ya que se parte de un modelo con un único átomo. Por ello, análogamente a [61], fijaremos este parámetro al comienzo del algoritmo a $\beta = 0.01\|\mathbf{y}\|^2$.

6.2. Algoritmo EM con *a posteriori* laplaciana

El algoritmo propuesto para encontrar la señal de representación *sparse* \mathbf{x} puede verse en alg. 4. Se trata de un algoritmo constructivo similar al usado en el BKSVD y su versión con diccionarios *sparse*. Partimos de un modelo vacío, es decir, $\mu_i = 0, \forall i$. En la primera iteración, el algoritmo tendrá que escoger óptimamente una de las N componentes de \mathbf{x} y añadirla al modelo. Esta elección se basa en buscar aquella solución $\{\mu_i, \delta_i\}$ del sistema (II.112) produce un mayor incremento de la ELBO. A partir de (II.108), (II.109) y (II.110) tenemos

$$\begin{aligned} \Delta \mathcal{L}_i = & -\frac{\beta}{2} (\|\mathbf{d}_i\|^2 [\mu_i^{2(t+1)} - \mu_i^{2(t)} + 2(\delta_i^{2(t+1)} - \delta_i^{2(t)})] - 2\mathbf{z}_i^T \mathbf{d}_i (\mu_i^{(t+1)} - \mu_i^{(t)})) \\ & - \frac{1}{\gamma} (\langle |x_i| \rangle^{(t+1)} - \langle |x_i| \rangle^{(t)}) + \log \frac{\delta_i^{(t+1)}}{\delta_i^{(t)}}. \end{aligned} \quad (\text{II.118})$$

El tratar la optimización separadamente por coordenadas nos permite garantizar una solución exactamente *sparse* además de posibilitar la refinación de estimaciones previas. Esta refinación se producirá cuando el mayor incremento de ELBO (II.118), siendo posible la reestimación de este valor, o su eliminación del modelo, en el caso de que el nuevo valor sea nulo o casi nulo.

Por último, la fig. 8 muestra el comportamiento del algoritmo, demostrando experimentalmente su convergencia. Los parámetros utilizados son $M = 100$, $N = 512$, $T = 20$. Es importante resaltar que todas las iteraciones del algoritmo producen un incremento de ELBO positivo.

6.3. Funcionamiento experimental del algoritmo

Para comprobar la validez del método generamos un diccionario sintético $\mathbf{D} \in \mathbb{R}^{P \times K}$ y un conjunto de señales *sparse* \mathbf{x} . Para la generación

CAPÍTULO II. *DENOISING* E *INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

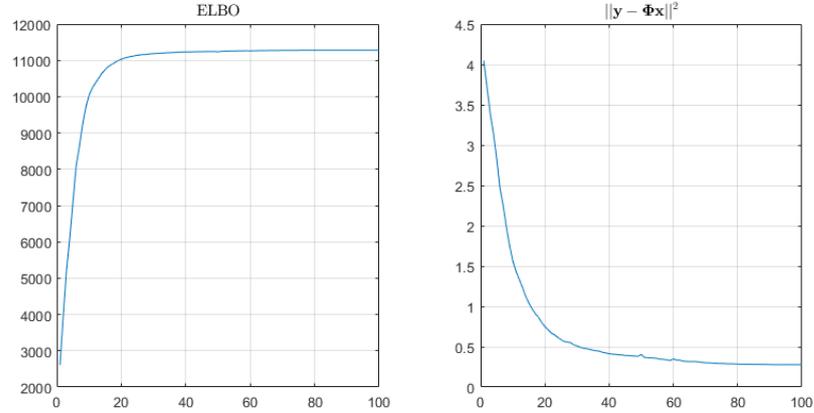


Figura 8: ELBO y término de fidelidad para un experimento sintético.

Algoritmo 4 Algoritmo de representación *sparse* con *a posteriori* laplaciana.

ENTRADA: \mathbf{D} e \mathbf{y} .

- 1: Establecer $\boldsymbol{\mu}^{(0)} = \mathbf{0}$.
- 2: **while** no converja **do**
- 3: Actualizar γ según (II.115).
- 4: **for** $i=1:N$ **do**
- 5: Encontrar $\mu_i^{(t+1)}$ y $\delta_i^{(t+1)}$ resolviendo (II.112) utilizando $\mu_j^{(t)}$ y $\delta_j^{(t)} \forall j \neq i$.
- 6: **end for**
- 7: Actualizar la componente de $\boldsymbol{\mu}$ y $\boldsymbol{\delta}$ que resulten en el mayor incremento de ELBO.

8: **end while**

SALIDA: $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\delta}}$ y $\hat{\gamma}$.

de las señales *sparse* generaremos un vector aleatorio con una distribución gaussiana $\mathcal{N}(0, \sigma)$. Una vez generada la señal seleccionaremos aleatoriamente solo T entradas, poniendo a cero el resto y, finalmente, generaremos las señales observadas sintéticas siguiendo el modelo $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{n}$, donde $\mathbf{n} \sim \mathcal{N}(0, 0.03^2)$. Compararemos el funcionamiento de nuestro método con el algoritmo presentado en [61]. Las figs. 9, 10 y 11 muestran tres ejemplos de reconstrucciones con $M = \{80, 100, 120\}$. Se puede observar que ambos algoritmos presentan un mayor error de reconstrucción para valores de M menores ($M = 80$). Lógicamente, cuanto menor sea el número de muestras tomadas M , más difícil será recuperar la señal *sparse* subyacente. Además, la matriz \mathbf{D} presentará una correlación mutua menor, dificultando la identificación exacta del soporte de \mathbf{x} . Para

6. REPRESENTACIONES *SPARSE* CON *A POSTERIORI* LAPLACIANA

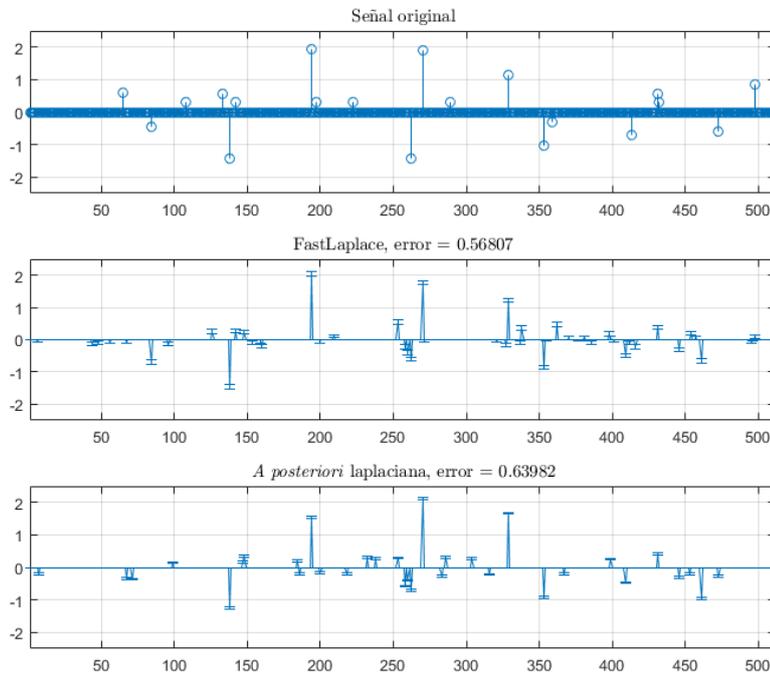


Figura 9: Ejemplo de reconstrucción con $T = 20$, $M = 80$.

los ejemplos con $M = 100$ y $M = 120$, los resultados obtenidos son similares, siendo levemente mejores para el algoritmo propuesto. Obsérvese que el método [61] añade al soporte algunas posiciones que no formaban parte de la señal *sparse* original, aunque con un valor pequeño.

A pesar de estos resultados, hay que realizar un estudio más exhaustivo del comportamiento de este algoritmo en escenarios más realistas y comprobar si es una alternativa viable dado su mayor tiempo de ejecución. Pese a ser algoritmos constructivos que proceden de forma análoga, la iteración de [61] tiene forma analítica cerrada, mientras que el algoritmo propuesto requiere la solución iterativa de un sistema de ecuaciones implícitas.

CAPÍTULO II. *DENOISING E INPAINTING* MEDIANTE APRENDIZAJE DE DICCIONARIOS

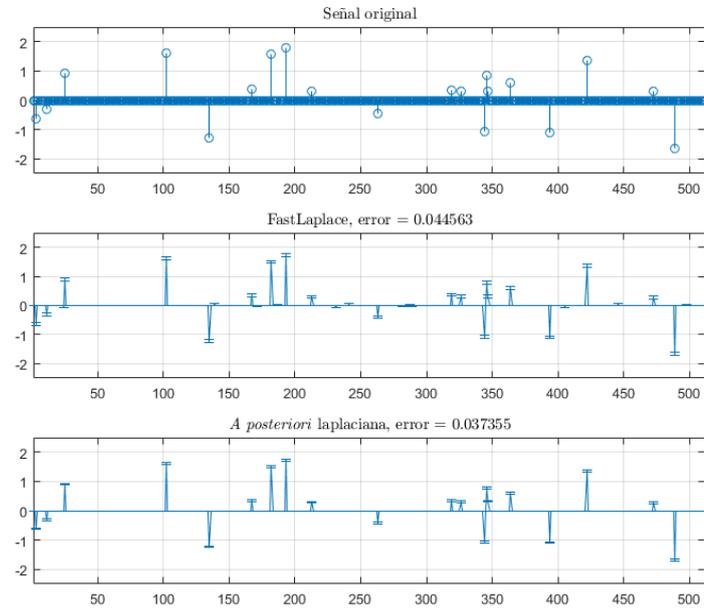


Figura 10: Ejemplo de reconstrucción con $T = 20$, $M = 100$.

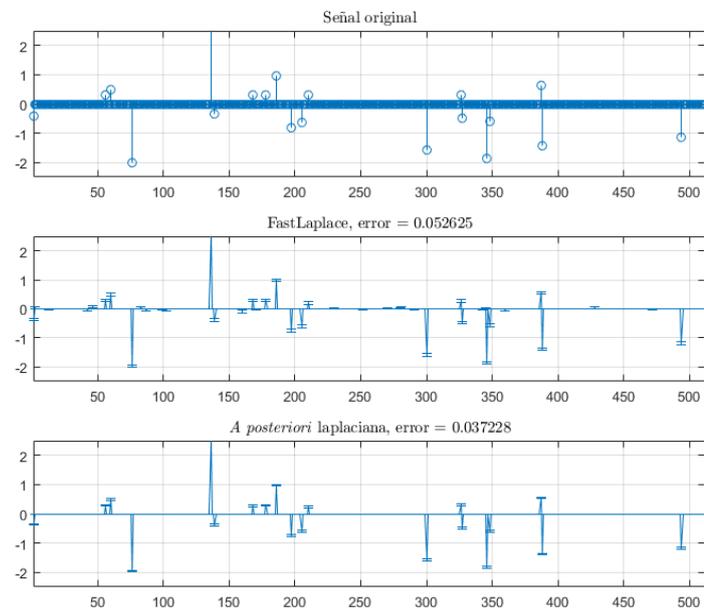


Figura 11: Ejemplo de reconstrucción con $T = 20$, $M = 120$.

Capítulo III

Deconvolución ciega de imágenes

En el capítulo anterior vimos cómo era posible la eliminación de ruido en imágenes milimétricas además del *inpainting* mediante el aprendizaje basado en diccionarios. En este capítulo abordaremos la eliminación del emborronamiento en las mismas. Desarrollaremos el modelo publicado en [73] que será evaluado cuantitativamente en este capítulo y sobre imágenes milimétricas en el siguiente.

Se propone un método de deconvolución ciega utilizando la distribución *a priori Spike-and-Slab* con un esquema eficiente de inferencia variacional EM (*Expectation Maximization*), utilizado por primera vez en este campo de investigación. Esta distribución constituye el máximo exponente del aprendizaje automático *sparse*.

1. Introducción

La deconvolución de imágenes tiene como objetivo obtener, dada una imagen borrosa y , comúnmente, con ruido, la imagen nítida subyacente. Tanto el emborronamiento como el ruido son, habitualmente, degradaciones indeseadas inherentes al proceso de adquisición de imágenes que afectan en una gran variedad de campos tales como la fotografía comercial, astronomía, imagen médica o teledetección, entre otros. El término «deconvolución» hace referencia al proceso inverso de emborronamiento de una imagen, el cual resulta de la convolución de una imagen con un filtro de emborronamiento, también llamado PSF, del inglés *Point Spread Function*. Formalmente, podemos expresar este proceso de degradación como

$$\mathbf{y} = \mathbf{h} * \mathbf{x} + \mathbf{n}, \quad (\text{III.1})$$

donde $\mathbf{y}, \mathbf{x}, \mathbf{n} \in \mathbb{R}^{H \times W}$ denotan la imagen borrosa, la imagen nítida subyacente y la matriz de ruido respectivamente. $\mathbf{h} \in \mathbb{R}^{h \times w}$ denota la PSF.

Nótese que se aplican dos restricciones habituales sobre las componentes h_i de \mathbf{h} , por un lado, la positividad de sus elementos y, por otro, se impone que la suma de estos sea uno. De esta forma se garantiza que las imágenes nítida y borrosa tengan, en caso de ruido de media nula, aproximadamente la misma media. Por último, $*$ denota la convolución 2D, que en este caso devuelve una imagen de tamaño igual a la de entrada, es decir, no se considera la convolución completa que produciría una imagen de mayor tamaño. Podemos ver en la fig. 1 un ejemplo de convolución de una imagen

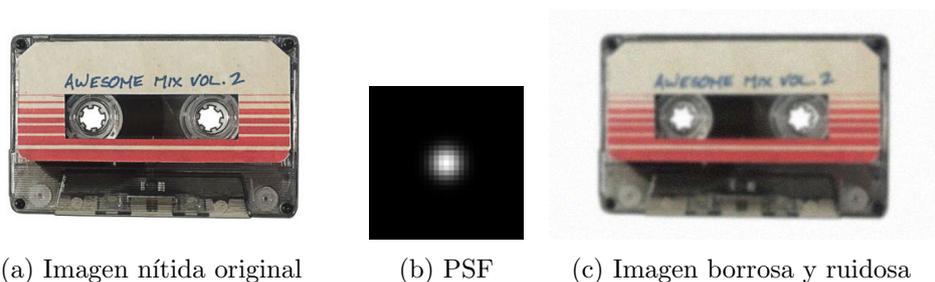


Figura 1: Ejemplo del proceso de emborronamiento.

nítida con un filtro de desenfoque gaussiano de desviación típica $\sigma = 2$ con adición de ruido gaussiano de media nula y desviación típica 0.03. En el caso de no conocer el filtro de emborronamiento hablaremos de deconvolución «ciega» de imágenes. Centraremos la atención de nuestro trabajo precisamente en este problema más complejo, al que nos referiremos abreviadamente y en lo sucesivo como BID, de sus siglas en inglés (*Blind Image Deconvolution*).

Conviene ahora observar que el modelo de degradación expuesto en (III.1) admite una representación matricial [74]

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad (\text{III.2})$$

donde $\mathbf{y}, \mathbf{x}, \mathbf{n}$ son ahora vectores columna de tamaño $HW \times 1$, y \mathbf{H} , de tamaño $HW \times HW$, es la matriz de convolución asociada a \mathbf{h} . Esta es una matriz es circulante por bloques cuyas columnas se obtienen a partir de los elementos de la PSF.

En deconvolución ciega tenemos que estimar tanto la imagen nítida como el filtro a partir, únicamente, de las observaciones, i. e., la imagen degradada \mathbf{y} . Este problema inverso es indeterminado en gran medida, puesto que intervienen mayor número de incógnitas que observaciones. Este hecho provoca la existencia de múltiples soluciones para el par imagen/PSF compatibles con la imagen observada. Peor aún, este problema está mal definido, según la definición de Hadamard [75], esto es, pequeñas variaciones en las observaciones pueden alterar enormemente la solución. Es más, pequeñas variaciones en la estimación del filtro de

emborronamiento también pueden causar grandes variaciones en la imagen estimada. Para aliviar este problema y obtener buenas estimaciones para PSF e imagen es necesario restringir el espacio de soluciones introduciendo conocimiento *a priori*, además de métodos de inferencia adecuados.

Por ejemplo, en el contexto de las imágenes naturales, este conocimiento *a priori* consiste en el hecho de que sabemos que solo algunos de los píxeles de estas corresponden a bordes, es decir, fronteras entre objetos con saltos bruscos de intensidad. Esta propiedad puede ser explotada en lo que se conoce como dominio filtrado, en contraposición al dominio de la imagen, al cual nos referíamos en lo previo, pudiéndose formular el problema en cualquiera de los dominios [74]. Los métodos de BID que trabajan en el dominio de la imagen estima la imagen latente y la PSF directamente a partir de la observación, la propia imagen borrosa. Sin embargo, el dominio filtrado recibe este nombre por la generación de un conjunto de pseudoobservaciones mediante filtrado paso alto de la imagen. Obviamente, las pseudoobservaciones serán *sparse* dado que, como ya comentábamos, presentan pocos bordes y, en consecuencia, muchas zonas planas o de variación suave que serán eliminadas por el filtrado. La literatura alberga cierta cantidad de trabajos que exponen las ventajas e inconvenientes de ambas aproximaciones [74, 76, 77]. Los métodos que trabajan en el dominio de la imagen son capaces de estimar la imagen latente y el filtro de emborronamiento directamente [78]. No obstante, las imágenes restauradas tienen típicamente una apariencia de dibujo, con bordes fuertemente marcados, dado que estimaciones precisas de la PSF dependen, a menudo, de la presencia de bordes muy marcados y pocos detalles en las imágenes. Es por esto que habitualmente se utiliza un algoritmo de deconvolución no ciega para obtener la imagen nítida final a partir de la PSF obtenida. En cambio, en el dominio de la frecuencia, los métodos disponen de más (pseudo)observaciones para estimar el emborronamiento, una por filtro. A pesar de esto, estos métodos suelen ser más sensibles al ruido. Tras la estimación de la PSF es necesario el uso de un algoritmo de deconvolución no ciega para estimar la imagen.

2. Revisión bibliográfica

El estado del arte en el campo de la deconvolución ciega está liderado por métodos estocásticos que, basándose en el modelo físico de degradación (III.1) o (III.2), formulan los distintos elementos en un marco probabilístico asignando distribuciones de probabilidad adecuadas para modelar todo el conocimiento *a priori* del problema. Podemos subdividir estos métodos, *grosso modo*, en dos grandes grupos: aquellos que utilizan inferencia MAP y los que usan inferencia variacional. Una comparación detallada de ambos métodos puede ser hallada en [79], donde se prueba la

conveniente habilidad de la VI para evitar máximos locales, que corresponden a malas soluciones del problema, cuando se utilizan distribuciones *a priori* altamente *sparse*. En contrapartida, los métodos MAP son computacionalmente más ligeros debido a que no buscan la distribución *a posteriori* completa de las variables latentes, sino solo su máximo, de aquí su nombre.

Existe una prolongada bibliografía de métodos MAP y métodos basado en regularización. Estos últimos se basan en la minimización de una función objetivo que suele incluir un término de fidelidad y otro/s de regularización sobre la imagen restaurada y, a veces, también sobre la PSF. Puede verse fácilmente la equivalencia de los métodos de regularización con los métodos MAP [80]. En estadios tempranos, se imponía una forma paramétrica para la PSF [81], con lo que la estimación del núcleo de emborronamiento se reducía al cálculos de sus parámetros [82]. Alternativamente, se imponían otras restricciones típicas sobre este como la no negatividad, suavidad o simetrías, véase [83, 84]. La observación de que los métodos MAP favorecían soluciones de emborronamiento nulo [85], condujo a los investigadores al diseño cuidadoso de distribuciones *a priori* para la imagen y PSF [86], así como al uso de imágenes intermedias con fronteras resaltadas [87–89]. En esta línea se encuentran muchos de los trabajos más recientes. Por ejemplo, trabajos como [90] utilizan distribuciones *a priori* hiperlaplacianas¹, además de incluir, en otros, un ajuste iteración a iteración de los parámetros de regularización [88, 91]. Otras distribuciones empleadas en la literatura incluyen las *a priori* log-TV [92], aproximaciones de la pseudonorma ℓ_0 utilizada en [77], trabajo extendido en [89] con el uso de la llamada «*a priori* del canal oscuro²», y combinaciones de regularizadores ℓ_0 y ℓ_2 para la imagen de gradientes y la PSF [93]. El método expuesto en [94], bajo la premisa de que los bordes pequeños en las imágenes conducen a estimaciones del emborronamiento poco precisas, propone una estimación de este en dos fases con selección de fronteras significantes. Con esta misma idea de selección de gradientes encontramos el trabajo de Gong *et al.* [95], obteniendo una estimación robusta de la PSF. Estos métodos de están íntimamente relacionados con los métodos *Spike-and-Slab* que presentaremos más adelante. Por otro lado, algunos de los métodos de deconvolución más recientes incorporan directamente la información de imágenes naturales entrenando las distribuciones *a priori* a partir de bases de datos de imágenes. Por ejemplo, en [96] emplean una *a priori* ℓ_p para la

¹Las distribuciones hiperlaplacianas son una extensión de la distribución laplaciana, en la que se sustituye el valor absoluto por una norma ℓ_p con $0 < p < 1$, es decir, $p(x) \propto e^{-k\|x\|_p^p}$. Estas distribuciones son más apuntadas que la laplaciana original.

²El canal oscuro de una imagen RGB viene a ser un filtrado de tipo mínimo, i. e., cada píxel en la imagen filtrada corresponde al valor mínimo, para todos los canales, en un vecindario en torno a ese píxel. El canal oscuro de una imagen nítida es *sparse*.

PSF e introducen la *a priori* «*Gaussian Scale Mixture Field of Experts*», cuyos filtros están preentrenados a partir de imágenes naturales, para la imagen. Cai *et al.* [97] reinciden en la idea de usar las fronteras más significativas además de utilizar representaciones *sparse*, mediante las que incorporan información preaprendida para la estimación tanto de la imagen como de la PSF. [98] desarrolla una *a priori* basada en una red convolucional profunda capaz de discernir entre imágenes nítidas y borrosas. Por último, [99] sigue esta misma filosofía, entrenando una *a priori* basada en un modelo de mezcla de gaussianas para diferentes clases de imágenes, como rostros o huellas dactilares.

El empleo de la inferencia variacional en problemas de deconvolución de imágenes surge con posterioridad a los primeros métodos MAP, su origen puede trazarse hasta el trabajo [100], véase también [78]. Similarmente a lo visto en los métodos MAP, los métodos varacionales de deconvolución ciega más actuales también hacen un uso extensivo de las distribuciones *a priori sparse*. La combinación de estas con la inferencia variacional ha ido suscitando cada vez mayor interés desde el trabajo de Fergus *et al.* [101] dada su demostrada efectividad. Este trabajo propuso el uso de una mezcla de gaussianas como *a priori* sobre la imagen de gradientes. Más adelante, *et al.* [102] generaliza este trabajo con el uso de distribuciones supergaussianas y mezclas infinitas de gaussianas. Nótese que este modelo engloba también otras distribuciones (*a priori sparse*) populares como la TV (*Total Variation*), ℓ_p , la mezcla (finita) de gaussianas o la *t* de Student [103]. Zhou *et al.* [104] proponen un método que utiliza supergaussianas de Huber como *a priori*: estas distribuciones salvan la singularidad en cero común en la mayoría de las supergaussianas. Recientemente se han propuesto mezclas de otras distribuciones exponenciales, como la laplaciana [105, 106], generalizando otros modelos de mezclas. En lugar de asumir una distribución *sparse* sobre la PSF y forzar posteriormente restricciones sobre la estimación, como veremos en la siguiente sección, [107] impone una distribución de Dirichlet sobre esta, lo que garantiza directamente que la estimación cumplirá con las restricciones deseadas.

La mayoría de estas distribuciones *sparse* tienen un punto en común: presentan una predilección por valores cercanos a cero, pero no exactamente cero, siendo, matemáticamente, nula la probabilidad de este evento³, por ello reciben habitualmente el apelativo de distribuciones «*soft*» *sparse*. La distribución gaussiana, la laplaciana, las mezclas de gaussianas o las supergaussianas en general son ejemplos que tienen cabida en esta categoría. Este comportamiento hace que no sean opciones ideales para promover soluciones *sparse*, véase [108]. Surge, pues, un interés

³En distribuciones continuas la probabilidad de que una variable tome exactamente un valor dado es cero, puesto que $p(x = a) = \int_a^a p(x)dx = 0$.

creciente en distribuciones *a priori* que combinen distribuciones continuas y discretas suponiendo, por tanto, una mejor aproximación de la regularización ℓ_0 [109, 110]. Las distribuciones *Spike-and-Slab* [111] son un ejemplo de estas combinaciones, siendo también referidas como distribuciones Bernoulli-gaussianas [110], dado que consisten en un producto de dos variables aleatorias: una discreta, que sigue una distribución de Bernoulli, y una gaussiana. Este tipo de distribuciones representan el máximo exponente del aprendizaje automático *sparse*, ya que tienen, por un lado, la capacidad de descartar variables irrelevantes y, por otro, regularizan moderadamente las relevantes [112]. Encontramos abundantes aplicaciones de estas distribuciones en la literatura incluyendo el campo de la representación *sparse* de señales ya comentado en el capítulo anterior. Selección de variables [113, 114], eliminación de ruido [108, 115], *inpainting* [108], modelos no supervisados de variables latentes [116] o la fusión de imágenes hiperespectrales son otros ejemplos de aplicación relevantes. Sin embargo, y pese al alto potencial de las distribuciones *Spike-and-Slab*, la literatura se ha encontrado con grandes obstáculos inherentes a estas. Nótese, en primer lugar, que esta distribución admite una representación alternativa mediante una mezcla de una distribución gaussiana, «*slab*», y una delta de Dirac «*spike*». La presencia de esta singular función complica en gran medida la inferencia, imposibilitando soluciones analíticas. Si bien es cierto que esta distribución se puede aproximar por una mezcla de dos gaussianas [117], una de ellas con una varianza muy pequeña, la distribución resultante sigue siendo continua y, por ello, no verdaderamente *sparse*. En segundo lugar, las simplificaciones habituales impuestas por la inferencia variacional con factorización *mean-field* eliminan dependencias cruciales inherentes a la verdadera distribución *a posteriori*. El modo habitual de salvar estos obstáculos ha sido el uso de la inferencia de Monte Carlo (MCMC, *Markov Chain Monte Carlo*). Desafortunadamente, este tipo de inferencia por muestreo tiene un coste computacional muy elevado. Sin embargo, recientemente se propuso una reparametrización de la distribución [108] en el contexto del aprendizaje multitarea con procesos gaussianos que posibilita la inferencia variacional de manera sencilla y eficiente salvando la presencia de la delta y conservando, además, dependencias deseables en la *a posteriori*. Es importante resaltar que la inferencia variacional establece un marco de trabajo que permite el cálculo automático de los parámetros del modelo, evitando así prácticas comunes poco eficientes como el método de ensayo/error, la fuerza bruta o validación cruzada [118] entre otras. Véase, por último, la inferencia usando el método de propagación de la esperanza con distribuciones *a priori* SnS en [119] en modelos de regresión lineal.

El trabajo presentado en esta tesis se apoya en el éxito de la distribución *Spike-and-Slab* como distribución *sparse* para modelar nuestro conocimiento

previo sobre los gradientes de imágenes naturales, es decir, se trabaja en el dominio filtrado. El uso de esta distribución permitirá al algoritmo descartar observaciones ruidosas produciendo estimaciones de la PSF más robustas y precisas. Además, por primera vez en la literatura, utilizamos un modelo de inferencia variacional inspirado en el trabajo [108], usando las distribuciones *Spike-and-Slab* en el problema de la estimación del emborronamiento.

3. Modelado bayesiano

Vamos a utilizar la formulación en el espacio filtrado del problema de deconvolución ciega expuesto en (III.2). En este modelo creamos L pseudoobservaciones $\mathbf{y}_\gamma \in \mathbb{R}^N$ filtrando la imagen degradada \mathbf{y} con un conjunto de filtros paso-alto $\{\mathbf{f}_\gamma\}_{\gamma=1}^L$, obteniendo

$$\mathbf{y}_\gamma = \mathbf{F}_\gamma \mathbf{y} = \mathbf{H} \mathbf{F}_\gamma \mathbf{x} + \mathbf{F}_\gamma \mathbf{n} = \mathbf{H} \mathbf{x}_\gamma + \mathbf{n}_\gamma, \quad (\text{III.3})$$

con $\mathbf{x}_\gamma = \mathbf{F}_\gamma \mathbf{x}$, donde \mathbf{F}_γ es la matriz de convolución circulante por bloques asociada al filtro \mathbf{f}_γ , que claramente conmuta con la matriz circulante por bloques \mathbf{H} .

Asumiendo que las pseudoobservaciones son independientes [74, 85] y ruido gaussiano con precisión de ruido β conocida, podemos escribir el modelo de observación de este problema como

$$\begin{aligned} p(\mathbf{y}_\Gamma | \mathbf{h}, \mathbf{x}_\Gamma) &= \prod_{\gamma} p(\mathbf{y}_\gamma | \mathbf{h}, \mathbf{x}_\gamma) = \prod_{\gamma} \mathcal{N}(\mathbf{y}_\gamma | \mathbf{H} \mathbf{x}_\gamma, \beta_\gamma^{-1} \mathbf{I}) \\ &\propto \prod_{\gamma} \exp \left\{ -\frac{\beta_\gamma}{2} \|\mathbf{y}_\gamma - \mathbf{H} \mathbf{x}_\gamma\|^2 \right\}, \end{aligned} \quad (\text{III.4})$$

donde $\mathbf{y}_\Gamma = \{\mathbf{y}_\gamma\}_{\gamma=1}^L$, $\mathbf{x}_\Gamma = \{\mathbf{x}_\gamma\}_{\gamma=1}^L$.

Una propiedad interesante relativa a las imágenes latentes $\{\mathbf{x}_\gamma\}_{\gamma=1}^L$ en el dominio filtrado es que presentan un número reducido de píxeles con valores significativos, correspondientes a los bordes de la imagen. Las zonas planas de la imagen tendrán un valor muy pequeño o cercano a cero debido a la aplicación de los filtros paso-alto, con lo que podemos asumir que estas son *sparse*. Al igual que hicimos con el problema de aprendizaje de diccionarios, podemos imponer esta propiedad en un contexto bayesiano utilizando distribuciones de probabilidad adecuadas. Concretamente, utilizaremos la distribución *Spike-and-Slab* sobre cada píxel $x_{\gamma i}$ para definir nuestra información *a priori* sobre \mathbf{x}_γ

$$p(x_{\gamma i} | \alpha_{\gamma i}, \pi_\gamma) = \pi_\gamma \mathcal{N}(x_{\gamma i} | 0, \alpha_{\gamma i}^{-1}) + (1 - \pi_\gamma) \delta(x_{\gamma i}), \quad (\text{III.5})$$

donde $\delta(\cdot)$ denota la función delta de Dirac. Es importante observar que esta distribución es verdaderamente *sparse*, es decir, la variable $x_{\gamma i}$ toma exactamente el valor cero con probabilidad $1 - \pi_\gamma$. Esto no sucede con

otras distribuciones *sparse*, que simplemente imponen que algunos valores sean muy pequeños pero no exactamente cero. No obstante, la función delta en (III.5) dificulta en gran medida el proceso de inferencia. Es posible salvar este obstáculo con una simple pero ingeniosa reparametrización de $x_{\gamma i}$, expresando esta como el producto de dos variables aleatorias [108], una gaussiana de media cero $\tilde{x}_{\gamma i} \sim \mathcal{N}(0, \alpha_{\gamma i}^{-1})$ y una segunda que sigue una distribución de Bernoulli $s_{\gamma i} \sim \text{Bern}(s_{\gamma} | \pi_{\gamma})$, i. e.,

$$x_{\gamma i} = s_{\gamma i} \tilde{x}_{\gamma i}. \quad (\text{III.6})$$

Podemos ahora reescribir la *a priori* sobre $x_{\gamma i}$ a partir de estas dos componentes

$$p(\tilde{x}_{\gamma i}, s_{\gamma i} | \alpha_{\gamma i}, \pi_{\gamma}) = \mathcal{N}(\tilde{x}_{\gamma i} | 0, \alpha_{\gamma i}^{-1}) \pi_{\gamma}^{s_{\gamma i}} (1 - \pi_{\gamma})^{1 - s_{\gamma i}}, \quad (\text{III.7})$$

donde $s_{\gamma i} \in \{0, 1\}$.

Una vez más, compactaremos la notación usando $\tilde{\mathbf{x}}_{\Gamma} = \{\tilde{\mathbf{x}}_{\gamma}\}_{\gamma=1}^L$, $\mathbf{s}_{\Gamma} = \{\mathbf{s}_{\gamma}\}_{\gamma=1}^L$, $\boldsymbol{\alpha}_{\Gamma} = \{\boldsymbol{\alpha}_{\gamma}\}_{\gamma=1}^L$, $\boldsymbol{\pi}_{\Gamma} = \{\pi_{\gamma}\}_{\gamma=1}^L$, $\boldsymbol{\Omega} = \{\mathbf{h}, \boldsymbol{\alpha}_{\Gamma}, \boldsymbol{\pi}_{\Gamma}\}$, y $\boldsymbol{\Theta} = \{\tilde{\mathbf{x}}_{\Gamma}, \mathbf{s}_{\Gamma}\}$. Con esto, podemos expresar la distribución conjunta como

$$p(\mathbf{y}_{\Gamma}, \boldsymbol{\Theta} | \boldsymbol{\Omega}) = \left[\prod_{\gamma} \prod_i p(\tilde{x}_{\gamma i}, s_{\gamma i} | \alpha_{\gamma i}, \pi_{\gamma}) \right] \prod_{\gamma} p(\mathbf{y}_{\gamma} | \mathbf{h}, \tilde{\mathbf{x}}_{\gamma}, \mathbf{s}_{\gamma}). \quad (\text{III.8})$$

4. Inferencia variacional EM

El objetivo principal consiste ahora en la estimación del emborronamiento. Para ello debemos maximizar la verosimilitud

$$p(\mathbf{Y} | \boldsymbol{\Omega}) = \int p(\mathbf{Y}, \boldsymbol{\Theta} | \boldsymbol{\Omega}) d\boldsymbol{\Theta}, \quad (\text{III.9})$$

donde hemos optado por utilizar \mathbf{Y} en lugar de \mathbf{y}_{Γ} para facilitar la comparación con el método de inferencia variacional utilizado en el capítulo anterior. Más adelante retomaremos la notación específica de este problema para seguir los cálculos con mayor claridad.

El modelo variacional considera todos los parámetros como variables latentes, desapareciendo así la dependencia con $\boldsymbol{\Omega}$ que observamos en la verosimilitud y la distribución conjunta, compárese con (II.25). El algoritmo EM encuentra las soluciones de máxima verosimilitud (o MAP) para los parámetros del modelo, estando el emborronamiento \mathbf{h} entre ellos en nuestro caso, si bien es cierto que obtiene distribuciones completas sobre las variables latentes. El método variacional, sin embargo, obtiene como solución distribuciones de probabilidad sobre todas las variables.

Podemos introducir, de nuevo, una distribución $q(\boldsymbol{\Theta})$ sobre las variables latentes [59], comprobándose que, para cualquier $q(\boldsymbol{\Theta})$, se cumple

$$\log p(\mathbf{Y} | \boldsymbol{\Omega}) = \mathcal{L}(q(\boldsymbol{\Theta}), \boldsymbol{\Omega}) + \text{KL}(q(\boldsymbol{\Theta}) || p(\boldsymbol{\Theta} | \mathbf{Y}, \boldsymbol{\Omega})), \quad (\text{III.10})$$

donde

$$\mathcal{L}(q(\Theta), \Omega) = \int q(\Theta) \log \frac{p(\mathbf{Y}, \Theta | \Omega)}{q(\Theta)} d\Theta, \quad (\text{III.11})$$

es la conocida ELBO. Resulta crucial observar que la ELBO depende de dos términos, por un lado las distribuciones $q(\Theta)$ sobre las variable latentes y, por otro, los parámetros del modelo. Esta doble dependencia da lugar a un proceso iterativo con dos etapas para maximizar la verosimilitud.

En primer lugar, para el llamado «paso E», consideremos que fijamos el valor de los parámetros a $\Omega^{(t)}$. Dado que la divergencia es una cantidad positiva (o nula, en caso de tratarse de distribuciones idénticas), podemos maximizar la verosimilitud maximizando la ELBO con respecto a $q(\Theta)$

$$q^{(t+1)}(\Theta) = \arg \max_{q(\Theta)} \mathcal{L}(q(\Theta), \Omega^{(t)}), \quad (\text{III.12})$$

cuya solución, en caso de utilizar una factorización *mean-field* es análoga a la vista en el capítulo anterior, véase (II.30). El mejor de los casos se da cuando la divergencia se anula, es decir, cuando la distribución $q(\Theta)$ es idéntica a la *a posteriori* $p(\Theta | \mathbf{Y}, \Omega^{(t)})$, lo que provocará que la ELBO iguale al logaritmo de la verosimilitud.

A continuación, fijaremos $q(\Theta)$ y procederemos a la maximizar la ELBO con respecto a los parámetros

$$\Omega^{(t+1)} = \arg \max_{\Omega} \mathcal{L}(q^{(t+1)}(\Theta), \Omega). \quad (\text{III.13})$$

Una vez descrito el algoritmo EM general, veamos como realmente maximiza la verosimilitud. Fijando los parámetros Ω a su valor actual $\Omega^{(t)}$, en el paso E se maximiza la ELBO con respecto a la distribución $q(\Theta)$, observamos que el valor óptimo para esta se alcanzará cuando la divergencia sea, en el mejor de los casos, nula. Esto es fácil de comprobar, puesto que la verosimilitud $p(\mathbf{Y} | \Omega^{(t)})$ no depende de $q(\Theta)$, por tanto, el máximo del primer término de (III.10), la ELBO, se alcanzará cuando el segundo, la divergencia, sea mínimo. A continuación, en el paso M se fija la distribución $q(\Theta)$ y se maximiza la ELBO con respecto a Ω . Esta operación incrementará el valor de la ELBO (a no ser que se encuentre ya en su valor máximo), incrementando necesariamente la verosimilitud, puesto que la divergencia de Kullback-Leibler es siempre positiva. Es más, al haber cambiado los parámetros a su nuevo valor $\Omega^{(t+1)}$, la divergencia, que era nula tras el paso E, tomará un valor mayor que cero, suponiendo un incremento adicional al aumento de verosimilitud recién conseguido.

Particularicemos ahora esta teoría para nuestro problema. Podemos expresar la ELBO vista en (III.11) como

$$\mathcal{L}(q(\tilde{\mathbf{x}}_{\mathbf{r}}, \mathbf{s}_{\mathbf{r}}), \Omega) = \sum_{\mathbf{s}_{\mathbf{r}}} \int q(\tilde{\mathbf{x}}_{\mathbf{r}}, \mathbf{s}_{\mathbf{r}}) \log \frac{p(\mathbf{y}_{\mathbf{r}}, \tilde{\mathbf{x}}_{\mathbf{r}}, \mathbf{s}_{\mathbf{r}} | \Omega)}{q(\tilde{\mathbf{x}}_{\mathbf{r}}, \mathbf{s}_{\mathbf{r}})} d\tilde{\mathbf{x}}_{\mathbf{r}}, \quad (\text{III.14})$$

donde la distribución conjunta viene dada en (III.8). La maximización alterna de esta cota con respecto a $q(\tilde{\mathbf{x}}_\gamma, \mathbf{s}_\gamma)$ y a Ω constituye la base del algoritmo de inferencia EM y se detallan a continuación.

4.1. Paso E

Análogamente a lo visto en el caso de la inferencia variacional en el capítulo anterior, resulta conveniente asumir algún tipo de factorización de la distribución *a posteriori* $q(\Theta)$ para facilitar la solución. La descomposición $q(\Theta) = \prod_\gamma q(\tilde{\mathbf{x}}_\gamma, \mathbf{s}_\gamma)$, en la que $q(\tilde{\mathbf{x}}_\gamma, \mathbf{s}_\gamma) = \prod_i q(\tilde{x}_{\gamma i})q(s_{\gamma i})$ se ha usado ampliamente en la literatura, véase [120] por ejemplo. No obstante, como se clarifica en [108], esto produce una distribución unimodal que no se corresponde con la naturaleza multimodal de la *a posteriori*. Obsérvese que la verdadera *a posteriori* $p(\tilde{\mathbf{x}}_\gamma | \mathbf{y}_\gamma, \Omega) = \sum_{\mathbf{s}_\gamma} p(\tilde{\mathbf{x}}_\gamma, \mathbf{s}_\gamma | \mathbf{y}_\gamma, \Omega)$ involucra la suma sobre todas las 2^N combinaciones posibles de \mathbf{s}_γ , lo cual posibilita la existencia de múltiples modas. En consecuencia, y puesto que las parejas $(\tilde{x}_{\gamma i}, s_{\gamma i})$ están fuertemente correladas (recuérdese que $x_{\gamma i} = \tilde{x}_{\gamma i} s_{\gamma i}$), haremos un tratamiento conjunto de estas variables utilizando la factorización

$$q(\tilde{\mathbf{x}}_\gamma, \mathbf{s}_\gamma) = \prod_i q(\tilde{x}_{\gamma i}, s_{\gamma i}). \quad (\text{III.15})$$

Fijando el valor de los parámetros Ω y dada esta factorización, podemos obtener las distribuciones óptimas $q(\tilde{x}_{\gamma i}, s_{\gamma i})$ que maximizan la ELBO (III.14) utilizando (II.30), de lo que resulta

$$\begin{aligned} q(\tilde{x}_{\gamma i}, s_{\gamma i}) &= \frac{1}{Z} e^{\langle \log p(\mathbf{y}_\gamma, \tilde{\mathbf{x}}_\gamma, \mathbf{s}_\gamma | \Omega) \rangle_{\Theta \setminus \tilde{x}_{\gamma i}, s_{\gamma i}}} \\ &= \frac{1}{Z} e^{\langle \log p(\mathbf{y}_\gamma | \mathbf{h}, \tilde{\mathbf{x}}_\gamma, \mathbf{s}_\gamma) \rangle_{\Theta \setminus \tilde{x}_{\gamma i}, s_{\gamma i}}} \mathcal{N}(\tilde{x}_{\gamma i} | 0, \alpha_{\gamma i}^{-1}) \pi_{\gamma i}^{s_{\gamma i}} (1 - \pi_{\gamma i})^{1 - s_{\gamma i}}, \end{aligned} \quad (\text{III.16})$$

donde

$$\begin{aligned} \langle \log p(\mathbf{y}_\gamma | \mathbf{h}, \tilde{\mathbf{x}}_\gamma, \mathbf{s}_\gamma) \rangle_{\Theta \setminus \tilde{x}_{\gamma i}, s_{\gamma i}} &= -\frac{\beta_\gamma}{2} \langle \|\mathbf{y}_\gamma - \mathbf{H}(\tilde{\mathbf{x}}_\gamma \odot \mathbf{s}_\gamma)\|^2 \rangle_{\Theta \setminus \tilde{x}_{\gamma i}, s_{\gamma i}} \\ &\quad + \frac{N}{2} \log \beta_\gamma - \frac{N}{2} \log 2\pi, \end{aligned} \quad (\text{III.17})$$

y

$$\begin{aligned} \langle \|\mathbf{y}_\gamma - \mathbf{H}(\tilde{\mathbf{x}}_\gamma \odot \mathbf{s}_\gamma)\|^2 \rangle_{\Theta \setminus \tilde{x}_{\gamma i}, s_{\gamma i}} &= \langle \|\mathbf{y}_\gamma - \sum_{k \neq i} \mathbf{h}_k s_{\gamma k} \tilde{x}_{\gamma k} - \mathbf{h}_i s_{\gamma i} \tilde{x}_{\gamma i}\|^2 \rangle_{\Theta \setminus \tilde{x}_{\gamma i}, s_{\gamma i}} \\ &= -2 \langle \mathbf{y}_\gamma - \sum_{k \neq i} \mathbf{h}_k (s_{\gamma k} \tilde{x}_{\gamma k}) \rangle_{\Theta \setminus \tilde{x}_{\gamma i}, s_{\gamma i}}^\top \mathbf{h}_i \tilde{x}_{\gamma i} s_{\gamma i} + \|\mathbf{h}\|^2 \tilde{x}_{\gamma i}^2 s_{\gamma i} + \langle \|\mathbf{y}_\gamma - \sum_{k \neq i} \mathbf{h}_k s_{\gamma k} \tilde{x}_{\gamma k}\|^2 \rangle_{\Theta \setminus \tilde{x}_{\gamma i}, s_{\gamma i}}. \end{aligned} \quad (\text{III.18})$$

El vector \mathbf{h}_i denota la i -ésima comlumna de la matriz de convolución \mathbf{H} . Nótese, además, que $\mathbf{h}_i^T \mathbf{h}_i = \|\mathbf{h}\|^2, \forall i$ ya que se ha supuesto emborronamiento espacialmente invariante, \odot representa el producto de Hadamard y que las esperanzas que aparecen en (III.18) se calculan usando las distribuciones $q(\tilde{x}_{\gamma k}, s_{\gamma k})$.

Para el cálculo explícito de la distribución *a posteriori* y, sin pérdida de generalidad, es importante observar que podemos expresar esta como

$$q(\tilde{x}_{\gamma i}, s_{\gamma i}) = q(\tilde{x}_{\gamma i} | s_{\gamma i}) q(s_{\gamma i}), \quad (\text{III.19})$$

y obtener separadamente $q(\tilde{x}_{\gamma i} | s_{\gamma i})$ y $q(s_{\gamma i})$.

En primer lugar, particularizando la expresión (III.16) para $s_{\gamma i} = 0$ y $s_{\gamma i} = 1$, una vez sustituidos los resultados (III.17) y (III.18), obtenemos

$$q(\tilde{x}_{\gamma i} | s_{\gamma i} = 0) \propto \exp \left\{ -\frac{\alpha_{\gamma i}}{2} \tilde{x}_{\gamma i}^2 \right\}, \quad (\text{III.20})$$

$$q(\tilde{x}_{\gamma i} | s_{\gamma i} = 1) \propto \exp \left\{ \beta_{\gamma} (\mathbf{y}_{\gamma} - \sum_{k \neq i} \mathbf{h}_k \langle s_{\gamma k} \tilde{x}_{\gamma k} \rangle)^T \mathbf{h}_i \tilde{x}_{\gamma i} - \frac{\beta_{\gamma}}{2} \|\mathbf{h}\|^2 \tilde{x}_{\gamma i}^2 - \frac{\alpha_{\gamma i}}{2} \tilde{x}_{\gamma i}^2 \right\}. \quad (\text{III.21})$$

Podemos deducir fácilmente de la forma cuadrática de los exponentes de (III.20) y (III.21) que se trata de dos distribuciones gaussianas,

$$q(\tilde{x}_{\gamma i} | s_{\gamma i} = 0) = \mathcal{N}(\tilde{x}_{\gamma i} | 0, \alpha_{\gamma i}^{-1}), \quad (\text{III.22})$$

$$q(\tilde{x}_{\gamma i} | s_{\gamma i} = 1) = \mathcal{N}(\tilde{x}_{\gamma i} | \mu_{x_{\gamma i}}, \rho_{\gamma i}^{-1}), \quad (\text{III.23})$$

que se pueden expresar de forma más compacta como

$$q(\tilde{x}_{\gamma i} | s_{\gamma i}) = \mathcal{N}(\tilde{x}_{\gamma i} | s_{\gamma i} \mu_{x_{\gamma i}}, s_{\gamma i} \rho_{\gamma i}^{-1} + (1 - s_{\gamma i}) \alpha_{\gamma i}^{-1}), \quad (\text{III.24})$$

donde

$$\mu_{x_{\gamma i}} = \frac{\beta_{\gamma}}{\rho_{\gamma}} \mathbf{h}_i^T (\mathbf{y}_{\gamma} - \sum_{k \neq i} \langle s_{\gamma k} \tilde{x}_{\gamma k} \rangle \mathbf{h}_k), \quad (\text{III.25})$$

$$\rho_{\gamma i} = \beta_{\gamma} \|\mathbf{h}\|^2 + \alpha_{\gamma i}, \quad (\text{III.26})$$

representan la media y la precisión de los *slabs* respectivamente. La ecuación (III.26) admite una interpretación interesante: por un lado, la varianza de los *slabs*, $\rho_{\gamma i}^{-1}$, es siempre menor que la varianza de la *a priori*, $\alpha_{\gamma i}^{-1}$, dado que incorpora información de las observaciones, pero, aún más relevante resulta la dependencia de la primera con el emborronamiento⁴. Valores altos de $\|\mathbf{h}\|^2$ implican que la PSF estará compuesta por un número reducido de píxeles significativos por lo que la (pseudo)imagen podrá estimarse fielmente. Según vaya incrementándose el número de

píxeles afectados por el emborronamiento, la norma de este decrecerá y, consecuentemente, así lo hará la confianza en la estimación.

Veamos ahora el cálculo de la distribución $q(s_{\gamma i})$, la cual es, obviamente, una distribución de Bernoulli y, por tanto, bastará con encontrar su parámetro, al que denotamos por $\omega_{\gamma i}$. Es bien conocido que el parámetro de una distribución de Bernoulli coincide con la probabilidad de que la variable tome el valor 1, esto es,

$$\omega_{\gamma i} = q(s_{\gamma i} = 1). \quad (\text{III.27})$$

Consecuentemente, podríamos obtener este valor marginalizando (III.16) sobre $\tilde{x}_{\gamma i}$, pero necesitaríamos conocer el valor de la función de partición, \mathcal{Z} , de la distribución $q(\tilde{x}_{\gamma i}, s_{\gamma i})$, véase (III.30). Para evitar este cálculo observemos que podemos reescribir $\omega_{\gamma i}$ como

$$\omega_{\gamma i} = \frac{1}{1 + e^{-u_{\gamma i}}}, \quad (\text{III.28})$$

con

$$u_{\gamma i} = \log q(s_{\gamma i} = 1) - \log q(s_{\gamma i} = 0). \quad (\text{III.29})$$

En esta sustracción se cancela la función de partición \mathcal{Z} , solucionando el mencionado problema. Partiendo de (III.16) tenemos

$$\begin{aligned} \log q(s_{\gamma i} = 1) &= \log \int q(\tilde{x}_{\gamma i}, 1) d\tilde{x}_{\gamma i} = \frac{N}{2} \log \beta_{\gamma} - \frac{N}{2} \log 2\pi + \frac{1}{2} \log \alpha_{\gamma i} \\ &\quad - \frac{\beta_{\gamma}}{2} \langle \|\mathbf{y}_{\gamma} - \sum_{k \neq i} \mathbf{h}_k x_{\gamma k}\|^2 \rangle + \log \pi_{\gamma} - \frac{1}{2} \log \rho_{\gamma i} \\ &\quad + \frac{\beta_{\gamma}^2}{2\rho_{\gamma i}} (\mathbf{h}_i^T (\mathbf{y}_{\gamma} - \sum_{k \neq i} \langle s_{\gamma k} \tilde{x}_{\gamma k} \rangle \mathbf{h}_k))^2 - \log \mathcal{Z}, \end{aligned} \quad (\text{III.30})$$

y

$$\begin{aligned} \log q(s_{\gamma i} = 0) &= \log \int q(\tilde{x}_{\gamma i}, 0) d\tilde{x}_{\gamma i} = \frac{N}{2} \log \beta_{\gamma} - \frac{N}{2} \log 2\pi \\ &\quad - \frac{\beta_{\gamma}}{2} \langle \|\mathbf{y}_{\gamma} - \sum_{k \neq i} \mathbf{h}_k x_{\gamma k}\|^2 \rangle + \log(1 - \pi_{\gamma}) - \log \mathcal{Z}, \end{aligned} \quad (\text{III.31})$$

lo que produce

$$u_{\gamma i} = \log \frac{\pi_{\gamma}}{1 - \pi_{\gamma}} + \frac{1}{2} \log \frac{\alpha_{\gamma i}}{\rho_{\gamma i}} + \frac{\rho_{\gamma i}}{2} \mu_{x_{\gamma i}}^2, \quad (\text{III.32})$$

⁴Nótese que, dada la restricción impuesta sobre el emborronamiento ($\sum_i h_i = 1, h_i \geq 0, \forall i$), los valores de $\|\mathbf{h}\|^2$ oscilan entre 1 y $1/I$, donde I es el número de valores distintos de cero en la PSF. El valor máximo, 1, se alcanza únicamente en el caso de que la PSF conste únicamente de un píxel, lo que supondría un emborronamiento nulo. El valor mínimo se puede obtener fácilmente imponiendo $\nabla \|\mathbf{h}\|^2 = 0$, sujeto a la restricción ya comentada, lo que resulta en $h_i = 1/I, \forall i$, que se corresponde con PSFs cuyos píxeles activos tienen todos el mismo valor.

y nos permite finalmente calcular el parámetro $\omega_{\gamma i} = \mathbf{q}(s_{\gamma i} = 1)$ de la distribución a partir de (III.28).

Finalmente, podemos expresar la *a posteriori* completa usando (III.19)

$$\begin{aligned} \mathbf{q}(\tilde{x}_{\gamma i}, s_{\gamma i}) &= \mathbf{q}(s_{\gamma i})\mathbf{q}(\tilde{x}_{\gamma i}|s_{\gamma i}) = \omega_{\gamma i}^{s_{\gamma i}}(1 - \omega_{\gamma i})^{1-s_{\gamma i}} \\ &\quad \times \mathcal{N}(\tilde{x}_{\gamma i}|s_{\gamma i}\mu_{x_{\gamma i}}, s_{\gamma i}\rho_{\gamma i}^{-1} + (1 - s_{\gamma i})\alpha_{\gamma i}^{-1}), \end{aligned} \quad (\text{III.33})$$

y usando esta distribución podemos obtener la media y valor cuadrático medio de las variables latentes $x_{\gamma i}$

$$\begin{aligned} \langle x_{\gamma i} \rangle &= \langle s_{\gamma i}\tilde{x}_{\gamma i} \rangle = \sum_{s_{\gamma i}} \int s_{\gamma i}\tilde{x}_{\gamma i}\mathbf{q}(\tilde{x}_{\gamma i}, s_{\gamma i})d\tilde{x}_{\gamma i} = \int \tilde{x}_{\gamma i}\mathbf{q}(\tilde{x}_{\gamma i}, 1)d\tilde{x}_{\gamma i} \\ &= \omega_{\gamma i} \int \tilde{x}_{\gamma i}\mathbf{q}(\tilde{x}_{\gamma i}|s_{\gamma i} = 1)d\tilde{x}_{\gamma i} = \omega_{\gamma i}\mu_{x_{\gamma i}}, \end{aligned} \quad (\text{III.34})$$

$$\begin{aligned} \langle x_{\gamma i}^2 \rangle &= \langle s_{\gamma i}^2\tilde{x}_{\gamma i}^2 \rangle = \sum_{s_{\gamma i}} \int s_{\gamma i}^2\tilde{x}_{\gamma i}^2\mathbf{q}(\tilde{x}_{\gamma i}, s_{\gamma i})d\tilde{x}_{\gamma i} = \int \tilde{x}_{\gamma i}^2\mathbf{q}(\tilde{x}_{\gamma i}, 1)d\tilde{x}_{\gamma i} \\ &= \omega_{\gamma i} \int \tilde{x}_{\gamma i}^2\mathbf{q}(\tilde{x}_{\gamma i}|s_{\gamma i} = 1)d\tilde{x}_{\gamma i} = \omega_{\gamma i}(\mu_{x_{\gamma i}}^2 + \rho_{\gamma i}^{-1}), \end{aligned} \quad (\text{III.35})$$

donde $\mu_{x_{\gamma i}}$, $\rho_{\gamma i}$ y $\omega_{\gamma i}$ se han definido en (III.25), (III.26) y (III.28) respectivamente. Estas cantidades que nos harán falta para los cálculos subsiguientes.

4.2. Paso M

Una vez calculada la distribución sobre las variables latentes $\mathbf{q}(\tilde{\mathbf{x}}_{\Gamma}, \mathbf{s}_{\Gamma})$, fijamos el valor de esta, de acuerdo con lo visto anteriormente, y procedemos a la actualización de los parámetros $\mathbf{\Omega} = \{\mathbf{h}, \boldsymbol{\alpha}_{\Gamma}, \boldsymbol{\pi}_{\Gamma}\}$ maximizando la ELBO (III.14) o, equivalentemente, su logaritmo

$$\mathcal{F}(\mathbf{\Omega}) = \log \mathcal{L}(\mathbf{q}^{(t+1)}(\tilde{\mathbf{x}}_{\Gamma}, \mathbf{s}_{\Gamma}), \mathbf{\Omega}) \quad (\text{III.36})$$

con respecto a $\mathbf{\Omega}$.

En primer lugar, tomando únicamente los términos que depende de π_{γ} en la ELBO, tenemos

$$\mathcal{F}(\pi_{\gamma}) = \sum_i \left[\langle s_{\gamma i} \rangle \log \pi_{\gamma} + (1 - \langle s_{\gamma i} \rangle) \log(1 - \pi_{\gamma}) \right] \quad (\text{III.37})$$

donde $0 < \pi_{\gamma} < 1$. Derivando esta expresión e igualando a cero obtenemos el valor óptimo en

$$\pi_{\gamma} = \frac{\sum_i \omega_{\gamma i}}{N}. \quad (\text{III.38})$$

Análogamente, para $\alpha_{\gamma i}$ tenemos que maximizar

$$\begin{aligned}\mathcal{F}(\alpha_{\gamma i}) &= \omega_{\gamma i} \int \mathcal{N}(\tilde{x}_{\gamma i} | \mu_{x_{\gamma i}}, \rho_{\gamma i}^{-1}) \log \mathcal{N}(\tilde{x}_{\gamma i} | 0, \alpha_{\gamma i}^{-1}) d\tilde{x}_{\gamma i} \\ &= \omega_{\gamma i} \left[\frac{1}{2} \log \alpha_{\gamma i} - \frac{1}{2} \alpha_{\gamma i} \langle \tilde{x}_{\gamma i}^2 \rangle_{\mathcal{N}(\tilde{x}_{\gamma i} | \mu_{x_{\gamma i}}, \rho_{\gamma i}^{-1})} \right] + \text{const},\end{aligned}\quad (\text{III.39})$$

lo que resulta en

$$\alpha_{\gamma i}^{-1} = \mu_{x_{\gamma i}}^2 + \rho_{\gamma i}^{-1}. \quad (\text{III.40})$$

Por último, para la estimación del emborronamiento, la maximización de $\mathcal{F}(\mathbf{h})$ es equivalente a la minimización de

$$f(\mathbf{h}) = \sum_{\gamma} \langle \|\mathbf{y}_{\gamma} - \mathbf{H}\mathbf{x}_{\gamma}\|^2 \rangle_{\Theta} = \sum_{\gamma} \left[\|\mathbf{y}_{\gamma} - \mathbf{H}\langle \mathbf{x}_{\gamma} \rangle\|^2 + \mathbf{h}^T \mathbf{D}_{\gamma} \mathbf{h} \right], \quad (\text{III.41})$$

restringido a $h_i \geq 0$, $\sum_i h_i = 1$, donde \mathbf{D}_{γ} es una matriz diagonal de la forma $\mathbf{D}_{\gamma} = \sum_i (\langle x_{\gamma i}^2 \rangle - \langle x_{\gamma i} \rangle^2) \mathbf{I}$. La optimización de esta función puede realizarse de manera eficiente utilizando programación cuadrática, véase [102].

4.3. Algoritmo BID

El algoritmo EM para la estimación del emborronamiento detallado en la sección anterior viene resumido en el algoritmo 5. Un aspecto crucial en deconvolución ciega es la estimación inicial del emborronamiento $\mathbf{h}^{(0)}$ dada la existencia de múltiples extremos locales [101], especialmente cuando trabajamos con filtros de gran tamaño. Una práctica habitual para atajar este problema consiste en utilizar una aproximación multiescala [101, 102, 104]. Este proceso consiste en realizar una primera estimación del emborronamiento a partir de una versión reducida de la imagen degradada original, donde la escala del emborronamiento también será menor, facilitando su estimación. Esta primera estimación será usada como estimación inicial para la siguiente escala, tras el correspondiente reescalado. Esta es la aproximación que utilizamos en los trabajos preliminares enviados a congresos [121, 122]. No obstante, el algoritmo final que presentamos y cuyos resultados aparecen expuestos en el siguiente apartado utiliza como estimación inicial el resultado producido por [104]. Este es un algoritmo multiescala, también en el dominio filtrado, que, como ya comentábamos en la revisión bibliográfica, utiliza distribuciones supergaussianas de Huber pero, más importantemente, utiliza el método ADMM (*Alternating Direction Method of Multipliers*) que permite el uso de la FFT en la inferencia, acelerando en gran medida los cálculos. El algoritmo recién presentado trabaja secuencialmente píxel a píxel, no pudiendo así hacer uso de cálculos basados en FFT. Por tanto, nuestro método obtiene una estimación razonable a partir de [104] aprovechando

Algoritmo 5 Algoritmo EM para la estimación de la PSF

ENTRADA: \mathbf{y}_Γ , β_Γ y valores iniciales $\mathbf{h}^{(0)}$, $\boldsymbol{\pi}_\Gamma^{(0)}$, $\boldsymbol{\alpha}_\Gamma^{(0)}$ and $\langle \mathbf{x}_\Gamma \rangle^{(0)}$.

- 1: Set $k = 0$.
 - 2: **while** $\Delta\text{ELBO} > \text{tol}$ **do**
 - 3: PASO E:
 - 4: Actualizar $\mu_{x_{\gamma i}}^{(t+1)}$, $\rho_{\gamma i}^{(t+1)}$ y $\omega_{\gamma i}^{(t+1)} \forall \gamma, \forall i$, según (III.25), (III.26) y (III.28) respectivamente.
 - 5: Actualizar $\langle x_{\gamma i} \rangle^{(t+1)} \forall i$ a partir de (III.34).
 - 6: PASO M:
 - 7: Update $\boldsymbol{\pi}_\Gamma^{(t+1)}$ from (III.38), and $\boldsymbol{\alpha}_\Gamma^{(t+1)}$ from (III.40).
 - 8: Update $\mathbf{h}^{(t+1)}$ using (III.41).
 - 9: Set $t = t + 1$.
 - 10: **end while**
- SALIDA:** $\hat{\mathbf{h}} = \mathbf{h}^{(t+1)}$.

su rapidez y trabaja únicamente en la última escala, refinando la solución gracias a las ventajas de la *a priori* utilizada.

Tras la estimación del emborronamiento, ha de estimarse la imagen nítida subyacente. Recuérdese que, al trabajar en el dominio filtrado, no obtenemos una estimación de la imagen latente. Por ende, utilizaremos un algoritmo de deconvolución no ciega para la estimación de esta. Estimaremos la imagen nítida resolviendo

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\hat{\mathbf{H}}\mathbf{x} - \mathbf{y}\|^2 + \frac{\lambda}{p} \sum_{\gamma} \|\mathbf{x}_{\gamma}\|_p, \quad (\text{III.42})$$

y utilizando el método iterativo de [123]. Utilizaremos un valor de p igual a 0.8, habitual en la literatura [104, 124, 125].

Como apunte final obsérvese que, para imágenes en color, la estimación del emborronamiento se realiza a partir de la luminancia, utilizándose posteriormente esta estimación para la deconvolución independiente de cada una de las bandas de color.

5. Experimentos

Desarrollamos un análisis del algoritmo en dos vertientes. Por un lado, estudiaremos el comportamiento experimental de la distribución *a priori* propuesta, analizando cómo afecta a la estimación de la PSF y, por otro, mostraremos una comparativa con métodos del estado del arte de su funcionamiento en términos de calidad de la imagen restaurada, incluyendo un breve análisis de tiempo.

Para evitar redundancia en los subsiguientes apartados, comentaremos aquí la inicialización de los diferentes parámetros del método:

- Los filtros \mathbf{f}_γ empleados para obtener las pseudoobservaciones son

$$\mathbf{f}_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad \mathbf{f}_2 = \mathbf{f}_1^T.$$

Determinamos empíricamente que el uso de más filtros no produce ganancias significativas en los resultados además de tener un alto coste computacional.

- $\langle \mathbf{x}_\Gamma \rangle$ se inicializa al valor de las pseudoobservaciones.
- Las variables π_γ se inicializan a 0.01, lo que equivale a suponer que habrá *a priori* pocos píxeles correspondientes a fronteras en la imagen nítida.
- $\alpha_{\gamma i} = 32, \forall i$. La estimación de $\alpha_{\gamma i}$ por medio de (III.40) puede producir estimaciones ruidosas. Por ello, se suavizan dichas estimaciones con el filtro

$$\frac{1}{12} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 8 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

- Consideramos conocida la precisión de ruido y fijamos su valor según [85] a $\beta = 10^4$, con $\beta_\gamma = \beta / \|\mathbf{f}_\gamma\|^2, \forall \gamma$. Nótese que podemos obtener estimaciones precisas de este parámetro a partir de regiones planas de la imagen o utilizando alguno de los múltiples métodos existentes en la literatura [126–129]. Experimentalmente encontramos una baja sensibilidad de nuestro algoritmo a este parámetro.
- La estimación inicial de la PSF, $\mathbf{h}^{(0)}$, se obtiene, como habíamos comentado, a partir de [104].

5.1. Comportamiento de la distribución *Spike-and-Slab*

Vamos a analizar en este apartado el funcionamiento de la *a priori Spike-and-Slab*. Comentábamos que esta distribución tiene la capacidad de descartar información irrelevante o perjudicial para la estimación del emborronamiento. Veremos numéricamente que efectivamente así es.

Para facilitar el entendimiento de este apartado, recordemos las expresiones de la *a priori* y la *a posteriori*

$$\begin{aligned} p(\tilde{x}_{\gamma i}, s_{\gamma i}) &= \mathcal{N}(\tilde{x}_{\gamma i} | 0, \alpha_{\gamma i}^{-1}) \pi_{\gamma}^{s_{\gamma i}} (1 - \pi_{\gamma})^{1 - s_{\gamma i}} \\ q(\tilde{x}_{\gamma i}, s_{\gamma i}) &= \mathcal{N}(\tilde{x}_{\gamma i} | s_{\gamma i} \mu_{x_{\gamma i}}, s_{\gamma i} \rho_{\gamma i}^{-1} + (1 - s_{\gamma i}) \alpha_{\gamma i}^{-1}) \omega_{\gamma i}^{s_{\gamma i}} (1 - \omega_{\gamma i})^{1 - s_{\gamma i}}, \end{aligned}$$

con $x_{\gamma i} = \tilde{x}_{\gamma i} s_{\gamma i}$, que podemos también expresar en términos de funciones delta deshaciendo la reparametrización

$$\begin{aligned} p(x_{\gamma i}) &= \pi_{\gamma} \mathcal{N}(x_{\gamma i} | 0, \alpha_{\gamma i}^{-1}) + (1 - \pi_{\gamma}) \delta(x_{\gamma i}) \\ q(x_{\gamma i}) &= \omega_{\gamma i} \mathcal{N}(x_{\gamma i} | \mu_{x_{\gamma i}}, \rho_{\gamma i}^{-1}) + (1 - \omega_{\gamma i}) \delta(x_{\gamma i}). \end{aligned}$$

La primera diferencia resulta ahora obvia: el valor medio de un píxel perteneciente a una frontera ($s_{\gamma i} = 1$) se ha actualizado a $\mu_{x_{\gamma i}}$ en la *a posteriori*, sobre el que tendremos una mayor o menor incertidumbre dependiendo de la PSF estimada (recuérdese que $\rho_{\gamma i} = \beta_{\gamma} \|\mathbf{h}\|^2 + \alpha_{\gamma i}$).

Veamos ahora un ejemplo numérico real. En la fig. 2 podemos ver para $\gamma = 1$, gradientes verticales, y $\gamma = 2$, los horizontales, los diferentes parámetros que gobiernan tanto la distribución *a priori* como la *a posteriori*, teniendo $\pi_{\Gamma} = \{0.057, 0.028\}$. En primer lugar, el parámetro π_{γ} , estimado como la media de los $\omega_{\gamma i}$ (III.38) representa la probabilidad *a priori* del *slab*. Por ejemplo, para $\gamma = 1$ tenemos una probabilidad *a priori* $1 - \pi_1 = 0.943$ de que el valor del píxel será exactamente cero, tratándose de una región llana y con un probabilidad $\pi_1 = 0.057$ el píxel provendrá de una gaussiana de media nula con varianza α_{1i}^{-1} . Podemos observar en los paneles correspondientes a $\alpha_{\gamma i}^{-1}$ de la fig. 2 que los valores estimados para esta variable son mayores en las regiones correspondientes a fronteras, indicando una mayor probabilidad de que estos píxeles sean distintos de cero. Adicionalmente, como podemos observar en la expresión de la *a posteriori*, esta sigue la misma distribución que la *a priori*. La media de esta distribución, que podemos ver en la misma figura denotada por $\langle \mathbf{x}_{\gamma} \rangle$, muestra que el algoritmo diseñado recupera imágenes con bordes nítidos.

Esta figura también refleja los valores estimados de ω_{γ} . Estos presentan solo unos pocos valores cercanos a uno, indicando que esos píxeles pertenecerán con gran probabilidad a fronteras en la imagen nítida, mientras la mayoría de los valores están muy cercanos a cero. Nótese que la mayoría de estos píxeles pertenecen a regiones planas o muy texturizadas de la imagen que, bien no aportan ninguna información para la estimación del emborronamiento, bien aportan información poco fiable, dado que es difícil discernir entre ruido y textura. Es importante resaltar que algunas fronteras marcadas de la imagen han sido descartadas o suavizadas, como, por ejemplo, aquellas pertenecientes a píxeles donde el ruido es predominante o están afectadas por artefactos de anillo. Ver, por ejemplo, los píxeles en torno a la cabeza o el codo en $\mu_{\mathbf{x}_1}$ que han sido puestos a cero en $\langle \mathbf{x}_1 \rangle$. Concluimos que únicamente los píxeles que aportan la información de mayor relevancia para la estimación de la PSF son escogidos por la distribución *Spike-and-Slab*, siendo moderadamente suavizados en la estimación de $\langle \mathbf{x}_{\gamma i} \rangle$ eliminado ruido y robusteciendo la estimación.

Por último, la figura 2 también muestra la varianza de la distribución

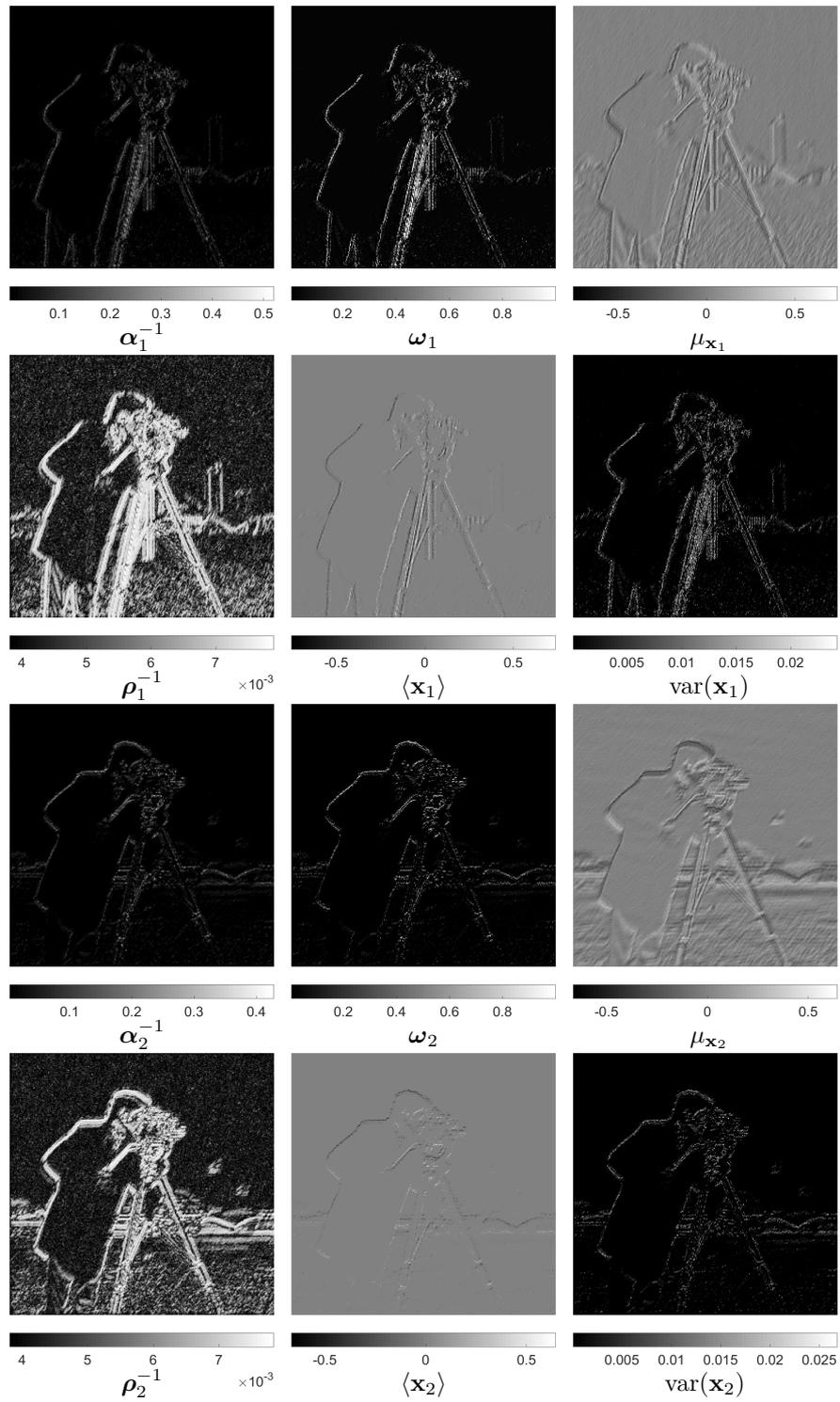


Figura 2: Parámetros de las distribuciones *a priori* y *a posteriori*.

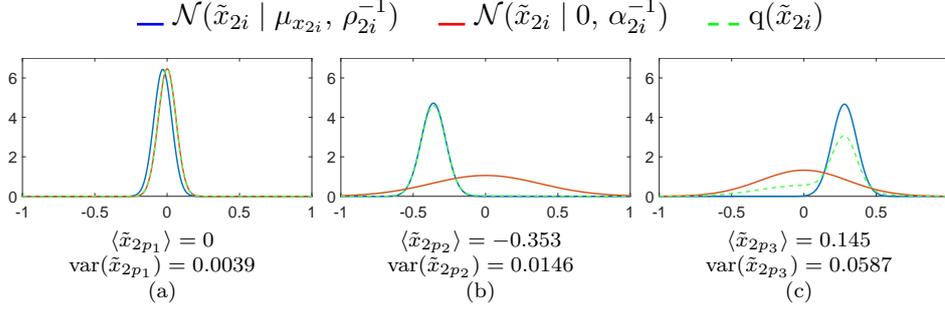


Figura 3: *A posteriori* $q(\tilde{x}_{\gamma i})$ para diferentes píxeles y $\gamma = 2$. (a) Píxel en región plana: $\mu_{x_{2p_1}} = -0.03$, $\rho_{2p_1}^{-1} = 0.0038$, $\alpha_{2p_1}^{-1} = 0.0038$, $\omega_{2p_1} = 0.008$. (b) Píxel de frontera en el codo: $\mu_{x_{2p_2}} = -0.36$, $\rho_{2p_2}^{-1} = 0.0071$, $\alpha_{2p_2}^{-1} = 0.1429$, $\omega_{2p_2} = 0.97$. (c) Píxel de frontera bajo el codo: $\mu_{x_{2p_3}} = 0.28$, $\rho_{2p_3}^{-1} = 0.0073$, $\alpha_{2p_3}^{-1} = 0.0909$, $\omega_{2p_3} = 0.58$.

a posteriori completa $\text{var}(x_{\gamma i}) = \omega_{\gamma i} \rho_{\gamma i}^{-1} + (1 - \omega_{\gamma i}) \omega_{\gamma i} \mu_{x_{\gamma i}}^2$ en los paneles denotados por $\text{var}(\mathbf{x}_{\gamma})$. Comprobamos que cuando $\omega_{\gamma i}$ tiene un valor de cero, la varianza también toma este valor ya que el píxel corresponde a un *spike* en la *a priori*. En cambio, cuando $\omega_{\gamma i}$ es uno, la varianza es igual $\rho_{\gamma i}^{-1}$ y, si toma un valor intermedio, la varianza aumenta con valores mayores para $\mu_{\gamma i}$ y cercanos a 0.5 para $\omega_{\gamma i}$, esto es, la varianza aumenta cuando tenemos una gran incertidumbre sobre si el píxel pertenece a una frontera o no. Podemos corroborar este comportamiento estudiando la distribución *a posteriori* sobre $\tilde{x}_{\gamma i}$, que podemos obtener por marginalización

$$\begin{aligned} q(\tilde{x}_{\gamma i}) &= \sum_{s_{\gamma i} \in \{0,1\}} q(\tilde{x}_{\gamma i} | s_{\gamma i}) q(s_{\gamma i}) \\ &= \omega_{\gamma i} \mathcal{N}(\tilde{x}_{\gamma i} | \mu_{x_{\gamma i}}, \rho_{\gamma i}^{-1}) + (1 - \omega_{\gamma i}) \mathcal{N}(\tilde{x}_{\gamma i} | 0, \alpha_{\gamma i}^{-1}), \end{aligned} \quad (\text{III.43})$$

que, como vemos, es una combinación lineal convexa de dos gaussianas, con media y varianza dadas por

$$\langle \tilde{x}_{\gamma i} \rangle = \omega_{\gamma i} \mu_{x_{\gamma i}}, \quad (\text{III.44})$$

$$\text{var}(\tilde{x}_{\gamma i}) = \omega_{\gamma i} \rho_{\gamma i}^{-1} + (1 - \omega_{\gamma i}) \alpha_{\gamma i}^{-1} + (1 - \omega_{\gamma i}) \omega_{\gamma i} \mu_{x_{\gamma i}}^2. \quad (\text{III.45})$$

Ilustramos esta distribución para tres píxeles arquetípicos de regiones distintas p_1 , p_2 y p_3 en la figura 3. Se puede observar que, en regiones planas, fig. 3(a), el valor de $\omega_{\gamma i}$ es prácticamente nulo y, por tanto, la *a priori* «tira» del valor de píxel hacia cero, eliminando así el ruido de la observación, teniendo además una varianza $\alpha_{\gamma i}^{-1}$ muy pequeña. En cambio, en regiones de frontera, fig. 3(b), donde $\omega_{\gamma i}$ es muy cercano a la unidad, la estimación depende, sobre todo, de la observación. En este caso, la varianza de la distribución es más cercana al valor de $\rho_{\gamma i}^{-1}$. Por último, podemos encontrar píxeles para los cuales $\omega_{\gamma i}$ presenta un valor más

intermedio, fig. 3(c). En estos casos, la varianza es superior a $\rho_{\gamma i}^{-1}$, lo que refleja incertidumbre en el valor estimado de $\tilde{x}_{\gamma i}$. Como vemos, la *a priori* nos permite la selección automática de los píxeles relevantes pertenecientes a fronteras, contribuyendo así a una mejor estimación del emborronamiento.

De haber utilizado una aproximación MAP en lugar de aproximar la *a posteriori* completa, no habríamos podido considerar la incertidumbre y, en consecuencia, $\omega_{\gamma i}$ habría sido exactamente cero, si $q(\tilde{x}_{\gamma i}|s_{\gamma i} = 0) < q(\tilde{x}_{\gamma i}|s_{\gamma i} = 1)$, o uno en otro caso. Esto supondría que $\tilde{x}_{\gamma i}$ tomaría el valor cero cuando $\omega_{\gamma i} = 0$ o $\mu_{\gamma i}$ en el caso complementario.

Como ya comentábamos en la revisión bibliográfica, existen diversos métodos que utilizan la idea de seleccionar solo unos pocos píxeles correspondientes a bordes en la imagen para estimar la PSF [94, 95] o el trabajo presentado en [130]. Sin embargo, estos métodos dependen de parámetros cuyos valores se fijan empíricamente teniendo incluso que escoger, en algunos de estos métodos, los píxeles de las fronteras manualmente. En nuestro algoritmo tanto los píxeles seleccionados para la estimación del filtro como los parámetros relativos al modelo se estiman automáticamente en el seno del marco de inferencia variacional.

5.2. Análisis de resultados

Veremos en este apartado una comparativa de las prestaciones del algoritmo propuesto comparado con tres métodos del estado del arte. Entre estos métodos incluimos un primer método que también utiliza inferencia variacional y otros dos métodos recientes con inferencia MAP. El primer método [104], al que nos referiremos por Zhou17 utiliza, como se vio, supergaussianas de Huber como *a priori* sobre la imagen, mientras que los segundos, Pan17 [89] y Perrone15 [92], utilizan el «canal oscuro» y distribuciones *a priori* logarítmicas respectivamente. El nombre elegido para los distintos métodos responde al patrón *<nombre del primer autor> + <año de publicación>*.

En primer lugar realizaremos una comparativa en un entorno controlado, en el que disponemos de las imágenes nítidas originales así como de los núcleos de emborronamiento, lo que nos permitirá realizar una comparación numérica. A continuación, se muestra un experimento con imágenes borrosas reales, para las que no disponemos de las imágenes nítidas originales ni de las PSFs causantes del emborronamiento. Por ello, la comparación de resultados será simplemente visual. En tercer lugar se hará un pequeño estudio del impacto que tiene el nivel de ruido en las observaciones en el proceso de estimación del emborronamiento y, por último, mostramos un breve análisis de tiempo de ejecución de los diferentes métodos. Nótese que, para garantizar una justa comparación de los diferentes métodos de convolución, una vez realizada la estimación de la

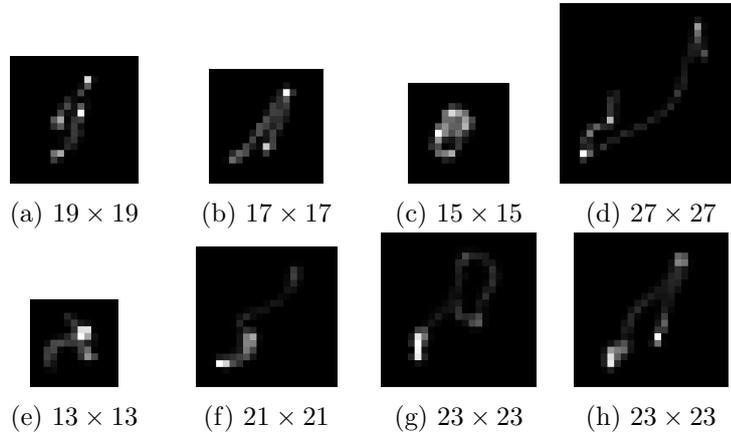


Figura 4: PSF utilizadas en el conjunto de imágenes de Levin.

PSF, la restauración final para todos los métodos se efectúa con el algoritmo de restauración no ciega presentado en [123].

Para el primer experimento utilizaremos el conjunto de imágenes propuesto en [85], el cual contiene 4 imágenes degradadas utilizando 8 núcleos de emborronamiento diferentes que podemos visualizar en la fig. 4. Todas las imágenes tienen un tamaño idéntico de 256×256 y las PSF tienen un tamaño variable que oscila entre 13×13 , la menor, y 27×27 , la mayor. Basamos la comparativa en dos métricas de referencia: PSNR y SSIM, mostrando gráficamente sus valores en las figuras 5 y 6 respectivamente. En cada uno de los paneles, para una PSF dada, se muestran los valores de PSNR o SSIM obtenidos por los diferentes métodos para cada una de las imágenes. Adicionalmente, mostramos un resumen de estos resultados en la tabla 1, donde para cada método, la columna de la izquierda corresponde a la PSNR media de las cuatro imágenes y la de la derecha al SSIM. Los valores entre paréntesis muestran la desviación típica de estas medidas. Fijándonos en el resumen de resultados de la tabla 1, vemos que el método propuesto obtiene una mayor PSNR en 4 de los 8 emborronamientos utilizados, con un comportamiento similar en SSIM. Observamos también la superioridad global del método propuesto reflejada tanto en los valores de PSNR y SSIM como en sus desviaciones típicas menores. Si comparamos los resultados del método propuesto con Zhou17, el cual sirve de inicialización, observamos unos mejores resultados para la amplia mayoría de imágenes y PSF, ver fig. 5 y fig. 6, obteniendo una ganancia considerable en términos de PSNR y SSIM en 7 de las 8 PSF. Concretamente, la ganancia global es de $+0.28$ dB con una desviación típica menor, teniendo una pequeña pérdida de calidad en los casos en que se empeora el resultado y una ganancia media de $+0.51$ dB en el resto. El algoritmo propuesto mejora los

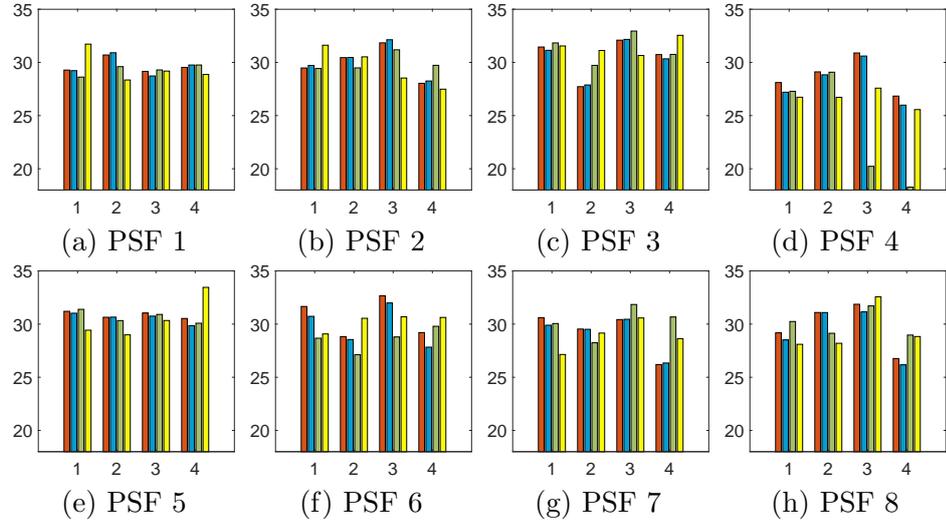


Figura 5: PSNR para cada imagen (eje x) y PSF (panel 1-8). Código de colores: ■ Propuesto, ■ Zhou17, ■ Pan17, and ■ Perrone15.

resultados de Perrone15 en prácticamente la totalidad de los casos. Por último, se observa que el método Pan17 es el menos robusto a la vista de la desviación típica global de sus resultados, obteniendo tanto en PSNR (2.93) como en SSIM (0.094) las desviaciones mayores.

Podemos comparar visualmente estos resultados en las figuras 7 y 8, donde mostramos un pequeño pero representativo subconjunto de los resultados. Las PSF superpuestas a las imágenes han sido representadas con un esquema de color no lineal que facilita la visualización de los valores más pequeños. Como se puede comprobar, el método propuesto produce imágenes nítidas sin artefactos visuales para tamaños de PSF razonable. Los emborronamientos de mayor tamaño perjudican gravemente la estimación del filtro, dado que afectan a grandes regiones de la imagen, empeorando naturalmente los resultados de todos los métodos, lo que puede corroborarse en la fig. 8(abajo), con emborronamientos menos precisos y, consecuentemente, peores restauraciones (véanse también los correspondientes valores de PSNR y SSIM en la tabla 1). No obstante, el método de deconvolución propuesto obtiene mejores resultados tanto numéricos como visuales, con fronteras mejor definidas y menos artefactos. Fijándonos en las PSF, el método propuesto y Pan17 producen las estimaciones menos ruidos, siendo las de Pan17 menos robustas, ver fig. 7(abajo) y 8(abajo).

En segundo lugar mostramos los resultados del experimento con imágenes reales en las figuras 9 y 10. En este caso, no se aprecian diferencias visuales significativas en las restauraciones producidas por los diferentes algoritmos, mostrando todas ellas una alta calidad. Las

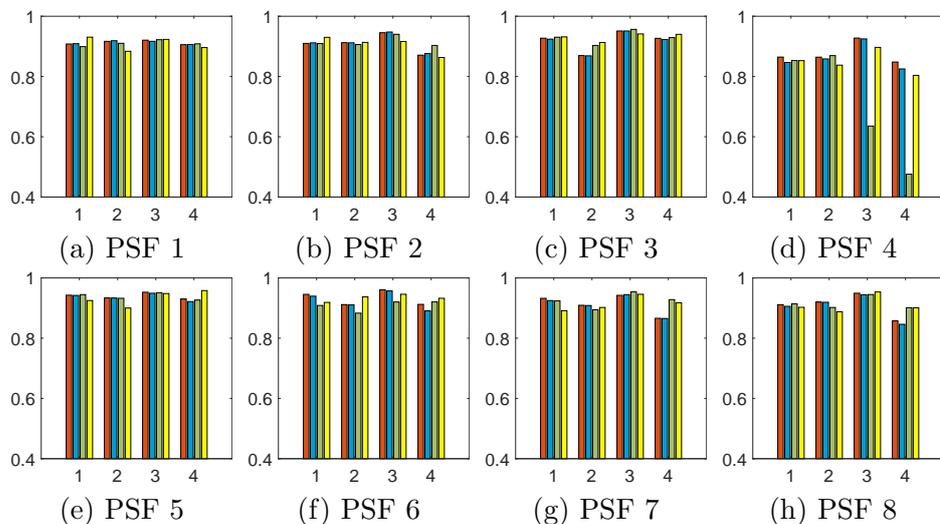


Figura 6: SSIM para cada imagen (eje x) y PSF (panel 1-8). Código de colores: ■ Propuesto, ■ Zhou17, ■ Pan17, and ■ Perrone15.

prestaciones del algoritmo propuesto son similares a las del estado del arte, si bien es cierto que los resultados son un tanto más nítidos debido a que produce estimaciones del emborronamiento menos ruidosas, repitiéndose la tendencia vista en el anterior experimento, en el que veíamos que las PSF de Zhou17 y Perrone15 también eran más ruidosas. Por otro lado, en la fig. 10, aunque se observa que todos los métodos de deconvolución obtienen una zona central nítida, aparecen artefactos en regiones alejadas del centro de la imagen. La razón de este comportamiento es que el emborronamiento presente en esta imagen es espacialmente variante, con lo que la PSF global estimada no es la óptima para todas las zonas de la imagen.

Aparte de la naturaleza espacialmente invariante de la imagen de la fig. 10, uno de los factores que comprometen la estimación del emborronamiento es el ruido presente en la observación, que como podemos comprobar en esta imagen, es especialmente notorio. Resulta por ello interesante realizar un estudio del impacto del nivel de ruido en la estimación de la PSF. La figura 11 muestra un conjunto de imágenes generadas sintéticamente a partir del modelo físico de degradación (III.2). En concreto, utilizamos la imagen *Lena*, a la cual, tras la convolución con el emborronamiento 7 del primer experimento, se le añade un ruido gaussiano de media nula y diferentes varianzas, produciendo un conjunto de 7 imágenes con BSNR (*Blurred Signal-to-Noise Ratio* de 40, 35, ..., 10 dB. En la fila inferior de la misma imagen se encuentran las PSF estimadas, donde observamos que para niveles de bajos hasta medio-altos (BSNR de 40 a 25 dB, el método propuesto estima la PSF con gran precisión. En mayor presencia de ruido el efecto de este comienza a hacerse

Cuadro 1: Media (y desviación típica) de PSNR y SSIM por PSF.

PSF	Propuesto		Zhou17		Pan17		Perrone15	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	29.67 (0.71)	0.913 (0.007)	29.66 (0.94)	0.913 (0.006)	29.32 (0.50)	0.910 (0.010)	29.53 (1.50)	0.908 (0.022)
2	29.95 (1.60)	0.910 (0.031)	30.15 (1.61)	0.912 (0.029)	29.96 (0.83)	0.915 (0.017)	29.54 (1.87)	0.906 (0.029)
3	30.50 (1.93)	0.919 (0.035)	30.38 (1.83)	0.917 (0.035)	31.32 (1.39)	0.930 (0.022)	31.47 (0.81)	0.931 (0.013)
4	28.74 (1.71)	0.876 (0.035)	28.16 (2.01)	0.864 (0.043)	23.72 (5.27)	0.708 (0.188)	26.65 (0.83)	0.848 (0.039)
5	30.85 (0.32)	0.940 (0.010)	30.57 (0.012)	0.936 (0.51)	30.68 (0.59)	0.939 (0.011)	30.55 (2.01)	0.933 (0.026)
6	30.59 (1.87)	0.932 (0.025)	29.77 (1.85)	0.924 (0.030)	28.59 (1.50)	0.908 (0.018)	30.23 (1.42)	0.933 (0.011)
7	29.19 (2.05)	0.912 (0.034)	29.05 (1.85)	0.911 (0.034)	30.20 (1.50)	0.925 (0.024)	28.87 (1.42)	0.914 (0.024)
8	29.72 (2.27)	0.909 (0.038)	29.23 (2.38)	0.904 (0.042)	30.01 (1.26)	0.915 (0.021)	29.42 (2.12)	0.911 (0.029)
media	29.90 (1.64)	0.914 (0.031)	29.62 (1.70)	0.910 (0.034)	29.23 (2.93)	0.894 (0.094)	29.53 (1.89)	0.911 (0.035)

patente. Para BSNR de 15 y 10 dB, donde el ruido es muy elevado (ampliense las imágenes observadas para una correcta visualización del ruido presente), la estimación se ve gravemente perjudicada, incrementándose este efecto según aumenta el ruido, como es natural. El motivo principal que afecta a la correcta estimación del emborronamiento en nuestro algoritmo reside en que, según aumenta el ruido, un mayor número de ω_{γ_i} se volverán nulos. Esto provocará el descarte de un mayor número de píxeles para la estimación de la PSF decrementando, por tanto, el valor de π_{γ} . Se puede concluir, pues, que el algoritmo es robusto a la presencia de ruido.

Concluimos este apartado con un breve análisis de tiempo de ejecución de los métodos. Todos los algoritmos han sido ejecutados con código no optimizado ni paralelizado implementado en Matlab[®] en un Intel[®] Core i7-3960X CPU @ 3.30GHz con 64 GB de RAM. La tabla 2 muestra los tiempos de ejecución medios en 10 ejecuciones. Nótese que, en primer lugar, se utilizaron diferentes imágenes y diferentes tamaños de PSF y, en segundo lugar, el tiempo mostrado para el método propuesto incluye el tiempo empleado en obtener la estimación inicial de la PSF con [104].

El algoritmo propuesto es mucho más rápido que el método MAP Perrone15 [92], ya que este necesita mucho tiempo de cómputo para el cálculo del gradiente TV (*Total Variation Gradient*). La comparación con Pan17 [89] es algo más justa, siendo este algo más lento que el método

Cuadro 2: Mean CPU time (in seconds) for different BID methods.

Proposed	Pan17	Zhou17	Perrone15
148	220	7	1031

propuesto. En este caso, el mayor problema reside en los cálculos asociados al «canal oscuro». No obstante, y como ya habíamos adelantado anteriormente, el método de Zhou17 [104] es mucho más rápido debido al uso de la FFT en los cálculos, uso que no es posible en nuestro método al iterar secuencialmente sobre los píxeles de la imagen. Es necesario un estudio que clarifique la posibilidad de estimar todos los píxeles paralelamente, lo que permitiría el uso de la FFT, lo que reduciría en gran medida el tiempo de ejecución.

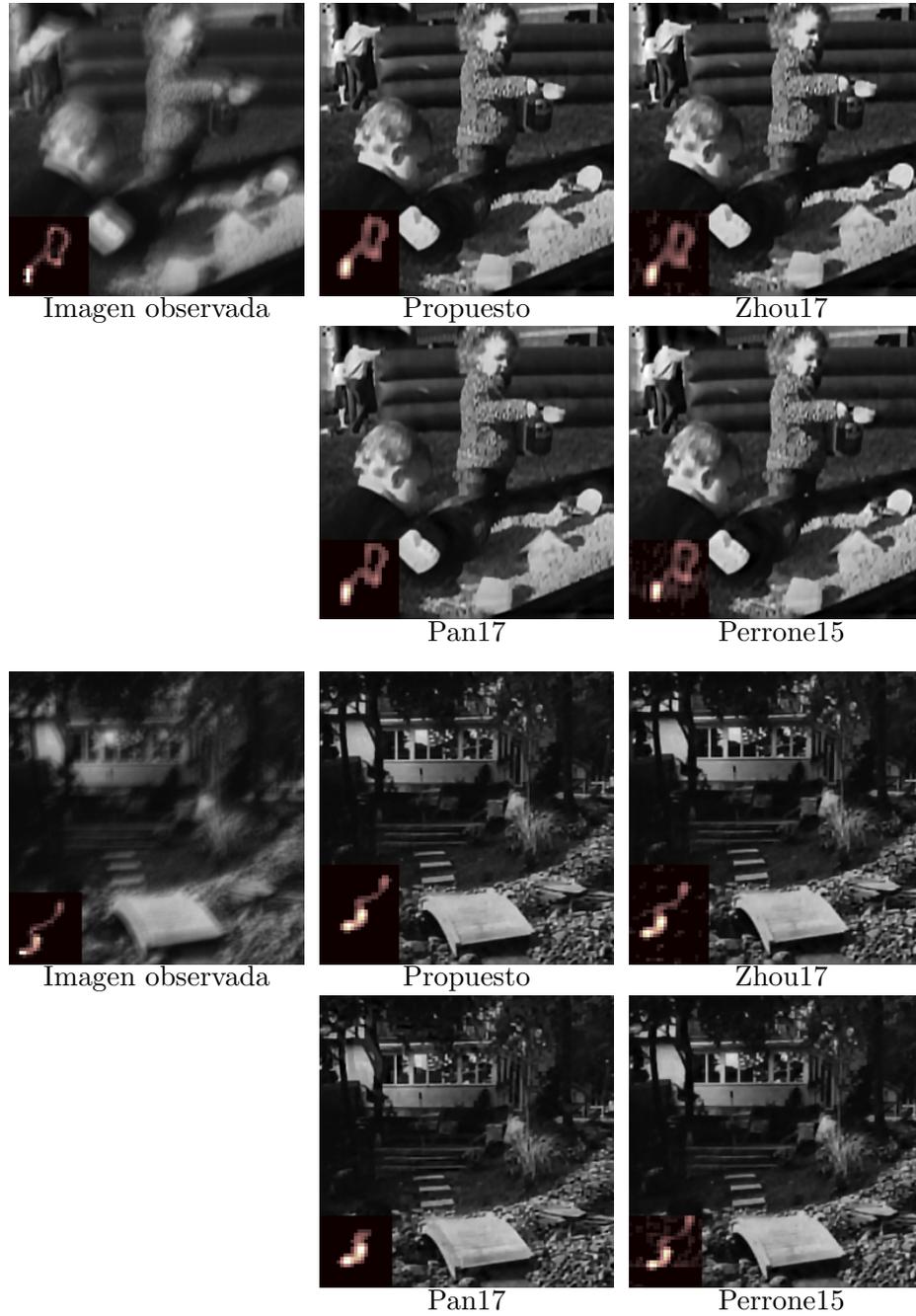


Figura 7: Muestra de resultados sobre el conjunto de imágenes de Levin. (Arriba) Imagen 1, PSF 7. (Abajo) Imagen 2, PSF 6.



Figura 8: Muestra de resultados sobre el conjunto de imágenes de Levin. (Arriba) Imagen 3, PSF 5. (Abajo) Imagen 4, PSF 4.

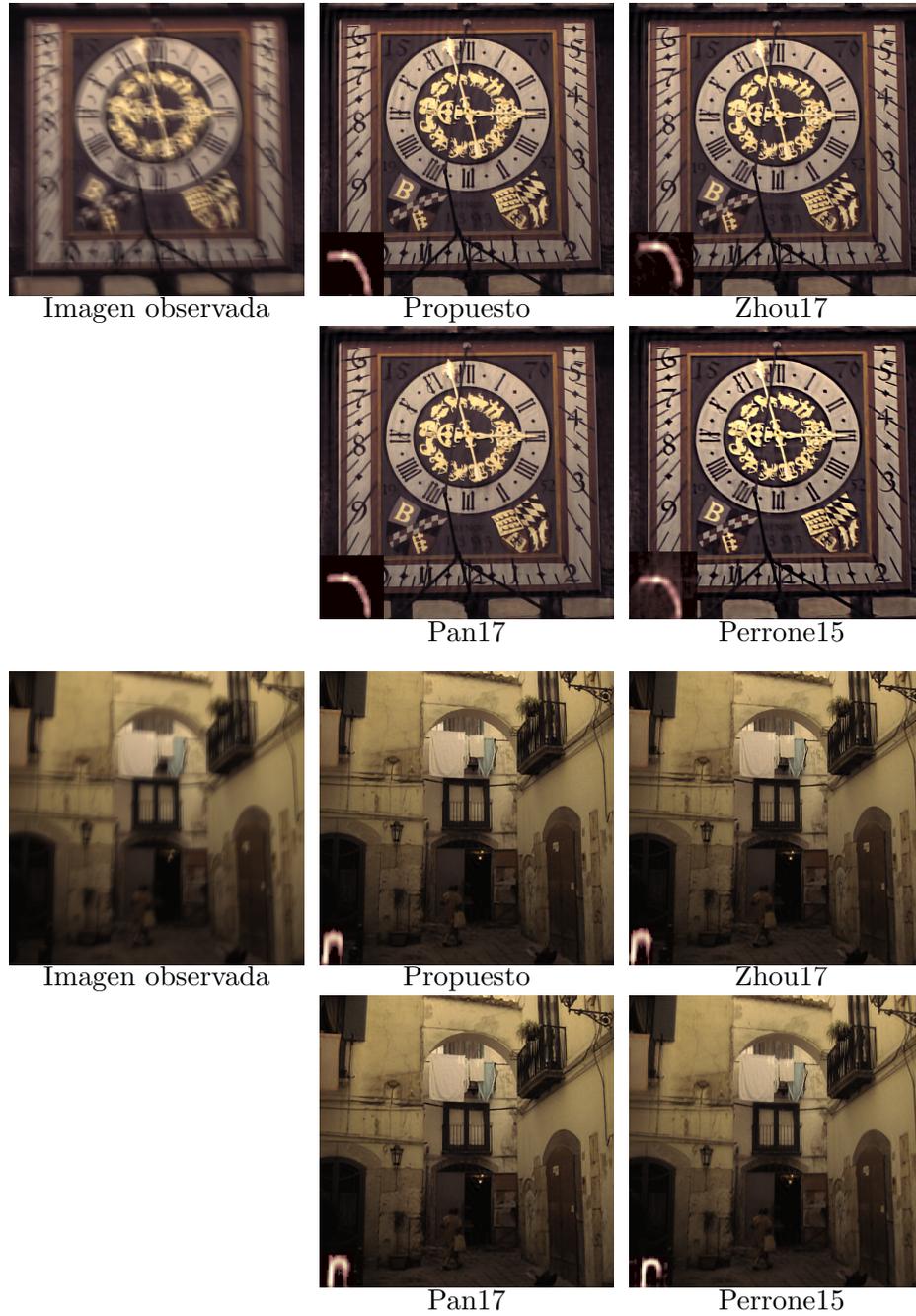


Figura 9: Resultados de la restauración sobre dos imágenes real del conjunto de imágenes de [131].



Figura 10: Resultados de la restauración de una imagen borrosa y ruidosa.

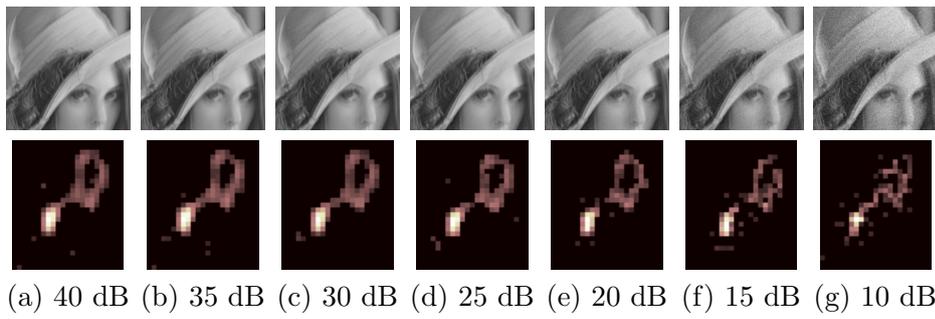


Figura 11: Imágenes sintéticas degradadas con diferentes valores de BSNR y las correspondientes estimaciones de la PSF.

Capítulo IV

Aplicación a imágenes milimétricas pasivas

En este capítulo aplicaremos los métodos de procesamiento y mejora de imágenes vistos en los capítulos II y III a imágenes milimétricas pasivas. Las imágenes milimétricas empleadas para la experimentación forman parte de una base de datos propia creada en la Universidad de Granada con la cámara Wavecam en el seno del proyecto «Imágenes milimétricas pasivas: captación, mejora y detección de amenazas» financiado por el Ministerio de Economía y Competitividad TIN2013-43880-R.

Comenzaremos describiendo la base de datos de imágenes adquirida, analizando visualmente las degradaciones presentes en esta. El objetivo principal del capítulo será la evaluación del preprocesamiento de las imágenes como medio para incrementar la precisión de los clasificadores. En primer lugar, veremos los resultados de aplicar los métodos desarrollados de *inpainting*, eliminación de ruido y deconvolución ciega.

Para la detección de amenazas utilizaremos dos modelos de detección diferente. El primero de ellos sigue una metodología clásica basada en extracción de rasgos y clasificación por bloque. En segundo lugar aplicaremos una red neuronal convolucional (CNN) con estructura de *autoencoder* basada en la red de segmentación Segnet. Por último, se describen los resultados de detección obtenidos por estas técnicas para los preprocesamientos realizados.

1. Base de datos

El conjunto de imágenes milimétricas pasivas utilizado está compuesto por un total de 3309 imágenes de tamaño 195×125 . 33 sujetos fueron escaneados de frente y de perfil portando 0, 1 o 2 objetos ocultos en distintas posiciones con tamaño variable entre 10×10 , el menor, y 39×35 , el de mayor tamaño. Estos objetos ocultos simulan potenciales amenazas,



Figura 1: Relación de amenazas simuladas utilizadas en la base de datos. 1. cúter, 2. 325 g de gel, 3. 200 g de arcilla, 4. pistola simulada, 5. 200 g de azúcar, 6. 200 g guisantes congelados congelados, 7. 150 mL de colonia, 8. 160 g de gel, 9. piezas metálicas, 10. 200 g de harina, 11. 500 mL de agua, 12. 250 mL de agua oxigenada.

presentando una gran diversidad de formas y materiales, ver fig. 1, incluyendo metales, líquidos y plásticos. La base de datos completa, así como una descripción más detallada de la misma puede encontrarse en <http://decsai.ugr.es/pi/pmmwi/testdata.html>.

2. Preprocesamiento de las imágenes

La señal captada por la cámara ha de procesarse para formar las imágenes milimétricas propiamente dichas, dado que la cámara está compuesta por un conjunto de 4 sensores que realizan un escaneo por barrido del espacio que no proporcionan directamente un imagen. Mostramos en la fig. 2 el aspecto de la señal de uno de estos sensores. Las regiones en negro de la imagen corresponden a fallos electrónicos del sistema de captación. Para el proceso de composición de la imagen resulta crucial la restauración de estos datos. Por esta razón, en primer lugar se utilizó el algoritmo de *inpainting* basado en BKSVD para rellenar estos huecos correspondientes a datos perdidos. Para ello, se escogió un tamaño de bloque 8×8 ($P = 64$) y se aprendió un diccionario con $K = 2P$ átomos. Cada diccionario fue aprendido de manera individual sobre cada imagen

2. PREPROCESAMIENTO DE LAS IMÁGENES

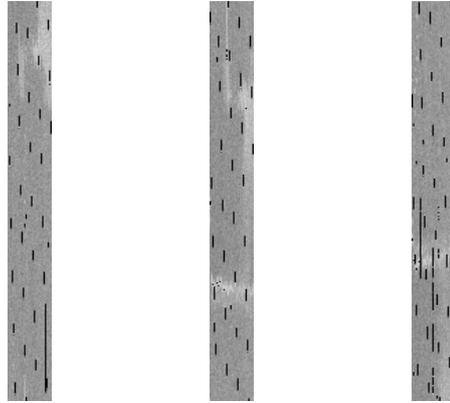


Figura 2: Imagen en formato *raw* 900×32 dividida en tres trozos generada por uno de los 4 sensores de la cámara.

utilizando el algoritmo visto en el apartado 4.3.1 considerando que no hay ruido en la imagen. A continuación, utilizamos este diccionario para reconstruir los bloques (con solapamiento máximo) que contienen datos perdidos. El valor final para un píxel perdido resulta de la media de los píxeles correspondientes en todos los bloques reconstruidos que lo contengan. Podemos ver los resultados de la aplicación de este método tras la posterior recomposición de la señal a formato imagen en la figura 3.

Las imágenes compuestas después de haber aplicado el método de *inpainting* sobre la señal *raw* presentan un alto nivel de ruido no estacionario. El método de *denoising* desarrollado en el capítulo II no está diseñado específicamente para este tipo de ruido pero, como veremos en las imágenes procesadas, el método basado en BKSVD es capaz de eliminar gran parte del ruido presente en ellas. Para la aplicación del algoritmo se han considerado 3 tamaños de bloque: 6×6 , 8×8 y 10×10 , lo que se corresponde con señales de tamaño $P = \{36, 64, 100\}$. El número de átomos utilizado ha sido $K = 4P$. En las figuras 4 y 5 podemos ver el resultado de aplicar este método para los tamaños de bloque 6×6 y 8×8 .

Por último mostramos los resultados de aplicar el método de deconvolución ciega con *a priori Spike-and-Slab* a las imágenes milimétricas tras el proceso de *denoising*, creando así otros tres conjuntos de imágenes procesadas puesto que se ha aplicado la deconvolución a las imágenes procesadas con los tres diferentes tamaños de bloque. En la figura 6 encontramos los resultados de aplicar el algoritmo propuesto en el capítulo anterior tras la eliminación de ruido con tamaño de bloque 6×6 . La inicialización de los parámetros e hiperparámetros del modelo es análoga a la utilizada en la sección experimental del método de deconvolución ciega propuesto, véase apartado 5. Se realizó un estudio de la influencia del parámetro de ruido β y los hiperparámetros pi_γ por

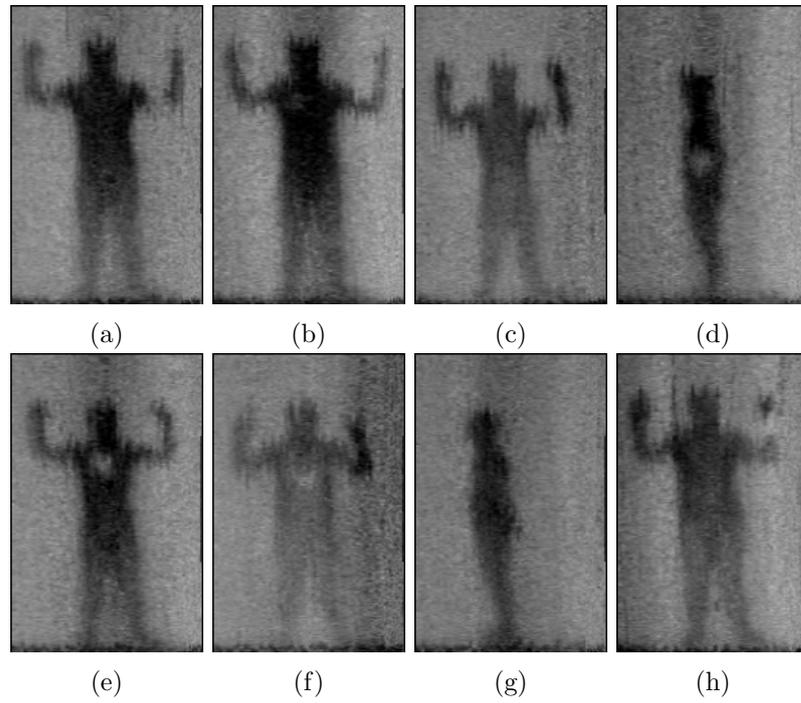


Figura 3: Imágenes preprocesadas con *inpainting* basado en BKSVD.

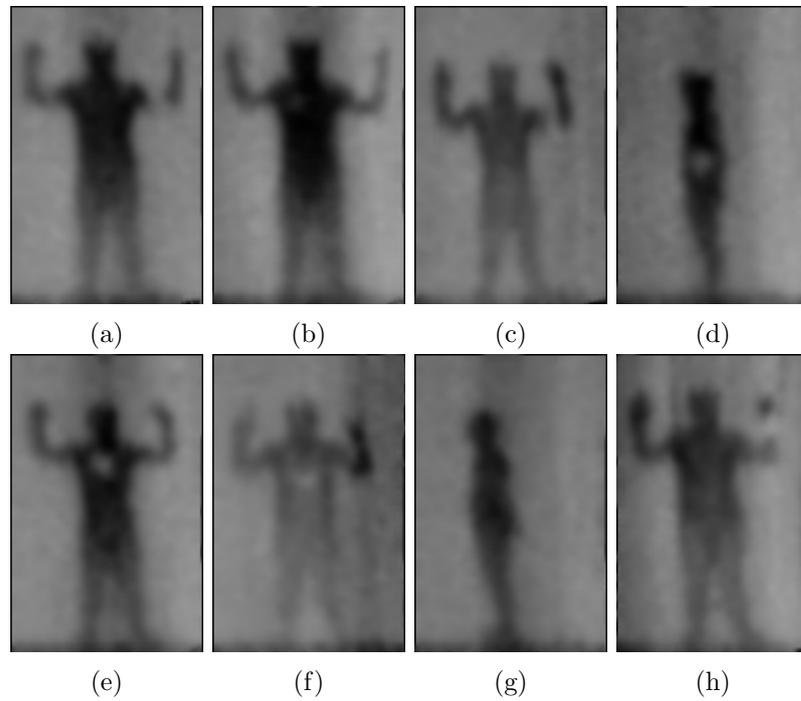


Figura 4: *Denoising* con BKSVD, con tamaño de bloque 6×6 .

3. DETECCIÓN BASADA EN EXTRACCIÓN DE RASGOS

estimación visual, dado que no se disponía de la solución verdadera. Aparentemente, una precisión de ruido menor y, probablemente, un menor valor de π_γ deberían ser más apropiados debido al aspecto de las imágenes: es todavía visible cierto nivel de ruido y el número de píxeles en estas que se corresponden a fronteras es muy reducido. No obstante, las diferencias visuales en los resultados no eran significativas, con lo que se ha elegido la configuración inicial de parámetros. Las fronteras pertenecientes a las siluetas de los sujetos escaneados aparecen mucho más nítidas y, en la mayoría de los ejemplos mostrados, las amenazas se muestran también más nítidas. Véase, por ejemplo, la amenaza presente en el pecho en la fig. 6b, que apenas aparece visible en su imagen procesada precedente, ver fig. 4b. Sin embargo, esta es una mejora local, pues resalta la aparición de gran cantidad de artefactos en las imágenes, manifestándose con una apariencia rugosa de las imágenes. Las PSF estimadas no presentan el aspecto limpio y suave que lucían en los experimentos previos sobre imágenes sintéticas y reales utilizadas. Los motivos principales de este *fallo* en la estimación de un emborronamiento limpio son la presencia de ruido no estacionario que ha quedado tras el *denoising* y las variaciones de intensidad de la señal en diferentes regiones de la imagen (por ejemplo, la parte inferior de las imágenes presenta, en muchas imágenes, una menor intensidad). Veremos en los siguientes apartados si la mejora en nitidez de la imagen ayuda a la detección o si, por el contrario, los artefactos introducidos contrarrestan esta mejora visual de las imágenes.

3. Detección basada en extracción de rasgos

En primer lugar adoptaremos una aproximación clásica a la detección de objetos en imágenes mediante la extracción de rasgos [5]. Es típico en detección de objetos en imágenes utilizar un procedimiento de ventana deslizante, ver fig. 7, transformando una imagen en un conjunto de bloques de tamaño $h \times w$ en los cuales buscaremos la presencia del objeto buscado. Las medidas del bloque dependerán de la relación de aspecto de las amenazas. Además, en caso de no conocer la escala en la se presentan estos objetos, se podría utilizar un procedimiento multiescala para la extracción de bloques de las imágenes. Sin embargo, dado el procedimiento de captación de las imágenes milimétricas, el sujeto siempre está a una distancia fija preestablecida, y por tanto, él y los posibles objetos ocultos presentaran una escala fija, por lo que bastará seleccionar un tamaño de bloque adecuado. Dada la variedad de las amenazas en la base de datos utilizada, escogeremos un tamaño de bloque que garantice el cubrimiento total de cualquier amenaza, en nuestro caso, 39×39 . El número total de bloques de tamaño $h \times w$ que obtendremos de una imagen $H \times W$ son un desplazamiento entre bloques consecutivos de Δx en la dirección horizontal

CAPÍTULO IV. APLICACIÓN A IMÁGENES MILIMÉTRICAS PASIVAS

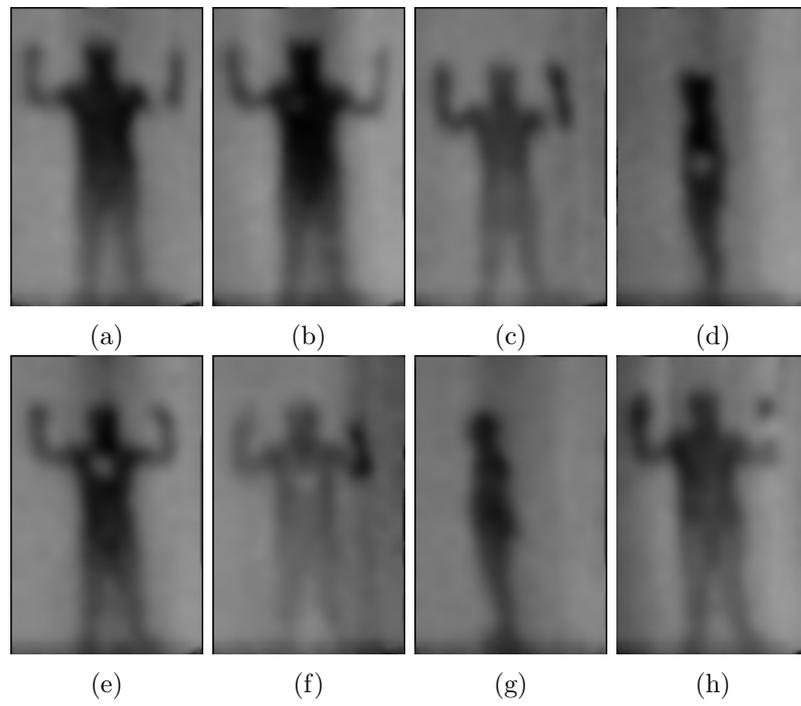


Figura 5: *Denoising* con BKSVD, con tamaño de bloque 8×8 .

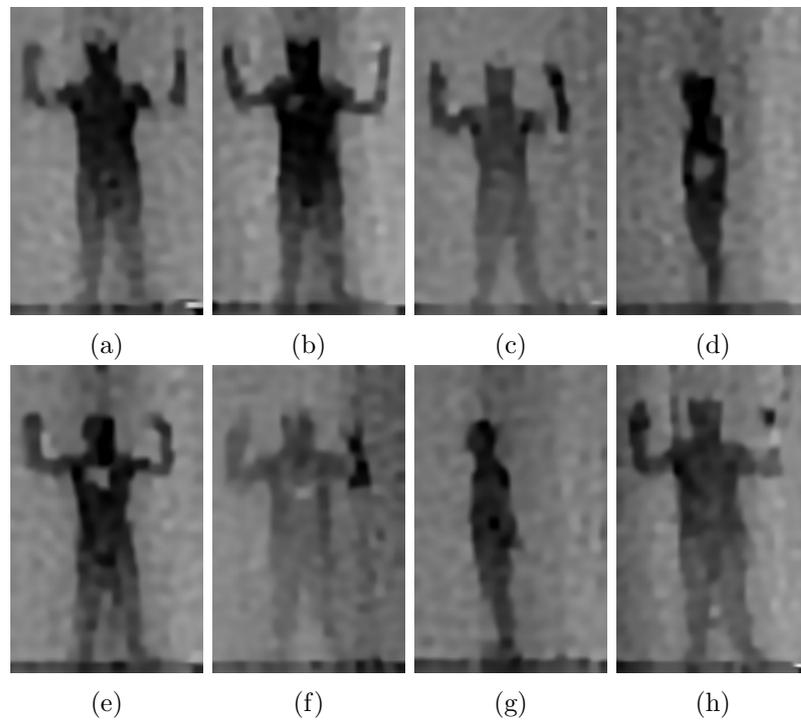


Figura 6: Deconvolución SnS tras *inpainting* y *denoising* BKSVD 6×6 .

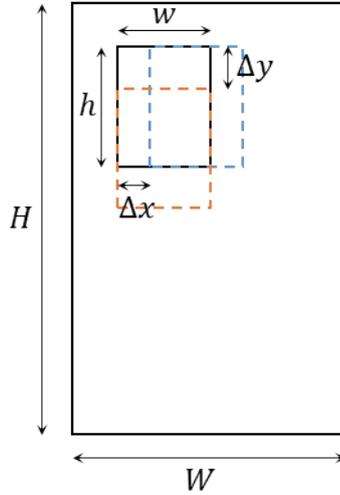


Figura 7: Extracción de parchas mediante ventana deslizante.

y Δy en la vertical es

$$N_B = \left(\left\lfloor \frac{W - w}{\Delta x} \right\rfloor + 1 \right) \left(\left\lfloor \frac{H - h}{\Delta y} \right\rfloor + 1 \right) \quad (\text{IV.1})$$

Las imágenes milimétricas de nuestra base de datos presentan un tamaño de 195×125 píxeles. Utilizando un desplazamiento de 2 píxeles en ambas direcciones, obtenemos un total de 3476 bloques por imagen. Cada uno de estos bloques será etiquetado con un 1 si el bloque cubre totalmente a la amenaza o con un 0 si el bloque no se solapa en absoluto con la amenaza. Los bloques que presenten un cierto solapamiento con la amenaza serán descartados para la fase de entrenamiento. Dado que las amenazas, en caso de haberlas, representan una pequeña zona de la imagen, el conjunto de datos está altamente desbalanceado. Para tratar este problema, se descartarán 1 de cada 4 bloques etiquetados como negativos, lo que es equivalente a haber seleccionado los bloques negativos con un paso de 4 en lugar de 2. Con esto, se reduce el número de bloques negativos a aproximadamente 5 veces el de positivos.

Cada uno de estos bloques puede entenderse como un vector de características de alta dimensión $\mathbf{x}_p \in \mathbb{R}^{1521}$, en el cual podemos encontrar información local de manera implícita, escondida en las correlaciones de diferentes conjuntos de píxeles. En lugar de trabajar directamente con las intensidades, niveles de gris, de los píxeles, y dejar que sea el clasificador el que aprenda dichas correlaciones, resulta más conveniente y eficaz la extracción previa de características locales que simplifiquen la función de detección y, consecuentemente, la tarea de aprendizaje. Existe una amplia literatura sobre la extracción de rasgos en imágenes. En este utilizaremos

únicamente las características Haar [132] y LBP [133] (*Local Binary Patterns*) puesto que presentan buenas capacidades para la detección de objetos y su cálculo puede realizarse de manera rápida y eficiente.

3.1. Filtros Haar

Las características Haar resultan de obtener la correlación de un conjunto de filtros con el bloque. Estos filtros, ver fig. 8, consisten en diferentes configuraciones de áreas con valor 1 en los píxeles blancos y -1 en los negros. La aplicación de uno de estos filtros será, consecuentemente, el resultado de promediar los píxeles del bloque que estén en las zonas blancas restando el promedio del resto. Este promedio implícito en el cálculo de estas características las dotará de una gran robustez frente al ruido. Se espera, por tanto, que la mejora obtenida tras el *denoising* con BKSVD no sea tan relevante en este caso. Se han utilizado un total de 115 filtros diferentes para un total de 3 escalas de bloque, 39×39 , 19×19 y 9×9 . Tras la concatenación de los valores resultantes para las tres escalas obtenemos el vector de características Haar. \mathbf{x}_p^{Haar} de tamaño $115 \times 3 = 345$.

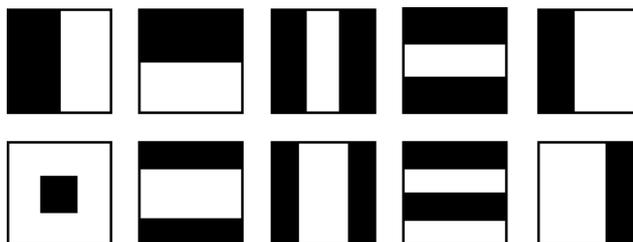


Figura 8: Ejemplos de filtros Haar utilizados.

3.2. *Local Binary Patterns*

LBP [133] es un descriptor utilizado ampliamente en tareas de clasificación de imágenes, especialmente útil para la identificación de texturas. Para el cálculo de este descriptor se asigna una etiqueta (valor, código LBP) a cada uno de los píxeles de, en nuestro caso, un bloque. El descriptor vendrá dado por el histograma de códigos presentes en ese bloque. Esto es, cada una de las características representa el número de ocurrencias de un código LBP o patrón dado. El cálculo de estos códigos se realiza por comparación de cierto número de píxeles vecinos alrededor de un píxel central, comúnmente estos puntos estarán situados sobre un círculo de un determinado radio. El valor decimal correspondiente al número binario resultante de la concatenación en un orden dado de las comparaciones $\{0, 1\}$ entre cada uno de los píxeles con el central determina

3. DETECCIÓN BASADA EN EXTRACCIÓN DE RASGOS

el código o etiqueta asignada al píxel central. En este trabajo utilizaremos códigos LBP uniformes con la extensión [134], que los hace invariantes a rotaciones. La figura 9 muestra algunos ejemplos de estos patrones. Los

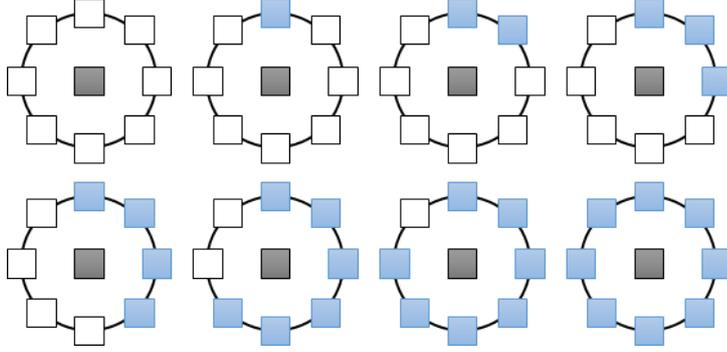


Figura 9: Patrones LBP uniformes invariantes a rotación.

cuadrados blancos representan posiciones cuya intensidad es inferior a la del píxel central en gris; se les asignará el valor 0. Por el contrario, los cuadrados azules (1) corresponden a valores de intensidad mayores o iguales que la del píxel central.

Utilizaremos vecindarios de radios $r \in \{1, 2, 3, 4\}$ con un número de vecinos dado por $n = 8r$. Teniendo en cuenta las tres escalas de bloques, y que el número de códigos para LBP uniforme invariante a rotaciones con n vecinos es $n + 2$, los vectores de características generados \mathbf{x}_p^{LBP} tendrán $(10 + 18 + 26 + 34) \cdot 3 = 264$ componentes.

3.3. Modelo de detección

Siguiendo la metodología de [5], planteamos el problema de detección como un problema de clasificación binaria en el que queremos etiquetar todos los píxeles de una imagen. Dado que el conjunto total de bloques extraídos \mathcal{B} tiene aproximadamente 5 veces más instancias negativas que positivas, se emplea una cascada de clasificadores, cada uno de los cuales se entrena con todas los bloques positivos \mathcal{B}_P y solo una quinta parte de las instancias negativas \mathcal{B}_N^c , siendo este un particionamiento aleatorio disjunto del conjunto de bloques negativos, es decir, $\mathcal{B}_N = \bigcup_{c=1}^5 \mathcal{B}_N^c$. Este proceso se repite t veces, obteniendo un total de $5t$ clasificadores, la salida de los cuales será promediada para obtener la predicción final $f_{\mathcal{B}}(\mathbf{x}_p) \in [0, 1]$. Como clasificador se ha elegido *Random Forest*, dado que es el que mejores resultados obtuvo en *santi1*. Tras esta clasificación, se aprende un umbral thr de detección que pondrá a cero el valor de la clasificación para los

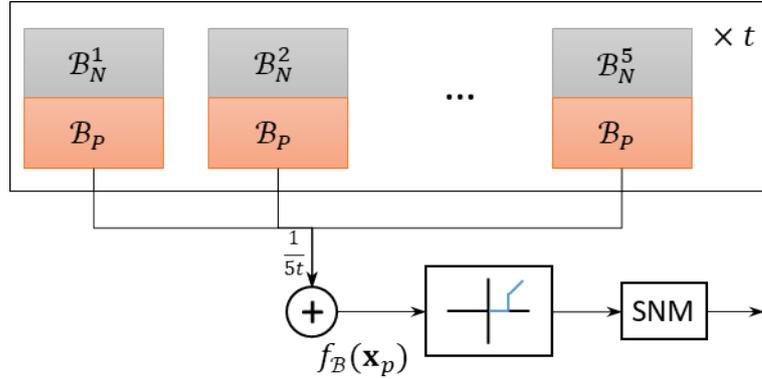


Figura 10: Modelo de detección basado en cascada de clasificadores, con umbralización y supresión de no máximos posterior.

bloques cuya estimación no supere este umbral, esto es,

$$f_B^{th}(\mathbf{x}_p) = \begin{cases} 0, & \text{si } f_B(\mathbf{x}_p) < thr \\ f_B(\mathbf{x}_p), & \text{en otro caso.} \end{cases} \quad (\text{IV.2})$$

Tras la umbralización se realiza la supresión de no máximos (SNM), que elimina detecciones múltiples de una misma amenaza. El proceso de SNM pondrá el valor de detección de un bloque a 0 siempre que este se solape en cierta medida con otro bloque con mayor valor de detección (más cercano a uno). La fig. 10 muestra el proceso completo de detección.

Para la validación de los resultados de detección, se considera que un bloque \mathbf{x}_p se ha clasificado correctamente cuando $f_B^{th}(\mathbf{x}_p)$, tras la supresión de no máximos, supera cierto umbral y, además, su intersección con la amenaza, delimitada por una región rectangular (*bounding box*) cubre, al menos, la mitad de esta. Veremos en el apartado 5 los resultados de detección obtenidos utilizando los descriptores previamente comentados.

4. Detección basada en red neuronal convolucional

El segundo de los métodos de detección utilizado es una red convolucional con estructura de *autoencoder* basada en la arquitectura *Segnet* [135] similar a la propuesta en el trabajo [6]. La figura 11 muestra la arquitectura utilizada por *Segnet*. Esta arquitectura fue propuesta para un problema de segmentación semántica a nivel de píxel. Esta arquitectura presenta un total de 13 capas convolucionales y ninguna totalmente conectada. Los mapas de características de baja resolución producidos por el codificador han de ser transformados a mapas de características de resolución completa que serán utilizados por el clasificador *soft-max* final. La novedad de este trabajo consiste en la introducción de la capa de

4. DETECCIÓN BASADA EN RED NEURONAL CONVOLUCIONAL

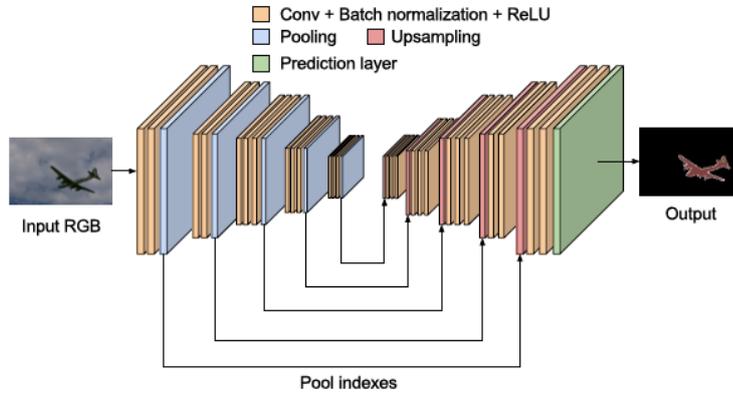


Figura 11: Arquitectura *Segnet* [135].

unpooling en el decodificador para la superresolución de los mapas de características. Esta capa genera mapas de características *sparse* guardando las posiciones de los máximos de las capas de *pooling* correspondientes.

En [6] se adaptó esta arquitectura para segmentación binaria para el problema de las imágenes milimétricas. En este trabajo, dado que se han utilizado un subconjunto de 500 imágenes de la base de datos, se ha adaptado la estructura aquí utilizada dado que el uso de la misma estructura de red producía sobreaprendizaje. La estructura de red que utilizaremos se muestra en la fig. 12.

	Input	Block output
Block 1	Conv. 11x11 16 ReLU	16x195x125
Block 2	Max-pooling 2x2	32x97x62
	Conv. 7x7 32 ReLU	
Block 3	Max-pooling 2x2	32x48x31
	Conv. 5x5 32 ReLU	
Block 4	Max-pooling 2x2	32x24x15
	Conv. 3x3 32 ReLU	
Block 5	Conv. 3x3 32 ReLU	32x48x31
	Max-unpooling 2x2	
Block 6	Conv. 5x5 32 ReLU	32x97x62
	Max-unpooling 2x2	
Block 7	Conv. 7x7 16 ReLU	16x195x125
	Max-unpooling 2x2	
Block 8	Conv. Sigmoid 11x11 1	195x125

Figura 12: Estructura de red convolucional para la segmentación de las imágenes milimétricas.

5. Resultados de detección

La tabla 1 muestra el área bajo la curva ROC (AUC) así como el número medio de falsos positivos por imagen para los diferentes métodos de detección explicados en los apartados 3 y 4. Los preprocesamientos propuestos se comparan con el de [5] que consiste en un *inpainting* previo sobre la imagen *raw* y un filtro de mediana 5×5 sobre la imagen reconstruida. La fig. 13 muestra, para una imagen milimétrica, el resultado de aplicar todos los preprocesamientos considerados. Los nombres de los preprocesamientos utilizados en la tabla 1 se han de entender como:

- *inpainting*: imagen reconstruida tras completado de la imagen intermedia (fig. 2) en formato raw.
- BKSVD b : eliminación de ruido sobre la imagen generada previamente mediante BKSVD con tamaño de bloque $b \times b$.
- *deconv. SnS b* : deconvolución ciega *Spike-and-Slab* tras la eliminación de ruido basada en BKSVD con tamaño de bloque $b \times b$.

Fijándonos en los resultados obtenidos por el detector basado en extracción de características se observa claramente que la eliminación de ruido mejora los resultados de detección (salvo en el caso Haar+BKSVD 10), siendo la ganancia más pronunciada para LBP. Este resultado era esperable, dado que los filtros Haar realizan un promedio de los píxeles pertenecientes a las regiones positivas y negativas de estos, eliminando de forma natural, parte del ruido de las imágenes. Así, pequeñas diferencias en el filtrado de ruido no afectarán a este tipo de características de forma tangible. Al contrario, LBP si es más sensible a la presencia de ruido, mostrándose en el amplio incremento de AUC conseguido en el caso LBP+BKSVD 6. Nótese que los mejores resultados para eliminación de

Cuadro 1: Área bajo la curva ROC y número medio de falsos positivos por imagen de los detectores utilizados.

Procesamiento	Haar + RF		LBP + RF		CNN	
	AUC	FP	AUC	FP	AUC	FP
<i>inpainting</i>	0.62	12 ± 8	0.59	15 ± 11	81.2 ± 1.5	0.03 ± 0.02
<i>denoising</i> [5]	0.63	10 ± 9	0.61	10 ± 9	79.4 ± 3.1	0.2 ± 0.14
BKSVD 6	0.63	10 ± 8	0.64	10 ± 8	87.1 ± 2.2	0.01 ± 0.0
BKSVD 8	0.62	11 ± 11	0.60	11 ± 11	85.6 ± 1.7	0.02 ± 0.02
BKSVD 10	0.61	12 ± 13	0.60	12 ± 13	84.9 ± 1.9	0.02 ± 0.03
<i>deconv. SnS</i> 6	0.62	10 ± 12	0.60	10 ± 12	88.3 ± 1.7	0.01 ± 0.0
<i>deconv. SnS</i> 8	0.63	10 ± 9	0.61	10 ± 9	83 ± 2.4	0.03 ± 0.01
<i>deconv. SnS</i> 10	0.59	15 ± 13	0.58	15 ± 13	78.3 ± 3.3	0.22 ± 0.18

ruido se obtienen, para ambos clasificadores, para el procesado con tamaño de bloque 6×6 . Por otro lado, por lo que a la deconvolución se refiere, se confirma que la introducción de artefactos en la imagen no es compensada por su mayor nitidez, reflejándose en una mayoría de casos que presentan un decremento de AUC. Sin embargo, para la deconvolución a partir de las imágenes preprocesadas con el método de eliminación de ruido con tamaño de bloque 8×8 se observa un incremento en AUC, respaldado visualmente por la fig. 13g en la que se observa nítidamente la amenaza con una silueta bien definida y con menos artefactos visuales que en los otros procesamientos, figs. 13f y 13h. Finalmente, en cuanto a los resultados de detección en sí, este método produce resultados no del todo satisfactorios con un número medio elevado de falsos positivos por imagen. Si bien es cierto que esto es parcialmente debido al uso de un subconjunto de la base de datos (500 imágenes de 3309), los resultados reportados en [5], aunque algo superiores, muestran una tendencia similar. Sin embargo, siempre se puede reducir el número de falsos positivos escogiendo un umbral mayor, aunque esto reducirá el ratio de verdaderos positivos.

El detector basado en CNN es claramente superior. Presente muy buenos resultados de detección, que podrían incrementarse considerando la base de datos completa, y con un número de falsos positivos por imagen despreciable. En este caso todos los procesamientos de *denoising* basados en BKSVD incrementan de forma sustancial el AUC del clasificador (casi 6 puntos porcentuales para BKSVD 6), superando holgadamente el preprocesamiento propuesto en [5] que, siguiendo la misma tendencia que se vio en [6], obtiene un AUC inferior que el conseguido al clasificar las imágenes sin la eliminación de ruido. Por otro lado, vemos que la deconvolución solo obtiene un incremento en AUC para el caso de *denoising* previo con tamaño de bloque 8×8 . En los otros dos casos, los artefactos generados empeoran la clasificación.

Podemos concluir, pues, que la selección de un buen preprocesamiento ayuda sustancialmente a los sistemas de detección, incrementando en buena medida su capacidad de detección y reduciendo el número de falsas alarmas.

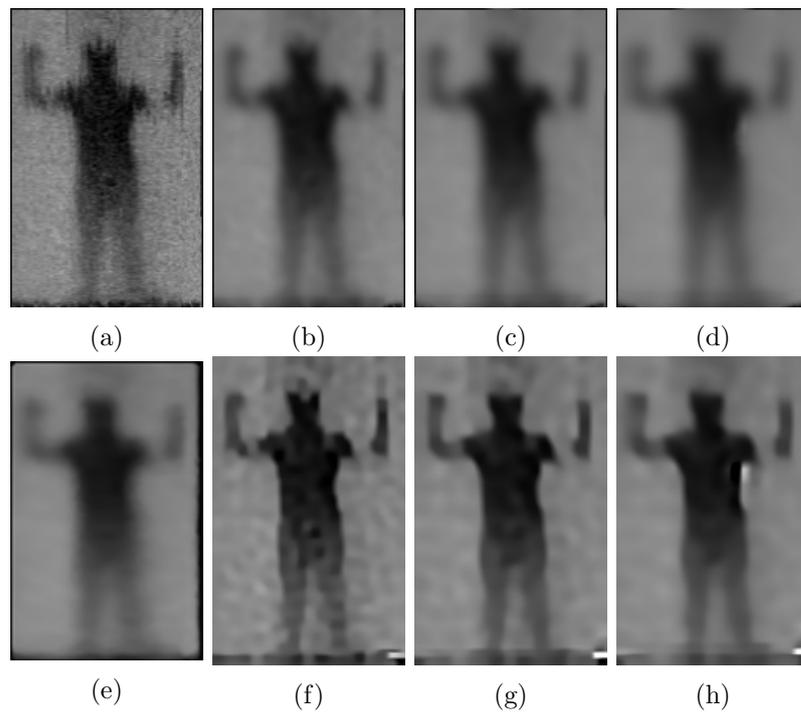


Figura 13: Muestra de diferentes preprocesamientos para una de las imágenes de la base de datos. (a) *inpainting*, (b) *denoising* 6×6 , (c) *denoising* 8×8 , (d) *denoising* 10×10 , (e) preprocesamiento [5], (f) *den.* 6×6 y deconvolución, (g) *den.* 8×8 y deconvolución, (h) *den.* 10×10 y deconvolución.

Capítulo V

Conclusiones y trabajo futuro

1. Conclusiones

La presente tesis ha abordado la detección de amenazas en imágenes milimétricas poniendo el foco en el preprocesamiento de las imágenes captadas dada su baja calidad. Concretamente, se han tratado los modelos que permiten abordar los problemas de *inpainting*, eliminación de ruido y deconvolución ciega, mostrando que estos problemas admiten un modelado bayesiano, obteniendo soluciones robustas gracias a la inferencia variacional. Los dos primeros problemas se han resuelto en el marco del aprendizaje de diccionarios, desarrollando un esquema de inferencia bayesiano para el modelo clásico y el doblemente *sparse*, en el que se consideran diccionarios *sparse* a su vez. En segundo lugar, se ha propuesto un método de estimación del emborronamiento para un algoritmo de deconvolución ciega con el uso novel de la inferencia variacional EM para la distribución *Spike-and-Slab* en esta área. Por último, hemos aplicado los algoritmos desarrollados a una base de datos propia de imágenes milimétricas para comprobar su efecto en el proceso de detección de amenazas, utilizando dos clasificadores con filosofías de trabajo diferentes. Un primero basado en la metodología clásica de extracción de características y detección por ventana deslizante y, un segundo que hace uso de técnicas de aprendizaje profundo. A continuación presentamos las conclusiones específicas que extraemos de estos tres grandes bloques en los que se ha dividido la memoria.

1.1. *Denoising* e *Inpainting* mediante aprendizaje de diccionarios

- En primer lugar se ha presentado el algoritmo BKSVD para resolver el problema de aprendizaje de diccionarios con norma ℓ_1 . La *a priori* impuesta sobre los vectores de coeficientes garantiza soluciones *sparse* permitiendo la inferencia variacional bayesiana. El uso de un marco bayesiano para la inferencia nos permite tener en cuenta la incertidumbre en las estimaciones para refinarlas. Este algoritmo es totalmente automático, estimando todas las variables, incluida la varianza de ruido, sin información adicional, más allá de las observaciones.
- Se ha aplicado el algoritmo a los problemas de eliminación de ruido e *inpainting*, obteniendo resultados competentes. Sin embargo, el K-SVD tiene dos grandes inconvenientes: la necesidad de una estimación precisa de la varianza de ruido y del conocimiento previo del número de componentes no nulas de cada señal. Ambos inconvenientes son resueltos de manera eficaz y precisa en el algoritmo propuesto, pero a costa de un mayor coste computacional. Junto con el método propuesto, el algoritmo BFPA es el que mejores resultados obtiene, aunque las imágenes que obtiene presentan un mayor número de artefactos.
- El *inpainting* se ha abordado de dos maneras diferentes. Es posible aprender un diccionario de mayor calidad utilizando una base de datos de imágenes limpias, resultando en mejores reconstrucciones pero, en caso de no disponer de este conjunto de datos modelo, también es posible aprender un diccionario a partir de datos corruptos: la propia imagen con datos perdidos.
- Los diccionarios aprendidos con BKSVD son mejores en términos de coherencia mutua, lo que resulta en una mayor calidad de las reconstrucciones, proporcionando soluciones *sparse* más óptimas.
- El tamaño preestablecido de los diccionarios influye en la calidad de las representaciones *sparse*. La estimación del tamaño óptimo no está integrada en la inferencia como variable del modelo, con lo que su estimación no es automática.
- Adicionalmente, se ha presentado un método de estimación de diccionarios *sparse* utilizando una modelización jerárquica análoga al BKSVD para los vectores de representación del diccionario. La principal ventaja frente a las técnicas doblemente *sparse* deterministas radica en nuestra capacidad para estimar automáticamente el ruido y el soporte de los vectores *sparse*,

teniendo en cuenta las incertidumbres de las estimaciones y produciendo valores competitivos de PSNR y SSIM. El modelo presenta una formulación general que posibilita su uso no solo para *denoising* e *inpainting*, sino también para *compressive sensing*.

- Por último, se ha demostrado la posibilidad de utilizar distribuciones *a posteriori* laplacianas sobre los vectores de coeficientes con *a priori* laplaciana, resultando en una inferencia tratable.

1.2. Deconvolución ciega de imágenes

- Se ha presentado un nuevo método de deconvolución ciega que trabaja en el dominio filtrado cuya principal aportación consiste en la introducción de la distribución *Spike-and-Slab* como *a priori* sobre las pseudoobservaciones obtenidas por filtrado paso alto de la imagen borrosa.
- La distribución *Spike-and-Slab* permite descartar píxeles poco relevantes y potencialmente perjudiciales para la estimación del emborronamiento.
- La inferencia variacional con esta *a priori* se hace posible gracias al uso de una potente a la vez que simple reparametrización. Además, la utilización de una adecuada factorización para la aproximación de la distribución *a posteriori* dota a la estimación de una mayor fidelidad a la *a posteriori* original de naturaleza multimodal.
- Se ha desarrollado un algoritmo variacional EM eficiente y preciso para la estimación de la PSF, estimando de manera automática todos los parámetros del modelo *a priori* dentro del mismo marco bayesiano.
- El estudio del comportamiento de la distribución *Spike-and-Slab* proporciona un conocimiento más profundo sobre la capacidad selección de píxeles relevantes de esta *a priori*, la cual permite una estimación más precisa del emborronamiento.
- Los resultados experimentales muestran la competitividad del método, produciendo estimaciones de la PSF más precisas que resultan en un incremento de la PSNR media con una menor varianza, indicando la robustez del método.
- El método de deconvolución ciega presentado presenta una buena tolerancia al ruido en las imágenes.

1.3. Aplicación a imágenes milimétricas pasivas

- En esta memoria se han empleado dos metodologías diferentes de abordar el problema de detección de objetos (amenazas) en imágenes. En primer lugar, siguiendo una metodología clásica, se han extraído bloques de las imágenes (ventana deslizante) y se han extraído diferentes características típicas, Haar y LBP, utilizando el método *Random Forest* para la clasificación. En segundo lugar hemos utilizado una red neuronal profunda para la segmentación binaria de la imagen.
- Las imágenes captadas presentaban un patrón repetitivo de regiones no observadas debido a deficiencias en el sistema de captación. Por ello, el primer paso del preprocesamiento ha sido el completado de estas regiones de las imágenes mediante el método de *inpainting* basado en BKSVD presentado en el capítulo II. Este paso resulta crucial sobre todo para la metodología basada en extracción de rasgos.
- La mejora en las imágenes producida por la deconvolución ciega depende en gran medida de la eliminación previa de ruido. Sin embargo, los artefactos producidos por el alto nivel de ruido no estacionario suelen limitar esta mejora.
- Los experimentos muestran una clara mejora en la tasa de detección tras aplicar la eliminación de ruido basada en BKSVD, siendo importante seleccionar adecuadamente los parámetros del método de *denoising*. Sin embargo, el alto nivel de ruido en las imágenes, además de su carácter no gaussiano, limitan el rendimiento del algoritmo al usar un detector basado en extracción de rasgos.
- La metodología basada en una red convolucional produce resultados de detección ostensiblemente mejores, siendo estos adecuados para la implementación práctica del sistema. Para la metodología clásica, sería necesario un estudio más amplio de estrategias de extracción de rasgos para mejorar su rendimiento.

2. Trabajo futuro

Concluimos la memoria con una enumeración de vías de trabajo futuro surgidas de la presente investigación.

- Adaptación de la estimación inicial del tamaño de diccionario basándonos en la aproximación propuesta en [37].
- Aplicación al problema de la superresolución de los métodos de aprendizaje de diccionarios desarrollados.

- Continuación del estudio de las distribuciones *a posteriori* laplacianas como alternativa para la fase de codificación *sparse*.
- Extensión del modelo de deconvolución propuesto considerando la misma varianza de la *a priori* para diferentes píxeles de la imagen como zonas planas o fronteras, en lugar de una varianza diferente por píxel e introducir información sobre estos parámetros mediante hiperdistribuciones *a priori*.
- Investigación del uso de diferentes heurísticas como, por ejemplo, poner a cero el valor estimado para ω_{γ_i} en píxeles saturados.
- Inclusión de la estimación de la varianza de ruido en el modelo de deconvolución.
- Adaptación del modelo a deconvolución espacialmente variante. Dado que la estimación de la imagen se realiza píxel a píxel, sería inmediata esta adaptación. En este caso, se podría adaptar el eficiente método de restauración espacialmente variante propuesto en [136, 137] para el proceso de estimación del núcleo.
- La paralelización del algoritmo de estimación de la imagen con *a priori Spike-and-Slab* necesita de más investigación para aclarar su viabilidad. Esto permitiría el uso de la FFT en el algoritmo acelerando en gran medida el algoritmo.
- Integración del preprocesamiento y la clasificación/detección. Se ha visto el preprocesamiento como un «bloque» separado de la detección misma, esperando que tras este, una mejora del funcionamiento de los clasificadores. Resulta interesante estudiar la mejora de las imágenes como un proceso integrado en la detección, es decir, el modelo deber aprender cómo mejorar la imagen para garantizar un incremento de la precisión.

Bibliografía

- [1] N. E. Alexander, C. Callejero Andres, and R. Gonzalo. Multispectral mm-wave imaging: materials and images. *Proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE)*, 6948:694803–694803–10, 2008.
- [2] Julie Accardo and M. Ahmad Chaudhry. Radiation exposure and privacy concerns surrounding full-body scanners in airports. *Journal of Radiation Research and Applied Sciences*, 7(2):198 – 200, 2014.
- [3] GB Rybicki and AP Lightman. Fundamentals of radiative transfer. *Radiative Processes in Astrophysics*, page 39, 1979.
- [4] Yang Meng, Anyong Qing, Chuan Lin, Jiefeng Zang, Yizhe Zhao, and Cheng Zhang. Passive millimeter wave imaging system based on helical scanning. *Scientific reports*, 8, 2018.
- [5] Santiago López-Tapia, Rafael Molina, and Nicolás Pérez de la Blanca. Using machine learning to detect and localize concealed objects in passive millimeter-wave images. *Engineering Applications of Artificial Intelligence*, 67:81 – 90, 2018.
- [6] S. López Tapia, R. Molina, and N. Pérez de la Blanca. Deep cnns for object detection using passive millimeter sensors. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
- [7] CD Haworth, BG Gonzalez, M Tomsin, R Appleby, P Coward, A Harvey, K Lebart, Y Petillot, and E Trucco. Image analysis for object detection in millimetre-wave images. In *Passive Millimetre-wave and Terahertz Imaging and Technology*, volume 5619, pages 117–128, December 2004.
- [8] C.D. Haworth, Y.R. Petillot, and E. Trucco. Image processing techniques for metallic object detection with millimetre-wave images. *Pattern Recognition Letters*, 27(15):1843–1851, 2006.
- [9] O. Martínez, L. Ferraz, X. Binefa, I. Gómez, and C. Dorronsoro. Concealed object detection and segmentation over millimetric waves

BIBLIOGRAFÍA

- images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 31–37, June 2010.
- [10] Seokwon Yeom, Dong-Su Lee, YuShin Jang, Mun-Kyo Lee, and Sang-Won Jung. Real-time concealed-object detection and recognition with passive millimeter wave imaging. *Optics Express*, 20(9):9371–9381, Apr 2012.
- [11] Seokwon Yeom, Dong-Su Lee, and Jung-Young Son. Shape feature analysis of concealed objects with passive millimeter wave imaging. *Progress In Electromagnetics Research Letters*, 57:131–137, 2015.
- [12] B. Kumar, P. Sharma, R. Upadhyay, D. Singh, and K. P. Singh. Optimization of image processing techniques to detect and reconstruct the image of concealed blade for mmw imaging system. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 76–79, July 2016.
- [13] Hoda Mohammadzade, Benyamin Ghojogh, Sina Faezi, and Mahdi Shabany. Critical object recognition in millimeter-wave images with robustness to rotation and scale. *Journal of the Optical Society of America A*, 34(6):846–855, Jun 2017.
- [14] J.G. Serra, M. Testa, R. Molina, and A.K. Katsaggelos. Bayesian k-svd using fast variational inference. *IEEE Transactions on Image Processing*, 26(7):3344–3348, July 2017.
- [15] J.G. Serra, S. Villena, R. Molina, and A.K. Katsaggelos. Greedy bayesian double sparsity dictionary learning. In *IEEE International Conference on Image Processing (ICIP 2017)*, pages 1935–1939. Beijing (China), September 2017.
- [16] Eero P. Simoncelli, William T. Freeman, Edward H. Adelson, and David J. Heeger. Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607, 1992.
- [17] Bruno A Olshausen and D J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.
- [18] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997.
- [19] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

-
- [20] Michael Elad, Mario AT Figueiredo, and Yi Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010.
- [21] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [22] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.
- [23] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2698, 2010.
- [24] Yue Huang, John Paisley, Qin Lin, Xinghao Ding, Xueyang Fu, and Xiao-ping Zhang. Bayesian Nonparametric Dictionary Learning for Compressed Sensing MRI. *IEEE Transactions on Image Processing*, 23(12):5007–5019, 2013.
- [25] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3501–3508, 2010.
- [26] T. H. Vu, H. S. Mousavi, V. Monga, G. Rao, and U. K. A. Rao. Histopathological image classification using discriminative feature-oriented dictionary learning. *IEEE Transactions on Medical Imaging*, 35(3):738–751, March 2016.
- [27] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311, 2006.
- [28] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696, 2009.
- [29] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric Bayesian dictionary learning for sparse image representations. *IEEE Transactions on Image Processing*, 2009.
- [30] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang. A non-convex relaxation approach to sparse dictionary learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1809–1816, 2011.

BIBLIOGRAFÍA

- [31] M S Lewicki and T J Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000.
- [32] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443–2446. IEEE, 1999.
- [33] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003.
- [34] Irina F Gorodnitsky and Bhaskar D Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, 1997.
- [35] David M Bradley and J Andrew Bagnell. Differential Sparse Coding. *Neural Information Processing Systems*, 2008.
- [36] M Girolami. A variational method for learning sparse and overcomplete representations. *Neural computation*, 13(11):2517–2532, 2001.
- [37] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, 21(1):130–144, 2012.
- [38] John Paisley and Lawrence Carin. Nonparametric factor analysis with beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 777–784. ACM, 2009.
- [39] David Knowles and Zoubin Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In *International Conference on Independent Component Analysis and Signal Separation*, pages 381–388. Springer, 2007.
- [40] Hemant Ishwaran and J Sunil Rao. Spike and slab variable selection: frequentist and bayesian strategies. *Annals of Statistics*, pages 730–773, 2005.
- [41] Abdul-Saboor Sheikh, Jacquelyn A Shelton, and Jörg Lücke. A truncated em approach for spike-and-slab sparse coding. *Journal of Machine Learning Research*, 15(1):2653–2687, 2014.

-
- [42] L. Li, J. Silva, M. Zhou, and L. Carin. Online Bayesian dictionary learning for large datasets. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2157–2160, 2012.
- [43] Michael R. Andersen, Ole Winther, and Lars K. Hansen. Bayesian inference for structured spike and slab priors. In *Advances in Neural Information Processing Systems*, pages 1745–1753, 2014.
- [44] M. R. Andersen, A. Vehtari, O. Winther, and L. K. Hansen. Bayesian inference for spatio-temporal spike and slab priors. *ArXiv e-prints*, September 2015.
- [45] H. S. Mousavi, V. Monga, and T. D. Tran. Iterative convex refinement for sparse recovery. *IEEE Signal Processing Letters*, 22(11):1903–1907, Nov 2015.
- [46] Tso-Jung Yen. A majorization-minimization approach to variable selection using spike and slab priors. *The Annals of Statistics*, pages 1748–1775, 2011.
- [47] Yizhe Zhang, Ricardo Henao, Chunyuan Li, and Lawrence Carin. Bayesian dictionary learning with gaussian processes and sigmoid belief networks. *unpublished*, 2016.
- [48] Miguel Lázaro-Gredilla and Michalis K. Titsias. Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in neural information processing systems*, pages 2339–2347, 2011.
- [49] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 399–406, 2010.
- [50] Arthur Szlam, Karol Gregor, and Yann LeCun. *Fast Approximations to Structured Sparse Coding and Applications to Object Classification*, pages 200–213. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [51] Mostafa Sadeghi, Massoud Babaie-Zadeh, and Christian Jutten. Learning overcomplete dictionaries based on atom-by-atom updating. *IEEE Transactions on Signal Processing*, 62(4):883–891, 2014.
- [52] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [53] Karl Skretting and Kjersti Engan. Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing*, 58(4):2121–2130, 2010.

BIBLIOGRAFÍA

- [54] Kjersti Engan, Karl Skretting, and J. H. Husø y. Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation. *Digital Signal Processing: A Review Journal*, 17(1):32–49, 2007.
- [55] Qiegen Liu, Shanshan Wang, Kun Yang, Jianhua Luo, Yuemin Zhu, and Dong Liang. 2013 Qiegen Liu Highly Undersampled Magnetic Resonance Image Reconstruction Using Two-Level Bregman Method With Dictionary Updating. 32(7):1290–1301, 2013.
- [56] Qiegen Liu, Dong Liang, Ying Song, Jianhua Luo, Yuemin Zhu, and Wenshu Li. Augmented Lagrangian-Based Sparse Representation Method with Dictionary Updating for Image Deblurring. 6(3):1689–1718, 2013.
- [57] Rémi Gribonval, Rodolphe Jenatton, and Francis Bach. Sparse and Spurious: Dictionary Learning With Noise and Outliers. *IEEE Transactions on Information Theory*, 61(11):6298–6319, 2015.
- [58] S Derin Babacan, Rafael Molina, Minh N Do, and Aggelos K. Katsaggelos. Bayesian blind deconvolution with general sparse image priors. In *Computer Vision–ECCV 2012*, pages 341–355. Springer, 2012.
- [59] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.
- [60] M. E. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, pages 3–6, 2003.
- [61] S. D. Babacan, R. Molina, and A. K. Katsaggelos. Bayesian compressive sensing using Laplace priors. *IEEE Transactions on Image Processing*, 19(1):53–63, Jan 2010.
- [62] Z. Chen, R. Molina, and A. K. Katsaggelos. Automated recovery of compressedly observed sparse signals from smooth background. *IEEE Signal Processing Letters*, 21(8):1012–1016, Aug 2014.
- [63] S D Babacan, L Mancera, R Molina, and A K Katsaggelos. Bayesian Compressive Sensing Using Non-Convex Priors. *European Signal Processing Conference 2009 EUSIPCO09*, 2009.
- [64] Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Publishing Company, Incorporated, 1st edition, 2010.

-
- [65] D. Barchiesi and M. D. Plumbley. Learning incoherent dictionaries for sparse approximation using iterative projections and rotations. *IEEE Transactions on Signal Processing*, 61(8):2055–2065, April 2013.
- [66] R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3):1553–1564, March 2010.
- [67] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York, 2009.
- [68] R. Liang, Z. Zhao, and S. Li. Image denoising using learned dictionary based on double sparsity model. In *2011 4th International Congress on Image and Signal Processing*, volume 2, pages 691–695, Oct 2011.
- [69] F. Li and S. Zhang. Double sparse dictionary learning for image super resolution. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 4344–4348, May 2016.
- [70] X. Zhan, R. Zhang, D. Yin, A. Hu, and W. Hu. Remote sensing image compression based on double-sparsity dictionary learning and universal trellis coded quantization. In *2013 IEEE International Conference on Image Processing*, pages 1665–1669, Sept 2013.
- [71] S. Li and H. Qi. Compressed dictionary learning for detecting activations in fmri using double sparsity. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 434–437, Dec 2014.
- [72] M. Abavisani, M. Joneidi, S. Rezaeifar, and S. B. Shokouhi. A robust sparse representation based face recognition system for smartphones. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6, Dec 2015.
- [73] J.G. Serra, J. Mateos, R. Molina, and A.K. Katsaggelos. Variational em method for blur estimation using the spike-and-slab image prior. *Digital Signal Processing*, 88:116–129, May 2019.
- [74] P. Ruiz, X. Zhou, J. Mateos, R. Molina, and A.K. Katsaggelos. Variational Bayesian blind image deconvolution: A review. *Digit. Signal Process.*, 47:116–127, 2015.
- [75] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. CRC Press, 1998.
- [76] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *2011 IEEE Conf.*

BIBLIOGRAFÍA

- Comput. Vision and Pattern Recognition (CVPR)*, pages 2657–2664, 2011.
- [77] L. Xu, S. C. Zheng, and J. Y. Jia. Unnatural l0 sparse representation for natural image deblurring. In *2013 IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pages 1107–1114, 2013.
- [78] R. Molina, J. Mateos, and A. K. Katsaggelos. Blind deconvolution using a variational approach to parameter, image, and blur estimation. *IEEE Trans. Image Process.*, 15(12):3715–3727, 2006.
- [79] David Wipf and Haichao Zhang. Revisiting Bayesian blind deconvolution. *Journal of Machine Learning Research*, 15:3775–3814, 2014.
- [80] R. Molina, A. K. Katsaggelos, and J. Mateos. Bayesian and regularization methods for hyperparameter estimation in image restoration. *IEEE Trans. Image Process.*, 8(2):231–246, 1999.
- [81] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Process. Mag.*, 14(2):24–41, 1997.
- [82] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Process. Mag.*, 13(3):43–64, 1996.
- [83] Yongyi Yang, Nikolas P. Galatsanos, and Henry Stark. Projection-based blind deconvolution. *J. Opt. Soc. Amer. A*, 11(9):2401–2409, 1994.
- [84] R. Molina, A. K. Katsaggelos, J. Abad, and J. Mateos. A Bayesian approach to blind deconvolution based on Dirichlet distributions. In *1997 IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, pages 2809–2812, 1997.
- [85] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pages 1964–1971, 2009.
- [86] Subhasis Chaudhuri, Rajbabu Velmurugan, and RM Rameshan. *Blind image deconvolution*. Springer, 2014.
- [87] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *2011 IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pages 233–240, 2011.
- [88] W. Zuo, Dongwei Ren, Shuhang Gu, Liang Lin, and L. Zhang. Discriminative learning of iteration-wise priors for blind

- deconvolution. In *2015 IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pages 3232–3240, 2015.
- [89] J. Pan, D. Sun, H. Pfister, and M. H. Yang. Deblurring images via dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.
- [90] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Inform. Process. Sys. (NIPS)*, pages 1033–1041, 2009.
- [91] J. Kotera, F. Šroubek, and P. Milanfar. Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors. In *2013 Int. Conf. Comp. Anal. Imag. and Patterns. (CAIP)*, pages 59–66, 2013.
- [92] Daniele Perrone, Remo Diethelm, and Paolo Favaro. Blind deconvolution via lower-bounded logarithmic image priors. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 112–125, 2015.
- [93] Wen-Ze Shao, Hai-Bo Li, and Michael Elad. Bi-10-12-norm regularization for blind motion deblurring. *Journal of Visual Communication and Image Representation*, 33:42 – 59, 2015.
- [94] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 157–170, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [95] D. Gong, M. Tan, Y. Zhang, A. v. d. Hengel, and Q. Shi. Blind image deconvolution by automatic gradient activation. In *2016 IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pages 1827–1836, 2016.
- [96] S. Tao, W. Dong, Z. Tang, and Q. Wang. Blind image deconvolution using the gaussian scale mixture fields of experts prior. In *2017 International Conference on Progress in Informatics and Computing (PIC)*, pages 190–195, Dec 2017.
- [97] Yu Cai, Jinshan Pan, and Zhixun Su. Blind image deblurring via salient structure detection and sparse representation. In Shin’ichi Satoh, editor, *Image and Video Technology*, pages 283–299, Cham, 2018. Springer International Publishing.

BIBLIOGRAFÍA

- [98] Lerenhan Li, Jinshan Pan, Wei-Sheng Lai, Changxin Gao, Nong Sang, and Ming-Hsuan Yang. Learning a discriminative prior for blind image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page accepted for publication, 03 2018.
- [99] M. Ljubenović and M. A. T. Figueiredo. Blind image deblurring using class-adapted image priors. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 490–494, Sept 2017.
- [100] C. L. Likas and N. P. Galatsanos. A variational approach for Bayesian blind image deconvolution. *IEEE Trans. Signal Process.*, 52(8):2222–2233, 2004.
- [101] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, 2006.
- [102] S. Babacan, R. Molina, M. Do, and A. K. Katsaggelos. Bayesian blind deconvolution with general sparse image priors. In *2012 Eur. Conf. Comput. Vision (ECCV)*, pages 341–355, 2012.
- [103] A. Mohammad-Djafari. Bayesian blind deconvolution of images comparing JMAP, EM and BVA with a Student-t a priori model. In *Int. Work. on Elec. Comp. Eng. Subfields*, pages 98–103, 2014.
- [104] X. Zhou, M. Vega, F. Zhou, R. Molina, and A. K. Katsaggelos. Fast Bayesian blind deconvolution with Huber super Gaussian priors. *Digit. Signal Process.*, 60:122–133, 2017.
- [105] R. Giri and B. Rao. Type I and type II Bayesian methods for sparse signal recovery using scale mixtures. *IEEE Trans. Signal Process.*, 64:3418–3428, 2016.
- [106] R. Giri and B. Rao. Learning distributional parameters for adaptive Bayesian sparse signal recovery. *IEEE Comput. Intell. Mag.*, 11:14–23, 2016.
- [107] X. Zhou, J. Mateos, F. Zhou, R. Molina, and A. K. Katsaggelos. Variational Dirichlet blur kernel estimation. *IEEE Trans. Image Process.*, 24:5127–5139, 2015.
- [108] M. K. Titsias and M. Lázaro-Gredilla. Spike and Slab variational inference for multi-task and multiple kernel learning. In *Advances in Neural Inform. Process. Sys. (NIPS)*, pages 2339–2347, 2011.
- [109] L. Chaari, J.-Y. Toumeret, and H. Batatia. Sparse Bayesian regularization using Bernoulli-Laplacian priors. In *Eur. Signal Process. Conf. (EUSIPCO)*, pages 1–5, 2013.

-
- [110] L. Chaari, J. Y. Toumeret, and C. Chaux. Sparse signal recovery using a Bernoulli generalized Gaussian prior. In *Eur. Signal Process. Conf. (EUSIPCO)*, pages 1711–1715, 2015.
- [111] H. Ishwaran and J. S. Rao. Spike and slab variable selection: Frequentist and Bayesian strategies. *The Annu. of Stat.*, 33(2):730–773, 2005.
- [112] S. A. Zilqurnain Naqvi. *Efficient sparse Bayesian learning using spike-and-slab priors*. PhD thesis, Purdue University, 2016.
- [113] G. Malsiner-Walli and H Wagner. Comparing spike and slab priors for Bayesian variable selection. *Austrian J. Stat.*, 40:241–264, 2011.
- [114] X. Huang, J. Wang, and F. Liang. A variational Bayesian algorithm for variable selection. *ArXiv e-prints*, 2016.
- [115] X. Lu, Y. Yuan, and P. Yan. Sparse coding for image denoising using spike and slab prior. *Neurocomput.*, 106:12–20, 2013.
- [116] S. Mohamed, K. A. Hellerand, and Z. Ghahramani. Bayesian and l_1 approaches for sparse unsupervised learning. In *Int. Conf. Mach. Learn. (ICML)*, pages 751–758, 2012.
- [117] V. Ročková and E. I. George. Negotiating multicollinearity with spike-and-slab priors. *Metron*, 72:217–229, 2014.
- [118] H. Liao and M. K. Ng. Blind deconvolution using generalized cross-validation approach to regularization parameter estimation. *IEEE Trans. Image Process.*, 20:670–680, 2011.
- [119] J. M. Hernández-Lobato, D. Hernández-Lobato, and A. Suárez. Expectation propagation in linear regression models with spike-and-slab priors. *Mach. Learn.*, 99:437–487, 2015.
- [120] Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John W. Paisley. Non-parametric Bayesian dictionary learning for sparse image representations. In *Advances in Neural Inform. Process. Sys. (NIPS)*, pages 2295–2303, 2009.
- [121] J.G. Serra, J. Mateos, R. Molina, and A.K. Katsaggelos. Spike and slab variational inference for blind image deconvolution. In *IEEE International Conference on Image Processing (ICIP 2017)*, pages 3765–3769. Beijing (China), September 2017.
- [122] J.G. Serra, J. Mateos, R. Molina, and A.K. Katsaggelos. Parameter estimation in spike and slab variational inference for blind image deconvolution. In *European Signal Processing Conference (EUSIPCO 2017)*, pages 1540–1544. Kos (Greece), August 2017.

BIBLIOGRAFÍA

- [123] X. Zhou, R. Molina, F. Zhou, and A. K. Katsaggelos. Fast iteratively reweighted least squares for l_p regularized image deconvolution and reconstruction. In *2014 IEEE Int. Conf. Image Process. (ICIP)*, pages 1783–1787, 2014.
- [124] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3):70, July 2007.
- [125] Xu Zhou, Fugen Zhou, Xiangzhi Bai, and Bindang Xue. A boundary condition based deconvolution framework for image deblurring. *Journal of Computational and Applied Mathematics*, 261:14–29, May 2014.
- [126] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise level estimation for blind denoising. *IEEE Trans. Image Process.*, 22(12):5226–5237, 2013.
- [127] C. Sutour, J. F. Aujol, and C. A. Deledalle. Automatic estimation of the noise level function for adaptive blind denoising. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 76–80, 2016.
- [128] M. Rakhshanfar and M. A. Amer. Estimation of Gaussian, Poissonian-Gaussian, and processed visual noise and its level function. *IEEE Trans. Image Process.*, 25(9):4172–4185, 2016.
- [129] L. Dong, J. Zhou, and Y. Y. Tang. Noise level estimation for natural images based on scale-invariant kurtosis and piecewise stationarity. *IEEE Trans. Image Process.*, 26(2):1017–1030, 2017.
- [130] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. *ACM Trans. Graph.*, 28(5):145:1–145:8, December 2009.
- [131] Rolf Köhler, Michael Hirsch, Betty Mohler, Bernhard Schölkopf, and Stefan Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 27–40, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [132] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision*, pages 555–562, Jan 1998.
- [133] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29:51–59, January 1996.

- [134] Timo Ahonen, Jiří Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Image Analysis*, volume 5575 of *Lecture Notes in Computer Science*, pages 61–70. Springer Berlin Heidelberg, 2009.
- [135] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec 2017.
- [136] David Miraut and Javier Portilla. Efficient shift-variant image restoration using deformable filtering (part I). *EURASIP Journal on Advances in Signal Processing*, 2012:100, May 2012.
- [137] David Miraut, Johannes Ballé, and Javier Portilla. Efficient shift-variant image restoration using deformable filtering (part II): PSF field estimation. *EURASIP Journal on Advances in Signal Processing*, 2012:193, Aug 2012.