UNIVERSIDAD DE GRANADA

E.T.S. DE INGENIERÍA INFORMÁTICA

Departamento de Ciencias de la Computación e Inteligencia Artificial

RECONSTRUCCIÓN AUTOMÁTICA DE IMÁGENES COMPRIMIDAS MEDIANTE TRANSFORMADA COSENO DISCRETA USANDO MÉTODOS BAYESIANOS

TESIS DOCTORAL

Javier Mateos Delgado

Granada, junio de 1.998

RECONSTRUCCIÓN AUTOMÁTICA DE IMÁGENES COMPRIMIDAS MEDIANTE TRANSFORMADA COSENO DISCRETA USANDO MÉTODOS BAYESIANOS

MEMORIA QUE PRESENTA

JAVIER MATEOS DELGADO

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

JUNIO 1.998

DIRECTORES

RAFAEL MOLINA SORIANO

AGGELOS K. KATSAGGELOS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
E INTELIGENCIA ARTIFICIAL

E.T.S. DE INGENIERÍA INFORMÁTICA UNIVERSIDAD DE GRANADA

La memoria titulada **Reconstrucción automática de imágenes comprimidas mediante transformada coseno discreta usando métodos bayesianos**, que presenta D. Javier Mateos Delgado para optar al grado de DOCTOR, ha sido realizada en el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección de los Doctores D. Rafael Molina Soriano y D. Aggelos K. Katsaggelos.

Granada, junio de 1.998

El doctorando — Los directores

Javier Mateos Delgado — Rafael Molina Soriano — Aggelos K. Katsaggelos

AGRADECIMIENTOS

Quiero agradecer a los directores de esta tesis, Rafael Molina Soriano y Aggelos K. Katsaggelos, el apoyo y los consejos recibidos durante la realización de la misma y el estímulo constante a la investigación que han realizado.

También quiero agradecer al grupo de "Análisis de imágenes digitales y sus aplicaciones" su apoyo económico que ha sufragado parte de los gastos para realizar esta tesis. Igualmente, hago extensivo mi agradecimiento a los miembros del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada que me apoyaron en mi trabajo.

Quisiera agradecer especialmente a Javier Abad Ortega tanto su ayuda prestada en la confección del software como su ánimo y apoyo moral.

Este trabajo ha sido parcialmente subvencionado por los proyectos de la Comisión Interministerial de Ciencia y Tecnología números PB93-1110 y TIC97-0989.

Parte de los contenidos de esta memoria han sido previamente publicados en varios congresos nacionales e internacionales [70, 71, 72, 74], en un capítulo de libro [80], y enviados a la revista IEEE Transaction on Image Processing [73].

Esta memoria está dedicada a mi familia y amigos y, muy especialmente, a Marian, que conoce bien el esfuerzo empleado en esta tesis y le da sentido.

Gracias a todos.

A mis padres

y

a Marian

RECONSTRUCCIÓN AUTOMÁTICA DE IMÁGENES COMPRIMIDAS MEDIANTE TRANSFORMADA COSENO DISCRETA USANDO MÉTODOS BAYESIANOS

JAVIER MATEOS DELGADO

Índice General

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción general

1.1 Introducción a la compresión de imágenes

A partir de la década de los 80 la comunicación visual mediante tecnología digital ha ido cobrando cada vez mayor importancia. Hoy por hoy las imágenes digitales, bien sean estáticas o en movimiento, forman parte de una gran variedad de formas de comunicación humana, siendo necesaria la transmisión y almacenamiento de estas imágenes. Sin embargo, las aplicaciones en las que se usan imágenes digitales son aún escasas debido a su alto coste relativo. El principal obstáculo para muchas aplicaciones es la enorme cantidad de datos necesarios para representar una imagen digital directamente. En televisión, por ejemplo, cada imagen tiene aproximadamente 600×400 pixels lo que, usando 8 bits por pixel para cada uno de las tres bandas de color y mostrando 30 fotogramas por segundo, nos obliga a disponer de un ancho de banda y memoria suficiente para transmitir sobre 180×10^6 bits por segundo. Dependiendo de las aplicaciones, el tamaño de las imágenes digitales sin procesar puede variar entre 10^5 y 10^8 bits por imagen. Debido al costo de almacenamiento o transmisión, el uso de imágenes digitales hace inviables muchas aplicaciones aunque los dispositivos de captación y visualización sean bastante asequibles. Esta gran cantidad de datos hace que la compresión de las imágenes sea necesaria tanto para disminuir las necesidades de canal como los de memoria.

La compresión de imágenes trata de minimizar el número de bits necesarios para representar una imagen explotando la redundancia de los datos de forma que la posterior reconstrucción de la imagen sea posible. Además de la transmisión de imágenes de televisión, la compresión de imágenes se usa, junto a otras técnicas, en teledetección, imágenes de radar y sonar, te-

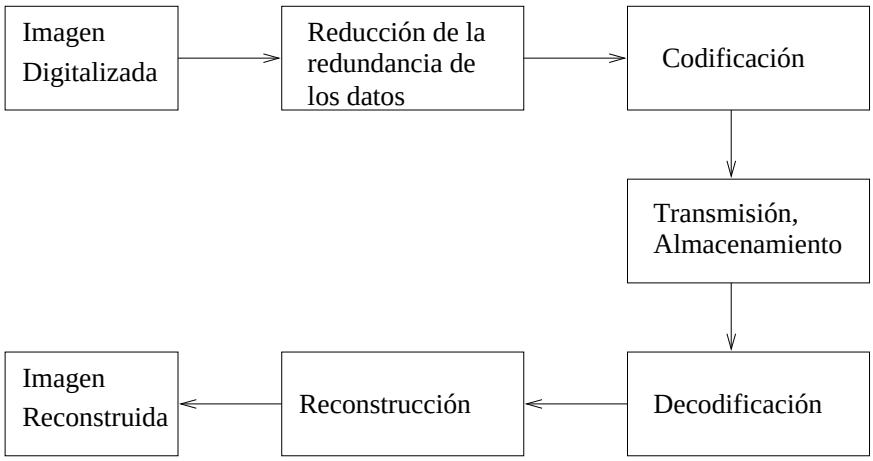


Figura 1.1: Compresión de los datos y reconstrucción de la imagen.

leconferencia, comunicación por computadora, transmisión por fax y otras. Reduciendo las necesidades de almacenamiento, la compresión de imágenes se usa sobre documentos, imágenes médicas, secuencias de imágenes, imágenes de satélite, aplicaciones multimedia, etc..

En la figura 1.1 [111] se presenta un algoritmo general para la compresión y descompresión de la imagen. El primer paso elimina la información redundante causada por la alta correlación de los datos de la imagen, posiblemente mediante técnicas basadas en transformaciones como la transformada coseno discreta. El segundo paso es la codificación de los datos transformados usando un código de longitud fija o variable. Después de transmitir o almacenar los datos comprimidos, estos se decodifican y se reconstruye la imagen. Normalmente, se suele llamar codificador a todo el sistema, bien físico o programa, que realiza la compresión y, decodificador, al que realiza la descompresión, siendo usual realizar una definición conjunta de ambos mediante el término codificador-decodificador o *codec*.

Los métodos de compresión de imágenes se pueden dividir en dos grupos: compresión que preserva la información o **sin pérdidas** y, por tanto, permite la reconstrucción de los valores de la imagen sin error y, métodos de compresión **con perdidas** de información, que no preservan completamente la información de la imagen, es decir, la imagen resultante después del proceso de compresión y descompresión no será exactamente la misma que se tenía antes de codificarla aunque, en la mayoría de los casos, la diferencia no se apreciará a simple vista. De hecho, para muchas de las aplicaciones de la compresión de imágenes la segunda aproximación es totalmente válida siempre que no se causen cambios significativos en la imagen. Este es

el caso, por ejemplo, de la televisión digital en el que no es tan importante que la imagen se pueda reconstruir sin pérdidas como que ocupe poco espacio y la calidad no baje de cierto límite visualmente aceptable. Sin embargo, es posible comprimir la imagen más allá de este límite, perdiendo más información, y llegando a una imagen visualmente desagradable o inaceptable. El problema a tratar en esta memoria es la reconstrucción de este tipo de imágenes altamente comprimidas y, por tanto, altamente degradadas, para que sean visualmente aceptables.

El primer grupo de métodos, los métodos de compresión sin pérdidas, se basa en la reducción de la redundancia estadística de la imagen. La redundancia estadística está relacionada con la similitud, correlación y predictibilidad de los valores de la imagen. Este tipo de redundancia tiene la propiedad de que puede ser eliminada, reduciendo el número de bits necesario para representar la imagen, sin destruir ningún tipo de información presente en la imagen. Esto significa que si se descomprime la imagen, el resultado será una imagen exactamente igual a la que se tenía antes de comprimirla. Estos métodos, sin embargo, no suelen ser suficientes por sí solos para la compresión de imágenes digitales debido a las bajas razones de compresión obtenidas, que suelen ser de 2:1 a 16:1 [42], aunque sí son útiles para comprimir los datos resultantes de aplicar un método de compresión con pérdidas. Así, es usual aplicar DPCM (*Differential Pulse Code Modulation*) y/o una codificación por entropía tras realizar una cuantificación de los datos de la imagen. En esta memoria no vamos a estudiar los métodos de compresión sin pérdidas, haciendo mención, solamente, a aquellos usados por algunos métodos estándar de codificación con pérdidas para obtener una mayor reducción de los datos. El lector puede encontrar una extensa visión de los métodos de codificación sin pérdidas, por ejemplo, en [42] y [91].

Los métodos de compresión con pérdidas, por otra parte, tratan de eliminar la redundancia subjetiva, es decir, aquella información que se puede eliminar sin que el observador humano note la diferencia. A diferencia de la redundancia estadística, la eliminación de la redundancia subjetiva no es reversible, es decir, los valores originales de la imagen no pueden ser recuperados exactamente tras la eliminación de este tipo de redundancia.

Los métodos de compresión de imágenes con pérdidas más usuales son los basados en el uso de la transformada coseno discreta (DCT), cuyo máximo exponente es JPEG, los basados en el uso de la transformada *wavelet*, los basados en el uso de cuantificación de vectores y los métodos de compresión fractal. Para compresión de vídeo también existen métodos basados en los mismos principios adaptados a las características propias de las secuencias de imágenes.

Entre ellos caben destacar los estándares de vídeoconferencia H.261 y H.263 y el estándar de uso doméstico MPEG, ambos basados en DCT, si bien existen otros métodos basados en cuantificación de vectores, como los formatos CinePak de Radius o INDEO de Intel, o basados en wavelets [49, 122].

En esta memoria nos vamos a centrar en los métodos basados en la transformada coseno discreta y, concretamente, en JPEG aunque estos métodos serán extensibles a métodos de compresión de vídeo como MPEG o H.261 y H.263. Como ya se comentó anteriormente, todos estos métodos de compresión con pérdidas producen, a altas razones de compresión, artificios que hacen que la imagen no sea agradable al sistema visual humano. Una descripción de los artificios posibles en los estándares de compresión de vídeo puede encontrarse en [133]. Muchos de estos artificios están también presentes en las imágenes comprimidas mediante JPEG, siendo el más visible llamado *artificio de bloques* pero, antes de entrar en la descripción de este artificio, es conveniente estudiar el funcionamiento de los métodos de compresión que lo producen. En las siguientes secciones estudiaremos el estándar de compresión de imágenes JPEG y haremos una introducción a los métodos H.261, H.263 y MPEG.

Por último, y antes de describir estos métodos de compresión estándar, necesitaremos tener medidas para representar la compresión alcanzada. Se suelen usar dos medidas para valorar la compresión obtenida por un método; una es la razón de compresión, medida como el cociente entre el tamaño de la imagen original y el tamaño de la imagen comprimida. Es usual representar este cociente como el número de bits de la imagen original que se corresponden con un bit de la imagen comprimida. Así, por ejemplo, una imagen con 256×256 pixels y una precisión de 8 bits por pixel, tiene un tamaño de 524288 bits. Si al comprimir esta imagen conseguimos una imagen con un tamaño de 16384 bits, la razón de compresión será 32:1, que se lee como *treinta y dos a uno*. La otra forma habitual de representar la compresión obtenida es el número de bits necesarios en la imagen comprimida para representar cada pixel. En el ejemplo anterior habremos obtenido una compresión a 0.25 bit por pixel (bpp).

1.2 Codificación mediante transformada coseno discreta. El estándar JPEG

Entre los métodos de compresión de imágenes con pérdidas, el más extendido es el creado por el grupo conjunto de expertos fotográficos, JPEG (*Joint Photographic Expert Group*). La

necesidad de un estándar internacional para compresión de imágenes estáticas de tonos continuos resultó, en 1.986, en la formación de JPEG. Este grupo fue constituido por ISO[1] y CCITT[2] para desarrollar un estándar de propósito general que se acomodase a tantas aplicaciones como fuera posible. Después de evaluar y probar un número de algoritmos propuestos para compresión de imágenes, el grupo acordó, en 1.988, una técnica basada en la DCT. Desde 1.988 a 1.990 el comité JPEG refinó varios métodos que incorporaban la DCT para compresión con pérdidas. Además, se definió un método para compresión sin pérdidas. Los trabajos del comité se publicaron en tres partes: "Part 1: Requeriments and guidelines" [34] que describe el método de compresión y descompresión de JPEG, "Part 2: Compliance Testing" [35] describe pruebas para verificar si un codificador-decodificador (codec) implementa los algoritmos de JPEG correctamente y "Part 3: Extensions" [36] donde se describen algunas extensiones de JPEG. La referencia [99] describe el método de compresión de JPEG en detalle.

En la siguiente sección daremos una idea general de los algoritmos JPEG y en las siguientes describiremos cada modo de operación en detalle.

1.3 Descripción general de los algoritmos JPEG

El comité JPEG no pudo satisfacer los requerimientos de todas las aplicaciones de las imágenes estáticas con un solo algoritmo. Como resultado, el comité definió cuatro modos de operación:

- *Secuencial basado en DCT* — Las figuras 1.2 y 1.3 presentan un diagrama simplificado de un codec secuencial basado en DCT. En este modo, la imagen se divide en bloques 8×8 que se examinan de izquierda a derecha y de arriba a abajo. Cada bloque consta de 64 valores de uno de los componentes de color que forman la imagen. Cada bloque de 64 valores se transforma a 64 coeficientes mediante la transformada coseno discreta. La DCT concentra la mayoría de la energía de los componentes del bloque en unos pocos

[1]International Organization for Standarization. Una federación mundial de las organizaciones de estándares nacionales de unos 90 países, una por cada país. ISO es una organización no gubernamental para promover el desarrollo de la estandarización, y otras actividades relacionadas, en el mundo. Los trabajos de ISO se traducen en acuerdos internacionales que son publicados como Estándares Internacionales.

[2]Comité Consultatif International Télégraphique et Téléphonique. Un comité de la Unión Internacional de telecomunicaciones (ITU o *International Telecommunications Union*), una agencia de la Naciones Unidas, responsable de hacer recomendaciones técnicas sobre telefonía y sistemas de comunicación de datos para proveedores y compañías de telecomunicaciones.

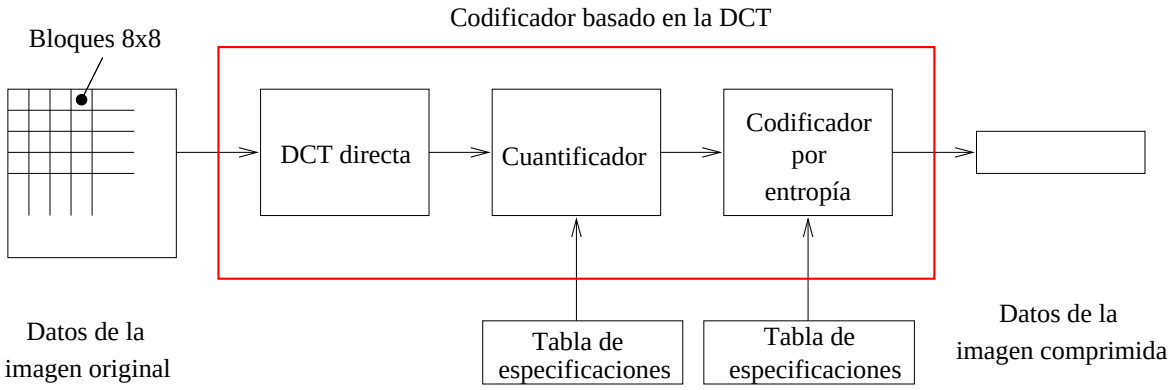


Figura 1.2: Pasos del proceso de codificación basado en la DCT.

coeficientes, normalmente en la esquina superior izquierda del bloque de la DCT [91]. El coeficiente en la esquina superior izquierda se denomina coeficiente DC y es proporcional a la intensidad media del bloque de muestras en el dominio espacial. Los coeficientes AC corresponden a frecuencias cada vez más altas conforme se alejan del coeficiente DC. Los coeficientes se cuantifican, se exploran en una secuencia en zigzag y codifican por entropía. Un modo restringido del modo secuencial es el llamado *modo base* o *baseline mode*, que se requiere que esté en todas las implementaciones de JPEG. La gran mayoría del software y hardware existente soporta sólo el modo base de JPEG.

- *Progresivo basado en DCT* — Este modo ofrece un medio de producir rápidamente una imagen decodificada "burda" cuando el medio que separa el codificador y el decodificador tiene un ancho de banda pequeño. Este método es similar al algoritmo secuencial basado en DCT pero los coeficientes cuantificados se codifican parcialmente en múltiples pasadas.

- *Sin pérdidas* — En este modo, el decodificador recibe una reproducción exacta de la imagen digital de la entrada. Las diferencias entre los valores de la entrada y sus predicciones se codifican por entropía. Las predicciones se obtienen como una combinación de uno a tres pixels vecinos.

- *Jerárquico* — Este modo, también conocido como codificación piramidal, se usa para codificar la imagen en una secuencia de fotogramas. El primer fotograma es una versión de resolución reducida de la imagen original. Los fotogramas siguientes se codifican como fotogramas diferenciales de mayor resolución.

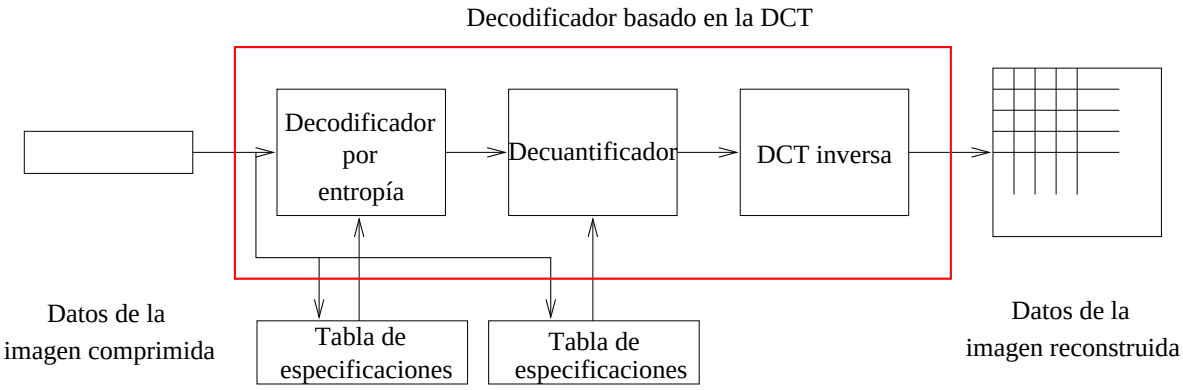


Figura 1.3: Pasos del proceso de decodificación basado en la DCT.

El espacio de color no es parte del estándar. En efecto, JPEG es independiente del espacio de color. Muchas imágenes (normalmente RGB) se convierten a una representación en luminosidad-cromatismo, como la YCbCr o YUV, antes de llevar a cabo este proceso de compresión. Como un primer paso en el proceso de compresión, muchos esquemas de compresión de imágenes explotan la baja sensibilidad del sistema visual humano a información de color de alta frecuencia para reducir la resolución de las bandas cromáticas muestreándolas a la mitad horizontalmente o en ambas direcciones. Los pixels de la entrada son enteros sin signo con precisión de n bits, es decir, con un valor entre 0 y $2^n - 1$.

Para el codificador por entropía se pueden usar técnicas de compresión aritmética o Huffman en cualquiera de los modos de operación (excepto en el modo base, en el que es obligatoria la codificación Huffman). No hay unos códigos Huffman estándar de JPEG sino que el codificador debe transmitir sus tablas de códigos Huffman (hasta 4 para los coeficientes AC y cuatro para los DC) siguiendo un formato específico. Un codificador Huffman comprime una serie de símbolos asignando un código corto a los símbolos que ocurren frecuentemente y un código largo a los poco probables. La salida de la codificación aritmética, por otro lado, es un número real para cada componente de la imagen y son, normalmente, más eficientes que los codificadores Huffman. Para las imágenes de prueba JPEG, la codificación de Huffman (usando tablas fijas) dieron como resultado datos comprimidos que requerían entre un 5 y un 10 por ciento más almacenamiento que la codificación aritmética [123].

Existen una serie de parámetros relacionados con la imagen original y el proceso de compresión que pueden modificarse según las necesidades del usuario. Así, una imagen codificada usando cualquier modo JPEG puede tener de 1 a 65.536 líneas y de 1 a 65.536 pixels por línea.

Cada pixel puede tener desde 1 a 255 componentes de color (se permiten 4 como máximo para el modo progresivo). El modo de operación determina la precisión del pixel de cada componente de color. Para los modos basados en DCT, se soportan 8 o 12 bits de precisión (sólo se permiten 8 bits para el modo base). Antes de la codificación los pixels se desplazan restándole 2^{n-1} para dar un rango con signo de -2^{n-1} a $2^{n-1}-1$. El modo de compresión sin pérdidas permite desde 2 a 16 bits de precisión. Si se usa un modo de operación basado en DCT también hay que definir la precisión del cuantificador. Para 8 bits de precisión de cada componente, la precisión del cuantificador se fija a 8 bits. Los componentes a 12 bits requieren 8 o 15 bits de precisión de cuantificación.

1.3.1 Modo base secuencial basado en DCT

El modo secuencial basado en DCT ofrece buenas razones de compresión manteniendo una calidad de imagen excelente. Un subconjunto de las capacidades del modo secuencial basado en DCT ha sido identificado por JPEG como *modo base* (8 bits por pixel, codificación Huffman, 8 bits de precisión para el cuantificador, hasta 2 tablas Huffman para los coeficientes AC y hasta 2 para los coeficientes DC). Todas las implementaciones de JPEG basadas en DCT requieren incluir la capacidad de modo base. Este requerimiento asegura la portabilidad entre diferentes codecs de diferentes vendedores.

DCT y cuantificación

Todos los codificadores basados en DCT empiezan el proceso de codificación dividiendo la imagen de entrada en bloques 8×8 no solapados. Después de desplazar cada muestra de 8 bits de forma que su rango esté entre -128 y +127, los bloques se transforman al dominio de las frecuencias usando la DCT. Las ecuaciones para la transformación coseno discreta ortogonal vienen dadas por

$$\text{DCT directa} \quad : \quad F(u,v) = \frac{C(u)}{2}\frac{C(v)}{2}\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}$$

$$\text{DCT inversa} \quad : \quad f(x,y) = \sum_{u=0}^{7}\frac{C(u)}{2}\sum_{v=0}^{7}\frac{C(v)}{2}F(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}$$

donde

$$C(u), C(v) = \begin{cases} 1/\sqrt{2} & \text{para } u, v = 0 \\ 1 & \text{en otro caso} \end{cases}$$

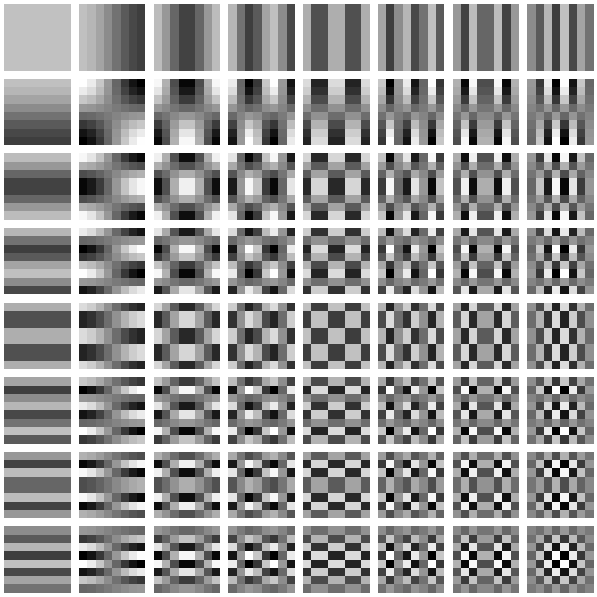


Figura 1.4: Funciones base de la DCT para un bloque 8×8. Cada matriz 8×8 muestra una función diferente 2-D. La frecuencia horizontal de las funciones base de la DCT aumenta de izquierda a derecha y la frecuencia vertical de las funciones de la DCT aumenta de arriba a abajo. La función base del coeficiente DC está, por tanto, en la esquina superior izquierda.

La figura 1.4 muestra la forma de las funciones base de la transformada coseno discreta para cada coeficiente de una matriz 8×8. Cada una de las matrices 8×8 muestra una función base 2-D diferente. La frecuencia horizontal de la DCT aumenta de izquierda a derecha y la frecuencia vertical aumenta de arriba a abajo. La función DC está, por tanto, en la esquina superior izquierda.

El siguiente paso en el proceso, la cuantificación, es la clave de la mayoría de la compresión de JPEG. Para cada imagen, se define una matriz de cuantificación, $\mathcal{Q}(u, v)$, donde cada elemento se corresponde con un coeficiente en el bloque de la DCT. La matriz de cuantificación se usa para reducir la amplitud de los coeficientes y así incrementar el número de coeficientes con valor cero después de la cuantificación. Aunque no hay matrices de cuantificación estándar de JPEG, el grupo incluye un conjunto que da buenos resultados para imágenes del tipo que se usa para la televisión digital. Estas matrices, para las bandas de luminosidad y cromatismo, se muestran en las tablas 1.1 y 1.2, respectivamente. El codificador debe transmitir sus matrices

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Tabla 1.1: Tabla de cuantificación de la luminosidad propuesta por JPEG.

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Tabla 1.2: Tabla de cuantificación del cromatismo propuesta por JPEG.

de cuantificación según un formato preestablecido. La cuantificación y decuantificación se realizan según las siguientes ecuaciones:

$$F\mathcal{Q}(u,v) = \text{redondeo}\left(\frac{F(u,v)}{\mathcal{Q}(u,v)}\right)$$
$$\tilde{F}(u,v) = F\mathcal{Q}(u,v) \times \mathcal{Q}(u,v).$$

Un diseño cuidadoso de las matrices de cuantificación producirán razones de compresión altas con una distorsión visualmente despreciable. Muchas implementaciones JPEG controlan la razón de compresión (y la calidad de la imagen de salida) mediante un factor de calidad (*q-factor*) que normalmente sólo es un factor de escala para las matrices de cuantificación.

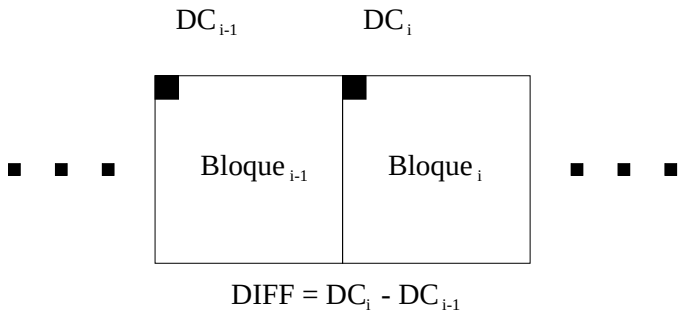


Figura 1.5: Codificación diferencial de los coeficientes DC.

Codificación de los coeficientes y secuencia zigzag

Tras la cuantificación, el coeficiente DC se trata de forma separada de los 63 coeficientes AC. El coeficiente DC es una medida del valor medio de los 64 pixels de la imagen y, ya que normalmente hay un fuerte correlación entre los coeficientes DC de dos bloques 8×8 adyacentes, los coeficientes DC se codifican como la diferencia entre el término DC del bloque previo en el orden de codificación, como se muestra en la figura 1.5. Este tratamiento especial merece la pena puesto que los coeficientes DC contienen una fracción significativa de la energía total de la imagen.

Finalmente, todos los coeficientes cuantificados se ordenan en una secuencia en zigzag, como se muestra en la figura 1.6. Esta ordenación ayuda a facilitar la codificación por entropía colocando los coeficientes de las bajas frecuencias, que normalmente no son cero, antes de los coeficientes de las altas frecuencias. De esta forma habrá una tendencia a tener cadenas largas de ceros y, puesto que solamente se codifican por entropía los coeficientes AC no nulos, la compresión alcanzada siguiendo la secuencia en zigzag será alta.

Codificación por entropía

El paso final de la codificación basada en la DCT es la codificación por entropía. Este paso consigue una compresión sin pérdidas adicional al codificar los coeficientes cuantificados de la DCT de una forma más compacta basándose en sus características estadísticas. La propuesta de JPEG especificaba dos métodos de codificación por entropía: Codificación de Huffman y codificación aritmética.

Es útil considerar la codificación por entropía como un proceso de dos pasos. El primero convierte la secuencia de coeficientes cuantificados en zigzag en una secuencia de símbolos

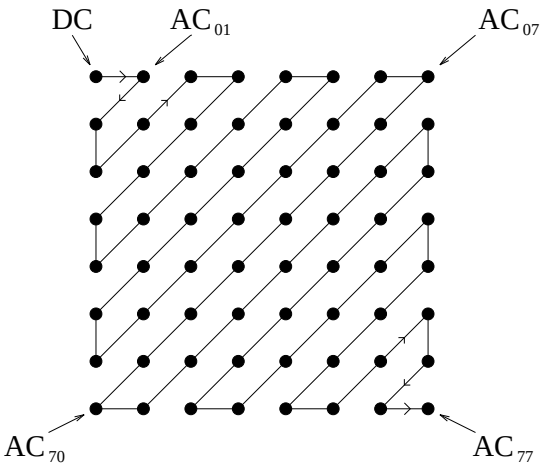


Figura 1.6: Secuencia zigzag para la preparación de los coeficientes cuantificados de cara a la codificación por entropía.

intermedia mediante codificación por recorridos (*run-lenght encoding*). El segundo paso convierte los símbolos a una secuencia de datos en la que los símbolos no tienen ya fronteras externamente identificables.

La codificación de Huffman requiere que la aplicación especifique uno o más conjuntos de tablas de códigos Huffman. Éstas pueden ser predefinidas por la aplicación o bien calculadas específicamente para cada imagen.

El método de codificación aritmética propuesto por JPEG, por el contrario, no requiere ninguna tabla externa porque es capaz de adaptarse a las características de la imagen conforme la codifica. Este tipo de codificación produce entre un 5% y un 10% más de compresión que los códigos Huffman pero es más complejo de implementar que la codificación Huffman, especialmente a nivel hardware.

1.3.2 Modo progresivo basado en DCT

JPEG definió un modo progresivo basado en DCT para satisfacer la necesidad de una decodificación rápida de la imagen cuando el decodificador estaba separado del codificador por un medio con bajo ancho de banda. Codificando parcialmente los coeficientes cuantificados de la DCT en múltiples pasadas, la calidad de la imagen decodificada crecía progresivamente de un nivel burdo hasta el nivel de calidad permitido por las matrices de cuantificación. Para codi-

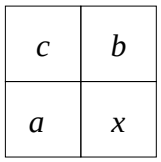


Figura 1.7: Pixels usados en la predicción, modo sin pérdidas.

ficar los coeficientes cuantificados se puede usar selección espectral, aproximación progresiva o una combinación de los dos.

En la selección espectral, los coeficientes cuantificados de un bloque la DCT se dividen en bandas no solapadas en el recorrido en zigzag. Las bandas se codifican en pasadas separadas. Antes de que se pueda codificar una banda de coeficientes AC se debe codificar el coeficiente DC del bloque. Con la aproximación sucesiva la precisión de los coeficientes se incrementa durante las diferentes pasadas. Después de una pasada con un número específico de bits más significativos de los coeficientes cuantificados, las siguientes pasadas incrementan la precisión en incrementos de un bit hasta que se codifica el último bit menos significativo.

1.3.3 Modo sin pérdidas

El modo sin pérdidas se definió para aplicaciones en las que los pixels de salida del decodificador deben ser idénticos a los pixels de entrada al codificador. Las razones de compresión que se logran con el modo sin pérdidas, típicamente 2:1, son mucho más pequeñas que aquellas que se pueden conseguir con el modo con pérdidas. Este modo no usa la DCT. En su lugar usa DPCM (*Differential pulse code modulation*) de una forma simular a la usada para los coeficientes DC en los modos basados en la DCT, excepto que el predictor se puede seleccionar entre varias opciones, como se muestra en la tabla 1.3. Las muestras a, b y c en la tabla corresponden a los vecinos del pixel x que se predice, como se muestra en la figura 1.7. Las entradas 1 a 3 en la tabla 1.3 se usan para codificación predictiva unidimensional y las 4 a 7 para predictores bidimensionales. La entrada 0 identifica codificación diferencial para el modo jerárquico. Como ocurrió anteriormente con los coeficientes DC, las diferencias entre el valor real y su predicción se codifican por entropía.

Valor	Predictor
0	Sin predicción
1	a
2	b
3	c
4	$a + b - c$
5	$a + ((b - c)/2)$
6	$b + ((a - c)/2)$
7	$(a + b)/2$

Tabla 1.3: Opciones de predictores para JPEG, modo sin pérdidas.

1.3.4 Modo jerárquico

En el modo jerárquico una imagen se codifica como un sucesión de fotogramas de resolución creciente. También conocida como codificación piramidal, esta aproximación ofrece una alternativa de alta calidad a los métodos previamente descritos para conseguir progresión. Sin embargo, es más costosa de implementar. También permite que decodificadores con diferentes capacidades de resolución usen los mismos datos comprimidos para su visualización en dispositivos con diferente resolución (monitores, impresoras, ...).

El primer fotograma codificado se crea reduciendo la resolución de la imagen de entrada por una potencia de dos en una o las dos dimensiones y procesando entonces la imagen de baja resolución mediante una de las técnicas con o sin pérdidas de los otros modos de operación. Los fotogramas siguientes se consiguen aumentando la resolución de la imagen en la(s) dimensión(es) con resolución reducida, restando la imagen aumentada de la imagen de entrada, a la misma resolución, y codificando la diferencia. Este proceso continua hasta que la imagen decodificada tiene la misma resolución que la imagen de entrada. Después se pueden codificar una o más imágenes de diferencias a máxima resolución. Un decodificador jerárquico puede abortar el proceso de decodificación después de decodificar un fotograma que tiene la resolución deseada.

Se puede usar cualquier método de codificación descrito en los otros tres modos de operación anteriores para codificar los fotogramas del modo jerárquico, con las siguientes restricciones:

- Si se escoge un modo con pérdidas, todos los fotogramas excepto el último deben co-

dificarse por ese método. El último fotograma se puede, opcionalmente, codificar sin pérdidas.

- Si se escoge un modo sin pérdidas, todos los fotogramas deben codificarse por ese método.
- La misma técnica de codificación por entropía (Huffman o aritmética) debe usarse para todos los fotogramas.

El proceso de codificación/decodificación jerárquico no es simétrico. Es más, el codificador debe incluir una gran parte del decodificador. Sin embargo, un decodificador jerárquico es más complejo que uno no jerárquico pues debe incluir métodos para ampliar y sumar los fotogramas. Este incremento en complejidad puede estar justificado dada la flexibilidad que se obtiene al ajustar el decodificador a la aplicación. Este tipo de codecs se ajusta bien a aplicaciones uno a muchos en los que un número de decodificadores (posiblemente con diferentes capacidades de resolución) accede a una base de datos de imágenes procesadas por un codificador jerárquico.

1.3.5 Compresión de imágenes en color

JPEG permite el uso de imágenes con varias componentes o bandas. Concretamente puede manejar entre 1 y 255 bandas para la misma imagen aunque lo normal es el uso de una banda, para el caso de imágenes en escala de grises, o de tres bandas diferentes en el caso de imágenes en color.

Aunque JPEG no define cual es el espacio de color en el que las imágenes deben representarse, de la misma forma que no almacena información sobre el tamaño del pixel o las características de la adquisición de la imagen, es usual transformar una imagen color de entrada a un espacio de color *adecuado*.

Es posible usar muchas representaciones o espacios de color diferentes aunque lo normal es usar un espacio con tres coordenadas de color. Este hecho viene fundamentado en la teoría tricromática que mantiene que la sensación de color en el ojo se produce excitando selectivamente tres clases de receptores oculares y, por tanto, sería suficiente una representación del color con tres componentes para representar una imagen color [99].

RGB es un ejemplo de una representación de color con tres componentes independientes para describir los colores. Esta representación asigna un valor a la cantidad de color rojo (R), verde (G) y azul (B) presente en cada pixel. Cada valor puede variar independientemente y, por tanto, tenemos un espacio tridimensional con las tres componentes, R, G y B, actuando

como coordenadas independientes. La escala de los grises se obtiene asignando el mismo valor a cada una de las tres componentes de color.

Para imágenes en color es usual transformar la imagen, que normalmente está en el espacio de color RGB, a un espacio de color de luminosidad/cromatismo como el YCbCr o YUV. En estos espacios el componente de luminosidad (Y) representa la escala de grises y los otros dos componentes (U y V o Cb y Cr) dan la información extra necesaria para convertir la imagen en escala de grises a color.

La razón de esta conversión es que se pueden permitir más pérdidas en la información de los componentes cromáticos que en el componente de luminosidad, puesto que el ojo humano no es tan sensible a las altas frecuencias cromáticas como a las altas frecuencias luminosas, permitiendo obtener una mayor compresión. Es preciso aclarar que esta transformación no es necesaria ya que el resto del algoritmo trabaja con las bandas independientemente y que conlleva una pequeña pérdida debido a los errores de redondeo aunque la magnitud de ese error es mucho menor que la que, normalmente, introduciremos después.

El espacio de color más usado suele ser el YCbCr ya que es el recogido en la recomendación 601 de CCIR [12], uno de los comités de la ITU, para la codificación de señales de televisión digital en un intento de hacer más compatibles los sistemas PAL, SECAM y NTCS. Según esta recomendación, una señal RGB analógica, con valores, E_R, E_G, E_B, en el rango $0-1$ y a los que le ha aplicado una corrección gamma de color, se transforma a una señal YCbCr en dos pasos. El primero es una conversión al espacio de color YUV, en el que se construyen las componentes de luminosidad y cromatismo de la siguiente forma:

$$\begin{aligned}
\hat{Y} &= 0.299E_R + 0.587E_G + 0.114E_B, \\
\hat{U} &= E_R - Y = 0.701E_R + 0.587E_G + 0.114E_B, \\
\hat{V} &= E_B - Y = -0.299E_R - 0.587E_G + 0.886E_B.
\end{aligned}$$

Estas componentes, así definidas, están en el rango 1.0 a 0.0 para $\hat{Y}$, $+0.701$ a -0.701 para $\hat{U}$ y en el rango $+0.886$ a -886 para $\hat{V}$. Para restaurar el rango de la señal, es usual redefinir U y V en el rango $+0.5$ a -0.5 mediante las siguientes ecuaciones:

$$\begin{aligned}
U &= 0.500E_R + 0.419E_G + 0.081E_B, \\
V &= -0.169E_R - 0.331E_G + 0.500E_B,
\end{aligned}$$

aunque también se suelen encontrar en su forma digital como enteros en el rango $[0-255]$.

Para realizar el segundo paso, en el caso de una cuantificación de color uniforme con una codificación binaria a 8 bits, se definen 256 niveles de cuantificación idénticamente espaciados. CCIR define que la señal de luminosidad Y debe ocupar sólo 220 de esos niveles y que el nivel cero es el 16. Así, el valor obtenido será

$$Y = \text{redondeo}\,(219\hat{Y} + 16).$$

Para las componentes de cromatismo, se definen 225 niveles, de forma que el cero sea el nivel 128. Por tanto, los valores de Cb y Cr se obtienen como

$$\begin{aligned} Cb &= \text{redondeo}\,(126\hat{V} + 128), \\ Cr &= \text{redondeo}\,(160\hat{U} + 128). \end{aligned}$$

Suponiendo que queramos derivar directamente las componentes YCbCr de una señal RGB digital (con la corrección gamma previamente realizada), entonces la cuantificación y transformación sería equivalente a

$$\begin{aligned} Y &= \text{redondeo}\left(\frac{77}{256}E_{R_D} + \frac{150}{256}E_{G_D} + \frac{29}{256}E_{B_D}\right), \\ Cb &= \text{redondeo}\left(-\frac{44}{256}E_{R_D} - \frac{87}{256}E_{G_D} + \frac{131}{256}E_{B_D}\right) + 128, \\ Cr &= \text{redondeo}\left(\frac{131}{256}E_{R_D} - \frac{110}{256}E_{G_D} - \frac{21}{256}E_{B_D}\right) + 128, \end{aligned}$$

donde los valores digitales de la señal RGB, E_{R_D}, E_{G_D} y E_{B_D}, han sido obtenidos a partir de la señal analógica como

$$\begin{aligned} E_{R_D} &= \text{redondeo}\,(219E_R) + 16, \\ E_{G_D} &= \text{redondeo}\,(219E_G) + 16, \\ E_{B_D} &= \text{redondeo}\,(219E_B) + 16. \end{aligned}$$

Esta misma recomendación también propone hacer un muestreo de cada componente de color tomando la media de grupos de pixels. La componente de luminosidad mantiene su resolución mientras que las componentes de cromatismo, a menudo, se reducen a la mitad horizontalmente o en ambas direcciones o, simplemente se dejan como están. Esas alternativas se suelen llamar 4:2:2, 4:1:1 y 4:4:4, respectivamente, aunque es frecuente encontrar también la nomenclatura 2h1v, 2h2v y 1h1v para cada una de ellas. Aunque en términos numéricos

esto supone una gran pérdida, en calidad visual casi no tiene impacto puesto que el ojo tiene menor resolución para la información cromática. Puesto que el muestreo no se puede aplicar a imágenes en escala de grises, las imágenes en color permiten mayor compresión que las imágenes en escala de grises.

JPEG incorpora la posibilidad de realizar este muestreo definiendo, para cada banda un coeficiente de muestreo horizontal y vertical, H_i y V_i, que especifican el número de muestras de la componente i-ésima relativa a las otras componentes de la imagen. Por tanto, si las dimensiones totales de la imagen son X y Y, las dimensiones de la i-ésima componente (el número de muestras horizontales, X_i, y verticales, Y_i) vienen dados por:

$$\begin{aligned} X_i &= \text{superior}\left(X\frac{H_i}{H_{\max}}\right), \\ Y_i &= \text{superior}\left(Y\frac{V_i}{V_{\max}}\right), \end{aligned}$$

donde $H_{\max}$ y $V_{\max}$ son el valor máximo de los coeficientes de muestreo horizontales y verticales, y "superior (θ)" indica un redondeo hacia arriba, es decir, el siguiente valor entero superior o igual a θ. Tanto los factores de muestreo como las dimensiones de la imagen se almacenan en la cabecera de la imagen.

1.3.6 Compresión y calidad de la imagen

Para imágenes en color con escenas moderadamente complejas, todos los modos de operación basados en la DCT producen los siguientes niveles de calidad de imagen en función de los niveles de compresión. Estos niveles son sólo guias ya que la calidad y la compresión pueden variar mucho dependiendo de las características de la imagen. Las unidades usadas son bits/pixel obtenidos como la media del número total de bits de la imagen comprimida dividido entre el número de pixels de la imagen. También, como referencia, se dan las razones de compresión en función de la relación entre la cantidad de datos original y la obtenida.

- 0.25 – 0.5 bits/pixel (96:1 – 48:1): calidad moderada a buena, suficiente para algunas aplicaciones;
- 0.5 – 0.75 bits/pixel (48:1 – 32:1): calidad buena a muy buena, suficiente para muchas aplicaciones;
- 0.75 – 1.5 bits/pixel (32:1 – 16:1): calidad excelente, suficiente para la mayoría de las aplicaciones;

- 1.5 – 2.0 bits/pixel (16:1 – 12:1): normalmente indistinguible de la original, suficiente para las aplicaciones que requieren máxima calidad.

1.4 Métodos de compresión de vídeo

JPEG, como todos los métodos de compresión de imágenes estáticas, aprovecha la redundancia existente en la imagen haciendo una compresión que, en técnicas de compresión de vídeo, se denomina intra-fotograma (*intraframe*). Pero además de la redundancia espacial, en las secuencias de imágenes, existe redundancia entre fotogramas consecutivos. Aprovechar este tipo de redundancia se denomina codificación inter-fotograma (*interframe*).

Puesto que los métodos basados en DCT, que se usan en los estándares H.261 y H.263 para vídeoconferencia y el estándar MPEG para uso doméstico, son los métodos más extendidos, a continuación los describiremos brevemente en relación con el método de compresión de imágenes estáticas JPEG.

1.4.1 CCITT H.261 y H.263. Estándares de codificación para vídeoconferencia

Integrated Services Digital Networks (ISDN) ofrece dos velocidades de transferencia que son aptas para la transmisión de vídeo. Se llaman canal B de 64 Kbits/seg. y el canal H0 de 384 Kbits/seg. .

Muchos países están ofreciendo la llamada *línea básica* ISDN que consiste en dos canales B. La línea básica ISDN permite transmisión de vídeo a 112 Kbits/seg. y audio a 16 Kbits/seg. H0 permite una codificación de vídeo a 320 Kbits/seg. y audio a 64 Kbits/seg. . Los algoritmos de compresión de vídeo internacionales se han establecido para acomodarse a las velocidades de transmisión de ISDN.

Desde un punto de vista algorítmico, la extensión de la codificación intra-fotograma mediante DCT de JPEG a la codificación de vídeo mediante DCT con compensación de movimiento de H.261 [33] o H.263 [32] es bastante natural. Sin embargo, históricamente H.261 se desarrolló bastante antes que JPEG.

Tanto el modo base de JPEG como H.261 usan la DCT y códigos de longitud variable. La mayor diferencia está en el modo en que manejan la información de movimiento. Mientras en JPEG las imágenes se procesan independientemente usando DCT intra-fotograma, en H.261

se usa un esquema de compensación de movimiento basado en bloques. Los pixels, aún sin codificar, de la imagen actual y los comprimidos del fotograma anterior, que se mantiene en memoria, se le pasan a un estimador de movimiento que calcula los vectores de desplazamiento de los bloques de luminosidad. Este estimador de movimiento se suele basar en emparejamiento de bloques y producen vectores de movimiento con precisión de un pixel y un rango máximo de desplazamiento de ± 15, tanto en la dirección vertical como en la horizontal. Como resultado, solo se necesita transmitir las diferencias, por lo usual pequeñas, entre el bloque anterior desplazado y el bloque actual.

Algunas características o consideraciones de diseño interesantes de H.261 son:

1. H.261 define esencialmente el decodificador aunque el codificador, que no está completa y explícitamente definido por el estándar, se espera que sea compatible con el bien definido decodificador.

2. Ya que H.261 se ha diseñado para comunicaciones en tiempo real, usa sólo el fotograma anterior para codificar la secuencia de imágenes en movimiento reduciendo, de esta forma, el retraso en la codificación.

3. Trata de equilibrar la complejidad hardware del codificador y del codificador puesto que ambos son necesarios para aplicaciones en tiempo real de vídeoconferencia. Otros esquemas, como los basados en VQ podrían haber tenido un decodificador más simple, pero un codificador mucho más complejo.

4. H.261 es un compromiso entre la eficiencia de la codificación, los requerimientos de tiempo real, la complejidad de la implementación y la robustez del sistema. La codificación mediante DCT con compensación de movimiento es un algoritmo maduro y, tras años de estudio, bastante general y robusto pudiendo manejar varios tipos de imágenes.

5. Las estructuras de codificación y los parámetros se han ajustado más hacia aplicaciones de codificación de baja razón de bits. Ésta es una elección lógica puesto que la elección de los parámetros y la estructura de codificación es más crítica a bajas razones de bits. A altas razones, unos parámetros no muy bien ajustados no afectan tanto a la eficiencia del codec.

H.263 es un estándar provisional de ITU-T. Fue diseñado para comunicaciones con un ancho de banda pequeño y, de hecho, ha reemplazando a H.261 en muchas aplicaciones.

La forma de codificación es bastante similar a la H.261 aunque con algunas mejoras y cambios para mejorar su eficiencia y la recuperación de errores. Las principales diferencias entre H.261 y H.263 se describen a continuación:

- Precisión de medio pixel en la compensación de movimiento mientras que H.261 usaba precisión de un pixel.
- Algunas partes de la estructura de los datos que se transmiten ahora son opcionales de forma que el codec puede configurarse para un menor consumo de ancho de banda o una mejor recuperación de errores.
- Ahora hay cuatro opciones negociables de codificación para mejorar la eficiencia: modo con vectores de movimiento sin restricciones, modo de predicción avanzada, modo con codificación aritmética y un sistema de predicción con el fotograma anterior y posterior, similar al que propone MPEG (ver sección 1.4.2) llamado fotogramas P-B. Cuando se usan las opciones avanzadas en H.263 a menudo se obtiene la misma calidad que en H.261 con menos de la mitad de bits. Todas estas opciones son *negociables* en el sentido de que el decodificador le indica al codificador que opciones puede manejar y, si el codificador las soporta, puede activarlas consiguiendo mayor calidad y menor razón de bits.
- H.263 soporta cinco resoluciones diferentes, mientras que H.261 sólo soportaba dos. Esto hace que el codec pueda competir con otros estándares con mayor razón de bits como los estándares MPEG.

1.4.2 ISO MPEG. Codificación para vídeo doméstico

El grupo de expertos en imágenes en movimiento (*Moving Picture Experts Group* o MPEG) fue encargado por la organización de estándares internacional (ISO) para estandarizar la representación de vídeo y su sonido asociado para el almacenamiento digital (DAT, CDROM, etc.) y transmisión en medios de comunicación (redes de telecomunicaciones, televisión por cable, etc.). Como resultado, MPEG produjo dos estándares, llamados coloquialmente MPEG1 y MPEG2.

MPEG1 [38] es un estándar internacional para la representación codificada de vídeo y su sonido asociado a velocidades sobre 1.5 Mbits/seg.. Si el vídeo se codifica a unos 1.1 Mbits/seg. y el sonido estéreo se codifica a 128 Kbits/seg. por canal, entonces la señal digital de audio

y vídeo podría tener requerir aproximadamente unos 1.4 Mbits/seg. que es la velocidad de transmisión de un CDROM (1x). Aunque el estándar se desarrolló para 1.5 Mbits/seg. esto no es un límite en absoluto pudiendo incluso llegar a velocidades de 100 Mbits/seg.. Sin embargo la calidad de la imagen durante el desarrollo del algoritmo MPEG se optimizó para 1.1M bits/seg. usando imágenes en color en modo no entrelazado (también llamado progresivo). Se propusieron dos Formatos de Entrada de Fuente (*Source Input Format* o SIF) para esta optimización. Uno correspondía al NTSC con 352 pixels, 240 lineas y 29.97 fotogramas/seg. y el otro correspondiente a PAL con 352 pixels, 288 líneas y 25 fotogramas por segundo. Al igual que H.261 y JPEG, MPEG1 usa un muestreo 2:1 de las bandas de color, tanto horizontal como verticalmente.

Para la realizar la compresión con MPEG1 hay que tener en cuenta la forma de codificación intra-fotograma y la forma inter-fotograma. Para realizar la codificación intra-fotograma, se usa una técnica como la propuesta por JPEG. A las imágenes que usan sólo codificación intra-fotograma se las llama *imágenes-I*.

Para poder explotar la correlación temporal, es decir, para realizar una codificación inter-fotograma, se definen *macrobloques*. Un macrobloque es un conjunto de cuatro bloques de luminosidad más uno de cada una de las bandas de cromatismo. Puesto que las bandas de cromatismo están muestreadas 2:1 tanto horizontal como verticalmente, el macrobloque contiene los bloques de luminosidad asociados a cada bloque de cromatismo.

Usando estimación del movimiento unidireccional, llamado *predicción hacia delante*, un bloque *destino* en el fotograma que está siendo codificado se empareja con un conjunto de bloques del mismo tamaño en un fotograma anterior llamado imagen de *referencia*. La estimación y codificación se hace de la misma forma que en H.261. Los fotogramas codificados usando predicción hacia delante se denominan *imágenes-P*.

Pero la característica clave de MPEG1 es la predicción temporal bidireccional, también llamada interpolación de compensación de movimiento. Las imágenes codificadas con predicción bidireccional usan dos fotogramas de referencia, uno en el pasado y otro en el futuro. Cada macrobloque en el fotograma actual se predice promediando el desplazamiento de dos macrobloques, uno en cada una de las imágenes de referencia. Las imágenes codificadas mediante esta forma de predicción se denominan *imágenes-B*. Las imágenes de referencia pueden ser tanto imágenes-P como imágenes-I pero no otras imágenes-B.

La predicción bidireccional presenta la ventaja de que se puede obtener mayor compresión

que usando compresión unidireccional, obteniendo la misma calidad. El inconveniente es una mayor complejidad en el codificador, que tiene que emparejar dos macrobloques para predecir uno de la imagen-B y un retraso extra en el proceso de codificación puesto que las imágenes se tienen que codificar fuera de su secuencia natural.

Entonces, una secuencia típica de fotogramas codificada tiene la forma

```
IBBPBBPBBPBBIBBPBBPBBPB...
```

Esta secuencia permite un acceso rápido hacia delante o hacia atrás decodificando sólo las imágenes-I.

MPEG2, por otra parte, está pensado para transmisión en medios con velocidades mayores a 2 Mbits/seg.. Originalmente se planteó un tercer trabajo, MPEG3, para televisión de alta definición pero acabó incorporándose a MPEG2. La principal diferencia con MPEG1 está en que MPEG2 es capaz de manejar vídeo entrelazado y su mayor número de bits lo hace útil para más aplicaciones. En esta memoria, no vamos a estudiar este método de compresión de vídeo, refiriéndose al lector a las citas [91] para una descripción básica y al documento ISO/IEC 13818 [37] para los detalles.

1.5 El problema del artificio de bloques

Puesto que, H.261 es el estándar mundial para vídeoconferencia y, dada la repercusión que esta tecnología va a tener en los próximos años, los esfuerzos destinados a mejorar la calidad de las secuencias de imágenes de este estándar no serán vanos.

Si el estudio de métodos de mejora de imágenes de H.261 es importante, no lo son menos los métodos de mejora de MPEG ya que este estándar es el que se usa para el almacenamiento de vídeo y audio en los discos compactos interactivos (CD-i) y, posiblemente, también se use para la televisión de alta definición. Esto nos lleva a pensar que los reproductores de este tipo de vídeos llevarán todos un decodificador MPEG estándar y la única diferencia entre las diferentes marcas que comercialicen los reproductores será el procesamiento posterior que realicen de la imagen para obtener una mejor calidad visual. Por tanto es interesante realizar métodos de reconstrucción que no modifiquen el estándar.

Si vemos un vídeo como una secuencia de imágenes podemos tratar de reconstruir cada una de estas imágenes mientras se muestran en la pantalla. Dado que, como hemos visto antes, cada una de esas imágenes se codifican mediante un método muy semejante al modo

base de JPEG, centraremos nuestro estudio en este método puesto que, teniendo las mismas características que los de compresión de vídeo, presenta una sencillez de manejo y conceptual mucho mayor.

A altas razones de compresión, la codificación de imágenes mediante JPEG (al igual que en MPEG y H.261), presenta una serie de artificios. El principal es el llamado artificio de bloques. Este artificio, que se manifiesta como una discontinuidad entre bloques adyacentes, es resultado directo del procesamiento independiente de los bloques sin tener en cuenta las correlaciones de los pixels entre los bloques y constituye un cuello de botella para muchas aplicaciones de comunicación visual que requieren una imágenes con cierta calidad visual a altas razones de compresión.

Para ilustrar el problema de este tipo de artificio vamos a usar un ejemplo. Si tenemos una imagen color 512×512 a 24 bits por pixel, como la imagen de "Lena" mostrada en la figura 1.8a, esta imagen contiene una cantidad de datos de 786432 bytes. Si, usando un programa de compresión JPEG estándar, comprimimos la imagen usando un coeficiente de calidad 50 (que supone usar las matrices propuestas por JPEG, ver tablas 1.1 y 1.2), la imagen obtenida (figura 1.8b) contiene 23329 bytes, que representan unos a 0.712 bits por pixel y una compresión de 34:1. Como se puede observar en la figura, esta imagen es muy semejante a la original sólo que ocupa 34 veces menos espacio.

Pero podemos seguir comprimiéndola más aún. Si usamos un factor de calidad 21, la imagen (figura 1.8c) ocupa 14374 bytes (0.439 bits por pixel y compresión de 55:1). Esta imagen ya empieza a ver mermada su calidad visual aunque aún puede ser útil para algunas aplicaciones. Pero una mayor compresión supone obtener imágenes severamente degradadas. Por ejemplo, con un factor de calidad 10 la imagen comprimida (figura 1.8d) sólo contiene 9694 bytes, a 0.296 bits por pixel y una razón de compresión mayor de 83:1 pero su calidad visual es mala. La imagen se ve como si fuese un mosaico formado con teselas cuadradas. Ese efecto es el llamado artificio de bloques. Si el factor de calidad es 5, la imagen (figura 1.8e), ocupa 7183 bytes, a 0.219 bits por pixel y una razón de compresión de 110:1. Estas dos últimas imágenes, en su estado actual, no son utilizables para prácticamente ninguna aplicación. Como se observa, la prominencia del artificio de bloques depende de la amplitud de los intervalos de cuantificación de los coeficientes de la DCT. Generalmente, el efecto no es tan visible en zonas espacialmente muy activas y en zonas muy oscuras.

La existencia del artificio de bloques que presentan las imágenes comprimidas con JPEG, o

cualquier otro método que use transformadas por bloques, para razones de compresión altas, ha dado lugar a comentarios y nuevos métodos de compresión. De hecho, algunos investigadores en compresión de imágenes usando wavelets creen que JPEG no representa el estándar actual. Por ejemplo, Hilton, *y otros* [27] comentan que "La calidad de reconstrucción de las imágenes comprimidas con wavelets ha ido más allá de las capacidades de JPEG que es, actualmente, el estándar internacional para la compresión de imágenes". Una evidencia más de este hecho es que el FBI ha adoptado un método de compresión de imágenes en escala de grises mediante wavelets para la compresión de imágenes de huellas digitales [6].

En esta memoria trataremos de reducir estos artificios mediante un proceso al que llamaremos *reconstrucción de la imagen*. Mediante este término describiremos el proceso de reducción del artificio de bloques en imágenes del tipo JPEG, una vez que la imagen ha sido ya descomprimida mediante el algoritmo estándar JPEG. Por tanto, los métodos descritos en esta memoria harán un post-procesamiento de la imagen una vez descomprimida aunque, como se estudiará más adelante, pueden hacer uso de la información recibida por el decodificador para extraer información que sea relevante al proceso de reconstrucción. Así, modelaremos el problema desde un punto de vista bayesiano y obtendremos métodos que reduzcan los artificios presentes entre los bloques de la imagen. Un ejemplo del resultado de este post-procesamiento se puede observar en la figura 1.9.

1.6 Objetivos

El objetivo de esta memoria es, obtener métodos automáticos de reconstrucción de imágenes altamente comprimidas mediante transformada coseno discreta, así como estimar los parámetros asociados a los mismos, empleando métodos bayesianos.

Este objetivo global se ha desglosado en el siguiente conjunto de objetivos:

- Obtención de métodos de reconstrucción que reduzcan el artificio de bloques mediante un post-procesamiento llevado a cabo en el decodificador. Estos métodos estimarán los parámetros asociados al proceso dentro del paradigma bayesiano jerárquico aplicado al problema de reconstrucción. Mostrar que usando este paradigma se puede realizar la reconstrucción de la imagen y la estimación de los parámetros de forma automática, sin intervención del usuario, usando técnicas robustas y bien fundamentadas.

- Realizar el proceso de estimación de los parámetros en el codificador usando la imagen

original, transmitirlos al decodificador y realizar la reconstrucción de la imagen en el mismo. De esta forma se podrá utilizar la información proveniente de la imagen original, la que tenemos en el codificador antes de comprimirla, para mejorar el comportamiento del método.

- Combinar la información sobre los parámetros obtenida en el codificador a partir de la imagen original con la información obtenida a partir de la imagen comprimida en un proceso que incorpore la información de la imagen original en la estimación de los parámetros en el decodificador y, así, obtener una reconstrucción con mayor información.

- Presentar una primera aproximación a la extensión de los resultados anteriores a la reconstrucción de imágenes en color.

1.7 Descripción por capítulos

Para llevar a cabo estos objetivos, esta memoria se divide como sigue:

En este primer capítulo hemos presentado brevemente el problema de la compresión de imágenes digitales. Hemos centrado nuestro estudio en los estándares de compresión de imágenes estáticas y en movimiento basados en la transformada coseno discreta por bloques. La necesidad de comprimir cada vez más las imágenes para que puedan usarse en un mayor número de aplicaciones hace que se reduzca la calidad visual de las mismas más allá de los límites aceptables y aparezcan artificios. En este capítulo hemos descrito el principal de estos artificios, el artificio de bloques, cuya eliminación o reducción es el objetivo básico de esta memoria.

En el capítulo segundo se hace una revisión de los diferentes métodos propuestos en la literatura para reducir el artificio de bloques en imágenes comprimidas con el estándar JPEG. Esta revisión divide los métodos en dos enfoques claramente diferenciados. El primero consiste en estimar los coeficientes de la DCT en el decodificador para que se parezcan lo más posible a los de la imagen original, consiguiendo de esta forma una reducción de los artificios y una mejora en la calidad de la imagen. El segundo enfoque consiste en filtrar la imagen en el dominio espacial para suavizar los bordes de los bloques y, de esta forma, crear una imagen más acorde con las características del sistema visual humano. La mayoría de los métodos revisados requieren la estimación de parámetros asociados a los modelos usados que, bien se estiman mediante técnicas ad hoc, o bien se dejan a la elección del usuario.

El capítulo 3 define matemáticamente el problema de la reconstrucción de imágenes comprimidas mediante JPEG y describe el paradigma bayesiano jerárquico, es decir, el modelo matemático que usaremos para representar nuestro conocimiento acerca de las imágenes y extraer las conclusiones necesarias sobre la imagen original. Este paradigma nos permitirá realizar la reconstrucción de la imagen y la estimación de los parámetros asociados usando técnicas robustas y bien fundamentadas. El paradigma bayesiano jerárquico incluye un modelo de fidelidad sobre los datos recibidos, llamado modelo de ruido, y un modelo adaptativo de imagen en el que se incluye nuestro conocimiento previo sobre la imagen original. Analizaremos diferentes alternativas para estos modelos de imagen y ruido. Dichos modelos dependerán de una serie de parámetros que habrá que estimar. Sobre dichos parámetros definiremos distribuciones de probabilidad que nos permitan incorporar conocimiento previo sobre el valor de estos parámetros en el proceso de estimación. Una vez definidos los elementos del paradigma bayesiano jerárquico estudiaremos las posibles soluciones del análisis bayesiano, escogiendo el llamado análisis basado en la evidencia para llevar a cabo la reconstrucción de la imagen y la estimación de los parámetros de forma simultánea.

Por claridad en el desarrollo del capítulo 3, algunas demostraciones sobre la forma de los modelos de imagen se presentan más detenidamente en el apéndice A. Los modelos de imagen definidos requieren el cálculo de una serie de pesos que capten la visibilidad de los artificios en cada zona de la imagen. En el apéndice B se exponen diversas formas de cálculo de los pesos.

Tras definir los elementos del paradigma bayesiano jerárquico y estudiar las diferentes soluciones al problema de la reconstrucción, en el capítulo 4 aplicaremos dicho paradigma a la estimación de los hiperparámetros y la reconstrucción simultánea de la imagen en el decodificador usando el análisis basado en la evidencia. Puesto que en el capítulo 3 definimos diferentes modelos de imagen y ruido, se propondrán diferentes métodos iterativos dependiendo de los modelos usados. Se analizará el funcionamiento de estos métodos sobre imágenes reales de diferentes tipos y se estudiará su comportamiento en función de la calidad visual de las soluciones, del pico de la relación señal-ruido y del número de iteraciones. Por último, se mostrarán una serie de perfiles sobre la imagen observada y la reconstruida que ilustren el comportamiento de los métodos dependiendo de las características locales de la imagen.

En el capítulo 5 estudiaremos como utilizar la información proveniente de la imagen original, la que tenemos en el codificador antes de comprimirla, para mejorar el comportamiento del método. Veremos cómo se pueden estimar los parámetros asociados a los modelos de imagen y

ruido en el codificador, transmitirlos, posiblemente tras un proceso de cuantificación, y cómo utilizarlos en el decodificador para obtener unos nuevos parámetros que incorporen también la información proveniente de esta imagen original. Esta combinación se hará dentro de la aproximación jerárquica bayesiana aplicada al problema de la reconstrucción de imágenes comprimidas, que nos permite, de una manera formal, incorporar este conocimiento previo sobre el valor de los parámetros proveniente del codificador cuando éstos se estiman en el decodificador. Se mostrará experimentalmente que esta combinación de información produce mejores reconstrucciones que cuando solamente se usa la información del decodificador, tanto en función del pico de la relación señal-ruido como de la calidad visual.

La mayoría de los algoritmos desarrollados para la reducción de los artificios de los bloques en la literatura tratan solamente con imágenes en escala de grises y, si se aplican a imágenes en color, sólo procesan la banda de luminosidad de la imagen pero no modifican las bandas de información cromática. En el capítulo 6 proponemos una primera aproximación a la extensión del método que mejores resultados nos ha dado en los experimentos llevados a cabo en esta memoria, para la reconstrucción de imágenes en color.

A continuación se exponen las conclusiones finales que se derivan de este trabajo, así como los futuros caminos de investigación que han surgido en el desarrollo del mismo.

La memoria concluye con la bibliografía utilizada en el trabajo de investigación que se ha llevado a cabo.

Figura 1.8: Imagen de Lena a diferentes razones de compresión. De izquierda a derecha y de arriba a abajo: imagen original, imagen comprimida a 40:1, imagen comprimida a 55:1, imagen comprimida a 83:1 e imagen comprimida a 110:1.

(a) (b)

Figura 1.9: Ejemplo de reconstrucción. (a) Imagen de Lena, comprimida a 83:1 y (b) reconstrucción con el algoritmo 6.1.

Capítulo 2

Revisión histórica

La reconstrucción de imágenes altamente comprimidas es un problema que ha sido abordado desde un gran número de puntos de vista a lo largo de los últimos veinte años. Cuando, en 1986, se propone la creación del grupo de trabajo JPEG se comienzan a proponer diferentes métodos de compresión, mucho de ellos basados en el uso de transformada coseno discreta. Los métodos que no cupieron en el estándar JPEG se han presentado y desarrollado posteriormente como una alternativa al mismo que no producen o palian el efecto de bloques propio de JPEG a altas razones de compresión aunque muchos de ellos tenían una complejidad computacional mucho mayor que JPEG o usaban técnicas que eran costosas de implementar en un sistema hardware, uno de los requerimientos que más peso tuvo en la elección del estándar de compresión de imágenes estáticas JPEG. En muchos de estos métodos, además, la calidad de la imagen a altas razones de compresión no es tan buena como la de JPEG. Este es el caso de la cuantificación de vectores, en la que el costo computacional del decodificador es muy pequeño pero la presencia de artificios, semejantes a los producidos por JPEG, se hace evidente a razones de compresión mas bajas.

Desde que el grupo JPEG saca sus conclusiones hasta hoy, la mayoría de los esfuerzos se han centrado en reconstruir las imágenes comprimidas mediante el estándar, usando un conjunto variopinto de técnicas. El hecho de que, hoy en día, existan codecs hardware que realizan la compresión y descompresión en tiempo real de imágenes y vídeos codificados mediante JPEG, MPEG, u otros estándares basados en cuantificación de los coeficientes de la DCT, hace deseable la utilización de métodos de reconstrucción que respeten el estándar, de forma que se pueda utilizar la implementación hardware del mismo, más un procesamiento adicional,

posiblemente también implementado en hardware, que nos devuelva la imagen con los artificios reducidos.

En esta línea, las investigaciones se han dirigido siguiendo dos enfoques claramente diferenciados. El primero consiste en estimar los coeficientes de la DCT en el decodificador (no olvidemos que al decodificador le llega una versión cuantificada de los coeficientes) para que se parezcan más a los que contenía la imagen original, siguiendo la propuesta del grupo JPEG en el anexo K del documento de requerimientos y directrices de JPEG [34], consiguiendo de esta forma una reducción de los artificios y una mejora en la calidad de la imagen. El segundo enfoque consiste en filtrar la imagen en el dominio espacial para suavizar los bordes de los bloques y, de esta forma, crear una imagen más acorde con las características del sistema visual humano.

Los investigadores que trabajan en el dominio de las frecuencias, tratando de estimar los coeficientes de la DCT, manifiestan que su aproximación es más rápida puesto que, normalmente, sólo se estiman los primeros pocos coeficientes de baja frecuencia no alterando el valor de los demás. De esta forma sólo se necesitan unos pocos cálculos por cada bloque de la imagen. Este procesamiento se basa en la observación de que los coeficientes de baja frecuencia contienen la mayor parte de la energía de la imagen y las variaciones en estos coeficientes, producidos por la cuantificación, se traducen en diferencias en las texturas y son responsables, en gran medida, de los escalones que se producen en las fronteras de los bloques dando lugar al efecto de bloques. La reconstrucción de estos coeficientes da lugar, no sólo a una reducción del efecto de bloques, sino que además mejora la calidad general de la imagen.

Sin embargo, la estimación de los pocos primeros coeficientes no suele resolver el problema de los bloques de una forma precisa necesitando, a veces, un procesamiento adicional, quizá en el dominio espacial. Una solución a este problema es estimar más coeficientes de la DCT pero esto lleva a que el procesamiento sea más lento y difícil de implementar.

Los trabajos en el dominio espacial, por otra parte, también se muestran como una alternativa eficaz para la reducción del efecto bloque. Algunos de los métodos propuestos no son más que filtros paso baja que permiten, a costa de un cierto emborronamiento de la imagen, la reducción del molesto efecto aunque muchos de los métodos usan técnicas que aplican un filtrado selectivo a diferentes zonas de la imagen dependiendo de las características locales de la misma. Además, trabajar sobre el dominio espacial permite modelizar de forma sencilla la restricción de suavidad presente entre los bordes de la imagen y, aunque la mayoría del

procesamiento se realice en este dominio, se puede usar la información que los coeficientes de la transformada coseno discreta aporta para discriminar zonas con diferentes características como zonas planas, con pocos detalles, etc.

A continuación haremos una revisión de los métodos propuestos para la reconstrucción de imágenes JPEG mediante estas dos aproximaciones así como de una serie de técnicas que realizan una reconstrucción en el dominio de otras transformadas, como la transformada wavelets.

2.1 Aproximaciones en el dominio de la DCT

El grupo JPEG, en el anexo K del documento de requerimientos y directrices de JPEG [34], propone una técnica para la estimación de los cinco primeros coeficientes de baja frecuencia de la DCT basada en el ajuste de una superficie cuadrática a una zona 3×3 alrededor del bloque a estimar. Si bien esta aproximación es sencilla se sabe que no funciona bien en zonas con fuertes transiciones de luz. Por tanto es conveniente estudiar otras técnicas para estimar la imagen.

Una de las formas más habituales para estimar una imagen con el artificio de bloques reducido en el dominio de las frecuencias es la estimación de los coeficientes de la DCT mediante la minimización de alguna medida de visibilidad de los bloques. Para ello, Jeong y Jeon [44] tratan de encontrar un término de corrección de los coeficientes que minimice una medida de discontinuidad cuando se añada este término a los coeficientes recibidos de la imagen. En el artículo sólo estiman el error en los tres primeros coeficientes realizando la minimización de la medida de discontinuidad por mínimos cuadrados localmente para cada uno de los bloques de la imagen.

En [131], Yang *y otros* proponen una aproximación de recuperación regularizada, basada en la teoría de mínimos cuadrados restringidos (*Constrained Least Squares* o CLS). Esta aproximación estimará todos los coeficientes de la transformada coseno discreta tratando de llegar a un compromiso entre la fidelidad a los datos recibidos y la suavidad de los mismos. Aunque el modelo se propone en el dominio espacial, la optimización se realiza iterativamente en el dominio de las frecuencia. El problema de este m´etodo es que el uso de un modelo no adaptativo a las características locales de la imagen y sólo un parámetro de regularización suaviza tanto el artificio por bloques como las zonas de altos detalles, obteniendo una imagen

un tanto borrosa. Crouse y Ramchandran [16] realizan una adaptación del método propuesto por Yang *y otros* [131] incorporando una cierta adaptatividad a las características locales de la imagen al estimar el parámetro de suavidad para cada bloque en lugar de hacerlo para la imagen completa. Un paso más en esta línea es la propuesta por Choy, Chan y Siu [14] que usan una estimación de los coeficientes por mínimos cuadrados con pesos. Los pesos se estiman localmente calculando medias y varianzas locales de cada coeficiente a partir de los coeficientes recibidos (cuantificados) permitiendo, de esta forma, la estimación de todos los coeficientes de la imagen al mismo tiempo.

En [76] se aplica una técnica de minimización mediante programación cuadrática para encontrar los M coeficientes de baja frecuencia de la DCT que minimizan la diferencia cuadrática media de la pendiente (*Mean Squared Difference of Slope* o MSDS). Básicamente, la MSDS es la energía a lo largo de la frontera de un bloque cuando se le aplica un filtro con respuesta de impulso $[1/2, -3/2, 3/2, -1/2]$. En el artículo sólo estiman los 3 primeros coeficientes y, por tanto, siguen presentes ciertos artificios de bloques. Joung, Chong y Kim [45] también estiman los 3 primeros coeficientes de la DCT que minimizan la pendiente entre dos bloques adyacentes, sujetos a los intervalos de cuantificación. Puesto que no consideran los errores de cuantificación en los componentes de alta frecuencia, el resultado es prometedor pero aún persiste el efecto bloque. Para solucionar el problema propone el uso de un filtro en el dominio de la transformada de un filtro de subbandas que acepta como entrada los coeficientes de la DCT (algunos de los cuales ya han sido ajustados) y devuelve la imagen en el dominio espacial. El filtrado de los bloques se hace simplemente modificando las imágenes de altas frecuencias alrededor de las fronteras de los bloques.

Paek y Lee [95] usan la teoría de proyección en conjuntos convexos (POCS) para suavizar las fronteras en el dominio de la DCT. El resultado es una combinación lineal convexa de los pares de coeficientes no nulos de la DCT a lo largo de una frontera de un bloque. Los parámetros que controlan ese alisamiento se eligen empíricamente. Siguiendo el mismo principio, Ahumada y Horn [1, 2, 30] estiman los coeficientes de la DCT reduciendo la varianza del borde del bloque. Esta reducción se hace en la dirección del gradiente de la varianza del borde.

Otra forma de estimar los coeficientes de la DCT es, por ejemplo, la propuesta por Jeong, Jeon y Jo en [43]. En él, la estimación de los coeficientes se hace añadiendo a los coeficientes recibidos los coeficientes estimados mediante máxima suavización de los pixels de los bordes de los bloques vecinos. Esta técnica de máxima suavización ha sido usada en [97] y [124] para

recuperar los bloques perdidos durante la transmisión en líneas ruidosas.

Una forma curiosa de reducir el artificio por bloques es la propuesta en [20, 125, 75]. La idea es introducir una distorsión en los coeficientes de la DCT. Esta distorsión (normalmente mediante *dithering*) hace que los bloques 'desaparezcan' aunque, como los autores reconocen, incrementa el error cuadrático medio y reducen el pico de la relación señal ruido. La técnica propuesta por Webb [125] incorpora esta distorsión sólo en las zonas con detalles, estimando los cuatro primeros coeficientes de cada bloque en las zonas suaves. La distinción entre zonas suaves o con detalles se hace usando la información cromática.

Silverstein y Klein [110] también proponen la introducción de una distorsión en los coeficientes de la DCT. La idea es clasificar en el codificador los bloques de la imagen en dos tipos; aquellos que se deben mejorar por un post-procesamiento y aquellos que no. Para transmitir el mapa de bloques sin alterar la cantidad de información, se propone modificar los coeficientes de cada bloque para que los bloques que necesiten ser post-procesados tengan paridad impar mientras que los que no lo necesiten tengan paridad par. De esta forma, la imagen puede ser reconstruida por un decodificador estándar, aunque con un poco más de distorsión, pero puede ser post-procesada por un decodificador más complejo que tenga en cuenta el mapa de bloques enviado.

Por último, la estimación de la matriz de cuantificación que debe usarse para codificar la imagen forma también un punto de atención entre los investigadores en reconstrucción de imágenes en el dominio de la DCT. En este campo Peterson *y otros* [100, 101] han propuesto métodos para reconstruir la imagen realizando el proceso de cuantificación mejor y proponen usar una matriz de cuantificación específica para cada imagen que se ajuste a las características de esa imagen. Recientemente, Prost *y otros* [102] propone un método que codifica la imagen usando la matriz de cuantificación estándar y, a la hora de transmitirla, ésta se sustituye por una nueva matriz de cuantificación que mantenga los detalles y la suavidad en las fronteras.

2.2 Aproximaciones en el dominio de la espacial

Respecto a las aproximaciones en el dominio espacial, las primeras propuestas datan de los 80 cuando Reeves y Lim [105], Ramamurthi y Gersho [104], Tzou [121] o Baskurt, Prost y Goutte [4] empezaron a proponer filtros y métodos de restauración de las imágenes comprimidas mediante transformadas por bloques. Pero no fue hasta principios de esta década cuando la

reconstrucción de imágenes altamente comprimidas toma su real importancia. En esta década ha aparecido un gran número de trabajos sobre recuperación de este tipo de imágenes usando una gran variedad de técnicas en el dominio espacial. Quizá los trabajos que más influencia han tenido en este proceso son los desarrollados por Sauer [108], Zakhor [134] y Yang, Galatsanos y Katsaggelos [131, 132].

Sauer [108] apunta algo que, por aquel entonces, ya empezaba a estar claro; las técnicas de filtrado espacialmente invariante clásicas son de poca ayuda al eliminar este error de reconstrucción dependiente de la señal. Poco después, Zakhor [134] presenta un algoritmo, comentado por Reeves y Eddins [106], basado en la proyección en conjuntos convexos, donde uno de los conjuntos es equivalente a la convolución con un filtro paso baja ideal y el otro representa la fidelidad a los intervalos de cuantificación de los coeficientes de la DCT. La proyección sobre este último conjunto convexo se realiza en el dominio de la DCT, teniendo que realizarse de nuevo la transformada coseno discreta directa e inversa para cada iteración. Las conclusiones obtenidas son semejantes a las apuntadas por Sauer; el filtro paso baja puede, por sií solo, eliminar el artificio de bloques pero a costa de incrementar el emborronamiento. Yang, Galatsanos y Katsaggelos [131], por su parte, también usan la teoría de POCS para resolver el problema pero, a diferencia de Zakhor, actúan solamente sobre los bordes de los bloques. El resultado obtenido de la proyección será una combinación lineal convexa de los pixels a ambos lados del bloque. Además se mantiene la fidelidad a los coeficientes de la DCT originales, al igual que hizo Zakhor. Sin embargo, el uso de unos modelos de imagen no adaptativos y la elección de un único parámetro de regularización hace que el método suavice por igual los bordes en zonas con gran nivel de detalle y en zonas suaves. Además, los parámetros usados en el modelo se estiman mediante técnicas ad hoc.

Ya teniendo en cuenta la idea de que el procesamiento debe ser adaptativo se proponen diferentes métodos para la reconstrucción.

Lai, Li y Kuo [56, 57] tratan de hacer adaptativo el modelo propuesto por Zakhor mediante el uso de varias técnicas. En primer lugar, realizan una clasificación de los bloques en función de su actividad espacial. Dicha clasificación, realizada en el dominio de la DCT, produce dos clases: suave y no suave. Usan una iteración de la técnica de filtrado no adaptativo de la imagen de Zakhor [134] más un ajuste de los coeficientes de la DCT mediante una interpolación con los de los bloques vecinos. Tras esto usan la clasificación realizada para aplicar el método de Zakhor a los bloques clasificados como suaves de forma iterativa hasta

que el algoritmo converja. Esta mezcla de técnicas, aseguran, da buenos resultados aunque su uso no está justificado y los parámetros para realizar estas operaciones deben ser introducidos por el usuario.

Stevenson [113] propone el uso de una técnica de regularización estocástica basada en la minimización de una función minimax de Huber convexa que mantiene las discontinuidades que producen las fronteras de la imagen mientras que suaviza los bordes de los bloques. Esta función es cuadrática para discontinuidades menores que un umbral, es decir, para los bordes de los bloques, y lineal para discontinuidades mayores que el umbral, como las fronteras naturales de la imagen. Puesto que la función es continua y convexa, la minimización se hace por gradiente descendente. Este trabajo se desarrolla en [92, 93] y [114] adaptándolo también a la codificación por cuantificación de vectores, mediante la aplicación, a cada paso de la minimización, de una proyección sobre un espacio de restricciones que dependerá del cuantificador o del libro de códigos escogido, según el método de compresión usado. Además, el trabajo original de Stevenson sirve como base para dos trabajos de Luo *y otros* [61, 62] en los que, partiendo de la función original de Stevenson, se crea un campo aleatorio de Huber-Markov que se optimiza por POCS [61] o por ICM [62]. En estos trabajos se asignan dos valores diferentes al umbral, uno para las fronteras de los bloques y otra para el interior de los mismos, que deben ser introducidos por el usuario. Otro uso de los campos aleatorios de Markov fue presentado por Özcelik, Brailean y Katsaggelos [94] obteniendo la solución iterativamente mediante enfriamiento del campo medio (*Mean Field Annealing* o MFA).

El trabajo presentado por Yang, Galatsanos y Katsaggelos [131], fue desarrollado posteriormete en [130] y [132] para hacerlo adaptativo. Basándose en estos métodos, Paek, Park y Lee [96] realiza un procesamiento similar al propuesto en [131] pero actualizando no sólo los pixels en las fronteras de los bloques sino también los pixels en siguiente la fila o columna más interior y, Kwak y Haddad [54], optimizan el método propuesto en [132] eliminando los pares DCT-IDCT que eran necesarios para mantener los valores de los coeficientes de la DCT dentro de sus intervalos de cuantificación.

En [71] se realiza una conversión del método propuesto en [132] al paradigma jerárquico bayesiano y realiza la estimación de los parámetros usando técnicas robustas y bien fundamentadas. Además, el método propuesto, si se conocen los parámetros, no es iterativo, a diferencia del propuesto por Yang, por lo que es adecuado para su implementación en sistemas de tiempo real. En [74] se desarrolla el trabajo anterior para estimar los parámetros necesarios

en el codificador y se propone un método para combinar esta información con la obtenida en el decodificador.

Jacovitti y Neri [41] transforman la imagen a un espacio de características visuales, obtenido mediante el filtrado con un filtro angular armónico de primer orden cuya salida compleja representa la orientación y la fuerza de las aristas. En este espacio, usa un estimador bayesiano no lineal para eliminar el artificio de bloques.

Nakajima, Hori y Kanoh [88] usa estimación de menor error cuadrático lineal (*Linear Least Square Error*) para encontrar una estimación de la imagen original con los artificios reducidos. La estimación se basa en la estimación de las medias y las varianzas locales de cada pixel mediante a partir de los datos recibidos y de los coeficientes de cuantificación.

Castagno y Villaroel [11] ajustan un spline de tercer orden a los pixels alrededor de los bordes de los bloques que maximiza la suavidad en el borde manteniendo los detalles en el interior. Si el bloque en cuestión presenta cierto nivel de detalle no se procesa en absoluto.

Otra técnica usada habitualmente consiste en la detección de las aristas de la imagen para, posteriormente, usar esta información como guia de un filtro o conjunto de ellos que dependan de las características del pixel que se estudie. Normalmente se usan métodos sofisticados para la detección de las aristas mientras que el filtrado posterior suele ser bastante sencillo. Esto es debido principalmente a que, puesto que conocemos la localización de los bloques y las aristas, podemos usar filtros diferentes para cada una de esas características. Ejemplos de esta aproximación son [53, 60, 128]. Linares *y otros* [59] usa la información proporcionada por el detector de Canny para controlar un filtro no lineal espacialmente adaptativo que reduce el artificio de bloques sin emborronar las aristas, que se realizan un filtrado direccional paralelo a éstas. Por contra, en [75] se usa un clasificador sencillo de aristas y un procesamiento más sofisticado basado en procesamiento no lineal de la imagen en el dominio espacial y una técnica de *dithering* en el dominio de las frecuencias.

Al igual que se opta por la detección de aristas con el fin de preservarlas, otros investigadores dividen la imagen en regiones con características similares, mediante clasificación de los bloques en regiones de alta o baja actividad basándose en estadísticas locales [40, 39, 117], clasificación de regiones mediante clustering [29], técnicas basadas en el histograma e información de gradientes [52], o dividir la imagen en pequeñas regiones de contenido muy similar y mezclarlas en zonas con características similares [19]. A cada una de las regiones se le aplican filtros adaptados a las características locales de la región. El problema principal de estos mé-

todos, al igual que los del grupo anterior es que los parámetros necesarios para realizar tanto la detección de las regiones o aristas como el filtrado deben ser, en la mayoría de los casos, controlados por el usuario siendo, por tanto, imposible realizar un tratamiento automático de la imagen.

Para acabar con esta revisión de los diferentes métodos propuestos en la literatura sobre reconstrucción de imágenes comprimidas, usando métodos en el dominio del espacio, queremos destacar una gran cantidad de filtros adaptativos.

Lynch, Reibman y Liu [63] usan la información de los coeficientes de la DCT para detectar los bloques con altas y bajas frecuencias y, cada uno de ellos, es procesado con diferentes filtros paso baja espacialmente variantes. En la misma línea de actuación, Sampson *y otros* [107] detectan las áreas con actividad espacial alta y baja. A las zonas con baja actividad espacial se les aplica un filtro gausiano y a las zonas con alta actividad espacial se le aplica otro filtro gausiano al que se le cambia de posición y de forma dependiendo de las características locales de la imagen. Derviaux *y otros* [18] aplican un filtro de alisamiento cuya forma y tamaño (fuerza) depende de una medida visual propuesta.

Yan [129] usa un filtro de mediana espacialmente variante para reconstruir la imagen sin emborronar las aristas presentes en la imagen. Suthaharan *y otros* [118] realiza un filtrado basado en una combinación de los ocho vecinos del pixel tratado en ese momento donde la importancia de cada vecino se determina mediante un algoritmo de interpolación de Lagrange modificado.

Estos métodos, generalmente, dan resultados aceptables si bien su fundamento es totalmente empírico y los parámetros que controlan la adaptatividad suelen ser estimados mediante métodos empíricos o, directamente, no se menciona la forma de escogerlos.

2.3 Aproximaciones basadas en otras transformadas

Además de las dos líneas de trabajo antes mencionadas, es decir, trabajar sobre el dominio de las frecuencias o trabajar sobre el dominio del espacio, hay investigadores usan otras transformadas sobre las que procesan la imagen. Hsung, Lun y Siu [31], por ejemplo, proponen una técnica basada en la representación de módulo máximo de la transformada wavelet. En esta representación se pueden caracterizar las fronteras de los bloques mediante una serie de singularidades en la representación y además permite el tratamiento simple y local de esas

singularidades. Finalmente la imagen se reconstruye usando la proyección sobre conjuntos convexos propuesta por Mallat y Zhong en [67] para la caracterización de aristas. Otro ejemplo de estas técnicas, propuesto en [127], usa la transformada de wavelets para detectar y suavizar los artificios mientras mantiene las aristas de la imagen intactas.

2.4 Otras técnicas para la reconstrucción de imágenes

Aunque los métodos de compresión que usan transformadas por bloques, como JPEG, son los más extendidos, técnicas como la cuantificación de vectores (*vector quantization* o VQ) también producen, a altas razones de compresión, el llamado artificio de bloques (véase [24] para una descripción del método de codificación y [89] para una revisión de las aplicaciones a codificación de imágenes). En la literatura se pueden encontrar algunos métodos para la reconstrucción de imágenes comprimidas mediante VQ. Todos ellos tratan de minimizar el artificio de bloques preservando características como las texturas o las aristas que existan en la imagen. Jové *y otros* [46] usa un post-procesamiento mediante un filtro de paso baja adaptativo unido a una combinación lineal con el bloque sin filtrar. Park, Kim y Lee [98] proponen un filtro paso baja similar, seguido de una proyección sobre un conjunto restringido de cuantificación de vectores. O'Rourke y Stevenson [93] usan un campo aleatorio de Huber-Markov que optimizan mediante gradiente descendente para realizar la reconstrucción de imágenes comprimidas por VQ realizando una proyección sobre un conjunto convexo tras cada iteración del algoritmo para que se sigan cumpliendo las características de la imagen original. Todos estos métodos usan parámetros que controlan el nivel de suavidad o para decidir si un pixel pertenece o no a una arista, que deben escoger el usuario final de la aplicación o se calculan a partir de los datos recibidos mediante técnicas ad hoc. Brailean *y otros* [7] usan proyecciones sobre un conjunto de restricciones sobre la cuantificación de vectores.

Por otro lado, como ya se comentó en el capítulo anterior, algunos investigadores piensan que JPEG, a pesar de ser un estándar ISO, no cumple con los criterios de calidad necesarios cuando se usan razones de compresión muy altas y proponen otras técnicas de codificación no acordes con el estándar JPEG, algunas también basadas en transformada coseno discreta, que reducen el molesto artificio.

Una de las técnicas más usadas es la llamada transformada por bloques solapados (*overlapping block transform* o OBT) en la que los bloques en que se divide la imagen tienen solapadas

unas filas o columnas con cada uno de los bloques a su alrededor, ver [109] y las referencias que en él se citan para más detalles. Así, al realizar la descompresión de la imagen, el artificio de bloques queda paliado puesto que se tiene en cuenta la correlación entre los diferentes bloques de la imagen. Sin embargo el decodificador se hace bastante más complejo y lento. Otro ejemplo de este tipo de técnicas es la llamada transformada ortogonal solapada (*Lapped Orthogonal Transform* o LOT) propuesta por Malvar y Staelin [68]. Véase también [120] y las referencias en él para más información sobre este tema.

Otras técnicas se basan en otras transformaciones, como transformación por bloques en malla (*mesh block transform*) [17]. En esta aproximación, una imagen $MN \times MN$ se divide en M^2 mallas de dimensión $N \times N$. Los pixels adyacentes en la malla están separados por N pixels en la imagen. Cada malla se transforma y los mismos coeficientes de transformación de cada malla se agrupan en bloques. Entonces se aplica una codificación semejante a la propuesta por JPEG a ese conjunto de bloques.

El uso de transformadas de bloques pequeños hace que la computación de la DCT sea eficiente pero introduce el efecto de bloques. Chan y Siu [13] proponen un algoritmo para simular la codificación mediante una transformada coseno discreta de gran tamaño a partir de transformaciones de pequeño tamaño haciendo uso de una técnica de codificación de subbandas (*subband coding*).

A pesar de que estas técnicas dan resultados aceptables, la muchos trabajos realizados modifican ligeramente el estándar JPEG para conseguir un método que sea más acorde con las características del sistema visual humano. En esta línea, Tan, Pang y Ngan [119], Ho y Kim [28] o Sultan y Latchman [116] proponen métodos de reconstrucción muy semejantes a JPEG aunque no siguen el estándar. Todos estos métodos dividen la imagen en zonas de diferente actividad (planas, con aristas, con texturas,...) o zonas en las que se centra o no la atención del espectador y cada una de ellas de codifica de acuerdo a las características propias, mediante un método como JPEG que use diferentes matrices de cuantificación para cada tipo de zona concreta o aplique un coeficiente de calidad diferente a la matriz de cuantificación estándar dependiendo de la zona en cuestión.

Capítulo 3

Elementos de la reconstrucción bayesiana de imágenes comprimidas

3.1 Introducción

Habiendo descrito en el capítulo 1 el problema de la reconstrucción de imágenes comprimidas y, en particular, la reducción de artificios por bloques (ver sección 1.5) y una vez analizados en el capítulo 2 las diferentes aproximaciones a la reducción de dichos artificios, en este capítulo estudiaremos los elementos necesarios para realizar la reconstrucción de las imágenes comprimidas mediante JPEG desde el punto de vista bayesiano jerárquico. Este paradigma nos permitirá realizar la reconstrucción de la imagen y la estimación de los parámetros asociados usando técnicas robustas y claramente establecidas.

El paradigma bayesiano jerárquico incluye un modelo de fidelidad sobre los datos recibidos, llamado modelo de ruido, y un modelo adaptativo de imagen en el que se incluye nuestro conocimiento previo sobre la imagen original. Definiremos diferentes posibilidades para estos modelos de imagen y ruido. Estos modelos dependerán de una serie de parámetros que hay que estimar. Definiremos modelos sobre estos parámetros que nos permitan realizar el análisis bayesiano tanto en situaciones en las que no se tenga conocimiento sobre el posible valor de los parámetros como en situaciones en las que si se tiene esta información, incorporándola al proceso de estimación. Una vez definidos los elementos del paradigma bayesiano jerárquico estudiaremos las posibles soluciones del análisis bayesiano para llevar a cabo la reconstrucción de la imagen y la estimación de los parámetros simultáneamente.

El resto del capítulo se organiza como sigue. En la sección 3.2 introduciremos la notación que se requiere. La sección 3.3 describe el modelo bayesiano jerárquico aplicado al problema de la reconstrucción. En las siguientes secciones estudiaremos los elementos que son necesarios en este paradigma. Así, la sección 3.4 describe los modelos de ruido, la sección 3.5 describe los modelos de imagen y la sección 3.6 describe los modelos que usaremos sobre los parámetros. Por último, en la sección 3.7 se describen las soluciones al problema de la reconstrucción mediante la realización del análisis bayesiano.

3.2 Definición matemática del problema de reconstrucción

La transformada coseno discreta por bloques (BDCT) para una imagen $M \times N$ puede verse como una transformación lineal de $R^{M\times N}$ en $R^{M\times N}$. Entonces para una imagen $\mathbf{f}$ podemos escribir

$$\mathbf{F} = \mathbf{B}\mathbf{f},$$

donde $\mathbf{F}$ es la BDCT de $\mathbf{f}$ y $\mathbf{B}$ es la matriz de la BDCT. La matriz $\mathbf{B}$ para una imagen $M \times N$ que se divide en bloques $k \times k$ es una matriz $(M \times N) \times (M \times N)$ diagonal por bloques con $\left(\frac{M\times N}{k}\right)^2$ matrices de tamaño $k^2 \times k^2$ en la diagonal. Estas matrices son idénticas y su expresión explícita es bien conocida (ver [42], por ejemplo). Además la matriz $\mathbf{B}$ es una matriz real y, debido a la propiedad unitaria de las matrices de la DCT, la matriz de la BDCT también es unitaria y su transformada inversa puede expresarse como su traspuesta, es decir,

$$\mathbf{B} = \mathbf{B}^* \qquad \text{y} \qquad \mathbf{B}^{-1} = \mathbf{B}^t,$$

donde el superíndice $*$ y el superíndice t indican el conjugado y la traspuesta de una matriz, respectivamente. Por tanto, la transformada inversa de una imagen $\mathbf{f}$ puede expresarse como

$$\mathbf{f} = \mathbf{B}^t\mathbf{F}.$$

Para conseguir una determinada reducción del número de bits para la transmisión, cada uno de los coeficientes de $\mathbf{F}$ se cuantifica. El operador de cuantificación se puede describir matemáticamente por una aplicación de $R^{M\times N}$ en $R^{M\times N}$ o un operador que hace corresponder a un conjunto de valores de entrada el mismo valor de salida. Sea $\mathcal{Q}$ este operador; la relación entrada/salida del codificador puede modelarse por

$$\mathbf{G} = \mathcal{Q}\mathbf{B}\mathbf{f}.$$

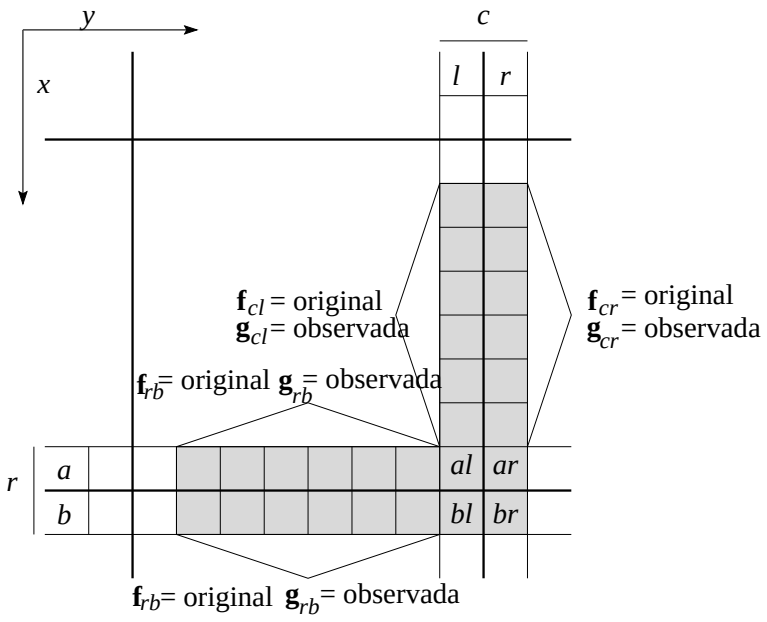


Figura 3.1: Distribución de los pixels en las fronteras de los bloques.

La aplicación definida por la cuantificación es *muchos a uno* siendo, por tanto, irreversible.

Además de la cuantificación, los coeficientes de $\mathbf{G}$ se codifican usando técnicas sin pérdidas, se transmiten y, en el receptor, se decodifican. En el receptor, por tanto, sólo están disponibles los coeficientes de la BDCT cuantificados y la salida de un decodificador convencional es

$$\mathbf{g} = \mathbf{B}^t \mathcal{Q}^{-1} \mathbf{G}.$$

El problema de la reconstrucción es encontrar un estimador de $\mathbf{f}$ dadas la imagen observada en el decodificador, $\mathbf{g}$, el cuantificador usado, $\mathcal{Q}$, y, posiblemente, algún conocimiento adicional sobre $\mathbf{f}$. En esta memoria usaremos bloques de tamaño $k \times k$ para codificar una imagen, $\mathbf{f}$, de tamaño $M \times N$, M y N múltiplos de k, usando transformada coseno discreta por bloques para obtener en el decodificador la imagen $\mathbf{g}$ y propondremos métodos para reconstruir los bordes de los bloques artificiales, donde el artificio es más visible, dejando el resto de la imagen intacta.

Puesto que la eliminación de los artificios de los bloques sólo se va a realizar en los pixels frontera de los bloques, introduzcamos la notación necesaria para caracterizar a esos pixels en la imagen.

Sea $\mathbf{f}_{cl}$ un vector columna definido por el apilamiento de todos los elementos de $\mathbf{f}$ que están a la izquierda de una frontera vertical de bloque, cl, pero no en la intersección de cuatro

bloques (ver figura 3.1), es decir,

$$\begin{aligned}\mathbf{f}_{cl} &= \{\mathbf{f}(u) \mid u=(x,y) \text{ con } x = k*i+l,\ y = k*j,\ i = 0,1,2,\ldots,M/k-1,\\ &\quad j = 1,2,\ldots,N/k-1,\ l = 2,3,\ldots,k-1\}.\end{aligned}$$

Análogamente definimos $\mathbf{f}_{cr}$, apilando todos los elementos de $\mathbf{f}$ que están a la derecha de una frontera vertical de bloque, cr, pero no en la intersección de cuatro bloques (ver figura 3.1). Usando los mismos procedimientos podemos obtener $\mathbf{g}_{cl}$ y $\mathbf{g}_{cr}$ a partir de la imagen degradada, $\mathbf{g}$.

También definimos los vectores

$$\mathbf{f}_c^t = \{(\mathbf{f}_{cl}(i)\ \mathbf{f}_{cr}(i),\ i = 1,\ldots,p)\}, \qquad \mathbf{g}_c^t = \{(\mathbf{g}_{cl}(i)\ \mathbf{g}_{cr}(i),\ i = 1,\ldots,p)\},$$

con

$$p = (k-2)(M/k) \times (N/k-1). \tag{3.1}$$

Nótese que $\mathbf{f}_{cl}$, $\mathbf{f}_{cr}$, $\mathbf{g}_{cl}$ y $\mathbf{g}_{cr}$ son vectores $p \times 1$.

De forma similar podemos definir $\mathbf{f}_{ra}$, $\mathbf{g}_{ra}$, $\mathbf{f}_{rb}$ y $\mathbf{g}_{rb}$, los vectores columna que representan las filas de encima y debajo de una frontera horizontal de bloque de $\mathbf{f}$ y $\mathbf{g}$, respectivamente (ver figura 3.1). Relacionados con éstos están los vectores

$$\mathbf{f}_r^t = \{(\mathbf{f}_{ra}(i)\ \mathbf{f}_{rb}(i),\ i = 1,\ldots,q)\}, \qquad \mathbf{g}_r^t = \{(\mathbf{g}_{ra}(i)\ \mathbf{g}_{rb}(i),\ i = 1,\ldots,q)\},$$

con

$$q = (k-2)(N/k) \times (M/k-1). \tag{3.2}$$

Además apilamos los elementos de $\mathbf{f}$ que están sobre una frontera horizontal y a la izquierda de una frontera vertical en una intersección de cuatro bloques, indicada por al en la figura 3.1, en el vector $\mathbf{f}_{al}$ definido como

$$\mathbf{f}_{al} = \{\mathbf{f}(u) \mid u = (x,y)\},$$

con $x = k*i$, $y = k*j$, $i = 1,2,\ldots,M/k-1$, $j = 1,2,\ldots,N/k-1$.

De forma semejante formamos los vectores $\mathbf{f}_{ar}$, formado por los elementos de $\mathbf{f}$ situados sobre una frontera horizontal y a la derecha de una frontera vertical, $\mathbf{f}_{br}$, bajo una frontera horizontal y a la derecha de una frontera vertical y, $\mathbf{f}_{bl}$, bajo una frontera horizontal y a la izquierda de una frontera vertical en una intersección de cuatro bloques, como se muestra en la figura 3.1. Siguiendo el mismo procedimiento, definimos los vectores de la observación para

esos pixels en la intersección de cuatro bloques, $\mathbf{g}_{al}$, $\mathbf{g}_{ar}$, $\mathbf{g}_{br}$ y $\mathbf{g}_{bl}$. Usando los vectores antes definidos, también definimos

$$\mathbf{f}_x^t = \{(\mathbf{f}_{al}(i)\ \mathbf{f}_{ar}(i)\ \mathbf{f}_{br}(i)\ \mathbf{f}_{bl}(i),\ i = 1, \ldots, m)\},$$
$$\mathbf{g}_x^t = \{(\mathbf{g}_{al}(i)\ \mathbf{g}_{ar}(i)\ \mathbf{g}_{br}(i)\ \mathbf{g}_{bl}(i),\ i = 1, \ldots, m)\},$$

con

$$m = (M/k - 1) \times (N/k - 1). \tag{3.3}$$

Cuando sea necesario, usaremos

$$\mathbf{f}^t = (\mathbf{f}_c^t\ \mathbf{f}_r^t\ \mathbf{f}_x^t) \quad \text{y} \quad \mathbf{g}^t = (\mathbf{g}_r^t\ \mathbf{g}_c^t\ \mathbf{g}_x^t).$$

Véase que originalmente $\mathbf{f}$ contenía la imagen original pero como los métodos que proponemos sólo modifican los pixels en las fronteras de los bloque podemos usar esta redefinición de $\mathbf{f}$.

Una vez definidos los elementos que necesitamos para realizar la reconstrucción y la notación que usaremos, veamos el paradigma bayesiano jerárquico que será el modelo matemático que usaremos para representar nuestro conocimiento sobre las imágenes y extraer las conclusiones necesarias sobre la imagen original.

3.3 El paradigma bayesiano jerárquico

El paradigma bayesiano jerárquico se usa actualmente en muchas áreas relacionadas con el análisis de imágenes. Buntine [8] aplica esta teoría a la construcción de árboles de clasificación. Spiegelhalter y Lauritzen [112] han aplicado el marco bayesiano jerárquico al problema de refinar redes de probabilidad. Buntine [9] y Copper y Herkovsits [15] usaron el mismo marco para tratar el problema de la construcción de esas redes. MacKay [65] y Buntine y Weigund [10] usan el marco bayesiano completo en redes neuronales con propagación hacia atrás. Este paradigma también se ha aplicado a problemas de interpolación (MacKay [64] o Gull [25]) y a problemas de restauración (Molina [77] o Molina *y otros* [79]), incluso cuando el emborronamiento presente en la imagen sólo era parcialmente conocido [23].

Este paradigma trata de modelizar un problema basándose en una aproximación estadística y está relacionado con la teoría de la decisión en presencia de conocimiento estadístico que puede arrojar luz sobre algunas incertidumbres involucradas en los problemas de decisión. La estadística clásica se dirige hacia el uso de la información proveniente de los datos obtenidos

de investigación estadística para hacer inferencias sobre datos desconocidos. La teoría de la decisión, por otra parte, intenta combinar la información de los datos con otros aspectos relevantes del problema para tomar decisiones mejores. Otro punto de vista del mismo problema, desde la teoría de la regularización, se puede observar en [47, 51, 79, 48]. Ver también [50] para otros enfoques.

Aplicado a la reconstrucción de imágenes comprimidas, el paradigma bayesiano jerárquico combina la información proveniente de los datos de la imagen recibida en el decodificador, expresada por $\mathrm{p}(\mathbf{g} \mid \mathbf{f}, \beta)$, con la información a priori estructural, proveniente de nuestra experiencia o conocimiento previo sobre el comportamiento estructural de la reconstrucción, en lo que se llama la distribución a posteriori de $\mathbf{f}$ dada $\mathbf{g}$, a partir de la cual se toman todas las decisiones y se hacen todas las inferencias.

Para el problema concreto de la reconstrucción de imágenes comprimidas tenemos, como información a priori estructural sobre $\mathbf{f}$, el conocimiento de que la imagen original, la que teníamos en el codificador antes de comprimirla, era suave en las zonas donde la imagen obtenida en el decodificador, $\mathbf{g}$, presenta el efecto de bloques. Esta información es la que representamos mediante la distribución de probabilidad $\mathrm{p}(\mathbf{f} \mid \alpha)$.

El paradigma bayesiano jerárquico aplicado a la reconstrucción de imágenes, tiene, al menos dos fases. En la primera fase, el conocimiento sobre la forma estructural del ruido y sobre el comportamiento estructural de la imagen reconstruida se usa para formar $\mathrm{p}(\mathbf{g} \mid \mathbf{f}, \beta)$ y $\mathrm{p}(\mathbf{f} \mid \alpha)$, respectivamente. Esos modelos de imagen y ruido suelen depender de unos parámetros o vectores de parámetros desconocidos α y β. En la segunda fase, el paradigma bayesiano jerárquico pone una distribución a priori sobre los parámetros desconocidos, $\mathrm{p}(\alpha)$ y $\mathrm{p}(\beta)$, donde se incluye nuestro conocimiento subjetivo sobre el posible valor de esos parámetros. Cuando se usa este paradigma, a los parámetros desconocidos se les suele denominar *hiperparámetros*. Es necesario notar que no existe ninguna razón teórica para limitar las distribuciones a priori jerárquicas a dos fases solamente, pero, en la práctica, raramente es útil usar más de dos [5].

Aunque a veces es posible conocer, por experiencias previas, relaciones entre los hiperparámetros, el modelo que estudiaremos en esta memoria usa una probabilidad global definida como

$$\mathrm{p}(\alpha, \beta, \mathbf{f}, \mathbf{g}) = \mathrm{p}(\alpha)\mathrm{p}(\beta)\mathrm{p}(\mathbf{f} \mid \alpha)\mathrm{p}(\mathbf{g} \mid \mathbf{f}, \beta).$$

Estudiemos ahora los elementos de cada una de las dos fases que hemos definido, es decir, los modelos de ruido e imagen y las distribuciones sobre los hiperparámetros.

3.4 Modelos de ruido

El primer elemento que debemos definir es el modelo de ruido que presenta la imagen.

El principal causante del ruido en la compresión de imágenes basada en transformada coseno discreta es el proceso de cuantificación. La cuantificación produce una diferencia en la estructura (textura) de cada uno de los bloques de la imagen comprimida así como en su media, obteniendo bloques con valores sensiblemente diferentes a los que presentaba la imagen original. Por tanto, el modelo de ruido debe tener en cuenta esta variación en los valores.

Por otra parte, el modelo de ruido debe mantener la fidelidad a los datos recibidos pues son el único punto de referencia que tenemos de los datos contenidos en la imagen original.

Siguiendo estos principios, el modelo de ruido que vamos a usar se define como

$$\begin{aligned} \mathrm{p}(\mathbf{g} \mid \mathbf{f}, \beta) &= \mathrm{p}(\mathbf{g}_c, \mathbf{g}_r, \mathbf{g}_x \mid \mathbf{f}_c, \mathbf{f}_r, \mathbf{f}_x, \beta) = \mathrm{p}(\mathbf{g}_c \mid \mathbf{f}_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \mathbf{f}_r, \beta)\mathrm{p}(\mathbf{g}_x \mid \mathbf{f}_x, \beta) \\ &\propto \beta^p \exp\left\{-N_c(\mathbf{g}_c \mid \mathbf{f}_c, \beta)\right\} \beta^q \exp\left\{-N_r(\mathbf{g}_r \mid \mathbf{f}_r, \beta)\right\} \\ &\quad \beta^{2m} \exp\left\{-N_x(\mathbf{g}_x \mid \mathbf{f}_x, \beta)\right\} \\ &= \beta^{p+q+2m} \exp\left\{-N(\mathbf{g} \mid \mathbf{f}, \beta)\right\}, \end{aligned} \tag{3.4}$$

donde

$$N(\mathbf{g} \mid \mathbf{f}, \beta) = N_c(\mathbf{g}_c \mid \mathbf{f}_c, \beta) + N_r(\mathbf{g}_r \mid \mathbf{f}_r, \beta) + N_x(\mathbf{g}_x \mid \mathbf{f}_x, \beta), \tag{3.5}$$

con

$$N_c(\mathbf{g}_c \mid \mathbf{f}_c, \beta) = \frac{1}{2}\beta \left\{ \| \mathbf{g}_{cl} - \mathbf{f}_{cl} \|^2 + \| \mathbf{g}_{cr} - \mathbf{f}_{cr} \|^2 \right\}, \tag{3.6}$$

$$N_r(\mathbf{g}_r \mid \mathbf{f}_r, \beta) = \frac{1}{2}\beta \left\{ \| \mathbf{g}_{ra} - \mathbf{f}_{ra} \|^2 + \| \mathbf{g}_{rb} - \mathbf{f}_{rb} \|^2 \right\}, \tag{3.7}$$

$$\begin{aligned} N_x(\mathbf{g}_x \mid \mathbf{f}_x, \beta) &= \frac{1}{2}\beta \left\{ \| \mathbf{g}_{al} - \mathbf{f}_{al} \|^2 + \| \mathbf{g}_{ar} - \mathbf{f}_{ar} \|^2 \right. \\ &+ \left. \| \mathbf{g}_{br} - \mathbf{f}_{br} \|^2 + \| \mathbf{g}_{bl} - \mathbf{f}_{bl} \|^2 \right\}, \end{aligned} \tag{3.8}$$

donde β se define como $\beta^{-1} = \sigma^2_{ruido}$ (la varianza del ruido de la imagen).

Existe la posibilidad de usar dos parámetros diferentes para controlar el ruido presente entre los bordes de los bloques de las columnas y filas, respectivamente. Este modelo de ruido se define como

$$\begin{aligned} \mathrm{p}(\mathbf{g} \mid \mathbf{f}, \beta_c, \beta_r) &= \mathrm{p}(\mathbf{g}_c, \mathbf{g}_r, \mathbf{g}_x \mid \mathbf{f}_c, \mathbf{f}_r, \mathbf{f}_x, \beta_c, \beta_r) \\ &= \mathrm{p}(\mathbf{g}_c \mid \mathbf{f}_c, \beta_c)\mathrm{p}(\mathbf{g}_r \mid \mathbf{f}_r, \beta_r)\mathrm{p}(\mathbf{g}_x \mid \mathbf{f}_x, \beta_c, \beta_r) \end{aligned}$$

$$
\begin{aligned}
&= \beta_c^p \exp\left\{-N_c(\mathbf{g}_c \mid \mathbf{f}_c, \beta_c)\right\} \beta_r^q \exp\left\{-N_r(\mathbf{g}_r \mid \mathbf{f}_r, \beta_r)\right\} \\
&\quad \beta_c^m \beta_r^m \exp\left\{-N_x(\mathbf{g}_x \mid \mathbf{f}_x, \beta_c, \beta_r)\right\} \\
&= \beta_c^{p+m} \beta_r^{q+m} \exp\left\{-N(\mathbf{g} \mid \mathbf{f}, \beta_c, \beta_r)\right\}, \qquad (3.9)
\end{aligned}
$$

con

$$
N(\mathbf{g} \mid \mathbf{f}, \beta_c, \beta_r) = N_c(\mathbf{g}_c \mid \mathbf{f}_c, \beta_c) + N_r(\mathbf{g}_r \mid \mathbf{f}_r, \beta_r) + N_x(\mathbf{g}_x \mid \mathbf{f}_x, \beta_c, \beta_r), \qquad (3.10)
$$

y

$$
\begin{aligned}
N_c(\mathbf{g}_c \mid \mathbf{f}_c, \beta_c) &= \frac{1}{2}\beta_c \left\{ \| \mathbf{g}_{cl} - \mathbf{f}_{cl} \|^2 + \| \mathbf{g}_{cr} - \mathbf{f}_{cr} \|^2 \right\}, \qquad (3.11) \\
N_r(\mathbf{g}_r \mid \mathbf{f}_r, \beta_r) &= \frac{1}{2}\beta_r \left\{ \| \mathbf{g}_{ra} - \mathbf{f}_{ra} \|^2 + \| \mathbf{g}_{rb} - \mathbf{f}_{rb} \|^2 \right\}, \qquad (3.12) \\
N_x(\mathbf{g}_x \mid \mathbf{f}_x, \beta_c, \beta_r) &= \frac{1}{2} \Big\{ \beta_c \| \mathbf{g}_{al} - \mathbf{f}_{al} \|^2 + \beta_r \| \mathbf{g}_{ar} - \mathbf{f}_{ar} \|^2 \\
&+ \beta_c \| \mathbf{g}_{br} - \mathbf{f}_{br} \|^2 + \beta_r \| \mathbf{g}_{bl} - \mathbf{f}_{bl} \|^2 \Big\}, \qquad (3.13)
\end{aligned}
$$

con β_c y β_r definidas como $\beta_c^{-1} = \sigma^2_{ruido\,c}$ (la varianza del ruido en la dirección vertical) y $\beta_r^{-1} = \sigma^2_{ruido\,r}$ (la varianza del ruido en la dirección horizontal), respectivamente.

A pesar de que este modelo pudiera parecer mejor para la modelización del ruido puesto que, al tener dos parámetros diferentes, permite un mejor control del proceso, el uso de un modelo de ruido con dos parámetros diferentes no está justificado desde el punto de vista teórico puesto que la degradación sufrida por todos los bloques es la misma ya que está basada en la misma matriz de cuantificación. Esto nos lleva a pensar que no hay ninguna razón para creer que la degradación sea diferente dependiendo del tipo de frontera (vertical u horizontal).

3.5 Modelos de imagen

Otro elemento hemos de definir es la distribución a priori o modelo de imagen en el que modelizamos nuestro conocimiento sobre la estructura de la imagen original. Nuestro conocimiento a priori nos indica que la imagen original $\mathbf{f}$ era suave en las fronteras de los bloques, es decir, no tenía, en general, saltos bruscos de intensidad en las mismas. Para incorporar este conocimiento, definimos la distribución de probabilidad sobre las fronteras verticales de los bloques de la imagen que no están en la intersección de cuatro bloques como

$$
\mathrm{p}(\mathbf{f}_c \mid \alpha) \propto \alpha^{\frac{p}{2}} \exp\left[-P_c(\mathbf{f}_c \mid \alpha)\right], \qquad (3.14)
$$

con

$$P_c(\mathbf{f}_c \mid \alpha) = \alpha \parallel \mathbf{f}_{cl} - \mathbf{f}_{cr} \parallel^2, \tag{3.15}$$

donde p se definió en la Ec. (3.1) y α, la inversa de la varianza, es una medida de suavidad de los pixels en las fronteras de los bloques de la imagen. Es importante ver que, dado que tenemos una distribución normal singular, usamos $\alpha^{\frac{p}{2}}$ en lugar de α^p (ver el apéndice A para más detalles).

De la misma forma, para las fronteras horizontales de los bloques de la imagen que no están en la intersección de cuatro bloques, definimos su distribución de probabilidad a priori como

$$\mathrm{p}(\mathbf{f}_r \mid \alpha) \propto \alpha^{\frac{q}{2}} \exp\left[-P_r(\mathbf{f}_r \mid \alpha)\right],$$

con

$$P_r(\mathbf{f}_r \mid \alpha) = \alpha \parallel \mathbf{f}_{ra} - \mathbf{f}_{rb} \parallel^2, \tag{3.16}$$

donde q se definió en la Ec. (3.2). Nótese que de nuevo tenemos una distribución normal singular y usamos $\alpha^{\frac{q}{2}}$ en lugar de α^q (ver apéndice A).

Finalmente, para los pixels en la intersección de cuatro bloques, tenemos

$$\mathrm{p}(\mathbf{f}_x \mid \alpha) \propto \alpha^{\frac{3m}{2}} \exp\left[-P_x(\mathbf{f}_x \mid \alpha)\right],$$

con

$$\begin{aligned} P_x(\mathbf{f}_x \mid \alpha) &= \frac{1}{2}\alpha \Big\{ \parallel \mathbf{f}_{al} - \mathbf{f}_{ar} \parallel^2 + \parallel \mathbf{f}_{ar} - \mathbf{f}_{br} \parallel^2 \\ &+ \parallel \mathbf{f}_{br} - \mathbf{f}_{bl} \parallel^2 + \parallel \mathbf{f}_{bl} - \mathbf{f}_{al} \parallel^2 \Big\}, \end{aligned} \tag{3.17}$$

donde m se definió en la Ec. (3.3). Véase que usamos $\alpha^{\frac{3m}{2}}$ en lugar de α^{2m} puesto que esta distribución también es singular. De nuevo se remite al lector al apéndice A para los detalles.

Así, el modelo a priori que definimos sobre la imagen original es

$$\begin{aligned} \mathrm{p}(\mathbf{f} \mid \alpha) &= \mathrm{p}(\mathbf{f}_c, \mathbf{f}_r, \mathbf{f}_x \mid \alpha) = \mathrm{p}(\mathbf{f}_c \mid \alpha)\mathrm{p}(\mathbf{f}_r \mid \alpha)\mathrm{p}(\mathbf{f}_x \mid \alpha) \\ &\propto \alpha^{\frac{p+q+3m}{2}} \exp\left\{-P(\mathbf{f} \mid \alpha)\right\}, \end{aligned} \tag{3.18}$$

con

$$P(\mathbf{f} \mid \alpha) = P_c(\mathbf{f}_c \mid \alpha) + P_r(\mathbf{f}_r \mid \alpha) + P_x(\mathbf{f}_x \mid \alpha). \tag{3.19}$$

Si bien mediante una modelización del problema de esta forma se puede obtener una imagen donde los artificios de bloque quedan reducidos, como se puede observar en [131] para una modelización similar, también presenta el importante problema de no ser adaptativo.

El problema de la adaptatividad ya se menciona en las primeras publicaciones sobre el problema de la reconstrucción de imágenes altamente comprimidas [108, 131, 134]. Como consecuencia de este problema, el proceso de reconstrucción produce un alisamiento excesivo de los bordes de regiones con alta actividad espacial, es decir, regiones con elementos que presentan cierto nivel de detalle, dando la impresión de una zona mucho más plana, perdiendo mucha información visual en esas zonas que quedan borrosas. Por otra parte, en las zonas con baja actividad espacial, es decir, zonas lisas o con poco detalle, no suaviza lo suficiente, siendo aún visibles los artificios.

Una solución es definir un modelo de imagen adaptativo, en el sentido que sea capaz de captar las propiedades locales de la imagen para obtener una imagen que sea más grata al sistema visual humano.

Para captar las propiedades locales de la imagen definimos una serie de matrices diagonales de pesos que ponderarán cada uno de los pixels en una frontera de bloque en función de las propiedades de la imagen en esa zona. Así, para captar las propiedades locales verticales de la imagen, definiremos la matriz diagonal, $\mathbf{W}_c$, de dimensión $p \times p$, como [132]

$$\mathbf{W}_c = \begin{bmatrix} \omega_c(1) & 0 & 0 & \cdots & 0 \\ 0 & \omega_c(2) & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \omega_c(p) \end{bmatrix},$$

donde los $\omega_c(i)$, $i = 1, 2, \ldots, p$, ponderarán cada diferencia entre los vectores que representan las columnas como se muestra en la figura 3.2. Nótese que los pixels en una intersección de cuatro bloques no se incluyen en esta matriz.

De forma análoga podemos definir $\mathbf{W}_r$ para captar las propiedades locales horizontales de la imagen, como se muestra en la figura 3.2, sin incluir los pixels en la intersección de cuatro bloques. La dimensión de esta matriz es $q \times q$.

Para los pixels en una intersección de cuatro bloques, vemos que tenemos que tener en cuenta las diferencias $\mathbf{f}_{al} - \mathbf{f}_{ar}$, $\mathbf{f}_{ar} - \mathbf{f}_{br}$, $\mathbf{f}_{br} - \mathbf{f}_{bl}$ y $\mathbf{f}_{bl} - \mathbf{f}_{al}$ y estas diferencias corresponden a diferencias entre filas, columnas, filas y columnas, respectivamente (ver figura 3.2). Para ponderarlas según la actividad de la región usamos los correspondientes pesos ω_c, ω_r, ω_c y ω_r. Las matrices correspondientes se denotarán por $\mathbf{W}_{x1}, \mathbf{W}_{x2}, \mathbf{W}_{x3}$ y $\mathbf{W}_{x4}$, respectivamente, como se puede observar en la figura 3.2. Todas esas matrices tienen tamaño $m \times m$.

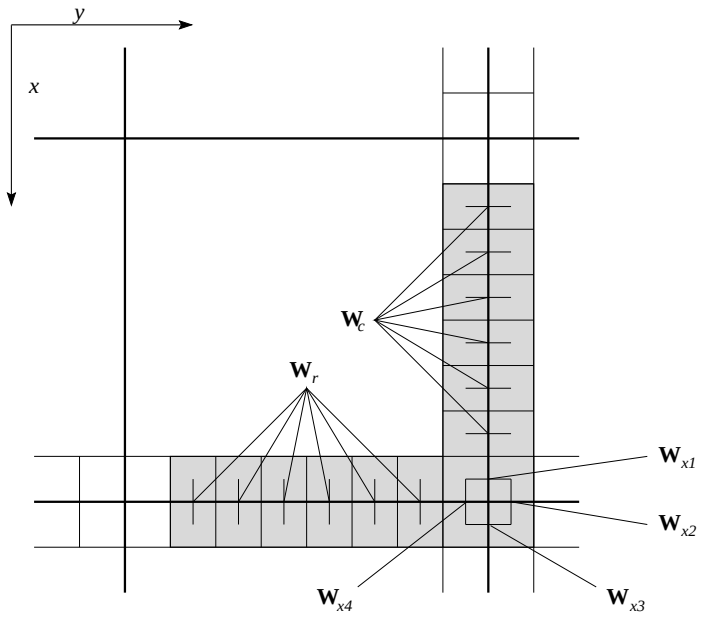


Figura 3.2: Distribución de los pesos respecto a las fronteras del bloque.

Al igual que hicimos para las imágenes $\mathbf{f}$ y $\mathbf{g}$, usaremos

$$\mathbf{W} = (\mathbf{W}_c\ \mathbf{W}_r\ \mathbf{W}_{x1}\ \mathbf{W}_{x2}\ \mathbf{W}_{x3}\ \mathbf{W}_{x4}), \tag{3.20}$$

cuando sea necesario para referirnos a la matriz de pesos que pondera cada diferencia entre las parejas de pixels.

En el apéndice B se ofrecen varias posibilidades para calcular los valores de los pesos.

Para incorporar este conocimiento sobre la actividad de las diferentes regiones de la imagen al modelo a priori definido en la Ec. (3.18), tenemos que redefinir las funciones de energía de la distribución, $P_c(\mathbf{f}_c \mid \alpha)$, $P_r(\mathbf{f}_r \mid \alpha)$ y $P_x(\mathbf{f}_x \mid \alpha)$, definidas en las Ecs. (3.15), (3.16) y (3.17), respectivamente. Esta redefinición se hace ponderando cada diferencia entre filas o columnas por el valor del peso correspondiente de la siguiente forma

$$P_c(\mathbf{f}_c \mid \alpha) = \alpha \parallel \mathbf{W}_c(\mathbf{f}_{cl} - \mathbf{f}_{cr}) \parallel^2, \tag{3.21}$$

$$P_r(\mathbf{f}_r \mid \alpha) = \alpha \parallel \mathbf{W}_r(\mathbf{f}_{ra} - \mathbf{f}_{rb}) \parallel^2, \tag{3.22}$$

$$\begin{aligned} P_x(\mathbf{f}_x \mid \alpha) &= \frac{1}{2}\alpha \Big\{ \parallel \mathbf{W}_{x1}(\mathbf{f}_{al} - \mathbf{f}_{ar}) \parallel^2 + \parallel \mathbf{W}_{x2}(\mathbf{f}_{ar} - \mathbf{f}_{br}) \parallel^2 \\ &+ \parallel \mathbf{W}_{x3}(\mathbf{f}_{br} - \mathbf{f}_{bl}) \parallel^2 + \parallel \mathbf{W}_{x4}(\mathbf{f}_{bl} - \mathbf{f}_{al}) \parallel^2 \Big\}. \end{aligned} \tag{3.23}$$

Por tanto, el modelo a priori propuesto sería

$$\mathrm{p}(\mathbf{f} \mid \alpha) = \mathrm{p}(\mathbf{f}_c, \mathbf{f}_r, \mathbf{f}_x \mid \alpha) = \mathrm{p}(\mathbf{f}_c \mid \alpha)\mathrm{p}(\mathbf{f}_r \mid \alpha)\mathrm{p}(\mathbf{f}_x \mid \alpha)$$

$$\propto \quad \alpha^{\frac{p+q+3m}{2}} \exp\left\{-P(\mathbf{f} \mid \alpha)\right\}, \tag{3.24}$$

con

$$P(\mathbf{f} \mid \alpha) = P_c(\mathbf{f}_c \mid \alpha) + P_r(\mathbf{f}_r \mid \alpha) + P_x(\mathbf{f}_x \mid \alpha). \tag{3.25}$$

Está claro que el modelo de imagen no adaptativo propuesto en las Ec. (3.18) puede verse como un caso particular de este modelo adaptativo cuando la matriz de pesos, $\mathbf{W}$, es la identidad, $\mathbf{W} = \mathbf{I}$.

Este modelo propone un único parámetro para controlar la suavidad tanto en las filas como en las columnas. Sin embargo, las características locales en las filas y las columnas pueden (y, de hecho, suelen) ser diferentes. Pensemos, por ejemplo en una imagen formada por barras horizontales. En este tipo de imágenes, la varianza de la imagen, medida entre las diferentes columnas, es muy baja mientras que la varianza horizontal, medida entre las filas, será mucho mayor. Este tipo de comportamiento lo presenta gran cantidad de imágenes como, por ejemplo, la clásica imagen de "Lena" donde la presencia de elementos verticales es muy superior al de elementos horizontales siendo, por tanto, la varianza vertical de la imagen mayor que la horizontal.

Para obtener un modelo que contemple esta característica presente en una gran cantidad de imágenes, se propone utilizar varios parámetros para controlar la suavidad en la imagen. En concreto, se puede usar un parámetro para controlar la suavidad para las columnas y otro parámetro para controlar la suavidad en las filas.

La definición del modelo de imagen en este caso, estará marcada por el uso de dos hiper-parámetros, α_c y α_r, que controlan la suavidad de la imagen entre las columnas y las filas de los bordes de los bloques, respectivamente. La definición del modelo es como sigue;

$$\begin{aligned}
\mathrm{p}(\mathbf{f} \mid \alpha_c, \alpha_r) &= \mathrm{p}(\mathbf{f}_c, \mathbf{f}_r, \mathbf{f}_x \mid \alpha_c, \alpha_r) \\
&= \mathrm{p}(\mathbf{f}_c \mid \alpha_c)\mathrm{p}(\mathbf{f}_r \mid \alpha_r)\mathrm{p}(\mathbf{f}_x \mid \alpha_c, \alpha_r) \\
&\propto \alpha_c^{\frac{p}{2}} \exp\left\{-P_c(\mathbf{f}_c \mid \alpha_c)\right\} \alpha_r^{\frac{q}{2}} \exp\left\{-P_r(\mathbf{f}_r \mid \alpha_r)\right\} \\
&\quad F(\alpha_c, \alpha_r) \exp\left\{-P_x(\mathbf{f}_x \mid \alpha_c, \alpha_r)\right\} \\
&= \alpha_c^{\frac{p}{2}} \alpha_r^{\frac{q}{2}} F(\alpha_c, \alpha_r) \exp\left\{-P(\mathbf{f} \mid \alpha_c, \alpha_r)\right\},
\end{aligned} \tag{3.26}$$

donde

$$P(\mathbf{f} \mid \alpha_c, \alpha_r) \quad = \quad P_c(\mathbf{f}_c \mid \alpha_c) + P_r(\mathbf{f}_r \mid \alpha_r) + P_x(\mathbf{f}_x \mid \alpha_c, \alpha_r), \tag{3.27}$$

$$F(\alpha_c, \alpha_r) = \prod_{i=1}^{m} \left[\alpha_c \alpha_r \left(\alpha_c \left(\frac{1}{\omega_{x2}^2(i)} + \frac{1}{\omega_{x4}^2(i)} \right) + \alpha_r \left(\frac{1}{\omega_{x1}^2(i)} + \frac{1}{\omega_{x3}^2(i)} \right) \right) \right]^{\frac{1}{2}}, \quad (3.28)$$

y

$$P_c(\mathbf{f}_c \mid \alpha_c) = \alpha_c \parallel \mathbf{W}_c(\mathbf{f}_{cl} - \mathbf{f}_{cr}) \parallel^2, \quad (3.29)$$

$$P_r(\mathbf{f}_r \mid \alpha_r) = \alpha_r \parallel \mathbf{W}_r(\mathbf{f}_{ra} - \mathbf{f}_{rb}) \parallel^2, \quad (3.30)$$

$$P_x(\mathbf{f}_x \mid \alpha_c, \alpha_r) = \frac{1}{2} \Big\{ \alpha_c \parallel \mathbf{W}_{x1}(\mathbf{f}_{al} - \mathbf{f}_{ar}) \parallel^2 + \alpha_r \parallel \mathbf{W}_{x2}(\mathbf{f}_{ar} - \mathbf{f}_{br}) \parallel^2 + \alpha_c \parallel \mathbf{W}_{x3}(\mathbf{f}_{br} - \mathbf{f}_{bl}) \parallel^2 + \alpha_r \parallel \mathbf{W}_{x4}(\mathbf{f}_{bl} - \mathbf{f}_{al}) \parallel^2 \Big\}, \quad (3.31)$$

y los hiperparámetros α_c y α_r controlan la suavidad entre las columnas y las filas de los bordes de los bloques, respectivamente. Nótese que, de nuevo, la distribución a priori es singular, por lo es necesario usar $\frac{p}{2}$ en lugar de p, $\frac{q}{2}$ en lugar de q y p($\mathbf{f}_x \mid \alpha_c, \alpha_r$) tiene sólo 3 autovalores no nulos para cada intersección de cuatro bloques, en lugar de cuatro. Véase el apéndice A para los detalles sobre la elección de los valores de los parámetros de normalización.

Los modelos de imagen definidos en esta sección son bastante simples aunque cumplen bastante bien la función para la que han sido diseñados. No obstante, se podrían utilizar modelos de imagen más complejos, como los propuestos en [81, 82, 83, 84], en los que se usa un proceso de línea para preservar las aristas de la imagen al mismo tiempo que se reconstruye la imagen.

3.6 Modelos de hiperparámetros

Dado el atractivo de la maquinaria bayesiana para realizar el análisis condicional, Berger [5] describe la posibilidad de usar la aproximación bayesiana incluso cuando se dispone de muy poca (o ninguna) información a priori sobre los hiperparámetros. En estas situaciones lo que se necesita es una *distribución a priori no informativa* que denota una distribución a priori que no contiene contiene información sobre los hiperparámetros o, dicho más crudamente, que no "favorece" un valor del hiperparámetro frente a otros. Por ejemplo, si tuviéramos dos posibles valores para el hiperparámetro, una distribución a priori que da una probabilidad $\frac{1}{2}$ a cada uno de ellos es claramente no informativa. Si el espacio en el que los hiperparámetros se encuentran definido es infinito, $\alpha, \beta \in [0, \infty)$, por ejemplo, y se desea una distribución a priori no informativa, parece razonable usar dar igual peso a todos los posibles valores de α y β y es

usual escoger, como distribución a priori, la definida como

$$\mathrm{p}(\alpha) \propto \mathrm{const} > 0, \qquad \mathrm{p}(\beta) \propto \mathrm{const} > 0. \tag{3.32}$$

Sin embargo, es posible incorporar conocimiento preciso sobre el valor de los hiperparámetros en la distribución a priori sobre los hiperparámetros. Examinemos las características de las distribuciones que usaremos.

En general, dependiendo de las distribuciones $\mathrm{p}(\alpha)$ y $\mathrm{p}(\beta)$ que se usen, $\mathrm{p}(\alpha, \beta \mid \mathbf{g})$ puede que no sea fácilmente computable. Una gran parte de la literatura bayesiana se dedica a encontrar distribuciones a priori para las que $\mathrm{p}(\alpha, \beta \mid \mathbf{g})$ pueda calcularse fácilmente. Estas son las llamadas distribuciones a priori conjugadas [5], que fueron extensivamente desarrolladas en Raiffa y Schlaifer [103].

Además de proporcionar un fácil cálculo de $\mathrm{p}(\alpha, \beta \mid \mathbf{g})$, las distribuciones a priori conjugadas tienen, como veremos más tarde, la característica intuitiva de permitir empezar con una cierta forma funcional de la distribución a priori y terminar con una distribución a posteriori con la misma forma funcional, pero con los parámetros actualizados con la información de los datos. Como describe Berger [5], al menos para un análisis inicial, las distribuciones a priori conjugadas como las que vamos a usar en nuestro problema, son bastante útiles en la práctica.

Esas atractivas propiedades de las distribuciones conjugadas son, sin embargo, sólo de importancia secundaria comparada a la cuestión básica de si puede escogerse una distribución a priori conjugada que de una aproximación razonable a la verdadera distribución a priori. Esto nos lleva al problema de la selección del modelo que no va a ser tratado en esta memoria (véase [64, 65]).

Teniendo en cuenta estas consideraciones sobre las distribuciones a priori conjugadas, usaremos como distribución a priori para los hiperparámetros una de esas distribuciones conjugadas, la distribución gamma definida por

$$\mathrm{p}(\theta) \propto \theta^{l-1} \exp[-al\theta], \tag{3.33}$$

donde θ denota un hiperparámetro, a es una constante cuyo significado se precisará después y l es una cantidad no negativa. Esta distribución tiene las siguientes propiedades.

$$E[\theta] = \frac{1}{a} \quad \text{y} \quad Var[\theta] = \frac{1}{a^2 l}.$$

Así, el hiperparámetro θ tiene como media $1/a$ y su varianza decrece conforme l crece. Esto

significa que l se puede ver como una medida de la certeza que tenemos en el conocimiento sobre la media del hiperparámetro (ver [78, 90]).

Una vez estudiados los modelos de imagen y ruido que usaremos en esta memoria estudiemos como realizar el análisis bayesiano.

3.7 Soluciones al problema de la reconstrucción

Habiendo definido $\mathrm{p}(\alpha, \beta, \mathbf{f}, \mathbf{g})$, se realiza el análisis bayesiano. Este análisis puede llevarse a cabo de dos formas diferentes. Una forma es la llamada análisis basado en la moda a posteriori (MAP) y, la otra, la llamada análisis basado en la evidencia. Estudiemos esos dos tipos de análisis en detalle.

3.7.1 Análisis basado en la moda a posteriori

Este análisis, sugerido recientemente por Buntine y Weigend [10], Strauss *y otros* [115], Wolpert [126], Molina [77] y comentado por Archer y Titterington [3], realiza la estimación de la imagen y los hiperparámetros simultáneamente integrando $\mathrm{p}(\alpha, \beta, \mathbf{f}, \mathbf{g})$ sobre α y β para obtener la verdadera verosimilitud y maximizando esta verosimilitud sobre $\mathbf{f}$. El proceso de estimación simultánea de $\mathbf{f}$, α y β es como sigue:

- Paso de estimación de parámetros:

$$\hat{\alpha}, \hat{\beta} = \arg\max_{\alpha,\beta} \mathrm{p}(\alpha, \beta \mid \hat{\mathbf{f}}, \mathbf{g}).$$

- Paso de reconstrucción:

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} \mathrm{p}(\mathbf{f}, \mathbf{g}) = \arg\max_{\mathbf{f}} \int_{\alpha} \int_{\beta} \mathrm{p}(\alpha, \beta, \mathbf{f}, \mathbf{g}) d\alpha \, d\beta.$$

Nótese que el análisis basado en el MAP no se preocupa realmente de la estimación de los hiperparámetros α y β y el primer paso del proceso puede entenderse realmente como un paso intermedio para el cálculo de $\hat{\mathbf{f}}$.

3.7.2 Análisis basado en la evidencia

Con el llamado análisis basado en la evidencia (ver [5, 66] para otros posibles nombres), $\mathrm{p}(\alpha, \beta, \mathbf{f}, \mathbf{g})$ se integra sobre $\mathbf{f}$ para obtener la evidencia $\mathrm{p}(\alpha, \beta \mid \mathbf{g})$ que entonces se maximiza

sobre los hiperparámetros y la reconstrucción se obtiene usando esos parámetros estimados. El proceso de estimación simultánea de $\mathbf{f}$, α y β es, en este caso, como sigue:

- Paso de estimación de parámetros:

$$\hat{\alpha}, \hat{\beta} = \arg\max_{\alpha,\beta} \mathrm{p}(\alpha, \beta \mid \mathbf{g}) = \arg\max_{\alpha,\beta} \int_{\mathbf{f}} \mathrm{p}(\alpha, \beta, \mathbf{f}, \mathbf{g}) d\mathbf{f}.$$

- Paso de reconstrucción:

$$\mathbf{f}^{(\hat{\alpha},\hat{\beta})} = \arg\max_{\mathbf{f}} \mathrm{p}(\mathbf{f} \mid \mathbf{g}, \hat{\alpha}, \hat{\beta}).$$

En esta memoria usaremos el análisis de la evidencia en lugar del análisis de la moda a posteriori. Hemos encontrado que este análisis produce mejores resultados para los problemas de reconstrucción-restauración como se puede ver en [79].

Capítulo 4

Reconstrucción en el decodificador basada en la evidencia

4.1 Introducción

Una vez definidos en el capítulo anterior los elementos del paradigma bayesiano jerárquico y estudiadas las diferentes soluciones al problema de la reconstrucción, en este capítulo estudiaremos los métodos para la reconstrucción de imágenes en el decodificador usando el análisis basado en el evidencia para la estimación de los hiperparámetros y la reconstrucción simultánea de la imagen. Dado que en el decodificador no poseemos información alguna sobre los valores que pueden tomar los hiperparámetros, usaremos como hiperdistribución a priori sobre los hiperparámetros la distribución uniforme, no informativa, definida en la Ec. (3.32). Puesto que, en el capítulo 3, hemos definido los diferentes modelos para el ruido (ver sección 3.4) y para la imagen (sección 3.5), en este capítulo estudiaremos métodos diferentes para el proceso de reconstrucción dependiendo del modelo de imagen y ruido usado.

El modelo de imagen no adaptativo, descrito por la Ec. (3.18), no es útil para la reconstrucción de imágenes altamente comprimidas, como se comentó en el capítulo 2 y en la sección 3.5 del capítulo anterior, y, por tanto, no vamos a estudiarlo, centrándonos en esta memoria en el uso de modelos de imagen adaptativos. Así, en la sección 4.2 se estudia el proceso de reconstrucción para un modelo de imagen con un único parámetro para controlar la suavidad en las filas y las columnas y un modelo de ruido con un único parámetro para las filas y las columnas. En la sección 4.3 se estudia el proceso de reconstrucción para un modelo de imagen

con un parámetro para las filas y otro diferente para las columnas y un modelo de ruido con un único parámetro para las filas y las columnas. En la sección 4.4 se estudia el proceso de reconstrucción para un modelo de imagen y un modelo de ruido con parámetros diferentes para las filas y las columnas y, por último, la sección 4.5 muestra los resultados de los experimentos realizados con los métodos propuestos sobre imágenes reales.

4.2 Reconstrucción adaptativa con los mismos parámetros para filas y columnas en modelos de imagen y ruido

En esta sección usaremos el modelo de imagen adaptativo con un único parámetro para controlar la suavidad de las filas y las columnas, definido en la Ec. (3.24) y el modelo de ruido, también con un único parámetro para las filas y las columnas, definido en la Ec. (3.4). Por tanto, el modelo que vamos a usar se define como

$$\mathrm{p}(\mathbf{f},\mathbf{g} \mid \alpha,\beta) \propto \alpha^{\frac{p+q+3m}{2}} \beta^{p+q+2m} \exp\left\{-C(\mathbf{f},\mathbf{g} \mid \alpha,\beta)\right\},$$

donde $C(\mathbf{f},\mathbf{g} \mid \alpha,\beta)$ se define como

$$C(\mathbf{f},\mathbf{g} \mid \alpha,\beta) = C_c(\mathbf{f}_c,\mathbf{g}_c \mid \alpha,\beta) + C_r(\mathbf{f}_r,\mathbf{g}_r \mid \alpha,\beta) + C_x(\mathbf{f}_x,\mathbf{g}_x \mid \alpha,\beta), \tag{4.1}$$

con

$$C_c(\mathbf{f}_c,\mathbf{g}_c \mid \alpha,\beta) = \alpha \parallel \mathbf{W}_c(\mathbf{f}_{cl}-\mathbf{f}_{cr}) \parallel^2 + \frac{1}{2}\beta\left\{\parallel \mathbf{g}_{cl}-\mathbf{f}_{cl} \parallel^2 + \parallel \mathbf{g}_{cr}-\mathbf{f}_{cr} \parallel^2\right\}, \tag{4.2}$$

$$C_r(\mathbf{f}_r,\mathbf{g}_r \mid \alpha,\beta) = \alpha \parallel \mathbf{W}_r(\mathbf{f}_{ra}-\mathbf{f}_{rb}) \parallel^2 + \frac{1}{2}\beta\left\{\parallel \mathbf{g}_{ra}-\mathbf{f}_{ra} \parallel^2 + \parallel \mathbf{g}_{rb}-\mathbf{f}_{rb} \parallel^2\right\}, \tag{4.3}$$

$$\begin{aligned} C_x(\mathbf{f}_x,\mathbf{g}_x \mid \alpha,\beta) &= \frac{1}{2}\alpha\left\{\parallel \mathbf{W}_{x1}(\mathbf{f}_{al}-\mathbf{f}_{ar}) \parallel^2 + \parallel \mathbf{W}_{x2}(\mathbf{f}_{ar}-\mathbf{f}_{br}) \parallel^2\right. \\ &+ \left.\parallel \mathbf{W}_{x3}(\mathbf{f}_{br}-\mathbf{f}_{bl}) \parallel^2 + \parallel \mathbf{W}_{x4}(\mathbf{f}_{bl}-\mathbf{f}_{al}) \parallel^2\right\} \\ &+ \frac{1}{2}\beta\left\{\parallel \mathbf{g}_{al}-\mathbf{f}_{al} \parallel^2 + \parallel \mathbf{g}_{ar}-\mathbf{f}_{ar} \parallel^2\right. \\ &+ \left.\parallel \mathbf{g}_{br}-\mathbf{f}_{br} \parallel^2 + \parallel \mathbf{g}_{bl}-\mathbf{f}_{bl} \parallel^2\right\}. \end{aligned} \tag{4.4}$$

Como se explicó en la sección 3.3, el análisis basado en la evidencia realiza la estimación simultánea de la imagen y los hiperparámetros en dos pasos. En el primero, se seleccionan $\hat{\alpha}$ y $\hat{\beta}$ como los estimadores de máxima verosimilitud (*mle*) de α y β a partir de $\mathrm{p}(\alpha,\beta \mid \mathbf{g})$ y estos parámetros se usan en el segundo paso para realizar la reconstrucción de la imagen.

En las siguientes secciones estudiaremos en detalle cada uno de estos pasos de la estimación mediante el análisis bayesiano jerárquico basado en la evidencia y propondremos un algoritmo iterativo para realizar la estimación simultánea de los hiperparámetros y la reconstrucción.

4.2.1 Paso de estimación de los hiperparámetros

Describamos en detalle la estimación de los hiperparámetros α y β para el caso de una distribución a priori no informativa.

Procedamos a estimar $\mathrm{p}(\alpha, \beta \mid \mathbf{g})$ teniendo en cuenta que

$$\mathrm{p}(\alpha, \beta \mid \mathbf{g}) \propto \mathrm{p}(\mathbf{g} \mid \alpha, \beta) = \alpha^{\frac{p+q+3m}{2}} \beta^{p+q+2m} \int_{\mathbf{f}} \exp[-C(\mathbf{f}, \mathbf{g} \mid \alpha, \beta)] d\mathbf{f}.$$

Fijemos α y β y expandamos $C(\mathbf{f}, \mathbf{g} \mid \alpha, \beta)$ alrededor de $\mathbf{f}^{(\alpha,\beta)}$. Entonces obtenemos,

$$\begin{aligned}
C(\mathbf{f}, \mathbf{g} \mid \alpha, \beta) &= \alpha \Big\{ \| \mathbf{W}_c(\mathbf{f}_{cl} - \mathbf{f}_{cr}) \|^2 + \| \mathbf{W}_r(\mathbf{f}_{ra} - \mathbf{f}_{rb}) \|^2 + \| \mathbf{W}_{x1}(\mathbf{f}_{al} - \mathbf{f}_{ar}) \|^2 \\
&+ \| \mathbf{W}_{x2}(\mathbf{f}_{ar} - \mathbf{f}_{br}) \|^2 + \| \mathbf{W}_{x3}(\mathbf{f}_{br} - \mathbf{f}_{bl}) \|^2 + \| \mathbf{W}_{x4}(\mathbf{f}_{bl} - \mathbf{f}_{al}) \|^2 \Big\} \\
&+ \frac{1}{2}\beta \Big\{ \| \mathbf{g}_{cl} - \mathbf{f}_{cl} \|^2 + \| \mathbf{g}_{cr} - \mathbf{f}_{cr} \|^2 + \| \mathbf{g}_{ra} - \mathbf{f}_{ra} \|^2 + \| \mathbf{g}_{rb} - \mathbf{f}_{rb} \|^2 \\
&+ \| \mathbf{g}_{al} - \mathbf{f}_{al} \|^2 + \| \mathbf{g}_{ar} - \mathbf{f}_{ar} \|^2 + \| \mathbf{g}_{br} - \mathbf{f}_{br} \|^2 + \| \mathbf{g}_{bl} - \mathbf{f}_{bl} \|^2 \Big\} \\
&= \alpha \Big\{ \| \mathbf{W}_c(\mathbf{f}_{cl}^{(\alpha,\beta)} - \mathbf{f}_{cr}^{(\alpha,\beta)}) \|^2 + \| \mathbf{W}_r(\mathbf{f}_{ra}^{(\alpha,\beta)} - \mathbf{f}_{rb}^{(\alpha,\beta)}) \|^2 \\
&+ \| \mathbf{W}_{x1}(\mathbf{f}_{al}^{(\alpha,\beta)} - \mathbf{f}_{ar}^{(\alpha,\beta)}) \|^2 + \| \mathbf{W}_{x2}(\mathbf{f}_{ar}^{(\alpha,\beta)} - \mathbf{f}_{br}^{(\alpha,\beta)}) \|^2 \\
&+ \| \mathbf{W}_{x3}(\mathbf{f}_{br}^{(\alpha,\beta)} - \mathbf{f}_{bl}^{(\alpha,\beta)}) \|^2 + \| \mathbf{W}_{x4}(\mathbf{f}_{bl}^{(\alpha,\beta)} - \mathbf{f}_{al}^{(\alpha,\beta)}) \|^2 \Big\} \\
&+ \frac{1}{2}\beta \Big\{ \| \mathbf{g}_{cl} - \mathbf{f}_{cl}^{(\alpha,\beta)} \|^2 + \| \mathbf{g}_{cr} - \mathbf{f}_{cr}^{(\alpha,\beta)} \|^2 + \| \mathbf{g}_{ra} - \mathbf{f}_{ra}^{(\alpha,\beta)} \|^2 \\
&+ \| \mathbf{g}_{rb} - \mathbf{f}_{rb}^{(\alpha,\beta)} \|^2 + \| \mathbf{g}_{al} - \mathbf{f}_{al}^{(\alpha,\beta)} \|^2 + \| \mathbf{g}_{ar} - \mathbf{f}_{ar}^{(\alpha,\beta)} \|^2 \\
&+ \| \mathbf{g}_{br} - \mathbf{f}_{br}^{(\alpha,\beta)} \|^2 + \| \mathbf{g}_{bl} - \mathbf{f}_{bl}^{(\alpha,\beta)} \|^2 \Big\} + \frac{1}{2}(\mathbf{f} - \mathbf{f}^{(\alpha,\beta)})^t \mathbf{Q}(\alpha, \beta)(\mathbf{f} - \mathbf{f}^{(\alpha,\beta)}) \\
&= C(\mathbf{f}^{(\alpha,\beta)}, \mathbf{g} \mid \alpha, \beta) + \frac{1}{2}(\mathbf{f} - \mathbf{f}^{(\alpha,\beta)})^t \mathbf{Q}(\alpha, \beta)(\mathbf{f} - \mathbf{f}^{(\alpha,\beta)})
\end{aligned}$$

donde $\mathbf{Q}(\alpha, \beta)$ es una matriz $(2p + 2q + 4m) \times (2p + 2q + 4m)$ diagonal por bloques compuesta por p matrices de orden 2×2 definidas por

$$\mathbf{q}_{(i,i)}(\alpha, \beta) = \mathbf{q}_{c\,(i,i)}(\alpha, \beta) = \alpha \mathbf{A}_c(i) + \beta \mathbf{I}_{2\times 2} \quad (i = 1, \ldots, p),$$

donde

$$\mathbf{A}_c(i) = \begin{bmatrix} 2\omega_c^2(i) & -2\omega_c^2(i) \\ -2\omega_c^2(i) & 2\omega_c^2(i) \end{bmatrix},$$

q matrices de orden 2×2 definidas por

$$\mathbf{q}_{(i,i)}(\alpha,\beta) = \mathbf{q}_{r\,(i,i)}(\alpha,\beta) = \alpha\mathbf{A}_r(i) + \beta\mathbf{I}_{2\times 2} \quad (i = 1,\ldots,q),$$

donde

$$\mathbf{A}_r(i) = \begin{bmatrix} 2\omega_r^2(i) & -2\omega_r^2(i) \\ -2\omega_r^2(i) & 2\omega_r^2(i) \end{bmatrix},$$

y m matrices de orden 4×4 definidas como

$$\mathbf{q}_{(i,i)}(\alpha,\beta) = \mathbf{q}_{x\,(i,i)}(\alpha,\beta) = \alpha\mathbf{A}_x(i) + \beta\mathbf{I}_{4\times 4} \quad (i = 1,\ldots,m),$$

donde

$$\mathbf{A}_x(i) = \begin{bmatrix} \omega_{x1}^2(i)+\omega_{x4}^2(i) & -\omega_{x1}^2(i) & 0 & -\omega_{x4}^2(i) \\ -\omega_{x1}^2(i) & \omega_{x1}^2(i)+\omega_{x3}^2(i) & -\omega_{x3}^2(i) & 0 \\ 0 & -\omega_{x2}^2(i) & \omega_{x2}^2(i)+\omega_{x4}^2(i) & -\omega_{x4}^2(i) \\ -\omega_{x1}^2(i) & 0 & -\omega_{x3}^2(i) & \omega_{x2}^2(i)+\omega_{x3}^2(i) \end{bmatrix}.$$

Entonces tenemos que

$$\begin{aligned} \mathrm{p}(\alpha,\beta \mid \mathbf{g}) &\propto \mathrm{p}(\mathbf{g} \mid \alpha,\beta) \propto \alpha^{\frac{p+q+3m}{2}}\beta^{p+q+2m}\exp\{-C(\mathbf{f}^{(\alpha,\beta)},\mathbf{g} \mid \alpha,\beta)\} \\ &\times \int_{\mathbf{f}} \exp\left\{-\frac{1}{2}(\mathbf{f}-\mathbf{f}^{(\alpha,\beta)})^t\mathbf{Q}(\alpha,\beta)(\mathbf{f}-\mathbf{f}^{(\alpha,\beta)})\right\} d\mathbf{f} \\ &= \alpha^{\frac{p+q+3m}{2}}\beta^{p+q+2m}\exp\{-C(\mathbf{f}^{(\alpha,\beta)},\mathbf{g} \mid \alpha,\beta)\}[\det\mathbf{Q}(\alpha,\beta)]^{-\frac{1}{2}}, \end{aligned}$$

Siguiendo el análisis basado en la evidencia, el valor estimado de α y β se obtiene como

$$\alpha,\beta = \arg\max_{\alpha,\beta} \mathrm{p}(\alpha,\beta \mid \mathbf{g}).$$

Diferenciando, entonces, $-\log \mathrm{p}(\alpha,\beta \mid \mathbf{g})$ respecto a α y β tenemos

$$\begin{aligned} \frac{p+q+3m}{\alpha} &= 2 \parallel \mathbf{W}_c(\mathbf{f}_{cl}^{(\alpha,\beta)} - \mathbf{f}_{cr}^{(\alpha,\beta)}) \parallel^2 + 2 \parallel \mathbf{W}_r(\mathbf{f}_{ra}^{(\alpha,\beta)} - \mathbf{f}_{rb}^{(\alpha,\beta)}) \parallel^2 \\ &+ \parallel \mathbf{W}_{x1}(\mathbf{f}_{al}^{(\alpha,\beta)} - \mathbf{f}_{ar}^{(\alpha,\beta)}) \parallel^2 + \parallel \mathbf{W}_{x2}(\mathbf{f}_{ar}^{(\alpha,\beta)} - \mathbf{f}_{br}^{(\alpha,\beta)}) \parallel^2 \\ &+ \parallel \mathbf{W}_{x3}(\mathbf{f}_{br}^{(\alpha,\beta)} - \mathbf{f}_{bl}^{(\alpha,\beta)}) \parallel^2 + \parallel \mathbf{W}_{x4}(\mathbf{f}_{bl}^{(\alpha,\beta)} - \mathbf{f}_{al}^{(\alpha,\beta)}) \parallel^2 \\ &+ \sum_{i=1}^{p} \mathrm{traza}\left([\mathbf{q}_{c(i,i)}(\alpha,\beta)]^{-1}\mathbf{A}_c(i)\right) \\ &+ \sum_{i=1}^{q} \mathrm{traza}\left([\mathbf{q}_{r(i,i)}(\alpha,\beta)]^{-1}\mathbf{A}_r(i)\right) \end{aligned}$$

$$+\sum_{i=1}^{m}\text{traza}\left([\mathbf{q}_{x(i,i)}(\alpha,\beta)]^{-1}\mathbf{A}_x(i)\right), \tag{4.5}$$

$$\begin{aligned}\frac{2(p+q+2m)}{\beta} &= \| \mathbf{f}_{cl}^{(\alpha,\beta)} - \mathbf{g}_{cl} \|^2 + \| \mathbf{f}_{cr}^{(\alpha,\beta)} - \mathbf{g}_{cr} \|^2 + \| \mathbf{f}_{ra}^{(\alpha,\beta)} - \mathbf{g}_{ra} \|^2 \\ &+ \| \mathbf{f}_{rb}^{(\alpha,\beta)} - \mathbf{g}_{rb} \|^2 + \| \mathbf{f}_{al}^{(\alpha,\beta)} - \mathbf{g}_{al} \|^2 + \| \mathbf{f}_{ar}^{(\alpha,\beta)} - \mathbf{g}_{ar} \|^2 \\ &+ \| \mathbf{f}_{bl}^{(\alpha,\beta)} - \mathbf{g}_{bl} \|^2 + \| \mathbf{f}_{br}^{(\alpha,\beta)} - \mathbf{g}_{br} \|^2 \\ &+ \sum_{i=1}^{p+q+m}\text{traza}\left([\mathbf{q}_{(i,i)}(\alpha,\beta)]^{-1}\right). \end{aligned} \tag{4.6}$$

Obviamente, las Ecs. (4.5) y (4.6) caracterizan a $\hat{\alpha}$ y $\hat{\beta}$ pero no nos proveen de una forma de estimarlos. El proceso para estimarlos se describe en la sección 4.2.3.

4.2.2 Paso de reconstrucción

Veamos ahora como puede realizar la estimación de la imagen. Dados α y β, $\mathbf{f}_c^{(\alpha,\beta)}$ se obtiene derivando la Ec. (4.2) respecto a $\mathbf{f}_c$, obteniendo

$$\begin{aligned} 2\alpha\mathbf{W}_c^t\mathbf{W}_c(\mathbf{f}_{cl}^{(\alpha,\beta)} - \mathbf{f}_{cr}^{(\alpha,\beta)}) + \beta(\mathbf{g}_{cl} - \mathbf{f}_{cl}^{(\alpha,\beta)}) &= 0, \\ 2\alpha\mathbf{W}_c^t\mathbf{W}_c(\mathbf{f}_{cr}^{(\alpha,\beta)} - \mathbf{f}_{cl}^{(\alpha,\beta)}) + \beta(\mathbf{g}_{cr} - \mathbf{f}_{cr}^{(\alpha,\beta)}) &= 0, \end{aligned}$$

que puede resolverse fácilmente puesto que sólo hay que invertir matrices 2×2 siendo la solución,

$$\mathbf{f}_{cl}(i) = \frac{1}{2}\left(1 + \frac{\beta}{\beta + 4\alpha\omega_c^2(i)}\right)\mathbf{g}_{cl}(i) + \frac{1}{2}\left(1 - \frac{\beta}{\beta + 4\alpha\omega_c^2(i)}\right)\mathbf{g}_{cr}(i), \tag{4.7}$$

$$\mathbf{f}_{cr}(i) = \frac{1}{2}\left(1 - \frac{\beta}{\beta + 4\alpha\omega_c^2(i)}\right)\mathbf{g}_{cl}(i) + \frac{1}{2}\left(1 + \frac{\beta}{\beta + 4\alpha\omega_c^2(i)}\right)\mathbf{g}_{cr}(i), \tag{4.8}$$

para $i = 1, 2, \ldots, p$. De igual forma, derivando la Ec. (4.3) respecto a $\mathbf{f}_r$ obtenemos como solución del sistema

$$\mathbf{f}_{ra}(i) = \frac{1}{2}\left(1 + \frac{\beta}{\beta + 4\alpha\omega_r^2(i)}\right)\mathbf{g}_{ra}(i) + \frac{1}{2}\left(1 - \frac{\beta}{\beta + 4\alpha\omega_r^2(i)}\right)\mathbf{g}_{rb}(i), \tag{4.9}$$

$$\mathbf{f}_{rb}(i) = \frac{1}{2}\left(1 - \frac{\beta}{\beta + 4\alpha\omega_r^2(i)}\right)\mathbf{g}_{ra}(i) + \frac{1}{2}\left(1 + \frac{\beta}{\beta + 4\alpha\omega_r^2(i)}\right)\mathbf{g}_{rb}(i), \tag{4.10}$$

para $i = 1, 2, \ldots, q$ y, para los pixels en una intersección de cuatro bloques, derivando la Ec. (4.4) respecto a $\mathbf{f}_x$, obtenemos $\mathbf{f}_{ar}(i)$, $\mathbf{f}_{al}(i)$, $\mathbf{f}_{br}(i)$ y $\mathbf{f}_{bl}(i)$ como la solución del siguiente sistema de ecuaciones:

$$\alpha[\omega_{x1}^2(i)(\mathbf{f}_{al}(i) - \mathbf{f}_{ar}(i)) + \omega_{x4}^2(i)(\mathbf{f}_{al}(i) - \mathbf{f}_{bl}(i))] + \beta(\mathbf{f}_{al}(i) - \mathbf{g}_{al}(i)) = 0$$

$$\begin{aligned}
\alpha[\omega_{x1}^2(i)(\mathbf{f}_{ar}(i)-\mathbf{f}_{al}(i))+\omega_{x2}^2(i)(\mathbf{f}_{ar}(i)-\mathbf{f}_{br}(i))]+\beta(\mathbf{f}_{ar}(i)-\mathbf{g}_{ar}(i)) &= 0\\
\alpha[\omega_{x3}^2(i)(\mathbf{f}_{br}(i)-\mathbf{f}_{bl}(i))+\omega_{x2}^2(i)(\mathbf{f}_{br}(i)-\mathbf{f}_{ar}(i))]+\beta(\mathbf{f}_{br}(i)-\mathbf{g}_{br}(i)) &= 0\\
\alpha[\omega_{x3}^2(i)(\mathbf{f}_{bl}(i)-\mathbf{f}_{br}(i))+\omega_{x4}^2(i)(\mathbf{f}_{bl}(i)-\mathbf{f}_{al}(i))]+\beta(\mathbf{f}_{bl}(i)-\mathbf{g}_{bl}(i)) &= 0. \qquad (4.11)
\end{aligned}$$

para $i = 1, 2, \ldots, m$, que puede resolverse fácilmente pues sólo hay que invertir matrices 4×4.

Las Ecs. (4.7)–(4.10) pueden reescribirse como

$$\begin{aligned}
\mathbf{f}_{cl}(i) &= \frac{\mathbf{g}_{cl}(i)+\mathbf{g}_{cr}(i)}{2}+\gamma_c(i)\frac{\mathbf{g}_{cl}(i)-\mathbf{g}_{cr}(i)}{2}\\
\mathbf{f}_{cr}(i) &= \frac{\mathbf{g}_{cl}(i)+\mathbf{g}_{cr}(i)}{2}-\gamma_c(i)\frac{\mathbf{g}_{cl}(i)-\mathbf{g}_{cr}(i)}{2}\\
\mathbf{f}_{ra}(i) &= \frac{\mathbf{g}_{ra}(i)+\mathbf{g}_{rb}(i)}{2}+\gamma_r(i)\frac{\mathbf{g}_{ra}(i)-\mathbf{g}_{rb}(i)}{2}\\
\mathbf{f}_{rb}(i) &= \frac{\mathbf{g}_{ra}(i)+\mathbf{g}_{rb}(i)}{2}-\gamma_r(i)\frac{\mathbf{g}_{ra}(i)-\mathbf{g}_{rb}(i)}{2}
\end{aligned}$$

con

$$\gamma_c(i)=\frac{\beta}{\beta+4\alpha\omega_c^2(i)} \qquad \text{y} \qquad \gamma_r(i)=\frac{\beta}{\beta+4\alpha\omega_r^2(i)}.$$

De estas ecuaciones se desprende que lo que realmente estamos haciendo para estimar el valor de los pixels en la frontera de cada bloque es decidir, para cada uno de esos pixels, cuanto deben parecerse al valor que teóricamente tendrían antes de comprimirse, que viene representado por la media de los dos pixels, una vez que sabemos la diferencia entre ellos. Está claro que en áreas donde los pesos, $\omega(i)$, sean grandes, la diferencia entre los pixels vecinos de la imagen reconstruida será menor que en las zonas donde los pesos son pequeños. Merece la pena estudiar los siguientes dos casos extremos: cuando el peso $\omega(i) = \infty$, la reconstrucción simplemente toma la media de valores de los dos pixels recibidos; cuando $\omega(i) = 0$ el pixel reconstruido no se modifica. Según esto, el valor de estos pesos deberá depender de la visibilidad de los artificios en la zona a la que afecten, siendo mayores en las zonas en que los artificios son más visibles y menores en las zonas en las que los artificios son menos visibles.

4.2.3 Procedimiento iterativo para estimar los hiperparámetros y reconstruir la imagen

Una vez caracterizados los hiperparámetros óptimos y tras mostrar como se puede reconstruir la imagen dados unos hiperparámetros usaremos el siguiente algoritmo para estimar los hiperparámetros y la imagen simultáneamente, asumiendo una hiperdistribución a priori uniforme sobre los hiperparámetros.

Algoritmo 4.1 Reconstrucción adaptativa con los mismos parámetros para las filas y las columnas.

1. *Escoger* α^0 *y* β^0.
2. *Calcular* $\mathbf{f}^{(\alpha^0,\beta^0)}$ *a partir de las Ecs. (4.7)–(4.10) y resolviendo el sistema de ecuaciones descrito en la Ec. (4.11).*
3. *Para* $k = 1, 2, \ldots$
   (a) *Estimar* α^k *y* β^k *sustituyendo los valores de* α^{k-1} *y* β^{k-1} *en la parte derecha de las Ecs. (4.5) y (4.6), respectivamente.*
   (b) *Calcular* $\mathbf{f}^{(\alpha^k,\beta^k)}$ *a partir de las Ecs. (4.7)–(4.10) y resolviendo el sistema de ecuaciones descrito en la Ec. (4.11).*
4. *Ir al paso 3 hasta que* $\| \mathbf{f}^{(\alpha^{k-1},\beta^{k-1})} - \mathbf{f}^{(\alpha^k,\beta^k)} \|$ *sea menor que un límite establecido.*

La convergencia de este algoritmo se establece viendo que corresponde a un algoritmo EM, donde los datos incompletos son las observaciones, $\mathbf{g}$, y los completos son las observaciones, $\mathbf{g}$, y la reconstrucción, $\mathbf{f}$, es decir, $\mathbf{z}$, los datos completos se definen como $\mathbf{z}^t = (\mathbf{f}^t, \mathbf{g}^t)$ y los datos completos e incompletos se relacionan como

$$\mathbf{g} = (0 \quad \mathbf{I})\,\mathbf{z},$$

donde $\mathbf{I}$ y 0 son matrices identidad y cero, respectivamente. Los detalles pueden consultarse en [55, 79].

4.3 Reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido

En esta sección se usa el modelo de imagen adaptativo con dos parámetros, uno para controlar la suavidad de las filas y otro de las columnas, definido en la Ec. (3.26) y el modelo de ruido, con un único parámetro para las filas y las columnas, definido en la Ec. (3.4). Por tanto, el modelo que vamos a usar se define como

$$\mathrm{p}(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta) \propto \alpha_c^{\frac{p}{2}} \alpha_r^{\frac{q}{2}} F(\alpha_c, \alpha_r) \beta^{p+q+2m} \exp\left\{-C(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta)\right\},$$

donde $C(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta)$ se define como

$$C(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta) = C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta) + C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta) + C_x(\mathbf{f}_x, \mathbf{g}_x \mid \alpha_c, \alpha_r, \beta), \tag{4.12}$$

con

$$\begin{aligned} C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta) &= P_c(\mathbf{f}_c \mid \alpha_c) + N_c(\mathbf{g}_c \mid \mathbf{f}_c, \beta), & (4.13)\\ C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta) &= P_r(\mathbf{f}_r \mid \alpha_r) + N_r(\mathbf{g}_r \mid \mathbf{f}_r, \beta), & (4.14)\\ C_x(\mathbf{f}_x, \mathbf{g}_x \mid \alpha_c, \alpha_r, \beta) &= P_x(\mathbf{f}_x \mid \alpha_c, \alpha_r) + N_x(\mathbf{g}_x \mid \mathbf{f}_x, \beta), & (4.15) \end{aligned}$$

con $P_u(\cdot)$, $u \in \{c, r, x\}$, definidos en las Ecs. (3.21), (3.22) y (3.23), $N_u(\cdot)$, $u \in \{c, r, x\}$, definidos en las Ecs. (3.6), (3.7) y (3.8) y $F(\alpha_c, \alpha_r)$ definido en la Ec. (3.28).

Describamos el proceso de estimación de los hiperparámetros y de reconstrucción de la imagen de forma detallada.

Recordemos que en la aproximación de la evidencia $\hat{\alpha}_c$, $\hat{\alpha}_r$ y $\hat{\beta}$ se seleccionan como

$$\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta} = \arg \max_{\alpha_c, \alpha_r, \beta} \mathrm{p}(\alpha_c, \alpha_r, \beta \mid \mathbf{g}),$$

y entonces la reconstrucción, $\mathbf{f}^{(\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta})}$, se obtiene como

$$\mathbf{f}^{(\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta})} = \arg \min_{\mathbf{f}} C(\mathbf{f}, \mathbf{g} \mid \hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta}). \tag{4.16}$$

El proceso de estimación de los hiperparámetros se basa, pues, en maximizar en α_c, α_r y β, $\mathrm{p}(\alpha_c, \alpha_r, \beta \mid \mathbf{g})$, definida como

$$\begin{aligned} \mathrm{p}(\alpha_c, \alpha_r, \beta \mid \mathbf{g}) &\propto \int_{\mathbf{f}} \mathrm{p}(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta) d\mathbf{f} \\ &= \int_{\mathbf{f}_c} \mathrm{p}(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta) d\mathbf{f}_c \int_{\mathbf{f}_r} \mathrm{p}(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta) d\mathbf{f}_r \int_{\mathbf{f}_x} \mathrm{p}(\mathbf{f}_x, \mathbf{g}_x \mid \alpha_c, \alpha_r, \beta) d\mathbf{f}_x \\ &= \mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta) \mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta) \mathrm{p}(\mathbf{g}_x \mid \alpha_c, \alpha_r, \beta), \end{aligned}$$

y nótese que incluir $\mathrm{p}(\mathbf{g}_x \mid \alpha_c, \alpha_r, \beta)$ en el proceso de estimación de α_c, α_r y β, presenta los siguientes dos problemas:

1. Los datos de las intersecciones de cuatro bloques no son fiables.

   Dado que los datos pertenecientes a las intersecciones de cuatro bloques, expresados en $\mathrm{p}(\mathbf{g}_x \mid \alpha_c, \alpha_r, \beta)$, tienen mezclada información sobre los dos hiperparámetros del modelo de imagen, no serán tan fiables como los de las filas o las columnas, donde solamente hay influencia de uno de ellos. Además, al ser pocos los datos de las intersecciones, extraer información útil y fiable de ellos es más difícil que hacerlo de las filas o de las columnas.

2. El método de reconstrucción se ralentiza.

   Para estimar la imagen tendremos que resolver un sistema de cuatro ecuaciones con cuatro incógnitas por cada una de las intersecciones de cuatro bloques, como ya sucedió con el método anterior. Esto implica realizar la inversión de m matrices 4×4 para la obtención de la imagen reconstruida en cada iteración del procedimiento. Tendremos también que realizar la inversión de las matrices envueltas en el cálculo de las trazas. En este caso tenemos tres hiperparámetros y, por tanto, tendremos que hacer la inversión de $3m$ matrices 4×4 para poder calcular las trazas que aparecen en la estimación de los hiperparámetros en cada iteración. Esto lleva a que el método se haga excesivamente lento y costoso.

Una solución a estos problemas es no tener en cuenta los datos en las intersecciones de cuatro bloques para realizar la estimación, es decir, puesto que los datos de $\mathrm{p}(\mathbf{g}_x \mid \alpha_c, \alpha_r, \beta)$ no son fiables y tenerlos en cuenta nos ralentiza el proceso de estimación los descartamos a la hora de hacer las estimación de los parámetros obteniendo así un método más rápido y fiable. Teniendo en cuenta estas consideraciones, procedemos a resolver el problema de la estimación usando los siguientes pasos

1. Estimar α_c, α_r y β mediante

$$\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta} = \arg \max_{\alpha_c, \alpha_r, \beta} \mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta), \tag{4.17}$$

   como se describirá en la sección 4.3.1.

2. Entonces usar $\hat{\alpha}_c$, $\hat{\alpha}_r$ y $\hat{\beta}$ en la Ec. (4.16) para obtener la imagen reconstruida (ver sección 4.3.2).

El algoritmo 4.2, en la sección 4.3.3, lleva a cabo el problema de optimización de la Ec. (4.17) y la estimación de la imagen original usando la Ec. (4.16).

4.3.1 Paso de estimación de los hiperparámetros

El proceso de estimación de α_c, α_r y β, para el caso de la hiperdistribución a priori no informativa, es como sigue.

Procedamos a calcular $\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta)$ en la Ec. (4.17), teniendo en cuenta que

$$\begin{aligned}&\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta) = \\ &\quad \alpha_c^{\frac{p}{2}} \beta^p \alpha_r^{\frac{q}{2}} \beta^q \int_{\mathbf{f}_c, \mathbf{f}_r} \exp[-C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta) + C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta)]\, d\mathbf{f}_c\, d\mathbf{f}_r.\end{aligned}$$

Fijemos α_c, α_r y β y expandamos $C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta) + C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta)$ alrededor de $\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)}$ y $\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)}$ obteniendo

$$\begin{aligned} C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta) + C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta) &= C_c(\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)}, \mathbf{g}_c \mid \alpha_c, \beta) + C_r(\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)}, \mathbf{g}_r \mid \alpha_r, \beta) \\ &+ \frac{1}{2}(\mathbf{f}_c - \mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)})^t \mathbf{Q}_c(\alpha_c, \beta)(\mathbf{f}_c - \mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)}) \\ &+ \frac{1}{2}(\mathbf{f}_r - \mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)})^t \mathbf{Q}_r(\alpha_r, \beta)(\mathbf{f}_r - \mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)}), \end{aligned}$$

donde $\mathbf{Q}_c(\alpha_c, \beta)$ es una matriz diagonal por bloques cuyas matrices diagonales 2×2 son $\mathbf{q}_{c(i,i)}(\alpha_c, \beta) = \alpha_c \mathbf{A}_c(i) + \beta \mathbf{I}_{2\times 2}$ con

$$\mathbf{A}_c(i) = \begin{bmatrix} 2\omega_c^2(i) & -2\omega_c^2(i) \\ -2\omega_c^2(i) & 2\omega_c^2(i) \end{bmatrix},$$

$i = 1, 2, \ldots, p$, y $\mathbf{Q}_r(\alpha_r, \beta)$ es una matriz diagonal por bloques cuyas matrices diagonales 2×2 son $\mathbf{q}_{r(i,i)}(\alpha_r, \beta) = \alpha_r \mathbf{A}_r(i) + \beta \mathbf{I}_{2\times 2}$ con

$$\mathbf{A}_r(i) = \begin{bmatrix} 2\omega_r^2(i) & -2\omega_r^2(i) \\ -2\omega_r^2(i) & 2\omega_r^2(i) \end{bmatrix},$$

$i = 1, 2, \ldots, q$.

Entonces tenemos que

$$\begin{aligned} &\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta) \propto \\ &\quad \alpha_c^{\frac{p}{2}} \beta^p \alpha_r^{\frac{q}{2}} \beta^q \exp\{-C_c(\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)}, \mathbf{g}_c \mid \alpha_c, \beta)\} \exp\{-C_r(\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)}, \mathbf{g}_r \mid \alpha_r, \beta)\} \\ &\times \quad [\det \mathbf{Q}_c(\alpha_c, \beta)]^{-\frac{1}{2}} [\det \mathbf{Q}_r(\alpha_r, \beta)]^{-\frac{1}{2}}. \end{aligned}$$

Entonces, diferenciando $-\log[\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta)]$ respecto a α_c, α_r y β tenemos como $\hat{\alpha}_c$, $\hat{\alpha}_r$ y $\hat{\beta}$

$$\frac{p}{\hat{\alpha}_c} = 2 \parallel \mathbf{W}_c(\mathbf{f}_{cl}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})} - \mathbf{f}_{cr}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})}) \parallel^2 + \sum_{i=1}^{p} \mathrm{traza}\,[\mathbf{q}_{c\,(i,i)}^{-1}(\hat{\alpha}_c, \hat{\beta})\mathbf{A}_c(i)], \quad (4.18)$$

$$\frac{q}{\hat{\alpha}_r} = 2 \parallel \mathbf{W}_r(\mathbf{f}_{ra}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})} - \mathbf{f}_{rb}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})}) \parallel^2 + \sum_{i=1}^{q} \mathrm{traza}\,[\mathbf{q}_{r\,(i,i)}^{-1}(\hat{\alpha}_r, \hat{\beta})\mathbf{A}_r(i)], \quad (4.19)$$

$$\begin{aligned} \frac{2(p+q)}{\hat{\beta}} &= \parallel \mathbf{g}_{cl} - \mathbf{f}_{cl}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})} \parallel^2 + \parallel \mathbf{g}_{cr} - \mathbf{f}_{cr}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})} \parallel^2 + \parallel \mathbf{g}_{ra} - \mathbf{f}_{ra}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})} \parallel^2 \\ &+ \parallel \mathbf{g}_{rb} - \mathbf{f}_{rb}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})} \parallel^2 + \sum_{i=1}^{p} \mathrm{traza}\,[\mathbf{q}_{c\,(i,i)}^{-1}(\hat{\alpha}_c, \hat{\beta})] + \sum_{i=1}^{q} \mathrm{traza}\,[\mathbf{q}_{r\,(i,i)}^{-1}(\hat{\alpha}_r, \hat{\beta})]. \quad (4.20) \end{aligned}$$

De nuevo, las Ecs. (4.18), (4.19) y (4.20) sólo caracterizan a $\hat{\alpha}_c$, $\hat{\alpha}_r$ y $\hat{\beta}$ pero no nos proveen de una forma de estimarlos. El proceso para estimarlos se describe en la sección 4.3.3.

4.3.2 Paso de reconstrucción

Examinemos ahora el paso de reconstrucción de la imagen. Una vez estimados los valores de los hiperparámetros α_c, α_r y β, $\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)}$ se calcula diferenciando la Ec. (4.13) respecto a $\mathbf{f}_c$, obteniendo

$$2\alpha_c \mathbf{W}_c^t \mathbf{W}_c (\mathbf{f}_{cl}^{(\alpha_c,\alpha_r,\beta)} - \mathbf{f}_{cr}^{(\alpha_c,\alpha_r,\beta)}) - \beta(\mathbf{g}_{cl} - \mathbf{f}_{cl}^{(\alpha_c,\alpha_r,\beta)}) = 0, \quad (4.21)$$

$$2\alpha_c \mathbf{W}_c^t \mathbf{W}_c (\mathbf{f}_{cr}^{(\alpha_c,\alpha_r,\beta)} - \mathbf{f}_{cl}^{(\alpha_c,\alpha_r,\beta)}) - \beta(\mathbf{g}_{cr} - \mathbf{f}_{cr}^{(\alpha_c,\alpha_r,\beta)}) = 0, \quad (4.22)$$

que puede resolverse fácilmente puesto que sólo hay que invertir matrices 2×2, siendo su solución

$$\mathbf{f}_{cl}^{(\alpha_c,\alpha_r,\beta)}(i) = \frac{1}{2}[1 + \frac{\beta}{\beta + 4\alpha_c\omega_c^2(i)}]\mathbf{g}_{cl}(i) + \frac{1}{2}[1 - \frac{\beta}{\beta + 4\alpha_c\omega_c^2(i)}]\mathbf{g}_{cr}(i), \quad (4.23)$$

$$\mathbf{f}_{cr}^{(\alpha_c,\alpha_r,\beta)}(i) = \frac{1}{2}[1 - \frac{\beta}{\beta + 4\alpha_c\omega_c^2(i)}]\mathbf{g}_{cl}(i) + \frac{1}{2}[1 + \frac{\beta}{\beta + 4\alpha_c\omega_c^2(i)}]\mathbf{g}_{cr}(i), \quad (4.24)$$

para $i = 1, 2, \ldots, p$. Además, diferenciando la Ec. (4.14) respecto a $\mathbf{f}_r$ obtenemos

$$\mathbf{f}_{ra}^{(\alpha_c,\alpha_r,\beta)}(i) = \frac{1}{2}[1 + \frac{\beta}{\beta + 4\alpha_r\omega_r^2(i)}]\mathbf{g}_{ra}(i) + \frac{1}{2}[1 - \frac{\beta}{\beta + 4\alpha_r\omega_r^2(i)}]\mathbf{g}_{rb}(i), \quad (4.25)$$

$$\mathbf{f}_{rb}^{(\alpha_c,\alpha_r,\beta)}(i) = \frac{1}{2}[1 - \frac{\beta}{\beta + 4\alpha_r\omega_r^2(i)}]\mathbf{g}_{ra}(i) + \frac{1}{2}[1 + \frac{\beta}{\beta + 4\alpha_r\omega_r^2(i)}]\mathbf{g}_{rb}(i), \quad (4.26)$$

para $i = 1, 2, \ldots, q$. Finalmente, $\mathbf{f}_x^{(\alpha_c,\alpha_r,\beta)}$ se obtiene diferenciando la Ec. (4.15) respecto a $\mathbf{f}_x$, obteniendo el siguiente sistema de ecuaciones,

$$\begin{aligned}
\alpha_c\omega_{x1}^2(i)(\mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta)}(i) - \mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta)}(i)) &+ \alpha_r\omega_{x4}^2(i)(\mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta)}(i) - \mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta)}(i)) \\
&- \beta(\mathbf{g}_{al}(i) - \mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta)}(i)) = 0, \\
\alpha_c\omega_{x1}^2(i)(\mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta)}(i) - \mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta)}(i)) &+ \alpha_r\omega_{x2}^2(i)(\mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta)}(i) - \mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta)}(i)) \\
&- \beta(\mathbf{g}_{ar}(i) - \mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta)}(i)) = 0, \\
\alpha_c\omega_{x3}^2(i)(\mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta)}(i) - \mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta)}(i)) &+ \alpha_r\omega_{x2}^2(i)(\mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta)}(i) - \mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta)}(i)) \\
&- \beta(\mathbf{g}_{br}(i) - \mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta)}(i)) = 0, \\
\alpha_c\omega_{x3}^2(i)(\mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta)}(i) - \mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta)}(i)) &+ \alpha_r\omega_{x4}^2(i)(\mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta)}(i) - \mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta)}(i)) \\
&- \beta(\mathbf{g}_{bl}(i) - \mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta)}(i)) = 0, \quad (4.27)
\end{aligned}$$

que puede resolverse fácilmente puesto que sólo tenemos invertir una matriz 4×4 para obtener cada uno de los valores para los pixels de las intersecciones de cuatro bloques, ($\mathbf{f}_{al}(i)$, $\mathbf{f}_{ar}(i)$, $\mathbf{f}_{br}(i)$, $\mathbf{f}_{bl}(i)$), $i = 1, 2, \ldots, m$.

De nuevo, las Ecs. (4.23)–(4.26) pueden reescribirse como

$$\begin{aligned}
\mathbf{f}_{cl}(i) &= \frac{\mathbf{g}_{cl}(i)+\mathbf{g}_{cr}(i)}{2} + \gamma_c(i)\frac{\mathbf{g}_{cl}(i)+\mathbf{g}_{cr}(i)}{2} \\
\mathbf{f}_{cr}(i) &= \frac{\mathbf{g}_{cl}(i)+\mathbf{g}_{cr}(i)}{2} - \gamma_c(i)\frac{\mathbf{g}_{cl}(i)+\mathbf{g}_{cr}(i)}{2} \\
\mathbf{f}_{ra}(i) &= \frac{\mathbf{g}_{ra}(i)+\mathbf{g}_{rb}(i)}{2} + \gamma_r(i)\frac{\mathbf{g}_{ra}(i)-\mathbf{g}_{rb}(i)}{2} \\
\mathbf{f}_{rb}(i) &= \frac{\mathbf{g}_{ra}(i)+\mathbf{g}_{rb}(i)}{2} - \gamma_r(i)\frac{\mathbf{g}_{ra}(i)-\mathbf{g}_{rb}(i)}{2}
\end{aligned}$$

con

$$\gamma_c(i) = \frac{\beta}{\beta + 4\alpha_c\omega_c^2(i)} \qquad \text{y} \qquad \gamma_r(i) = \frac{\beta}{\beta + 4\alpha_r\omega_r^2(i)},$$

pudiéndose realizar sobre ellas el mismo análisis que se hizo en la sección 4.2.2.

4.3.3 Procedimiento iterativo para estimar los hiperparámetros y reconstruir la imagen

Una vez caracterizados los hiperparámetros óptimos y tras mostrar como se puede reconstruir la imagen dados unos hiperparámetros usaremos el siguiente algoritmo para estimar los hiperparámetros y la imagen simultáneamente, asumiendo una hiperdistribución a priori uniforme sobre los hiperparámetros.

Algoritmo 4.2 Reconstrucción adaptativa con diferentes parámetros para las filas y las columnas en el modelo de imagen y un único parámetro en el modelo de ruido.

1. *Escoger* α_c^0, α_c^0 *y* β^0.
2. *Calcular* $\mathbf{f}_c^{(\alpha_c^0,\alpha_r^0,\beta^0)}$ *y* $\mathbf{f}_r^{(\alpha_c^0,\alpha_r^0,\beta^0)}$ *a partir de las Ecs. (4.23)–(4.26).*
3. *Para* $k = 1, 2, \ldots$
   (a) *Estimar* α_c^k, α_r^k *y* β^k *sustituyendo los valores de* α_c^{k-1}, α_c^{k-1} *y* β^{k-1} *en la parte derecha de las Ecs. (4.18), (4.19) y (4.20), respectivamente.*
   (b) *Calcular* $\mathbf{f}_c^{(\alpha_c^k,\alpha_r^k,\beta^k)}$ *y* $\mathbf{f}_r^{(\alpha_c^k,\alpha_r^k,\beta^k)}$ *a partir de las Ecs. (4.23)–(4.26).*
4. *Ir al paso 3 hasta que* $\| \mathbf{f}_c^{(\alpha_c^{k-1},\alpha_r^{k-1},\beta^{k-1})} - \mathbf{f}_c^{(\alpha_c^k,\alpha_r^k,\beta^k)} \| + \| \mathbf{f}_r^{(\alpha_c^{k-1},\alpha_r^{k-1},\beta^{k-1})} - \mathbf{f}_r^{(\alpha_c^k,\alpha_r^k,\beta^k)} \|$ *sea menor que un límite establecido.*

5. *Usando* α_c^k, α_r^k *y* β^k, *calcular* $\mathbf{f}_x^{(\alpha_c^k, \alpha_r^k, \beta^k)}$ *resolviendo el sistema de ecuaciones descrito en la Ec. (4.27).*

La convergencia de este algoritmo se establece, de nuevo, viendo que corresponde a un algoritmo EM, donde los datos incompletos son las observaciones, $\mathbf{g}$, y los completos son las observaciones, $\mathbf{g}$, y la reconstrucción, $\mathbf{f}$, es decir, $\mathbf{z}$, los datos completos se definen como $\mathbf{z}^t = (\mathbf{f}^t, \mathbf{g}^t)$ y los datos completos e incompletos se relacionan como

$$\mathbf{g} = (0 \quad \mathbf{I})\,\mathbf{z},$$

donde $\mathbf{I}$ y 0 son matrices identidad y cero, respectivamente. Los detalles pueden consultarse en [55, 79].

4.4 Reconstrucción adaptativa con parámetros distintos para filas y columnas en modelos de imagen y ruido

En esta sección se usa el modelo de imagen adaptativo con dos parámetros, uno para controlar la suavidad de las filas y otro de las columnas, definido en la Ec. (3.26) y el modelo de ruido, también con dos parámetros diferentes para para las filas y las columnas, definido en la Ec. (3.9). Por tanto, el modelo que vamos a usar se define como

$$\mathrm{p}(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta_c, \beta_r) \propto \alpha_c^{\frac{p}{2}} \alpha_r^{\frac{q}{2}} F(\alpha_c, \alpha_r) \beta_c^{p+m} \beta_r^{q+m} \exp\left\{-C(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta_c, \beta_r)\right\},$$

donde $C(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta_c, \beta_r)$ se define como

$$\begin{aligned} C(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta_c, \beta_r) &= C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta_c) + C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta_r) \\ &+ C_x(\mathbf{f}_x, \mathbf{g}_x \mid \alpha_c, \alpha_r, \beta_c, \beta_r), \end{aligned} \tag{4.28}$$

con

$$C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta_c) = P_c(\mathbf{f}_c \mid \alpha_c) + N_c(\mathbf{g}_c \mid \mathbf{f}_c, \beta_c), \tag{4.29}$$

$$C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta_r) = P_r(\mathbf{f}_r \mid \alpha_r) + N_r(\mathbf{g}_r \mid \mathbf{f}_r, \beta_r), \tag{4.30}$$

$$C_x(\mathbf{f}_x, \mathbf{g}_x \mid \alpha_c, \beta_c, \alpha_r, \beta_r) = P_x(\mathbf{f}_x \mid \alpha_c, \alpha_r) + N_x(\mathbf{g}_x \mid \mathbf{f}_x, \beta_c, \beta_r), \tag{4.31}$$

$P_u(\cdot)$ y $N_u(\cdot)$, $u \in \{c, r, x\}$, definidas en las Ecs. (3.29), (3.30) y (3.31) y Ecs. (3.11), (3.12) y (3.13), respectivamente, y $F(\alpha_c, \alpha_r)$ definido en la Ec. (3.28).

Describamos el proceso de estimación de los hiperparámetros y de reconstrucción de la imagen de forma detallada.

Recordemos que en la aproximación de la evidencia $\hat{\alpha}_c$, $\hat{\alpha}_r$, $\hat{\beta}_c$ y $\hat{\beta}_r$, para este caso, se seleccionan como

$$\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta}_c, \hat{\beta}_r = \arg \max_{\alpha_c, \alpha_r, \beta_c, \beta_r} \mathrm{p}(\alpha_c, \alpha_r, \beta_c, \beta_r \mid \mathbf{g}),$$

y entonces la reconstrucción, $\mathbf{f}^{(\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta}_c, \hat{\beta}_r)}$, se obtiene como

$$\mathbf{f}^{(\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta}_c, \hat{\beta}_r)} = \arg \min_{\mathbf{f}} C(\mathbf{f}, \mathbf{g} \mid \hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta}_c, \hat{\beta}_r). \tag{4.32}$$

El proceso de estimación de los hiperparámetros se basa, pues, en maximizar en α_c, α_r, β_c y β_r, $\mathrm{p}(\alpha_c, \alpha_r, \beta_c, \beta_r \mid \mathbf{g})$, definida como

$$\begin{aligned}
&\mathrm{p}(\alpha_c, \alpha_r, \beta_c, \beta_c \mid \mathbf{g}) \propto [\text{para una distrib. uniforme}] \int_{\mathbf{f}} \mathrm{p}(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta_c, \beta_r) d\mathbf{f} \\
&= \int_{\mathbf{f}_c} \mathrm{p}(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta_c)\, d\mathbf{f}_c \int_{\mathbf{f}_r} \mathrm{p}(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta_r)\, d\mathbf{f}_r \int_{\mathbf{f}_x} \mathrm{p}(\mathbf{f}_x, \mathbf{g}_x \mid \alpha_c, \alpha_r, \beta_c, \beta_r)\, d\mathbf{f}_x \\
&= \mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta_c)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta_r)\mathrm{p}(\mathbf{g}_x \mid \alpha_c, \alpha_r, \beta_c, \beta_r),
\end{aligned}$$

y nótese que incluir $\mathrm{p}(\mathbf{g}_x \mid \alpha_c, \alpha_r, \beta_c, \beta_r)$ en el proceso de estimación de α_c, α_r, β_c y β_r, presenta los mismos problemas enumerados en la sección 4.3, agravados incluso por el hecho de tener cuatro parámetros que estimar en lugar de tres.

Usando la misma solución que en la sección anterior, procederemos a resolver el problema de la estimación usando los siguientes pasos:

1. Estimar α_c, α_r, β_c y β_r mediante

$$\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta_c)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta_r), \tag{4.33}$$

   como se describirá en la sección 4.4.1.

2. Entonces usar $\hat{\alpha}_c$, $\hat{\alpha}_r$, $\hat{\beta}_c$ y $\hat{\beta}_r$ en la Ec. (4.32) para obtener la imagen reconstruida (ver sección 4.4.2).

El algoritmo 4.3, en la sección 4.4.3, lleva a cabo el problema de optimización de la Ec. (4.33) y la estimación de la imagen original usando la Ec. (4.32).

4.4.1 Paso de estimación de los hiperparámetros

Para caracterizar la estimación de los parámetros procedamos a calcular $\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta_c)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta_c)$ de la Ec. (4.33), teniendo en cuenta que

$$\begin{aligned}\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta_c)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta_r) &= \alpha_c^{\frac{p}{2}} \beta_c^p \alpha_r^{\frac{q}{2}} \beta_r^q \\ &\times \int_{\mathbf{f}_c, \mathbf{f}_r} \exp[-C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta_c) - C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta_r)]\, d\mathbf{f}_c\, d\mathbf{f}_r.\end{aligned}$$

Fijemos α_c, α_r, β_c y β_r y expandamos $C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta_c) + C_r(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta_r)$ alrededor de $\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}$ y $\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}$. Entonces tenemos,

$$\begin{aligned}&C_c(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta_c) + C_c(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta_r) = \\ &= C_c(\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}, \mathbf{g}_c \mid \alpha_c, \beta_c) + C_r(\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}, \mathbf{g}_r \mid \alpha_r, \beta_r) \\ &+ \frac{1}{2}(\mathbf{f}_c - \mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)})^t \mathbf{Q}_c(\alpha_c, \beta_c)(\mathbf{f}_c - \mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}) \\ &+ \frac{1}{2}(\mathbf{f}_r - \mathbf{f}_r^{(\alpha_c,\alpha_r,\beta_c,\beta_r)})^t \mathbf{Q}_r(\alpha_r, \beta_r)(\mathbf{f}_r - \mathbf{f}_r^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}),\end{aligned}$$

donde $\mathbf{Q}_c(\alpha_c, \beta_c)$ es una matriz diagonal por bloques cuyas matrices diagonales 2×2 son $\mathbf{q}_{c(i,i)}(\alpha_c, \beta_c) = \alpha_c \mathbf{A}_c(i) + \beta_c \mathbf{I}_{2\times 2}$ con

$$\mathbf{A}_c(i) = \begin{bmatrix} 2\omega_c^2(i) & -2\omega_c^2(i) \\ -2\omega_c^2(i) & 2\omega_c^2(i) \end{bmatrix},$$

$i = 1, 2, \ldots, p$, y $\mathbf{Q}_r(\alpha_r, \beta_r)$ es una matriz diagonal por bloques cuyas matrices diagonales 2×2 son $\mathbf{q}_{r(i,i)}(\alpha_r, \beta_r) = \alpha_r \mathbf{A}_r(i) + \beta_r \mathbf{I}_{2\times 2}$ con

$$\mathbf{A}_r(i) = \begin{bmatrix} 2\omega_r^2(i) & -2\omega_r^2(i) \\ -2\omega_r^2(i) & 2\omega_r^2(i) \end{bmatrix},$$

$i = 1, 2, \ldots, q$.

Entonces tenemos

$$\begin{aligned}&\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta_c)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta_r) \propto \\ &\quad \alpha_c^{\frac{p}{2}} \beta_c^p \alpha_r^{\frac{q}{2}} \beta_r^q \exp\{-C_c(\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}, \mathbf{g}_c \mid \alpha_c, \beta_c)\} \exp\{-C_r(\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}, \mathbf{g}_r \mid \alpha_r, \beta_r)\} \\ &\times \int_{\mathbf{f}_c} \exp\left\{-\frac{1}{2}(\mathbf{f}_c - \mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)})^t \mathbf{Q}_c(\alpha_c, \beta_c)(\mathbf{f}_c - \mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)})\right\} d\mathbf{f}_c \\ &\times \int_{\mathbf{f}_r} \exp\left\{-\frac{1}{2}(\mathbf{f}_r - \mathbf{f}_r^{(\alpha_c,\alpha_r,\beta_c,\beta_r)})^t \mathbf{Q}_r(\alpha_r, \beta_r)(\mathbf{f}_r - \mathbf{f}_r^{(\alpha_c,\alpha_r,\beta_c,\beta_r)})\right\} d\mathbf{f}_r \\ &= \alpha_c^{\frac{p}{2}} \beta_c^p \alpha_r^{\frac{q}{2}} \beta_r^q \exp\{-C_c(\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}, \mathbf{g}_c \mid \alpha_c, \beta_c)\} \exp\{-C_r(\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}, \mathbf{g}_r \mid \alpha_r, \beta_r)\} \\ &\times [\det \mathbf{Q}_c(\alpha_c, \beta_c)]^{-\frac{1}{2}} [\det \mathbf{Q}_r(\alpha_r, \beta_r)]^{-\frac{1}{2}}.\end{aligned}$$

Entonces, diferenciando $-\log(\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta_c)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta_r))$ respecto a α_c, α_r, β_c y β_r tenemos como $\hat{\alpha}_c$, $\hat{\alpha}_r$, $\hat{\beta}_c$ y $\hat{\beta}_r$,

$$\frac{p}{\hat{\alpha}_c} = 2 \parallel \mathbf{W}_c(\mathbf{f}_{cl}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta}_c,\hat{\beta}_r)} - \mathbf{f}_{cr}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta}_c,\hat{\beta}_r)}) \parallel^2 + \sum_{i=1}^{p} \mathrm{traza}\,[\mathbf{q}_{c(i,i)}^{-1}(\hat{\alpha}_c, \hat{\beta}_c)\mathbf{A}_c(i)], \quad (4.34)$$

$$\frac{q}{\hat{\alpha}_r} = 2 \parallel \mathbf{W}_r(\mathbf{f}_{ra}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta}_c,\hat{\beta}_r)} - \mathbf{f}_{rb}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta}_c,\hat{\beta}_r)}) \parallel^2 + \sum_{i=1}^{q} \mathrm{traza}\,[\mathbf{q}_{r(i,i)}^{-1}(\hat{\alpha}_r, \hat{\beta}_r)\mathbf{A}_r(i)], \quad (4.35)$$

$$\frac{2p}{\hat{\beta}_c} = \parallel \mathbf{g}_{cl} - \mathbf{f}_{cl}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta}_c,\hat{\beta}_r)} \parallel^2 + \parallel \mathbf{g}_{cr} - \mathbf{f}_{cr}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta}_c,\hat{\beta}_r)} \parallel^2 + \sum_{i=1}^{p} \mathrm{traza}\,[\mathbf{q}_{c(i,i)}^{-1}(\hat{\alpha}_c, \hat{\beta}_c)], \quad (4.36)$$

$$\frac{2q}{\hat{\beta}_r} = \parallel \mathbf{g}_{ra} - \mathbf{f}_{ra}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta}_c,\hat{\beta}_r)} \parallel^2 + \parallel \mathbf{g}_{rb} - \mathbf{f}_{rb}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta}_c,\hat{\beta}_r)} \parallel^2 + \sum_{i=1}^{q} \mathrm{traza}\,[\mathbf{q}_{r(i,i)}^{-1}(\hat{\alpha}_r, \hat{\beta}_r)], \quad (4.37)$$

ecuaciones que caracterizan a α_c, α_r, β_c y β_r.

4.4.2 Paso de reconstrucción

El proceso de reconstrucción de la imagen, dados α_c, α_r, β_c y β_r, se realiza como en las secciones anteriores. Así, $\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}$ se calcula diferenciando la Ec. (4.29) respecto a $\mathbf{f}_c$, obteniendo la solución

$$\mathbf{f}_{cl}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) = \frac{1}{2}\left(1 + \frac{\beta_c}{\beta_c + 4\alpha_c\omega_c^2(i)}\right)\mathbf{g}_{cl}(i) + \frac{1}{2}\left(1 - \frac{\beta_c}{\beta_c + 4\alpha_c\omega_c^2(i)}\right)\mathbf{g}_{cr}(i), \quad (4.38)$$

$$\mathbf{f}_{cr}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) = \frac{1}{2}\left(1 - \frac{\beta_c}{\beta_c + 4\alpha_c\omega_c^2(i)}\right)\mathbf{g}_{cl}(i) + \frac{1}{2}\left(1 + \frac{\beta_c}{\beta_c + 4\alpha_c\omega_c^2(i)}\right)\mathbf{g}_{cr}(i), \quad (4.39)$$

para $i = 1, 2, \ldots, p$. Además, diferenciando la Ec. (4.30) respecto a $\mathbf{f}_r$ tenemos

$$\mathbf{f}_{ra}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) = \frac{1}{2}\left(1 + \frac{\beta_r}{\beta_r + 4\alpha_r\omega_r^2(i)}\right)\mathbf{g}_{ra}(i) + \frac{1}{2}\left(1 - \frac{\beta_r}{\beta_r + 4\alpha_r\omega_r^2(i)}\right)\mathbf{g}_{rb}(i), \quad (4.40)$$

$$\mathbf{f}_{rb}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) = \frac{1}{2}\left(1 - \frac{\beta_r}{\beta_r + 4\alpha_r\omega_r^2(i)}\right)\mathbf{g}_{ra}(i) + \frac{1}{2}\left(1 + \frac{\beta_r}{\beta_r + 4\alpha_r\omega_r^2(i)}\right)\mathbf{g}_{rb}(i), \quad (4.41)$$

para $i = 1, 2, \ldots, q$. Finalmente, $\mathbf{f}_x^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}$ se obtiene diferenciando la Ec. (4.31) respecto a $\mathbf{f}_x$, obteniendo el siguiente sistema de ecuaciones,

$$\begin{aligned}
\alpha_c\omega_{x1}^2(i)(\mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) - \mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) &+ \alpha_r\omega_{x4}^2(i)(\mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) - \mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) \\
&- \beta_c(\mathbf{g}_{al}(i) - \mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) = 0, \\
\alpha_c\omega_{x1}^2(i)(\mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) - \mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) &+ \alpha_r\omega_{x2}^2(i)(\mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) - \mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) \\
&- \beta_r(\mathbf{g}_{ar}(i) - \mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) = 0, \\
\alpha_c\omega_{x3}^2(i)(\mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) - \mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) &+ \alpha_r\omega_{x2}^2(i)(\mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)(i)} - \mathbf{f}_{ar}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i))
\end{aligned}$$

$$
\begin{aligned}
& & - \quad & \beta_c(\mathbf{g}_{br}(i) - \mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) = 0, \\
\alpha_c\omega_{x3}^2(i)(\mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i) - \mathbf{f}_{br}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) & + & \alpha_r\omega_{x4}^2(i)(\mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)(i)} - \mathbf{f}_{al}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) \\
& & - \quad & \beta_r(\mathbf{g}_{bl}(i) - \mathbf{f}_{bl}^{(\alpha_c,\alpha_r,\beta_c,\beta_r)}(i)) = 0, \qquad (4.42)
\end{aligned}
$$

que puede resolverse fácilmente puesto que sólo tenemos invertir una matriz 4×4 para obtener cada uno de los valores para los pixels de las intersecciones de cuatro bloques, ($\mathbf{f}_{al}(i)$, $\mathbf{f}_{ar}(i)$, $\mathbf{f}_{br}(i)$, $\mathbf{f}_{bl}(i)$), $i = 1, 2, \ldots, m$.

4.4.3 Procedimiento iterativo para estimar los hiperparámetros y reconstruir la imagen

Una vez caracterizados los hiperparámetros óptimos y tras mostrar como se puede reconstruir la imagen dados unos hiperparámetros usaremos el siguiente algoritmo para estimar los hiperparámetros y la imagen simultáneamente.

Algoritmo 4.3 Reconstrucción adaptativa con diferentes parámetros para las filas y las columnas en los modelos de imagen y ruido.

1. *Escoger* α_c^0, α_c^0, β_c^0 *y* β_r^0.
2. *Calcular* $\mathbf{f}_c^{(\alpha_c^0,\alpha_r^0,\beta_c^0,\beta_r^0)}$ *y* $\mathbf{f}_r^{(\alpha_c^0,\alpha_r^0,\beta_c^0,\beta_r^0)}$ *a partir de las Ecs. (4.38)-(4.41).*
3. *Para* $k = 1, 2, \ldots$
   (a) *Estimar* α_c^k, α_r^k, β_c^k *y* β_r^k *sustituyendo los valores de* α_c^{k-1}, α_c^{k-1}, β_c^{k-1} *y* β_r^{k-1} *en la parte derecha de las Ecs. (4.34), (4.35), (4.36) y (4.37), respectivamente.*
   (b) *Calcular* $\mathbf{f}_c^{(\alpha_c^k,\alpha_r^k,\beta_c^k,\beta_r^k)}$ *y* $\mathbf{f}_r^{(\alpha_c^k,\alpha_r^k,\beta_c^k,\beta_r^k)}$ *a partir de las Ecs. (4.38)-(4.41).*
4. *Ir al paso 3 hasta que* $\| \mathbf{f}_c^{(\alpha_c^{k-1},\alpha_r^{k-1},\beta_c^{k-1},\beta_r^{k-1})} - \mathbf{f}_c^{(\alpha_c^k,\alpha_r^k,\beta_c^k,\beta_r^k)} \| + \| \mathbf{f}_r^{(\alpha_c^{k-1},\alpha_r^{k-1},\beta_c^{k-1},\beta_r^{k-1})} - \mathbf{f}_r^{(\alpha_c^k,\alpha_r^k,\beta_c^k,\beta_r^k)} \|$ *sea menor que un límite establecido.*
5. *Usando* α_c^k, α_r^k, β_c^k *y* β_r^k, *calcular* $\mathbf{f}_x^{(\alpha_c^k,\alpha_r^k,\beta_c^k,\beta_r^k)}$ *resolviendo el sistema de ecuaciones descrito en la Ec. (4.42).*

La prueba de la convergencia de este algoritmo se basa de nuevo en el hecho de que es un algoritmo EM (ver [58, 79]).

4.5 Resultados experimentales

Habiendo expuesto tres métodos para la reconstrucción de la imagen y estimación simultánea de los parámetros, a saber:

1. reconstrucción adaptativa con los mismos parámetros para filas y columnas en modelos de imagen y ruido (algoritmo 4.1),

2. reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido (algoritmo 4.2), y

3. reconstrucción adaptativa con parámetros distintos para filas y columnas en modelos de imagen y ruido (algoritmo 4.3),

en esta sección estudiaremos su comportamiento sobre un conjunto de imágenes reales.

4.5.1 Imágenes utilizadas y criterios de bondad

Como conjunto de prueba para los experimentos usaremos doce imágenes frecuentemente utilizadas como banco de pruebas en problemas de compresión de imágenes. Estas imágenes se han agrupado en tres tipos de semejantes características en cuanto a su actividad espacial. Así se ha escogido un primer grupo de imágenes (que llamaremos G1) con una actividad espacial alta, es decir, con un gran número de pequeños detalles, mostrado en la figura 4.1, un segundo grupo (G2) con una actividad espacial media, mostrado en la figura 4.2 y un tercer grupo, G3, de imágenes suaves sin gran cantidad de detalles, mostrado en la figura 4.3. Todas estas imágenes están digitalizadas a 8 bits por pixel.

Teniendo en cuenta que los experimentos realizados presentan resultados similares para las imágenes en los grupos G1, G2 y G3, respectivamente, estos resultados se presentarán para una imagen representativa de cada grupo. En concreto, se usará la imagen *boats* como representante del grupo G1, la imagen *couple* como representante del grupo G2 y la imagen *lena* como representante del grupo G3.

El lector puede consultar los resultados completos de todas las imágenes a través de Internet en el siguiente URL

`ftp://nesos.ugr.es/pub/tesis/resultados/decodificador`

Para determinar la bondad de los resultados obtenidos será necesario usar una medida objetiva de la distancia entre la imagen original y la imagen reconstruida. Esta medida será el pico de la relación señal-ruido (*peak signal-to-noise ratio* o $PSNR$) que, para dos imágenes $\mathbf{f}$ y $\mathbf{g}$ de tamaño $M \times N$ con un rango $[0, 255]$, se define como

$$PSNR = 10 \log_{10} \left[\frac{M \times N \times 255^2}{\| \mathbf{g} - \mathbf{f} \|^2} \right]. \tag{4.43}$$

También se usará, como medida de la bondad subjetiva, la calidad visual de la imagen.

4.5.2 Niveles de compresión utilizados

Cada una de las imágenes del conjunto de prueba se comprimen mediante la implementación de JPEG llevada a cabo por *Independent JPEG Group, IJG*, a diferentes razones de compresión. Como muestra en este estudio hemos utilizado tres razones de compresión diferentes: una razón alta, en la que la imagen presenta una gran prominencia del efecto de bloques, una razón de compresión media, y una razón de compresión baja que prácticamente no produce este efecto bloque. Usaremos, para los experimentos, imágenes comprimidas a, aproximadamente, 0.20bpp, 0.30bpp y 0.50bpp, o, equivalentemente, comprimidas a, aproximadamente, 39:1, 26:1 y 16:1, correspondiéndose estas razones a una compresión alta, media y baja compresión, respectivamente.

4.5.3 Matriz de pesos

Antes de aplicar los métodos de reconstrucción sobre esta imágenes es necesario definir la forma de calcular la matriz de pesos $\mathbf{W}$, definida en la Ec. (3.20), que pondera cada diferencia de pixels en los diferentes modelos a priori adaptativos. El valor de cada uno de los elementos de la matriz, $\omega(i)$, se define como

$$\omega(i) = \log \left(1 + \frac{\sqrt{\mu(i)}}{1 + \sigma(i)} \right),$$

tal como se describe en la Ec. (B.1) del Apéndice B. Véase este apéndice para una mayor información sobre otras posibilidades de elección de los pesos y el cálculo de los valores de $\mu(i)$ y $\sigma(i)$. Después de escalarlos adecuadamente, en la figura 4.4 se muestran ejemplos de los pesos las imágenes de muestra del conjunto de prueba. En las zonas brillantes de esta imagen los bloques son más visibles que en las zonas oscuras.

4.5.4 Criterio de parada

A cada una de las imágenes del conjunto de prueba se les aplicó cada uno de los métodos de reconstrucción descritos en este capítulo, es decir, algoritmo 4.1, algoritmo 4.2 y algoritmo 4.3. El criterio de parada fue, en todos los casos, que la diferencia de las reconstrucciones (en norma) entre dos iteraciones consecutivas fuese menor que 0.1, es decir,

$$\| \mathbf{f}^{(\alpha^{k-1},\beta^{k-1})} - \mathbf{f}^{(\alpha^{k},\beta^{k})} \| < 0.1, \tag{4.44}$$

donde α y β representa el vector de hiperparámetros correspondiente al modelo de imagen y ruido respectivamente.

4.5.5 Análisis visual

Las figuras 4.5, 4.7 y 4.9 muestran la zona central 256×256 de cada imagen original junto al resultado de la compresión mediante la implementación de JPEG llevada a cabo por *Independent JPEG Group, IJG* para las razones de compresión alta, media y baja, y las figuras 4.6, 4.8 y 4.10 muestran un ejemplo de reconstrucción de la zona central de las imágenes con cada uno de los tres métodos de reconstrucción. Por economía de espacio no se incluyen todas imágenes resultantes, resumiéndose los resultados en las figuras 4.11, 4.12 y 4.13 donde se muestra, para cada razón de compresión de la imagen (alta, media y baja), el pico de la razón señal-ruido de la imagen comprimida y de la reconstrucción mediante cada uno de los algoritmos propuestos.

En la totalidad de los casos se mejora en la calidad visual con respecto a la imagen comprimida si bien es interesante ver las diferencias entre los diferentes métodos de reconstrucción.

Es preciso observar que todos hacen un buen trabajo mejorando la calidad visual de las imágenes y es difícil descubrir diferencias entre ellos en muchas de las imágenes de prueba. Sin embargo, en muchas de las imágenes, especialmente aquellas que poseen una mayor presencia de elementos en una dirección, queda patente que el método de reconstrucción adaptativa con los mismos parámetros para filas y columnas en modelos de imagen y ruido (algoritmo 4.1) produce un mayor emborronamiento en una de las direcciones de la imagen mientras que en la otra no llega a eliminar suficiente los artificios de los bloques. Obsérvese, por ejemplo, la imagen *couple* donde el brazo del hombre, entre otras zonas, aún presenta el artificio de los bloques en la dirección horizontal mientras que en el resto de las imágenes éste ha quedado mermado. Esto es debido a que, al tener un único parámetro para controlar la suavidad en filas

y columnas, éste sobreestima la suavidad en una de las direcciones mientras que la subestima en la otra.

Entre los otros dos métodos de reconstrucción las diferencias son menos visibles si bien el método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en el modelo de imagen y el mismo parámetro para filas y columnas en el modelo de ruido (algoritmo 4.2) suele ser algo mejor, especialmente para razones de compresión más altos (0.20bpp y 0.30bpp).

4.5.6 Análisis del *PSNR*

En cuanto a los resultados respecto al pico de la relación señal-ruido, casi en la totalidad de las reconstrucciones aumenta el valor de esta medida (ver figuras 4.11, 4.12 y 4.13), con la excepción de la imagen *couple* comprimida a 0.50bpp en la que se observa una ligera disminución en el $PSNR$ aunque la calidad visual presente una ligera mejora. Esto es debido, en parte, a que el $PSNR$ es una medida de distancia entre imágenes que no siempre equivale a la percepción visual para la calidad de la imagen, como se observa en algunos de los experimentos anteriores en los que, aunque los artificios de los bloques se reducen visiblemente, la diferencia de $PSNR$ no refleja tan claramente esta mejora. Sin embargo, esta medida es la más usada en la comunidad científica para medir la mejora producida por los métodos de reconstrucción.

El método de reconstrucción adaptativa con los mismos parámetros para filas y columnas en modelos de imagen y ruido (algoritmo 4.1) suele producir una menor mejora en el $PSNR$ que los otros métodos si bien, para una mayoría de las imágenes comprimidas a 0.50bpp, donde la imagen comprimida y la real son bastantes parecidas este método suele presentar un mejor comportamiento que los otros dos métodos a nivel de pico de la relación señal ruido puesto que puede tener en cuenta los pixels de las intersecciones de cuatro bloques para realizar la estimación de los parámetros, lo que le aporta al método una valiosa información extra.

Entre los otros dos métodos de reconstrucción, las diferencias son bastante menos visibles, tanto a nivel de mejora de calidad visual como de $PSNR$, si bien el método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido (algoritmo 4.2) suele ser algo mejor, especialmente para razones de compresión más altos (0.20bpp y 0.30bpp) tanto en calidad visual como en $PSNR$. La mejora del método de reconstrucción adaptativa con parámetros distintos para filas y columnas en modelos de imagen y ruido (algoritmo 4.3) respecto al método

de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido, para algunas imágenes, conforme baja la razón de compresión podría ser debida a que el ruido producido por la cuantificación puede ser un poco diferente para las filas y las columnas si la cuantificación no es muy pronunciada, especialmente en imágenes con muchos detalles, pudiendo este método captarlo algo mejor.

No obstante, en general el método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido (algoritmo 4.2) presenta un mejor comportamiento para la mayoría de las imágenes tanto a nivel visual como de $PSNR$.

4.5.7 Análisis del número de iteraciones

Respecto al número de iteraciones necesarias para que el método converja (según el criterio establecido en la Ec. (4.44)), en las tablas 4.1, 4.2 y 4.3 se muestra el resumen de resultados para cada una de las imágenes de muestra. A la vista de estos resultados, está claro que las más de 300 iteraciones necesarias en algunos casos hace que los diferentes métodos sean lentos.

Sin embargo, en las primeras iteraciones el método converge rápidamente, como se puede observar en la figura 4.14, y es en estas iteraciones cuando se consigue la mayor parte de la ganancia en calidad visual en la reconstrucción por lo que, a la hora de implementarlo en un sistema real se puede ver que con las primeras, digamos 4, iteraciones es suficiente para conseguir una calidad visual mucho mejor que la obtenida con la imagen comprimida como se puede observar en las figuras 4.15, 4.16 y 4.17.

Las figuras 4.18, 4.19 y 4.20 muestran la comparación de resultados en $PSNR$ entre los métodos tras ejecutar 4 iteraciones y ejecutarlos hasta la condición de convergencia antes indicada obteniendo unos resultados con 4 iteraciones bastante cercanos a los resultados a convergencia. Se puede observar que, en algunos casos, el valor del $PSNR$ con 4 iteraciones es mayor que el obtenido a la convergencia del método. Esto es debido a que el método no busca una mejora del $PSNR$ sino un compromiso entre una imagen sin fronteras visuales en las fronteras de los bloques y la imagen recibida desde el codificador. Este compromiso no siempre obtiene valores máximos de $PSNR$, pudiendo producir en las primeras iteraciones una imagen en la que los valores de las fronteras se aproxime a la media entre los pixels a cada lado de la frontera, produciendo posiblemente una imagen con un $PSNR$ más alto, pero

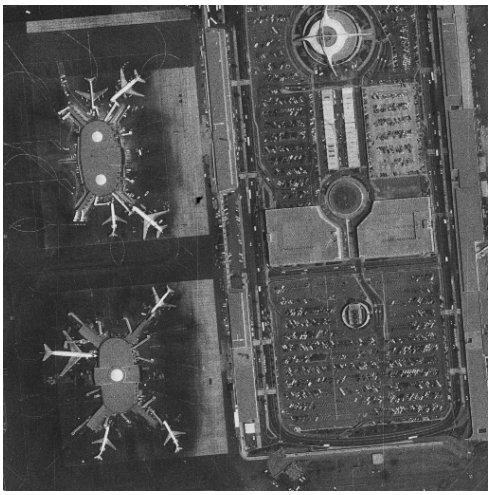
borrosa en las zonas con detalles, e ir ajustando los parámetros en las sucesivas iteraciones hasta una imagen suave en las fronteras y que preserve los detalles.

4.5.8 Análisis de perfiles

Por último, en las figuras 4.21 a 4.23 se muestran perfiles sobre la imagen comprimida y la reconstrucción con el Algoritmo 4.2. Como se puede observar el método de reconstrucción suaviza las fronteras artificiales de los bordes de los bloques en función de la actividad espacial de la zona en la que éste está. Así, en las zonas con escasa actividad espacial (ver figura 4.23) se produce un escalonamiento de la frontera haciéndola menos visible mientras que en las zonas con un gran nivel de detalle (ver figura 4.21 y parte central de la figura 4.22) apenas modifica la frontera, preservando de esta forma los detalles presentes en la imagen.

A la vista de los análisis realizados podemos concluir que el método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido presenta un mejor comportamiento tanto a nivel de calidad visual de las reconstrucciones como de aumento del $PSNR$ para la gran mayoría de las imágenes de prueba utilizadas, siendo éste método el que usaremos y desarrollaremos en los siguientes capítulos.

boats 720×576



lax 512×512

goldhill 720×576

harbour 512×512

Figura 4.1: Conjunto de imágenes de prueba con actividad espacial alta (G1). Bajo cada imagen se muestra su nombre y tamaño.

$airplane\ 512 \times 512$

$couple\ 512 \times 512$

$crowd\ 512 \times 512$

$man\ 512 \times 512$

Figura 4.2: Conjunto de imágenes de prueba con actividad espacial media (G2). Bajo cada imagen se muestra su nombre y tamaño.

lena 512×512 *peppers* 512×512

woman1 512×512 *woman2* 512×512

Figura 4.3: Conjunto de imágenes de prueba con actividad espacial baja (G3). Bajo cada imagen se muestra su nombre y tamaño.

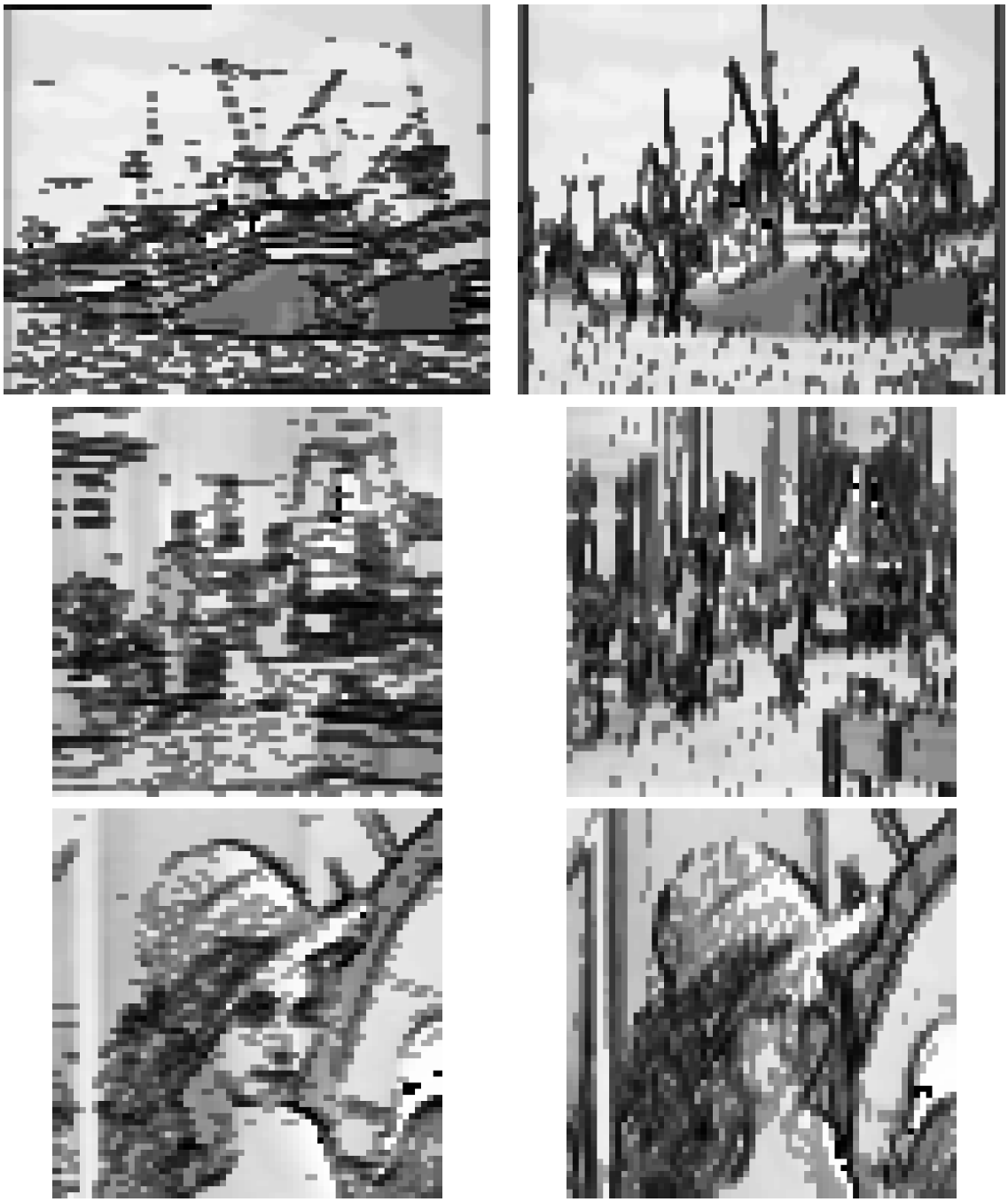


Figura 4.4: Matrices de pesos verticales y horizontales para las imágenes de muestra. De arriba a abajo: Para la imagen $boats$. Para la imagen $couple$. Para la imagen $lena$.

Figura 4.5: De izquierda a derecha y de arriba a abajo: Imagen $boats$ original. Imagen comprimida a 0.20bpp. Imagen comprimida a 0.30bpp. Imagen comprimida a 0.50bpp.

Figura 4.6: De izquierda a derecha y de arriba a abajo: Imagen $boats$ comprimida a 0.20bpp. Reconstrucción con el Algoritmo 4.1. Reconstrucción con el Algoritmo 4.2. Reconstrucción con el Algoritmo 4.3.

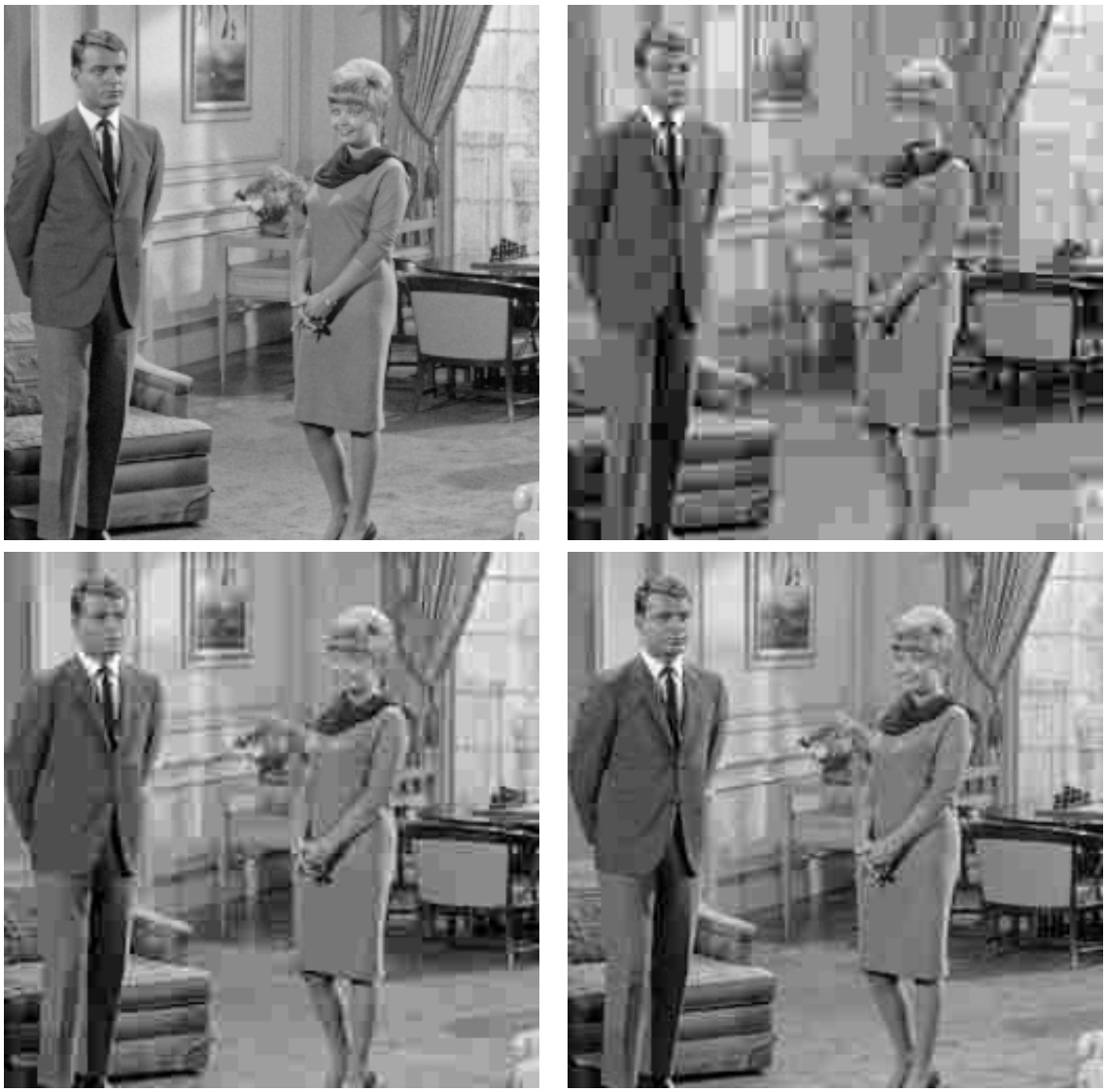


Figura 4.7: De izquierda a derecha y de arriba a abajo: Imagen $couple$ original. Imagen comprimida a 0.20bpp. Imagen comprimida a 0.30bpp. Imagen comprimida a 0.50bpp.

Figura 4.8: De izquierda a derecha y de arriba a abajo: Imagen $couple$ comprimida a 0.30bpp. Reconstrucción con el Algoritmo 4.1. Reconstrucción con el Algoritmo 4.2. Reconstrucción con el Algoritmo 4.3.

Figura 4.9: De izquierda a derecha y de arriba a abajo: Imagen $lena$ original. Imagen comprimida a 0.20bpp. Imagen comprimida a 0.30bpp. Imagen comprimida a 0.50bpp.

Figura 4.10: De izquierda a derecha y de arriba a abajo: Imagen $lena$ comprimida a 0.30bpp. Reconstrucción con el Algoritmo 4.1. Reconstrucción con el Algoritmo 4.2. Reconstrucción con el Algoritmo 4.3.

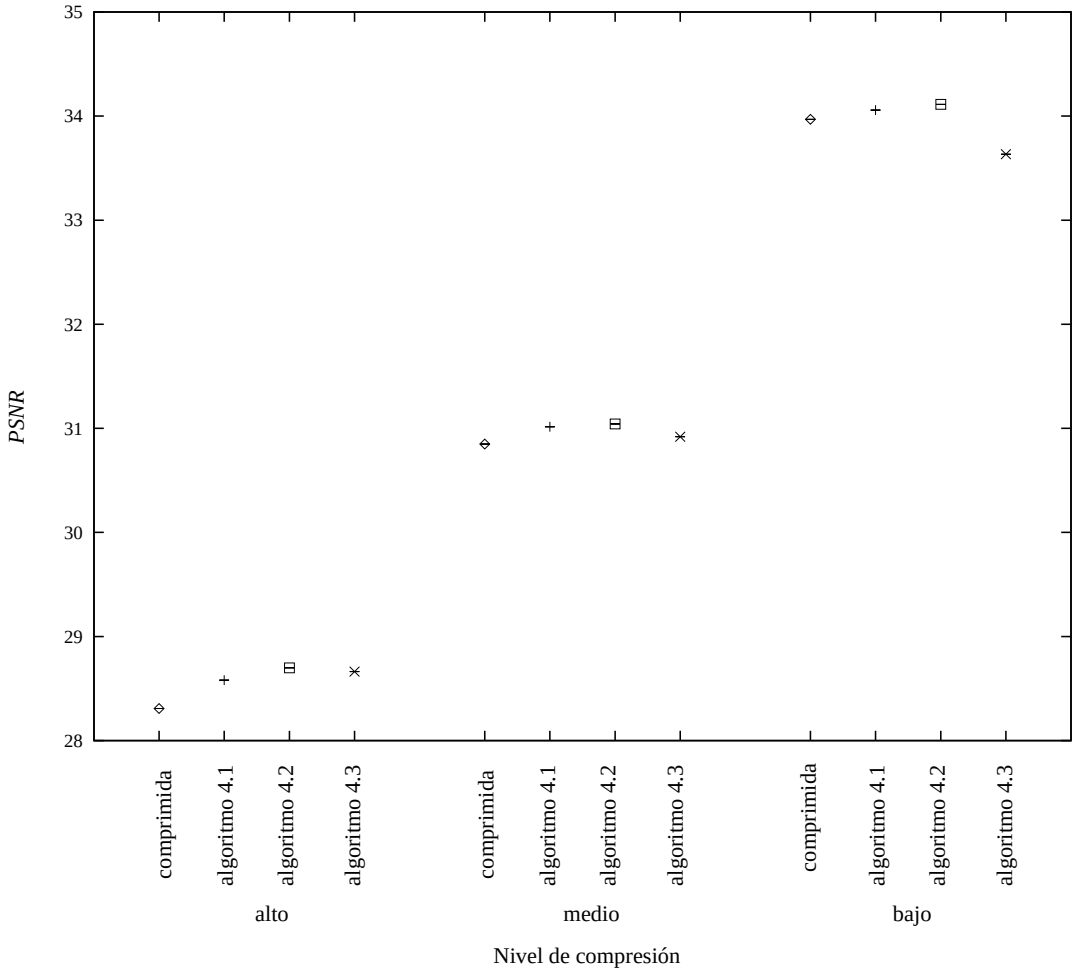


Figura 4.11: Resultados de la reconstrucción con los diferentes algoritmos, expresados en $PSNR$, para la imagen $boats$ a diferentes razones de compresión.

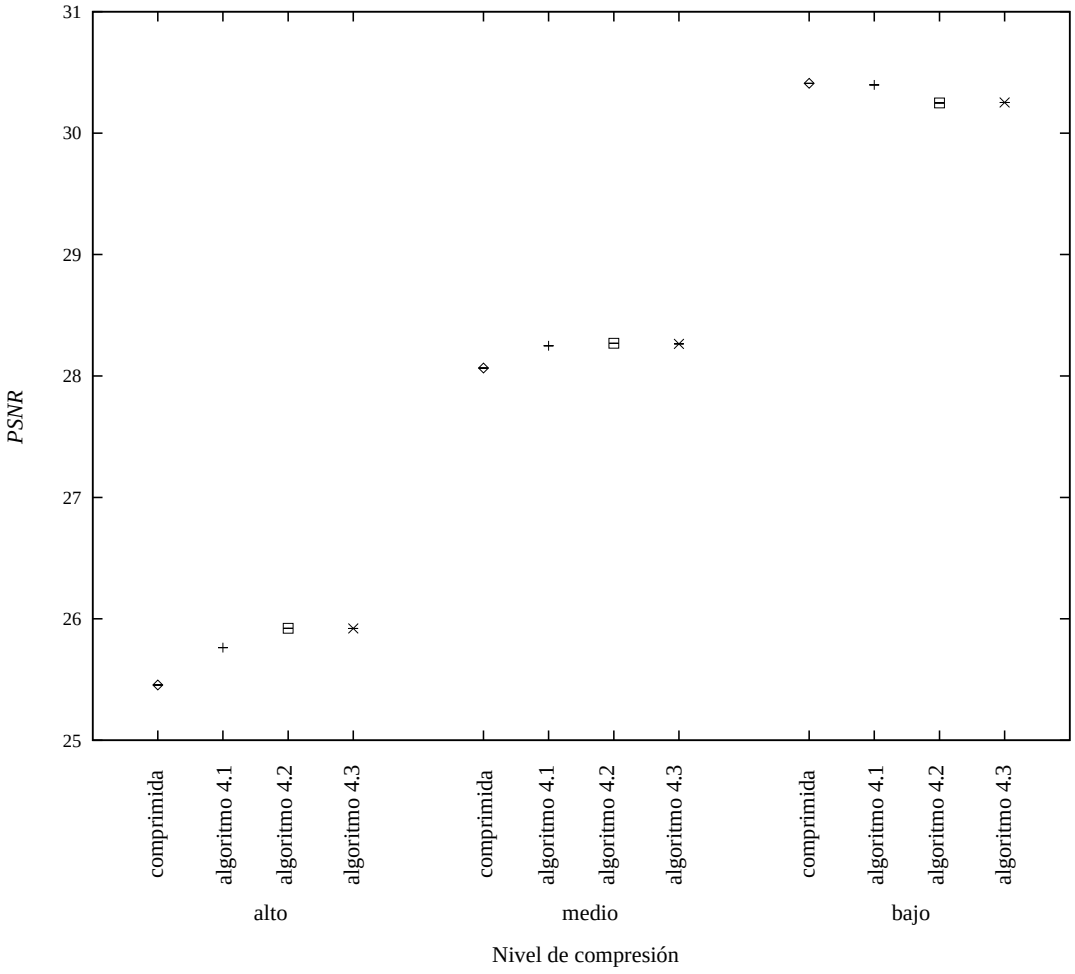


Figura 4.12: Resultados de la reconstrucción con los diferentes algoritmos, expresados en $PSNR$, para la imagen $couple$ a diferentes razones de compresión.

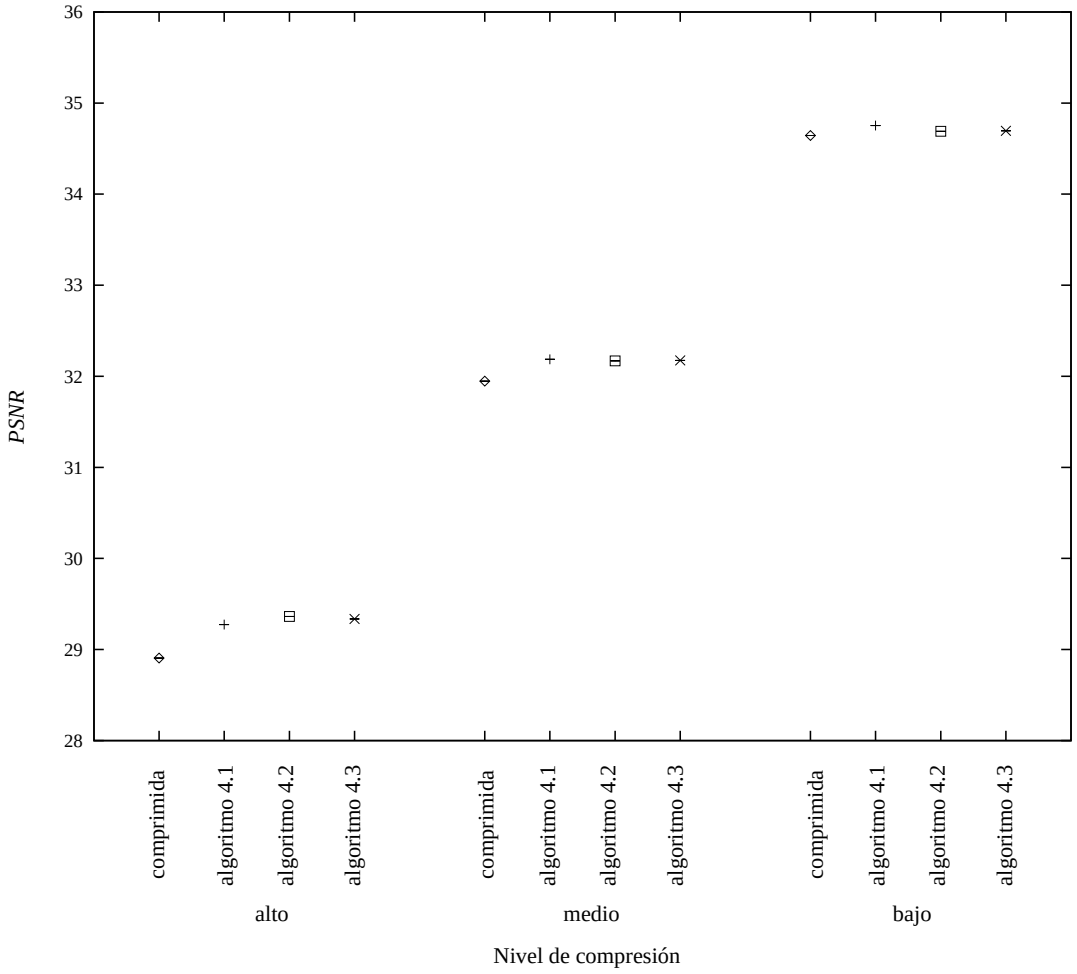


Figura 4.13: Resultados de la reconstrucción con los diferentes algoritmos, expresados en $PSNR$, para la imagen $lena$ a diferentes razones de compresión.

Compresión (bpp)	Algoritmo 4.1	Algoritmo 4.2	Algoritmo 4.3
0.206	41	78	103
0.297	51	84	110
0.500	83	78	362

Tabla 4.1: Número de iteraciones necesarios para que converjan los diferentes métodos de reconstrucción en el decodificador sobre la imagen $boats$.

Compresión (bpp)	Algoritmo 4.1	Algoritmo 4.2	Algoritmo 4.3
0.197	35	106	108
0.310	41	106	113
0.495	55	118	123

Tabla 4.2: Número de iteraciones necesarios para que converjan los diferentes métodos de reconstrucción en el decodificador sobre la imagen $couple$.

Compresión (bpp)	Algoritmo 4.1	Algoritmo 4.2	Algoritmo 4.3
0.203	48	82	56
0.307	76	114	74
0.497	204	340	349

Tabla 4.3: Número de iteraciones necesarios para que converjan los diferentes métodos de reconstrucción en el decodificador sobre la imagen $lena$.

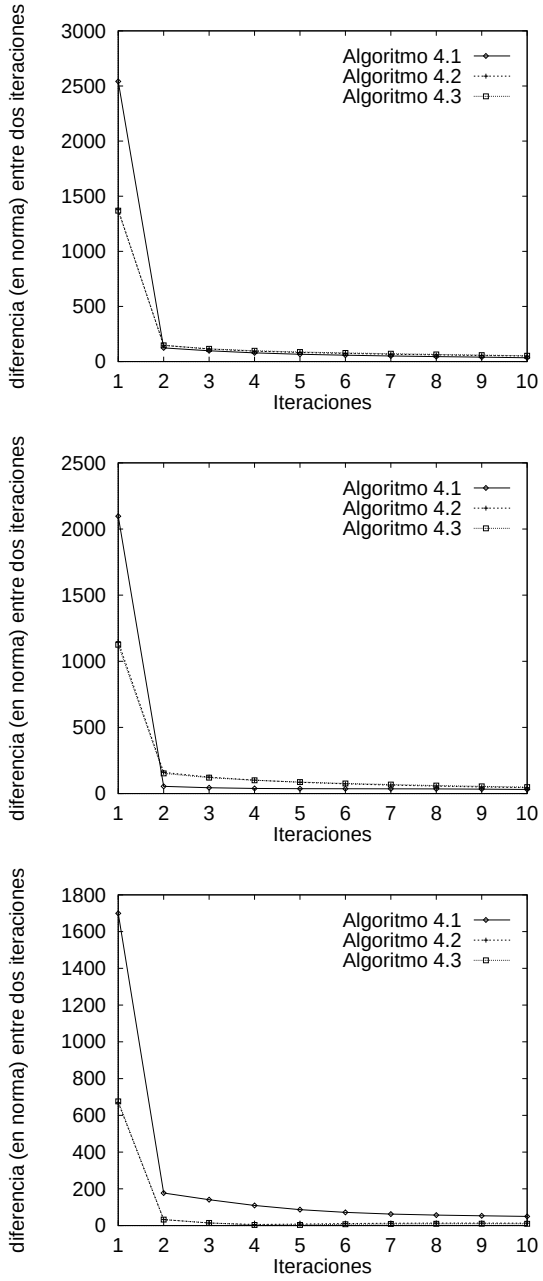


Figura 4.14: Convergencia de los métodos propuestos. En el eje de ordenadas se representa $\| \mathbf{f}^{(\alpha^{k-1},\beta^{k-1})} - \mathbf{f}^{(\alpha^{k},\beta^{k})} \|$ y en el eje de abscisas la iteración k. De arriba a abajo: Para $boats$ comprimida a 0.20bpp. Para $couple$ comprimida a 0.30bpp. Para $lena$ comprimida a 0.30bpp.

Figura 4.15: De izquierda a derecha y de arriba a abajo: Imagen $boats$ comprimida a 0.20bpp. Reconstrucción con 4 iteraciones del Algoritmo 4.1. Reconstrucción con 4 iteraciones del Algoritmo 4.2. Reconstrucción con 4 iteraciones del Algoritmo 4.3.

Figura 4.16: De izquierda a derecha y de arriba a abajo: Imagen $couple$ comprimida a 0.30bpp. Reconstrucción con 4 iteraciones del Algoritmo 4.1. Reconstrucción con 4 iteraciones del Algoritmo 4.2. Reconstrucción con 4 iteraciones del Algoritmo 4.3.

Figura 4.17: De izquierda a derecha y de arriba a abajo: Imagen $lena$ comprimida a 0.30bpp. Reconstrucción con 4 iteraciones del Algoritmo 4.1. Reconstrucción con 4 iteraciones del Algoritmo 4.2. Reconstrucción con 4 iteraciones del Algoritmo 4.3.

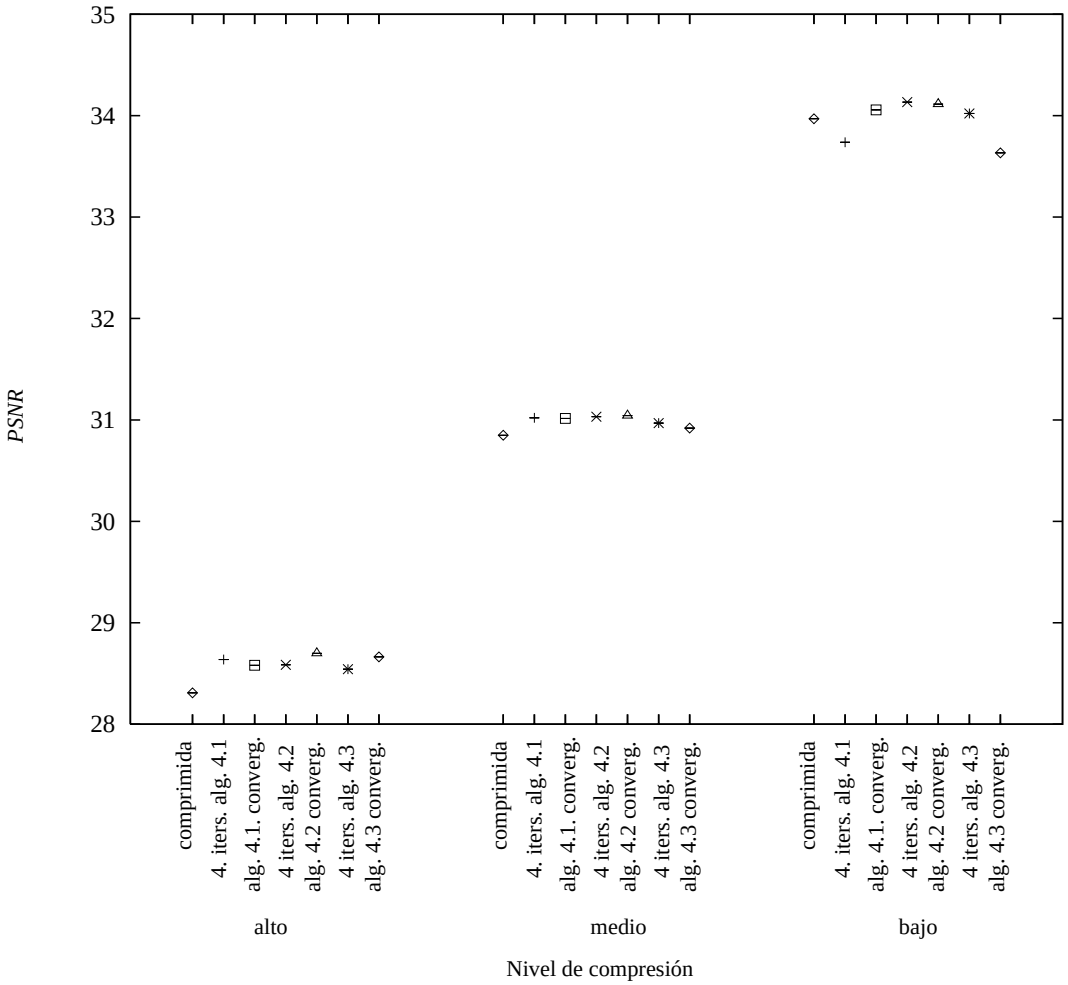


Figura 4.18: Comparación de los resultados de la reconstrucción con 4 iteraciones de los diferentes algoritmos y a convergencia, expresados en $PSNR$, para la imagen $boats$ a diferentes razones de compresión.

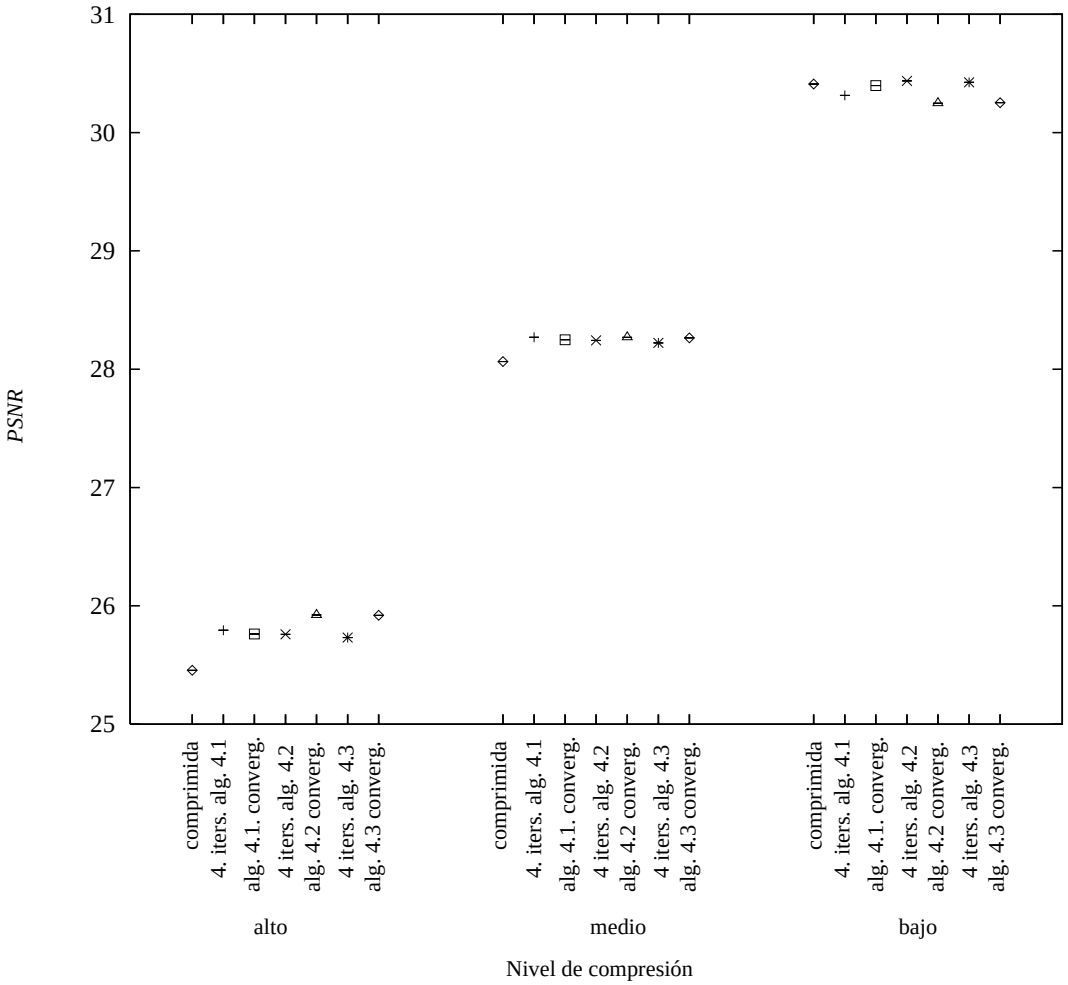


Figura 4.19: Comparación de los resultados de la reconstrucción con 4 iteraciones de los diferentes algoritmos y a convergencia, expresados en $PSNR$, para la imagen $couple$ a diferentes razones de compresión.

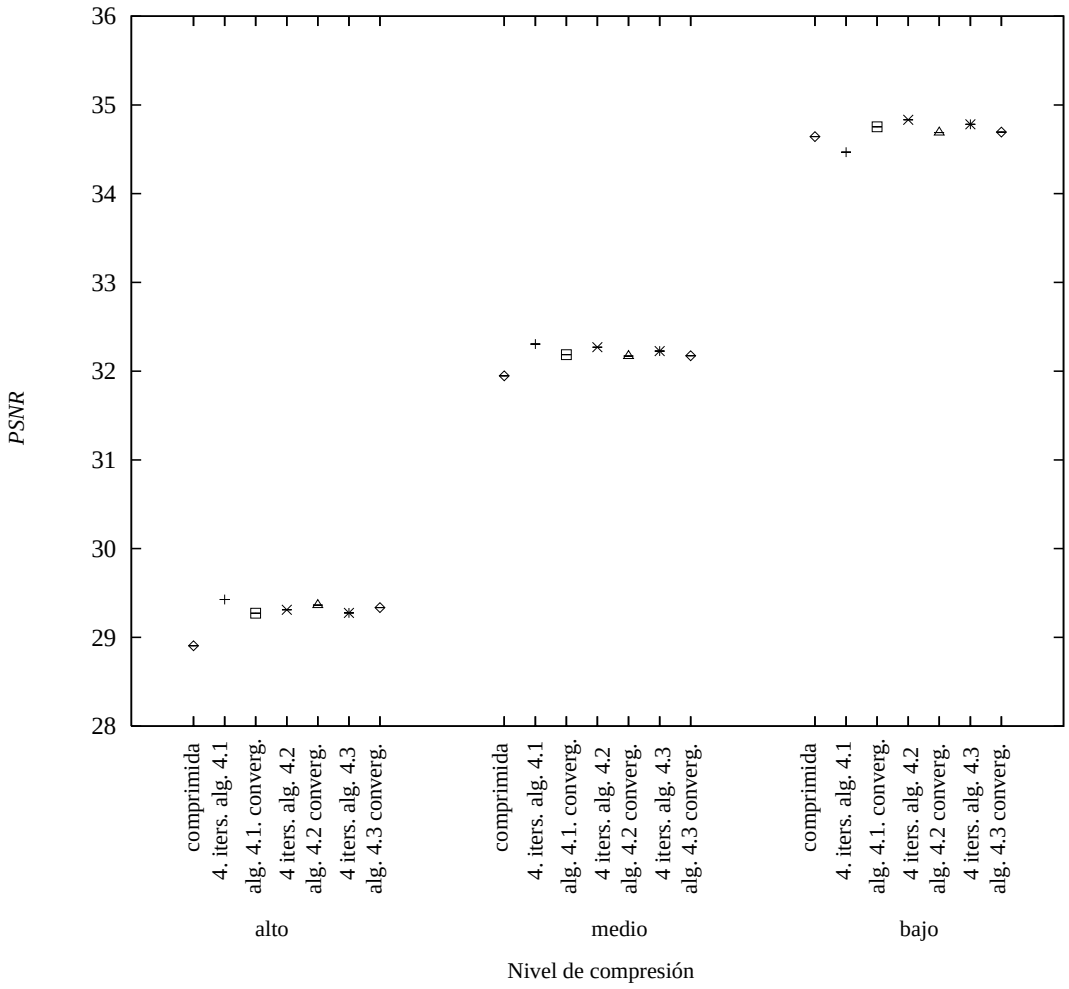


Figura 4.20: Comparación de los resultados de la reconstrucción con 4 iteraciones de los diferentes algoritmos y a convergencia, expresados en $PSNR$, para la imagen $lena$ a diferentes razones de compresión.

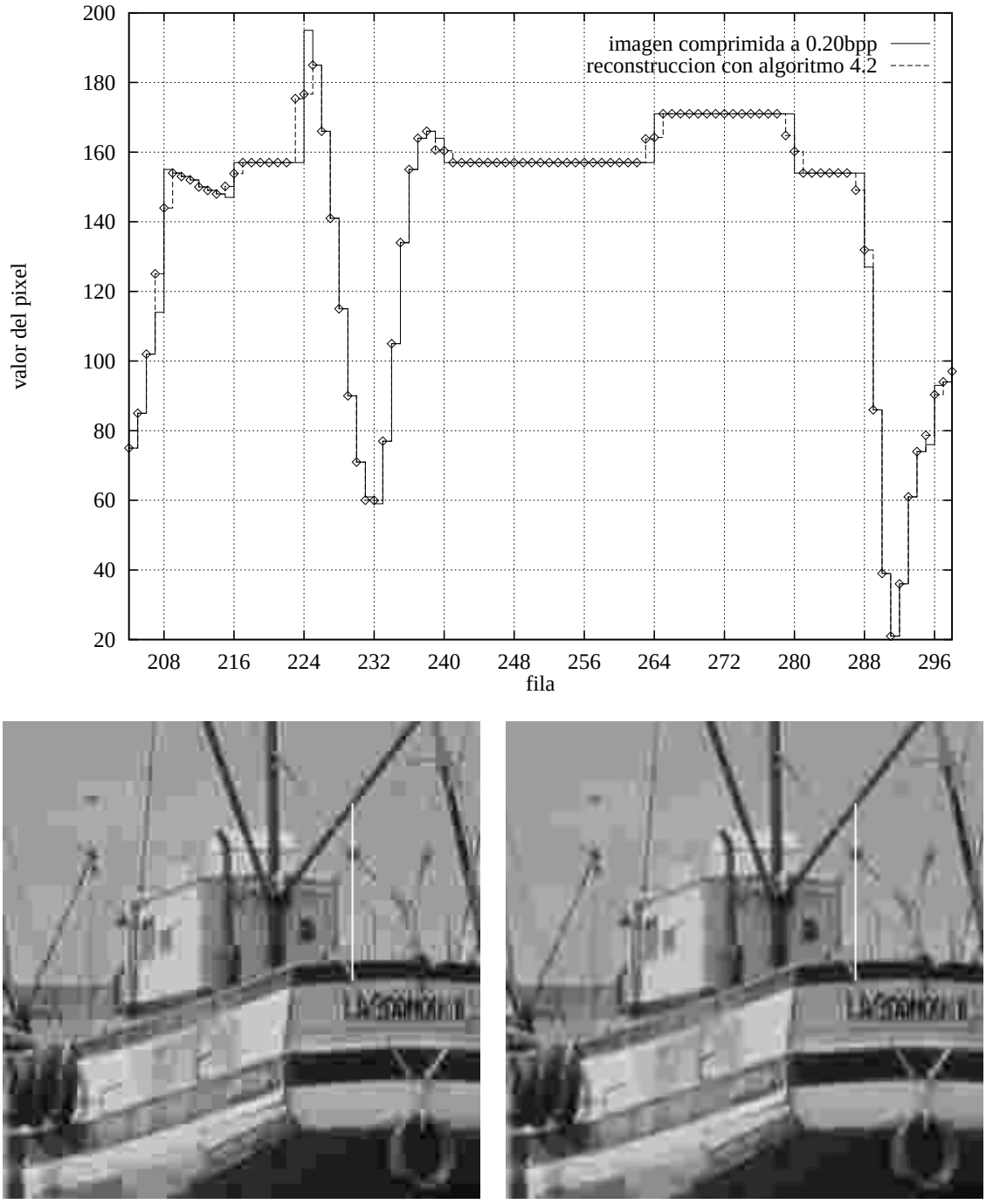


Figura 4.21: Perfil de la columna 419 (marcada en blanco sobre la imagen) de la imagen $boats$ comprimida a 0.20bpp y de su reconstrucción con el Algoritmo 4.2.

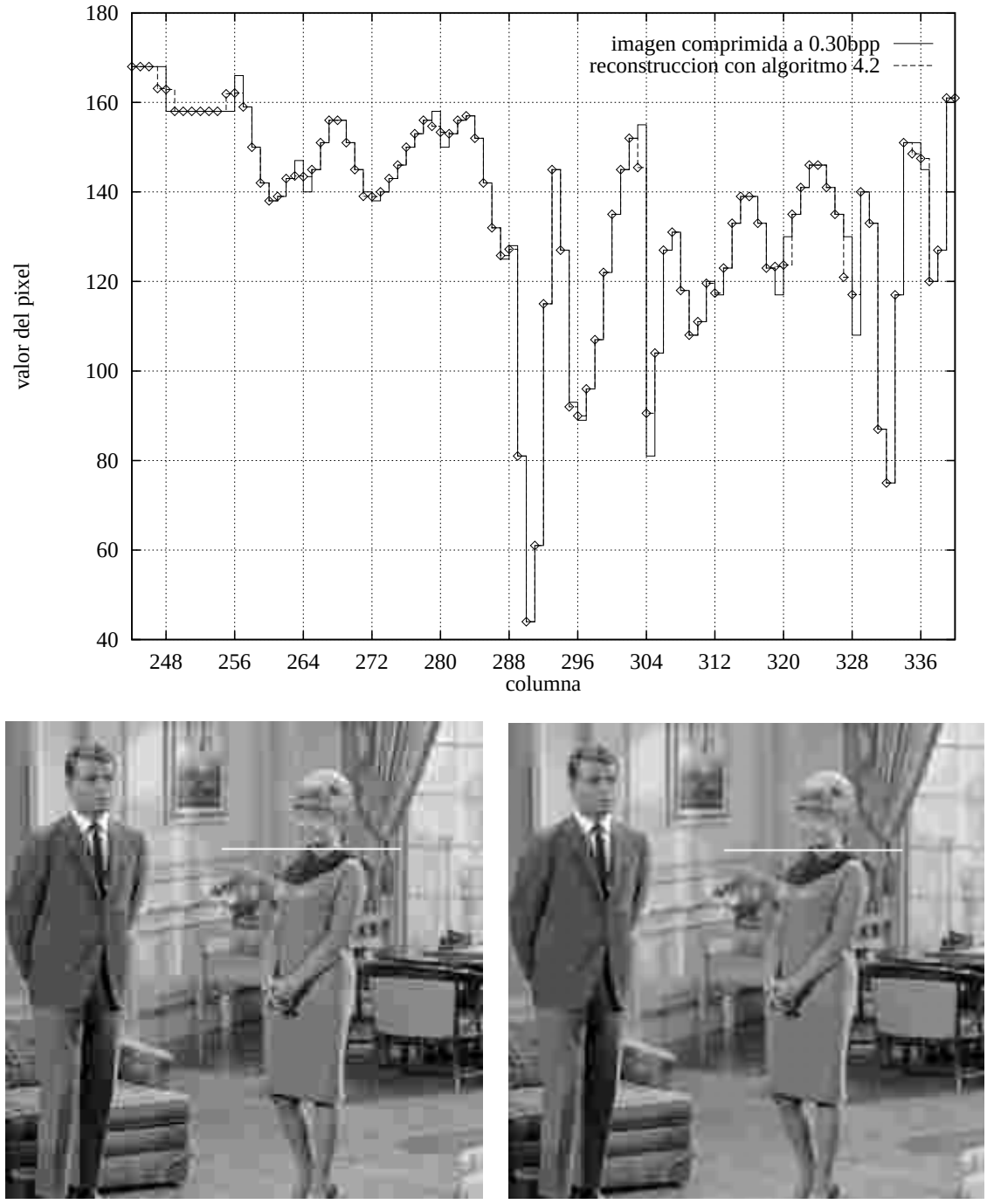


Figura 4.22: Perfil de la fila 196 (marcada en blanco sobre la imagen) de la imagen $couple$ comprimida a 0.30bpp y de su reconstrucción con el Algoritmo 4.2.

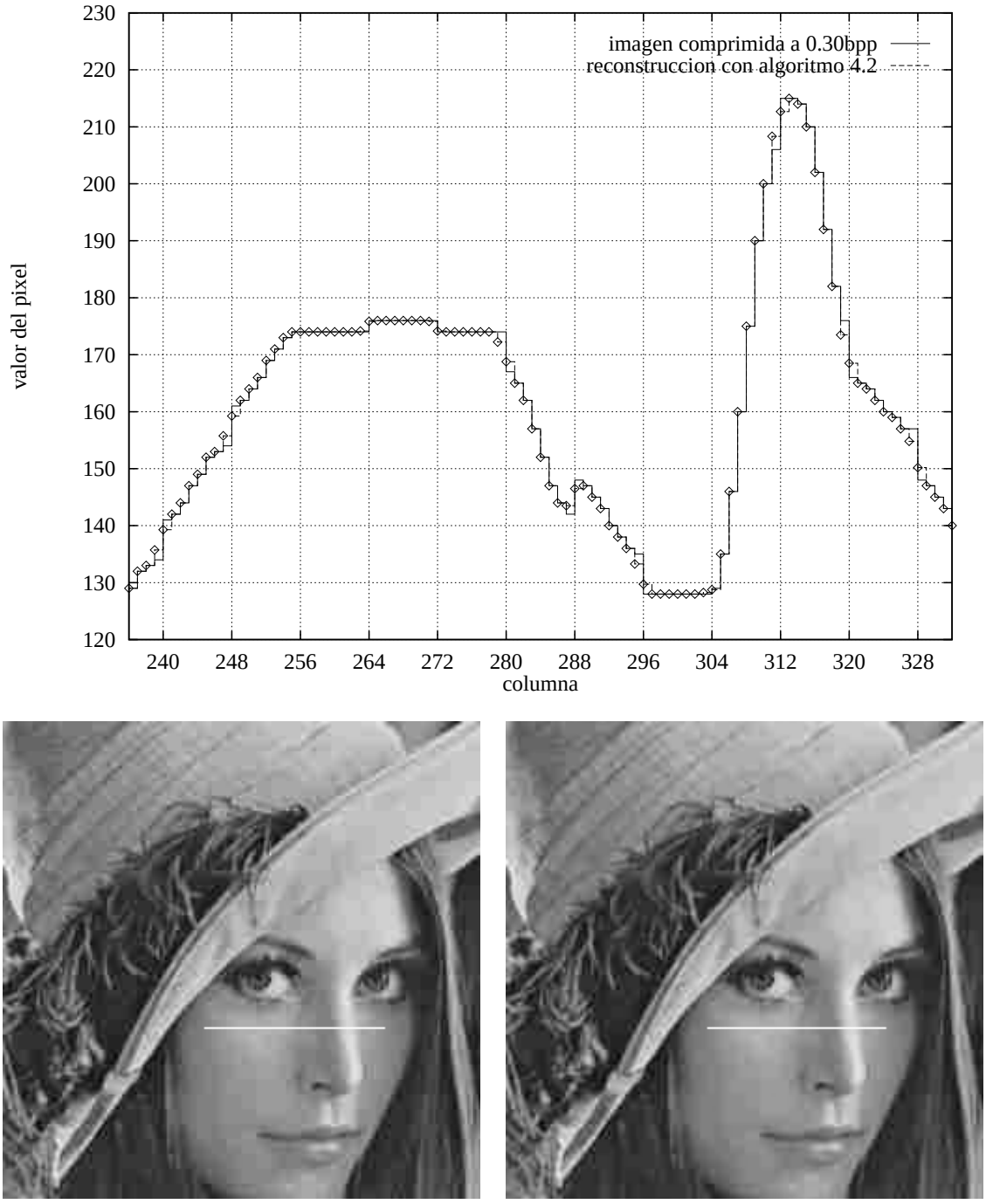


Figura 4.23: Perfil de la fila 232 (marcada en blanco sobre las imágenes) de la imagen $lena$ comprimida a 0.30bpp y de su reconstrucción con el Algoritmo 4.2.

Capítulo 5

Reconstrucción basada en la evidencia con información del codificador

5.1 Introducción

En el capítulo anterior hemos expuesto diferentes métodos para la estimación de los parámetros y de la imagen en el decodificador obteniendo una mejora tanto en el pico de la relación señal-ruido como en calidad visual. Sin embargo, puesto que *en la reconstrucción de imágenes comprimidas disponemos de la imagen original, sin distorsionar, en el codificador, podría ser interesante utilizar la información proveniente de esta imagen para mejorar el comportamiento del método.* Utilizando este planteamiento, podríamos *estimar unos parámetros en el codificador a partir de la imagen original y transmitirlos al decodificador.* Nótese que realizar la transmisión de los parámetros al decodificador no supone la modificación del codificador estándar JPEG puesto que el formato de imágenes definido por el grupo JPEG permite la inclusión, en la cabecera de la imagen, de información usada por la aplicación que produce la codificación. Además, existe un campo de comentarios sobre la imagen en el que, en cualquier caso, se podría introducir esta información.

Sin embargo, es posible que los parámetros estimados en el codificador usando la imagen original no sean los mejores parámetros para la imagen comprimida. En concreto, basarnos en la imagen original para realizar la estimación, podría darnos una información más precisa sobre los parámetros de suavidad entre los bloques de la imagen que la estimada al basarnos en la imagen obtenida en el decodificador pero, por contra, la estimación del parámetro de

ruido puede no ser, en la mayoría de los casos, tan precisa como si se hiciese sobre la imagen comprimida debido, principalmente, a que la imagen original no presenta el efecto de bloques.

Sería, por tanto, interesante encontrar una forma de *combinar la información obtenida en el codificador a partir de la imagen original con la información obtenida en el decodificador a partir de la imagen comprimida en un proceso que incorpore la información de la imagen original en la estimación de los parámetros en el decodificador.*

La aproximación jerárquica bayesiana aplicada al problema de la reconstrucción de imágenes comprimidas nos permite, de una manera formal, incorporar conocimiento previo sobre los hiperparámetros cuando estimamos en el decodificador. Podremos, por tanto, utilizando una modelización bien fundamentada, estimar los parámetros en el codificador usando la imagen original, transmitirlos junto a la imagen comprimida y utilizarlos en el decodificador para obtener unos nuevos parámetros que incorporen también la información proveniente de esta imagen comprimida.

En este capítulo extenderemos el método que mejores resultados nos ha dado para la estimación de la imagen y de los hiperparámetros en el decodificador (es decir, el método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido, descrito en la sección 4.3), para combinar la estimación de los parámetros en el codificador y en el decodificador y así obtener una reconstrucción con mayor información.

El resto del capítulo se organiza de la siguiente forma. En la sección 5.2 presentamos la forma en la que la estimación de los parámetros se puede realizar en el codificador, usando la imagen original, mediante un análisis basado en la evidencia usando una distribución uniforme (no informativa) sobre los hiperparámetros. En la sección 5.3 combinaremos los parámetros obtenidos en el codificador y en el decodificador para obtener la imagen reconstruida utilizando el modelo basado en la evidencia. La sección 5.4 muestra resultados experimentales sobre imágenes reales usando este método y se extraen conclusiones sobre estos resultados.

5.2 Reconstrucción en el codificador basada en la evidencia usando una distribución uniforme sobre los hiperparámetros

En el codificador tenemos la información de la imagen original, antes de ser comprimida, y es posible usarla para estimar los parámetros a partir de ella, como veremos, usando cualquiera de los métodos expuestos en el capítulo anterior que permiten estimar los parámetros y la imagen reconstruida. Por simplicidad, nos centraremos, en uno de esos métodos, el método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido descrito en la sección 4.3, aunque es posible realizar todos los desarrollos de manera semejante para los demás métodos. Mediante este método realizaremos la estimación de los parámetros en el codificador dentro del paradigma jerárquico bayesiano, usando la imagen original como imagen observada.

Estudiemos como puede hacerse esta estimación dentro del análisis de la evidencia cuando se usa una distribución uniforme como modelo a priori sobre los hiperparámetros.

En general, si α_c, α_r y β son los parámetros estimados, podemos estimar la imagen en el codificador como

$$\mathbf{f}^{(\alpha_c,\alpha_r,\beta)cod} = \arg\max_{\mathbf{z}} \mathrm{p}(\mathbf{z},\mathbf{f} \mid \alpha_c,\alpha_r,\beta)$$

donde se ha usado la imagen original, $\mathbf{f}$, como imagen observada. Una vez que sabemos como estimar la imagen $\mathbf{f}^{(\alpha_c,\alpha_r,\beta)cod}$ podemos estimar también los parámetros usando la imagen original como observación y una distribución no informativa sobre los hiperparámetros, es decir,

$$\hat{\alpha}_c^{cod},\hat{\alpha}_r^{cod},\hat{\beta}^{cod} = \arg\max_{\alpha_c,\alpha_r,\beta} \int_{\mathbf{z}} \mathrm{p}(\mathbf{z},\mathbf{f} \mid \alpha_c,\alpha_r,\beta)\, d\mathbf{z}.$$

Está claro que para obtener los valores de los parámetros $\hat{\alpha}_c^{cod}$, $\hat{\alpha}_r^{cod}$ y $\hat{\beta}^{cod}$, los estimadores de α_c, α_r y β en el codificador, sólo tenemos que aplicar el algoritmo 4.2 descrito en la sección 4.3.3.

Una vez estimados, los hiperparámetros $\hat{\alpha}_c^{cod}$, $\hat{\alpha}_r^{cod}$ y $\hat{\beta}^{cod}$ se transmiten, posiblemente tras un proceso de cuantificación, con el resto de la imagen comprimida, y en el decodificador se puede reconstruir la imagen, $\mathbf{f}^{(\alpha_c^{cod},\alpha_r^{cod},\beta^{cod})}$ usando esos parámetros, usando la solución caracterizada en la sección 4.3.2.

Pero es posible que los parámetros estimados en el codificador no sean la mejor estimación

posible de los parámetros de la imagen comprimida o que, como consecuencia de una cuantificación, no resulten en los mejores parámetros en el decodificador. Por ello, en la siguiente sección estudiaremos como, una vez estimados los parámetros en el codificador y transmitidos (posiblemente con pérdidas), combinarlos con los estimados en el decodificador mediante un proceso robusto y bien fundamentado.

5.3 Incorporando información del codificador. Distribuciones a priori gamma

Para el problema que nos ocupa en esta sección usaremos el paradigma jerárquico bayesiano para utilizar la información de los parámetros estimados en el codificador como información a priori sobre el valor de los hiperparámetros de la imagen que se estimarán en el decodificador.

Utilizaremos, de nuevo, el método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido. Una vez estimados los valores de α_c^{cod}, α_r^{cod} y β^{cod} mediante el algoritmo 4.2 aplicado en el codificador, en la forma en que se ha descrito en la sección anterior, estos parámetros se transmitirán, posiblemente con pérdidas, junto a la imagen comprimida hacia el decodificador. Sean

$$\begin{aligned} m_c^{cod} &= \mathcal{Q}\alpha_c^{cod}, \\ m_r^{cod} &= \mathcal{Q}\alpha_r^{cod}, \\ n^{cod} &= \mathcal{Q}\beta^{cod}, \end{aligned}$$

los parámetros transmitidos por el codificador, posiblemente tras un proceso de cuantificación representado por $\mathcal{Q}$, y que queremos usar para guiar la estimación de los hiperparámetros en el decodificador, que denotaremos por α_c, α_r y β.

Entonces para combinar la información sobre los hiperparámetros proveniente del codificador con las estimaciones en el decodificador, usaremos de nuevo el paradigma jerárquico bayesiano, usando los mismos modelos de imagen y ruido que usamos en la sección 4.3.3 aunque usando la distribución gamma, definida por

$$\mathrm{p}(\alpha_c) \propto \alpha_c^{l(m_c^{cod})-1} \exp[-l(m_c^{cod})\alpha_c/m_c^{cod}], \tag{5.1}$$

$$\mathrm{p}(\alpha_r) \propto \alpha_r^{l(m_r^{cod})-1} \exp[-l(m_r^{cod})\alpha_r/m_r^{cod}], \tag{5.2}$$

$$\mathrm{p}(\beta) \propto \beta^{l(n^{cod})-1} \exp[-l(n^{cod})\beta/n^{cod}], \tag{5.3}$$

como modelo a priori sobre los hiperparámetros, en lugar de la distribución plana.

Recordemos que la distribución gamma tiene las siguientes propiedades:

$$E[\alpha_c] = \frac{1}{m_c^{cod}} \qquad \text{y} \qquad Var[\alpha_c] = \frac{1}{{m_c^{cod}}^2 l(m_c^{cod})}.$$
$$E[\alpha_r] = \frac{1}{m_r^{cod}} \qquad \text{y} \qquad Var[\alpha_r] = \frac{1}{{m_r^{cod}}^2 l(m_r^{cod})}.$$
$$E[\beta] = \frac{1}{n^{cod}} \qquad \text{y} \qquad Var[\beta] = \frac{1}{{n^{cod}}^2 l(n^{cod})}.$$

Así, cada hiperparámetro, α_c, α_r y β, tiene como media m_c^{cod}, m_r^{cod} y n^{cod}, respectivamente, y su varianza decrece conforme crece $l(m_c^{cod})$, $l(m_r^{cod})$ y $l(n^{cod})$, respectivamente. Esto significa que $l(m_c^{cod})$, $l(m_r^{cod})$ y $l(n^{cod})$ pueden verse como una medida de la certeza que tenemos en el conocimiento sobre la media de cada hiperparámetro.

Habiendo definido los elementos necesarios, de nuevo usaremos el análisis basado en la evidencia para estimar la imagen y el valor de los hiperparámetros en el decodificador. Estudiemos ahora los pasos de estimación de los parámetros y de reconstrucción de la imagen que caracterizan las soluciones al problema y, una vez caracterizadas las soluciones, propondremos una algoritmo iterativo para estimar los parámetros y reconstruir la imagen simultáneamente.

5.3.1 Paso de estimación de los parámetros

El proceso de estimación de los parámetros es, en este caso, como sigue. Primero tengamos en cuenta que

$$\begin{aligned}
\mathrm{p}(\alpha_c, \alpha_r, \beta \mid \mathbf{g}) &\propto \mathrm{p}(\alpha_c)\mathrm{p}(\alpha_r)\mathrm{p}(\beta) \int_{\mathbf{f}} \mathrm{p}(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta)\, d\mathbf{f} \\
&= \mathrm{p}(\alpha_c)\mathrm{p}(\alpha_r)\mathrm{p}(\beta) \int_{\mathbf{f}_c} \mathrm{p}(\mathbf{f}_c, \mathbf{g}_c \mid \alpha_c, \beta)\, d\mathbf{f}_c \int_{\mathbf{f}_r} \mathrm{p}(\mathbf{f}_r, \mathbf{g}_r \mid \alpha_r, \beta)\, d\mathbf{f}_r \int_{\mathbf{f}_x} \mathrm{p}(\mathbf{f}_x, \mathbf{g}_x \mid \alpha_c, \alpha_r, \beta)\, d\mathbf{f}_x \\
&= \mathrm{p}(\alpha_c)\mathrm{p}(\alpha_r)\mathrm{p}(\beta)\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta)\mathrm{p}(\mathbf{g}_x \mid \alpha_c, \alpha_r, \beta).
\end{aligned}$$

Puesto que tenemos los mismos problemas que en la aproximación usando la hiperdistribución a priori plana si incluimos $\mathrm{p}(\mathbf{g}_x \mid \alpha_c, \alpha_r, \beta)$ en la estimación de los parámetros, estimaremos α_c, α_r y β como

$$\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta} = \arg \max_{\alpha_c, \alpha_r, \beta} \mathrm{p}(\alpha_c)\mathrm{p}(\alpha_r)\mathrm{p}(\beta)\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta). \tag{5.4}$$

Entonces,

$$\mathrm{p}(\alpha_c)\mathrm{p}(\alpha_r)\mathrm{p}(\beta)\mathrm{p}(\mathbf{g}_c \mid \alpha_c, \beta)\mathrm{p}(\mathbf{g}_r \mid \alpha_r, \beta) =$$

$$
\begin{aligned}
&= \alpha_c^{\frac{p}{2}+l(m_c^{cod})-1}\alpha_r^{\frac{q}{2}+l(m_r^{cod})-1}\beta^{p+q+l(n^{cod})-1}\int_{\mathbf{f}_c,\mathbf{f}_r}\exp\Bigg[-M_c(\mathbf{f}_c,\mathbf{g}_c\mid\alpha_c,\beta)\\
&- M_r(\mathbf{f}_r,\mathbf{g}_r\mid\alpha_r,\beta)-\frac{l(m_c^{cod})\alpha_c}{m_c^{cod}}-\frac{l(m_r^{cod})\alpha_r}{m_r^{cod}}-\frac{l(n^{cod})\beta}{n^{cod}}\Bigg]\,d\mathbf{f}_c\,d\mathbf{f}_r.
\end{aligned}
$$

Por tanto tenemos que

$$
\begin{aligned}
&\mathrm{p}(\alpha_c)\mathrm{p}(\alpha_r)\mathrm{p}(\beta)\mathrm{p}(\mathbf{g}_c\mid\alpha_c,\beta)\mathrm{p}(\mathbf{g}_r\mid\alpha_r,\beta)\propto\\
&\propto \alpha_c^{\frac{p}{2}+l(m_c^{cod})-1}\alpha_r^{\frac{q}{2}+l(m_r^{cod})-1}\beta^{p+q+l(n^{cod})-1}\exp\{-M_c(\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)},\mathbf{g}_c\mid\alpha_c,\beta)\}\\
&\times \exp\{-M_r(\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)},\mathbf{g}_r\mid\alpha_r,\beta)\}\exp\left\{-\frac{l(m_c^{cod})\alpha_c}{m_c^{cod}}\right\}\exp\left\{-\frac{l(m_r^{cod})\alpha_r}{m_r^{cod}}\right\}\\
&\times \exp\left\{-\frac{l(n^{cod})\beta}{n^{cod}}\right\}\int_{\mathbf{f}_c}\exp\left\{-\frac{1}{2}(\mathbf{f}_c-\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)})^t\mathbf{Q}_c(\alpha_c,\beta)(\mathbf{f}_c-\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)})\right\}d\mathbf{f}_c\\
&\times \int_{\mathbf{f}_r}\exp\left\{-\frac{1}{2}(\mathbf{f}_r-\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)})^t\mathbf{Q}_r(\alpha_r,\beta)(\mathbf{f}_r-\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)})\right\}d\mathbf{f}_r\\
&= \alpha_c^{\frac{p}{2}+l(m_c^{cod})-1}\alpha_r^{\frac{q}{2}+l(m_r^{cod})-1}\beta^{p+q+l(n^{cod})-1}\exp\{-M_c(\mathbf{f}_c^{(\alpha_c,\alpha_r,\beta)},\mathbf{g}_c\mid\alpha_c,\beta)\}\\
&\times \exp\{-M_r(\mathbf{f}_r^{(\alpha_c,\alpha_r,\beta)},\mathbf{g}_r\mid\alpha_r,\beta)\}\exp\left\{-\frac{l(m_c^{cod})\alpha_c}{m_c^{cod}}\right\}\exp\left\{-\frac{l(m_r^{cod})\alpha_r}{m_r^{cod}}\right\}\\
&\times \exp\left\{-\frac{l(n^{cod})\beta}{n^{cod}}\right\}[\det\,\mathbf{Q}_c(\alpha_c,\beta)]^{-\frac{1}{2}}[\det\,\mathbf{Q}_r(\alpha_r,\beta)]^{-\frac{1}{2}}.
\end{aligned}
$$

Entonces, diferenciando $-\log[\mathrm{p}(\alpha_c)\mathrm{p}(\alpha_r)\mathrm{p}(\beta)\mathrm{p}(\mathbf{g}_c\mid\alpha_c,\beta)\mathrm{p}(\mathbf{g}_r\mid\alpha_r,\beta)]$ respecto a α_c, α_r y β tenemos en $\hat{\alpha}_c$, $\hat{\alpha}_r$ y $\hat{\beta}$

$$
\begin{aligned}
(\frac{p}{2}+l(m_c^{cod})-1)\frac{1}{\hat{\alpha}_c} &= \parallel\mathbf{W}_c(\mathbf{f}_{cl}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})}-\mathbf{f}_{cr}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})})\parallel^2+\frac{l(m_c^{cod})}{m_c^{cod}}\\
&+ \frac{1}{2}\sum_{i=1}^{p}\mathrm{traza}\,[\mathbf{q}_c^{-1}(\hat{\alpha}_c,\hat{\beta})(i)\mathbf{A}_c(i)], \qquad (5.5)\\
(\frac{q}{2}+l(m_r^{cod})-1)\frac{1}{\hat{\alpha}_r} &= \parallel\mathbf{W}_r(\mathbf{f}_{ra}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})}-\mathbf{f}_{rb}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})})\parallel^2+\frac{l(m_r^{cod})}{m_r^{cod}}\\
&+ \frac{1}{2}\sum_{i=1}^{q}\mathrm{traza}\,[\mathbf{q}_r^{-1}(\hat{\alpha}_r,\hat{\beta})(i)\mathbf{A}_r(i)], \qquad (5.6)\\
2(p+q+l(n^{cod})-1)\frac{1}{\hat{\beta}} &= \parallel\mathbf{g}_{cl}-\mathbf{f}_{cl}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})}\parallel^2+\parallel\mathbf{g}_{cr}-\mathbf{f}_{cr}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})}\parallel^2\\
&+ \parallel\mathbf{g}_{ra}-\mathbf{f}_{ra}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})}\parallel^2+\parallel\mathbf{g}_{rb}-\mathbf{f}_{rb}^{(\hat{\alpha}_c,\hat{\alpha}_r,\hat{\beta})}\parallel^2+\frac{2l(n^{cod})}{n^{cod}}\\
&+ \sum_{i=1}^{p}\mathrm{traza}\,[\mathbf{q}_c^{-1}(\hat{\alpha}_c,\hat{\beta})(i)]+\sum_{i=1}^{q}\mathrm{traza}\,[\mathbf{q}_r^{-1}(\hat{\alpha}_r,\hat{\beta})(i)]. \qquad (5.7)
\end{aligned}
$$

5.3.2 Paso de reconstrucción

Puesto que la reconstrucción se basa en la estimación de la imagen dado el valor de los parámetros, α_c, α_r y β, no depende del tipo de distribución a priori que introducimos sobre éstos. Esto significa que el paso de reconstrucción de la imagen para una distribución a priori gamma sobre los parámetros es exactamente igual que en el caso de la distribución plana, remitiéndose al lector a la sección 4.3.2 para los detalles sobre la reconstrucción de la imagen.

5.3.3 Procedimiento iterativo para estimar los hiperparámetros y reconstruir la imagen

Una vez caracterizadas las soluciones óptimas para los hiperparámetros y mostrado como reconstruir la imagen dado un conjunto de hiperparámetros, usamos el siguiente algoritmo para estimar los hiperparámetros y la imagen simultáneamente asumiendo una distribución gamma.

Algoritmo 5.1 Reconstrucción adaptativa con diferentes parámetros para las filas y las columnas en el modelo de imagen y un único parámetro en el modelo de ruido usando una distribución gamma sobre los hiperparámetros.

1. *Escoger* α_c^0, α_c^0 *y* β^0.
2. *Calcular* $\mathbf{f}_c^{(\alpha_c^0,\alpha_r^0,\beta^0)}$ *y* $\mathbf{f}_r^{(\alpha_c^0,\alpha_r^0,\beta^0)}$ *a partir de las Ecs. (4.23)–(4.26).*
3. *Para* $k = 1, 2, \ldots$
   (a) *Estimar* α_c^k, α_r^k *y* β^k *sustituyendo los valores de* α_c^{k-1}, α_c^{k-1} *y* β^{k-1} *en la parte derecha de las Ecs. (5.5), (5.6) y (5.7), respectivamente.*
   (b) *Calcular* $\mathbf{f}_c^{(\alpha_c^k,\alpha_r^k,\beta^k)}$ *y* $\mathbf{f}_r^{(\alpha_c^k,\alpha_r^k,\beta^k)}$ *a partir de las Ecs. (4.23)–(4.26).*
4. *Ir al paso 3 hasta que* $\| \mathbf{f}_c^{(\alpha_c^{k-1},\alpha_r^{k-1},\beta^{k-1})} - \mathbf{f}_c^{(\alpha_c^k,\alpha_r^k,\beta^k)} \| + \| \mathbf{f}_r^{(\alpha_c^{k-1},\alpha_r^{k-1},\beta^{k-1})} - \mathbf{f}_r^{(\alpha_c^k,\alpha_r^k,\beta^k)} \|$ *sea menor que un límite establecido.*
5. *Usando* α_c^k, α_r^k *y* β^k, *calcular* $\mathbf{f}_x^{(\alpha_c^k,\alpha_r^k,\beta^k)}$ *resolviendo el sistema de ecuaciones descrito en la Ec. (4.27)*

La prueba de la convergencia de este algoritmo se basa de nuevo en el hecho de que es un algoritmo EM (ver [55, 79]).

Asumiendo que $p \simeq p-2$ y $q \simeq q-2$, podemos escribir las Ecs. (5.5), (5.6) y (5.7) como

$$\frac{1}{\alpha_c^k} = \mu_c \frac{1}{m_c^{cod}} + (1-\mu_c)\frac{1}{\alpha_c^{k,dec}}, \tag{5.8}$$

$$\frac{1}{\alpha_r^k} = \mu_r \frac{1}{m_r^{cod}} + (1-\mu_r)\frac{1}{\alpha_r^{k,dec}}, \tag{5.9}$$

$$\frac{1}{\beta^k} = \nu \frac{1}{n^{cod}} + (1-\nu)\frac{1}{\beta^{k,dec}}, \tag{5.10}$$

donde $\alpha_c^{k,dec}$, $\alpha_r^{k,dec}$ y $\beta^{k,dec}$ se calculan sustituyendo α_c^{k-1}, α_r^{k-1} y β^{k-1} en la parte derecha de las Ecs. (4.18), (4.19) y (4.20), y

$$\mu_c = \frac{2l(m_c^{cod})}{2l(m_c^{cod})+p}, \tag{5.11}$$

$$\mu_r = \frac{2l(m_r^{cod})}{2l(m_r^{cod})+q}, \tag{5.12}$$

$$\nu = \frac{l(n^{cod})}{l(n^{cod})+p+q}. \tag{5.13}$$

Nótese que μ_c, μ_r y ν pueden entenderse como un parámetro de confianza normalizado. Todos ellos toman valores en el intervalo $[0,1)$. Cuando toman el valor cero significa que no se tiene ninguna confianza en los valores transmitidos (se realiza la estimación completa en el decodificador o, lo que es lo mismo, m_c^{cod}, m_r^{cod} y n^{cod} o no se han transmitido o no se usan), mientras que si los parámetros de confianza normalizados son uno refuerzan totalmente el conocimiento a priori sobre la media de los hiperparámetros, no realizándose ninguna estimación de los hiperparámetros en el decodificador. Está claro, por tanto, que $l(m_c^{cod})$, $l(m_r^{cod})$ y $l(n^{cod})$ dependen de la confianza que tengamos en sus correspondientes medias, m_c^{cod}, m_r^{cod} y n^{cod}.

El procedimiento iterativo descrito en el Algoritmo 5.1 puede reescribirse entonces, para reflejar estas ideas, de la siguiente forma:

Algoritmo 5.2 Reconstrucción adaptativa con diferentes parámetros para las filas y las columnas en el modelo de imagen y un único parámetro en el modelo de ruido usando una distribución gamma sobre los hiperparámetros.

1. *Escoger* α_c^0, α_c^0 *y* β^0.
2. *Calcular* $\mathbf{f}_c^{(\alpha_c^0,\alpha_r^0,\beta^0)}$ *y* $\mathbf{f}_r^{(\alpha_c^0,\alpha_r^0,\beta^0)}$ *a partir de las Ecs. (4.23)–(4.26).*
3. *Para* $k = 1, 2, \ldots$

(a) *Estimar* $\alpha_c^{k,dec}$, $\alpha_r^{k,dec}$ *y* $\beta^{k,dec}$ *sustituyendo* α_c^{k-1}, α_r^{k-1} *y* β^{k-1} *en la parte izquierda de las Ecs. (4.18), (4.19) y (4.20), respectivamente.*

(b) *Estimar* α_c^k, α_r^k *y* β^k *sustituyendo los valores de* α_c^{k-1}, α_c^{k-1} *y* β^{k-1} *en la parte derecha de las Ecs. (4.18), (4.19) y (4.20), respectivamente.*

(c) *Actualizar nuestro conocimiento sobre los hiperparámetros mediante las Ecs. (5.8), (5.9) y (5.10).*

(d) *Calcular* $\mathbf{f}_c^{(\alpha_c^k,\alpha_r^k,\beta^k)}$ *y* $\mathbf{f}_r^{(\alpha_c^k,\alpha_r^k,\beta^k)}$ *a partir de las Ecs. (4.23)–(4.26).*

4. *Ir al paso 3 hasta que* $\| \mathbf{f}_c^{(\alpha_c^{k-1},\alpha_r^{k-1},\beta^{k-1})} - \mathbf{f}_c^{(\alpha_c^k,\alpha_r^k,\beta^k)} \| + \| \mathbf{f}_r^{(\alpha_c^{k-1},\alpha_r^{k-1},\beta^{k-1})} - \mathbf{f}_r^{(\alpha_c^k,\alpha_r^k,\beta^k)} \|$ *sea menor que un límite establecido.*

5. *Usando* α_c^k, α_r^k *y* β^k, *calcular* $\mathbf{f}_x^{(\alpha_c^k,\alpha_r^k,\beta^k)}$ *resolviendo el sistema de ecuaciones descrito en la Ec. (4.27)*

Está claro que el modo en que los parámetros se ponderan depende de como mejoren la calidad visual o el pico de la razón señal-ruido y de la forma en que se transmiten. Por ejemplo, si se transmiten sin pérdidas y se obtiene una mejora en la relación señal-ruido o en la calidad visual, se podrían usar en lugar de los obtenidos en el decodificador. Si son cuantificados, de la misma forma en que se cuantifica la imagen, es posible que se obtengan mejores resultados combinándolos con los obtenidos en el decodificador.

También haremos notar que el algoritmo propuesto es no lineal en los hiperparámetros pero es lineal en la inversa de esos hiperparámetros, que son las varianzas de los modelos.

Se podría argumentar que la introducción de la distribución gamma como modelo de hiperparámetros hace que, de nuevo, tengamos que estimar unos parámetros, los parámetros de confianza normalizados, no siendo ya un método automático como era el que teníamos para el caso de una distribución plana. Pero, si bien el método no proporciona una forma de estimar esos parámetros, tampoco es necesario puesto que nos permite usar bien los parámetros estimados en el codificador, bien los estimados en el decodificador, de forma totalmente automática y sin necesidad de estimar estos parámetros de confianza. Además permite al usuario realizar una combinación ambos conjuntos de parámetros, dependiendo de su experiencia previa, permitiendo por tanto una flexibilidad mucho mayor que el método propuesto en el capítulo anterior.

Por supuesto, aunque en esta memoria sólo desarrollemos el proceso de incorporación de información del codificador para el método de reconstrucción adaptativa con diferentes pará-

imagen	$\alpha_c^{cod^{-1}}$	$\alpha_r^{cod^{-1}}$	$\beta^{cod^{-1}}$
boats	100.215	13.977	30.839
couple	19.007	14.323	78.728
lena	174.795	44.967	2.619

Tabla 5.1: Valores de los parámetros obtenidos en el codificador para las diferentes imágenes de muestra del conjunto de prueba

metros para las filas y las columnas en el modelo de imagen y un único parámetro en el modelo de ruido, este mismo proceso se puede aplicar a los demás métodos obteniendo, en todos los casos, una combinación lineal entre las inversas de los parámetros transmitidos y los estimados en el decodificador.

En la siguiente sección estudiaremos el funcionamiento del método propuesto sobre un conjunto de imágenes reales.

5.4 Resultados experimentales

En esta sección presentaremos experimentos que muestren el funcionamiento de los métodos reconstrucción propuestos. Para ello usaremos el mismo conjunto de imágenes de prueba que en el capítulo anterior, escogiendo también los mismos representantes de cada grupo de imágenes de alta (G1), media (G2) y baja (G3) actividad espacial.

El lector puede consultar los resultados completos sobre todas las imágenes en el URL

`ftp://nesos.ugr.es/pub/tesis/resultados/codificador`

5.4.1 Resultados con los parámetros estimados en el codificador

El primer experimento realizado consiste en estimar los parámetros en el codificador, transmitirlos y realizar la reconstrucción de la imagen en el decodificador usando estos parámetros, tal y como se describió en la sección 5.2. La tabla 5.1 muestra los valores de los hiperparámetros obtenidos en el codificador, usando la imagen original como observación, para las diferentes imágenes de muestra.

Análisis del $PSNR$ en el codificador y decodificador

Las figuras 5.1, 5.2 y 5.3 muestran la comparación entre los valores del $PSNR$ de la imagen original y las reconstrucciones con los parámetros estimados en el codificador y en el decodificador.

Cuando se usan los parámetros estimados en el codificador, casi en la totalidad de las reconstrucciones aumenta el valor de esta medida respecto a las imágenes comprimidas, con la excepción de la imagen *couple* comprimida a 0.50bpp en la que, como ya ocurrió en los experimentos realizados en el capítulo anterior, se observa una ligera disminución en el $PSNR$ aunque la calidad visual presente una ligera mejora.

Sin embargo, el $PSNR$ suele ser algo menor con estos parámetros estimados en el codificador que usando los parámetros estimados en el decodificador. Esto es debido, en la mayoría de los casos, a que el valor del parámetro de ruido estimado en el codificador suele subestimar el ruido que presenta la imagen comprimida puesto que ha sido estimado a partir de la imagen original que no presenta el efecto de bloques, causante del ruido.

5.4.2 Resultados incorporando información de los parámetros estimados en el codificador

Procedemos, entonces, a combinar los hiperparámetros obtenidos en el codificador con los obtenidos en el decodificador usando el algoritmo 5.2. Esa combinación se obtiene variando los parámetros μ_c, μ_r y ν (definidos en las Ecs. 5.11, 5.12 y 5.13), en el intervalo $[0, 1]$. Observemos que su un parámetro, por ejemplo ν, es cero, significa que no se tiene ninguna confianza en los valores transmitidos o, lo que es lo mismo, n^{cod} bien no se ha transmitido bien no se usa, mientras que $\nu = 1$ no realizándose ninguna estimación de los hiperparámetros en el decodificador, usándose sólo el valor de n^{cod} transmitido. Lo mismo es aplicable a μ_c y μ_r. En estos experimentos usamos $\mu = \mu_c = \mu_r$ y ν pertenecientes a {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}.

Análisis del $PSNR$

Las figuras 5.4, 5.5 y 5.6 muestran la evolución del $PSNR$ en función de los valores de $\mu = \mu_c = \mu_r$ y ν para las imágenes de muestra de cada uno de los grupos.

Cuando se usa la información los parámetros estimados en el codificador para dirigir la

estimación de los parámetros en el decodificador se observa, en la mayoría de los casos, el mismo comportamiento; se produce una subida significativa en el valor del $PSNR$ al aumentar la confianza en los parámetros del modelo de imagen estimados en el codificador (α_c^{cod} y α_r^{cod}), es decir, el valor del $PSNR$ es una función creciente de μ salvo para valores de μ cercanos a 1 donde se produce un leve decrecimiento. Por otra parte, confiar en el valor del parámetro del modelo de ruido estimado en el codificador, β^c, hace decrecer el valor del $PSNR$ de forma notable. Esto es debido, como se comentó anteriormente, a que la estimación en el codificador se realiza sobre la imagen original, que suele presentar menos ruido que la imagen comprimida, al menos para razones de compresión altas y medias, por lo que se tiende a subestimar el valor del parámetro de ruido. Sin embargo, la estimación de los parámetros del modelo de imagen se realiza de forma más precisa en el codificador puesto que la imagen original no presenta el efecto de bloques pudiéndose medir de forma más real la diferencia entre los pixels adyacentes.

Muchas de las imágenes comprimidas con una razón baja, sin embargo, no presentan este comportamiento general. En estos casos aumenta el $PSNR$ al aumentar la confianza tanto en los parámetros del modelo de imagen como en los del modelo de ruido, alcanzando su máximo para $\mu = \nu = 1$, o en un entorno cercano a estos valores. Esto es debido a que, para razones de compresión bajas, el valor del ruido estimado en el codificador se acerca más al ruido real de la imagen comprimida produciéndose, por tanto, mejores resultados.

La excepción a estas reglas la presenta, de nuevo, la imagen *couple* en la que la forma de la gráfica se invierte respecto a la regla general, es decir, el $PSNR$ aumenta conforme ν aumenta disminuyendo conforme aumenta μ (ver figura 5.5). En este caso concreto, esta tendencia se puede explicar teniendo en cuenta que muchas de las fronteras naturales de la imagen están situadas justamente en fronteras de bloques por lo que, al reconstruir la imagen no solamente estamos eliminando el efecto bloque sino que, además estamos suavizando las fronteras naturales de la imagen. Por eso, cuanto más tenemos en cuenta los parámetros de suavidad obtenidos en el codificador, que nos dicen que, en general, la imagen es suave (puesto que muchos de los bloques están situados en zonas con poca actividad espacial) la imagen reconstruida presenta un menor $PSNR$.

Este problema no es en sí, defecto del método de reconstrucción, que en la mayoría de los casos mejora el $PSNR$ respecto a la imagen comprimida, sino a la elección de los pesos que ponderan las diferencias en el modelo de imagen. Para esta imagen, los pesos de los bloques que están a ambos lados de las fronteras naturales son relativamente altos, debido a

que la varianza de los bloques es pequeña puesto que las fronteras naturales coinciden con las fronteras de los bloques y no están, como en la mayoría de las otras imágenes, en el interior de los mismos.

Posibles soluciones a este problema pasarían por escoger una forma de cálculo de los pesos que tuviese en cuenta las fronteras naturales de la imagen además de las características del sistema visual humano. Sin embargo, este tema queda fuera del ámbito de esta memoria y se propone como un trabajo a realizar en el futuro.

5.4.3 Resultados incorporando información cuantificada proveniente del codificador

Se realizaron también los mismos experimentos usando dos versiones cuantificadas de los parámetros estimados en el codificador. Suponiendo que se usan 8 bits para representar el valor de cada parámetro, la primera versión de los parámetros cuantificados se obtuvo transmitiendo sólo los 3 bits más significativos (lo que equivale a una coeficiente de cuantificación 32) y la segunda versión se obtuvo transmitiendo los 6 bits más significativos (equivalente a un coeficiente de cuantificación 4).

La evolución del $PSNR$ en función de los valores de μ y ν cuando se aplica el algoritmo 5.2 sobre las imágenes de muestra de cada grupo, usando los parámetros estimados en el codificador cuantificados a 3 bits y a 6 bits, se presentan en las figuras 5.11, 5.12 y 5.13. Nótese que si, al cuantificar, nos sale un valor 0 para el parámetro usamos un valor 0.1 para evitar divisiones por cero cuando se aplica el proceso de reconstrucción.

Cuando se usan los valores cuantificados de los parámetros obtenidos en el codificador, los resultados son bastante semejantes a los obtenidos con los parámetros originales aunque se observa una disminución general en el $PSNR$ de las reconstrucciones.

5.4.4 Análisis visual

La parte central de cada imagen comprimida así como las reconstrucciones usando los parámetros estimados en el codificador, en el decodificador y la mejor reconstrucción en términos de $PSNR$ obtenida mediante la aplicación del Algoritmo 5.2 con los diferentes valores de μ y ν, se muestran en las figuras 5.7, 5.8 y 5.9.

Todas las reconstrucciones muestran un gran aumento en su calidad visual tanto al usar los parámetros estimados en el codificador como al combinar éstos con los obtenidos en el

decodificador y, si bien todas tiene una calidad visual muy semejante, especialmente en las imágenes con razones de compresión media y baja, se puede observar una pequeña disminución de la calidad visual en algunas de las imágenes reconstruidas, especialmente a altas razones de compresión, cuando se usan los parámetros estimados en el codificador. Esto es debido, al igual que la disminución que se producía en el $PSNR$, a que el parámetro de ruido estimado en el codificador no es la mejor estimación posible para la imagen obtenida en el decodificador.

Cuando se combinan los parámetros estimados en el codificador con los obtenidos en el decodificador, en la mayoría de las imágenes de prueba se ha observado una ligera mejora. Esta mejora es más clara en las imágenes comprimidas a una razón alta (0.20bpp) donde las zonas sin gran número de detalles se ven más suaves mientras que se mantiene una buena calidad en las zonas con alto nivel de detalle dado que en la mayoría de los casos se usa la información de los parámetros del modelo de ruido estimados en el codificador mientras que se usa el valor del parámetro de ruido estimado en el decodificador.

La excepción, de nuevo, es la imagen *couple*. En este caso, la imagen con peor $PSNR$ presenta mejor calidad visual en las zonas sin fronteras naturales que la que tiene mejor $PSNR$, como se puede observar en la figura 5.10 la imagen reconstruida con $\mu = 1$ y $\nu = 0$, es decir, la que produce un $PSNR$ más bajo presenta mejor calidad visual en muchas zonas que la imagen con mejor $PSNR$ (obtenida con $\mu = 0$ y $\nu = 0.2$). Obsérvese, por ejemplo, las piernas y cara del hombre, las piernas de las mujer o el sillón tras el hombre. Sin embargo, también difumina más las fronteras naturales de la imagen. Obsérvese, por ejemplo, la cabeza de los dos personajes en la imagen o los contornos generales de los objetos.

Cuando se usaron los valores cuantificados de los parámetros obtenidos en el codificador como entrada al algoritmo 5.2, la calidad visual de las reconstrucciones presentó una pequeña disminución si bien ésta es difícilmente apreciable para las razones de compresión media y baja.

Por tanto, podemos ver que, en general, obtener el valor de los parámetros del modelo de imagen en el codificador, transmitirlos y usarlos en el decodificador producirá imágenes con mayor $PSNR$ y mayor calidad visual que las obtenidas estimando todos los parámetros en el decodificador, incluso cuando se transmite una versión cuantificada de estos parámetros.

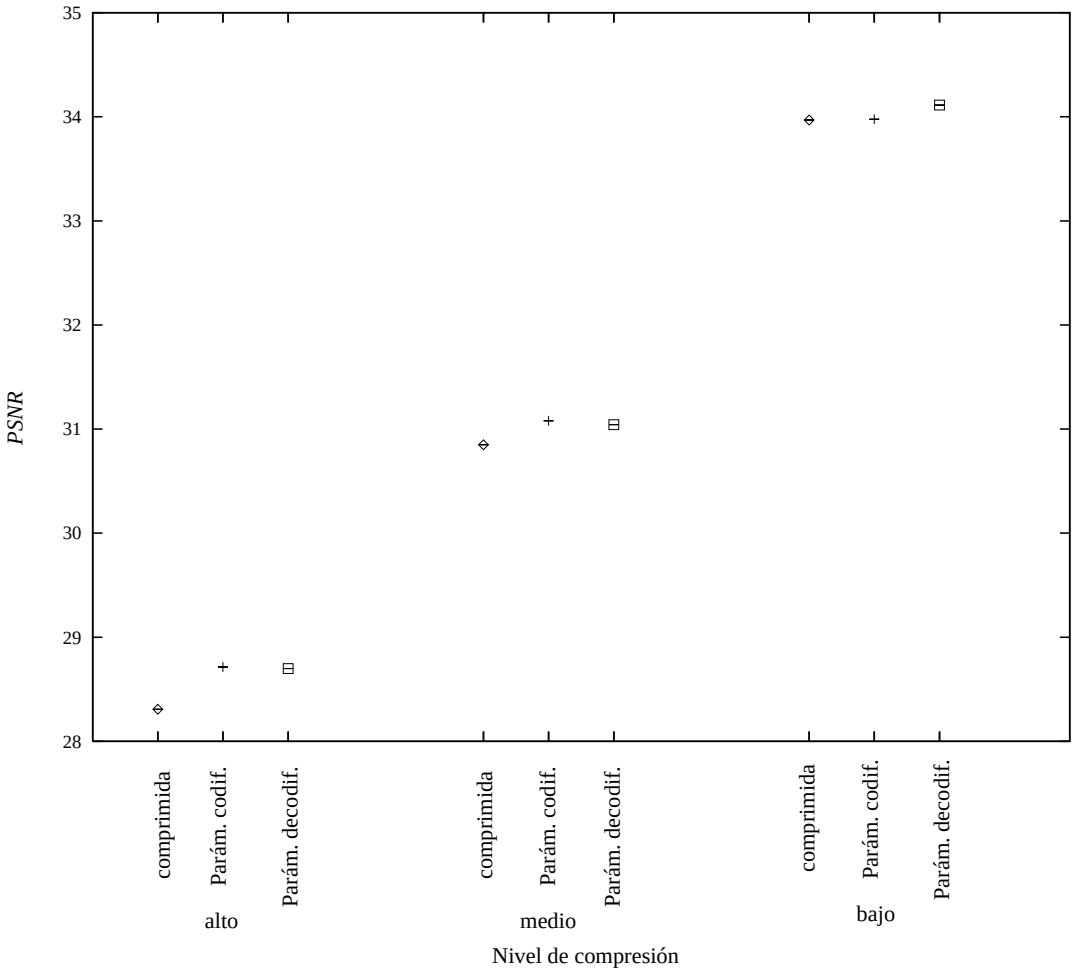


Figura 5.1: Comparación de los resultados de la reconstrucción, expresados en $PSNR$, con los parámetros estimados en el codificador y en el decodificador para la imagen $boats$ a diferentes razones de compresión.

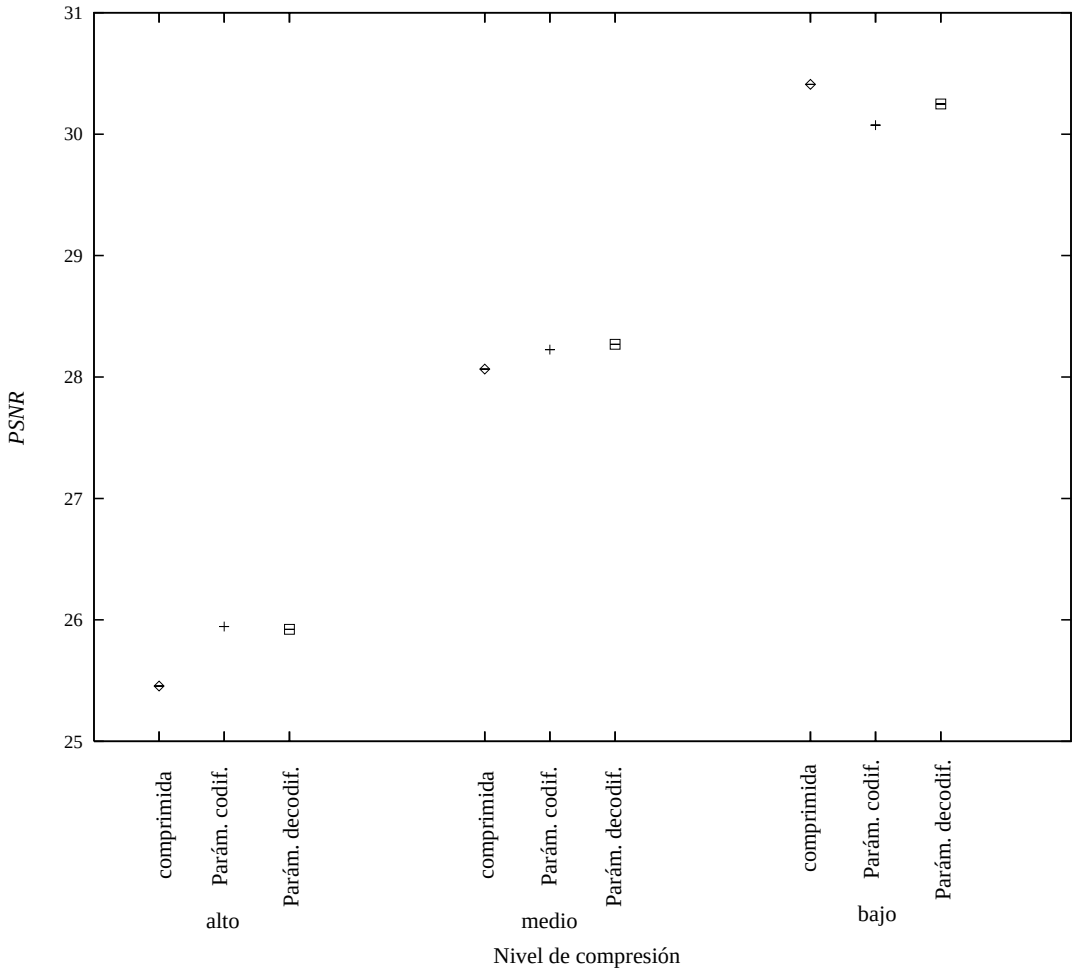


Figura 5.2: Comparación de los resultados de la reconstrucción, expresados en $PSNR$, con los parámetros estimados en el codificador y en el decodificador para la imagen $couple$ a diferentes razones de compresión.

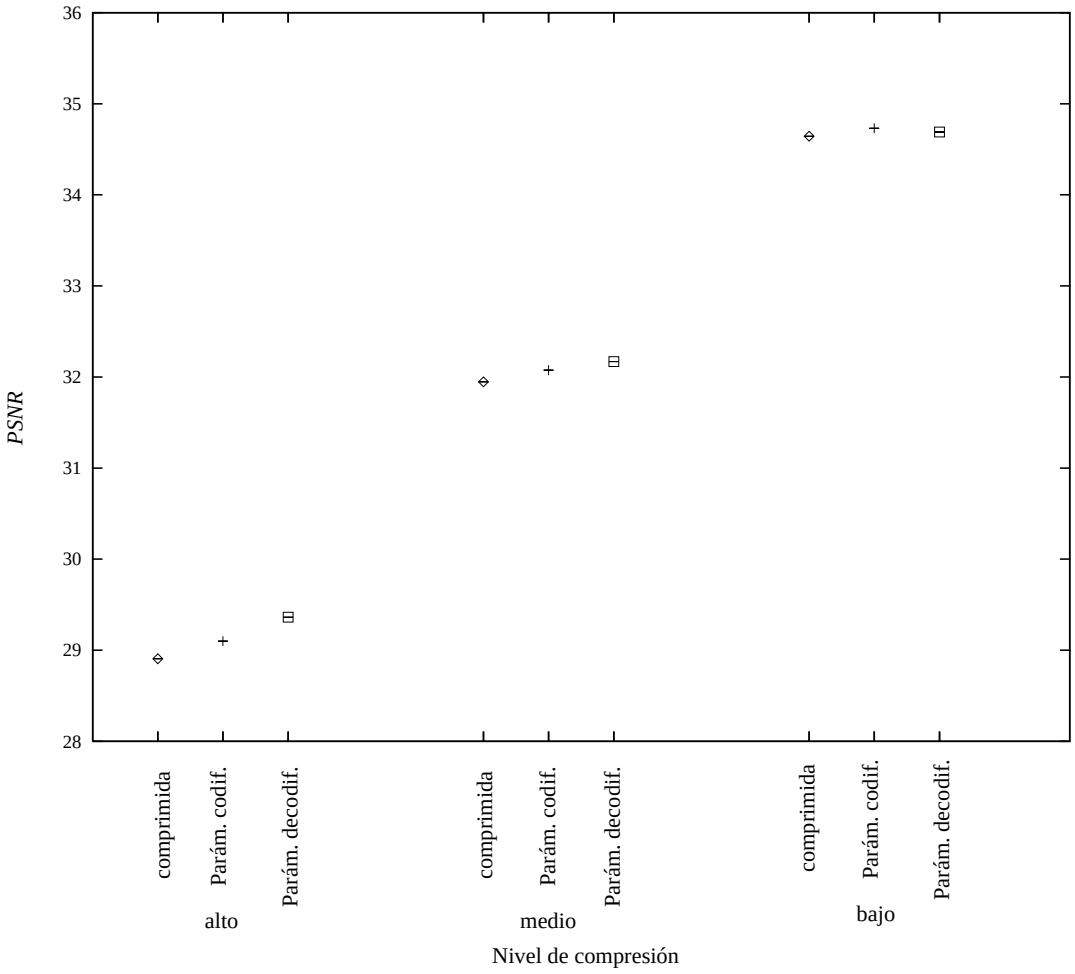


Figura 5.3: Comparación de los resultados de la reconstrucción, expresados en $PSNR$, con los parámetros estimados en el codificador y en el decodificador para la imagen $lena$ a diferentes razones de compresión.

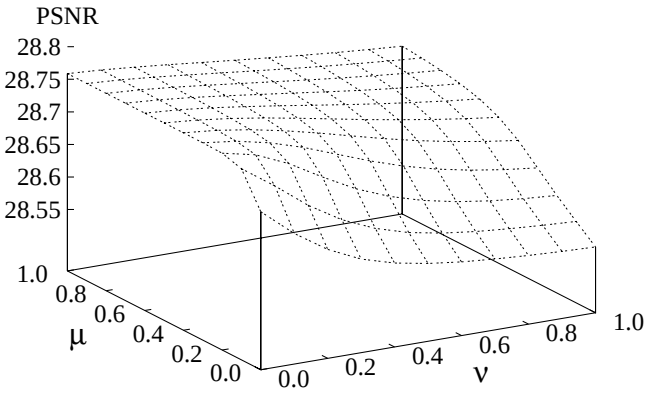


Figura 5.4: Evolución del $PSNR$ en función de los valores de μ y ν para la imagen $boats$ comprimida a 0.20bpp usando los parámetros obtenidos en el codificador ($m_c^{cod} = \alpha_c^{cod} = 100.215^{-1}$, $m_r^{cod} = \alpha_r^{cod} = 13.977^{-1}$ y $n^{cod} = \beta^{cod} = 30.839^{-1}$).

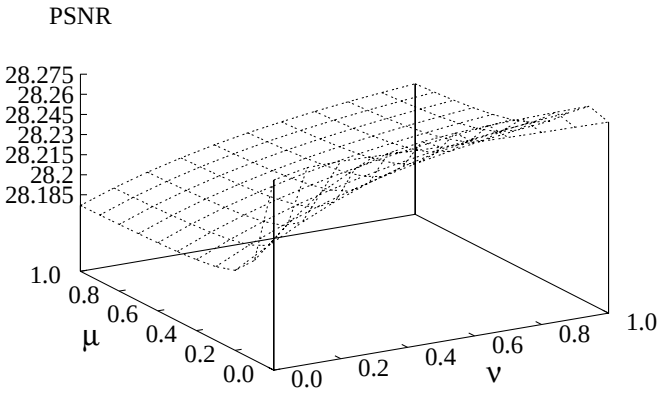


Figura 5.5: Evolución del $PSNR$ en función de los valores de μ y ν para la imagen $couple$ comprimida a 0.30bpp usando los parámetros obtenidos en el codificador ($m_c^{cod} = \alpha_c^{cod} = 19.007^{-1}$, $m_r^{cod} = \alpha_r^{cod} = 14.323^{-1}$ y $n^{cod} = \beta^{cod} = 78.728^{-1}$).

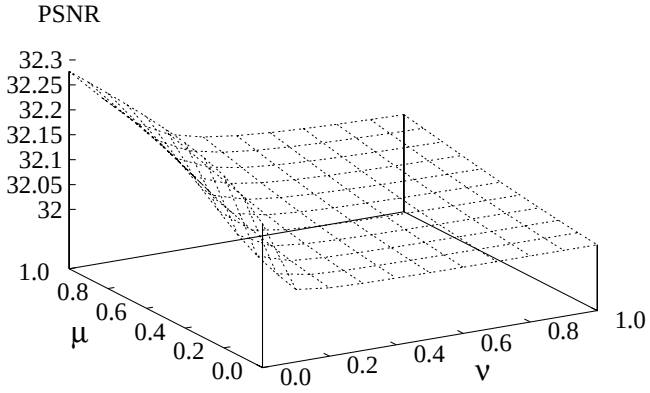


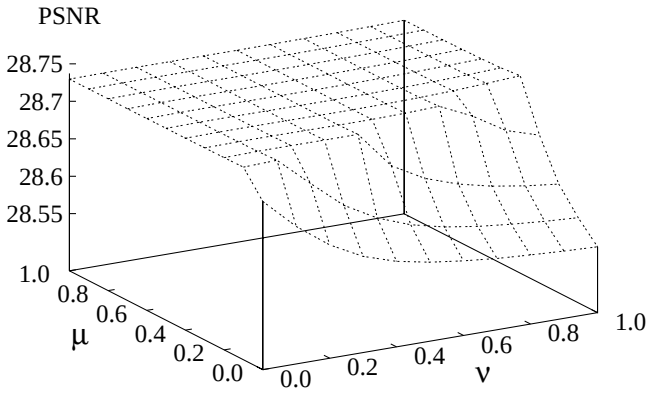
Figura 5.6: Evolución del $PSNR$ en función de los valores de μ y ν para la imagen $lena$ comprimida a 0.30bpp usando los parámetros obtenidos en el codificador ($m_c^{cod} = \alpha_c^{cod} = 174.795^{-1}$, $m_r^{cod} = \alpha_r^{cod} = 44.967^{-1}$ y $n^{cod} = \beta^{cod} = 2.619^{-1}$).

Figura 5.7: De izquierda a derecha y de arriba a abajo: Imagen $boats$ comprimida a 0.20bpp, $PSNR = 28.308$. Reconstrucción con el Algoritmo 4.2 con los parámetros estimados en el decodificador, $PSNR = 28.698$. Reconstrucción con el Algoritmo 4.2 con los parámetros estimados en el codificador, $PSNR = 28.713$. Mejor reconstrucción en términos de $PSNR$ con el Algoritmo 5.2, $\mu = 0.4$, $\nu = 0.0$, $PSNR = 28.759$.

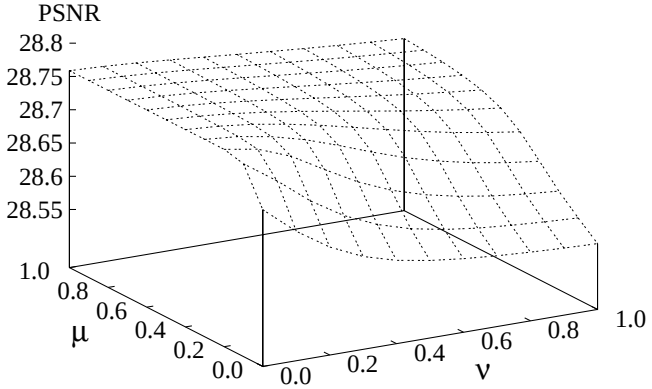
Figura 5.8: De izquierda a derecha y de arriba a abajo: Imagen $couple$ comprimida a 0.30bpp, $PSNR = 28.065$. Reconstrucción con el Algoritmo 4.2 con los parámetros estimados en el decodificador, $PSNR = 28.270$. Reconstrucción con el Algoritmo 4.2 con los parámetros estimados en el codificador, $PSNR = 28.225$. Mejor reconstrucción en términos de $PSNR$ con el Algoritmo 5.2, $\mu = 0.0$, $\nu = 0.2$, $PSNR = 28.275$.

Figura 5.9: De izquierda a derecha y de arriba a abajo: Imagen $lena$ comprimida a 0.30bpp, $PSNR = 31.947$. Reconstrucción con el Algoritmo 4.2 con los parámetros estimados en el decodificador, $PSNR = 32.169$. Reconstrucción con el Algoritmo 4.2 con los parámetros estimados en el codificador, $PSNR = 32.076$. Mejor reconstrucción en términos de $PSNR$ con el Algoritmo 5.2, $\mu = 1.0$, $\nu = 0.0$, $PSNR = 32.277$.

Figura 5.10: De izquierda a derecha y de arriba a abajo: Imagen $couple$ comprimida a 0.30bpp, $PSNR = 28.065$. Reconstrucción con el Algoritmo 4.2 con los parámetros estimados en el codificador, $PSNR = 28.225$. Mejor reconstrucción en términos de $PSNR$ con el Algoritmo 5.2, $\mu = 0.0$, $\nu = 0.2$, $PSNR = 28.275$. Reconstrucción en términos de $PSNR$ con el Algoritmo 5.2, $\mu = 1.0$, $\nu = 0.0$, $PSNR = 28.177$.

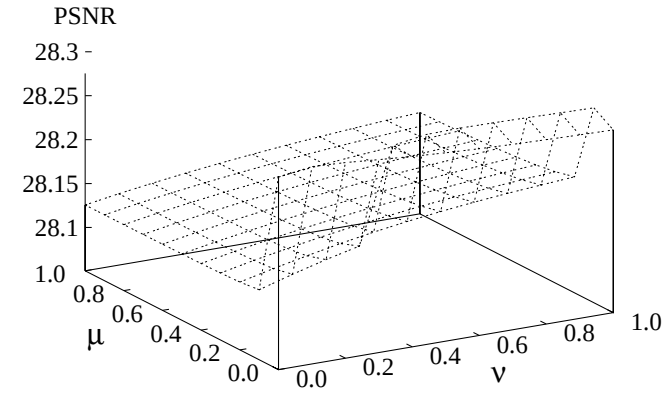


(a)

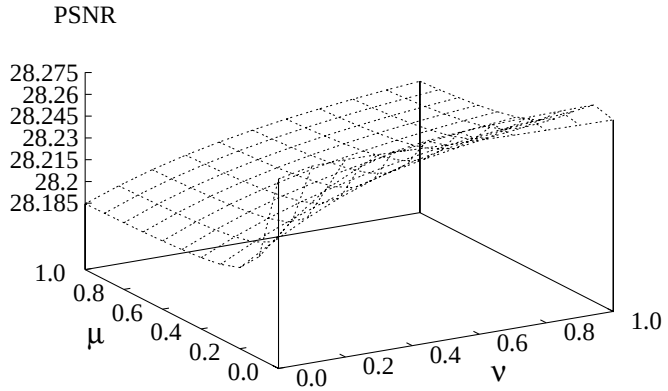


(b)

Figura 5.11: Evolución del $PSNR$ en función de los valores de μ y ν para la imagen $boats$ comprimida a 0.20bpp. (a) usando los parámetros estimados en el codificador cuantificados a 3 bits ($m_c^{cod} = 96^{-1}$, $m_r^{cod} = 0.1^{-1}$ y $n^{cod} = 32^{-1}$). (b) usando los parámetros estimados en el codificador cuantificados a 6 bits ($m_c^{cod} = 100^{-1}$, $m_r^{cod} = 12^{-1}$ y $n^{cod} = 32^{-1}$).

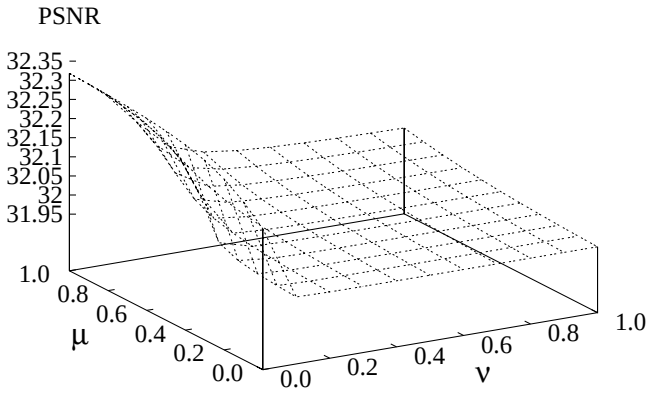


(a)

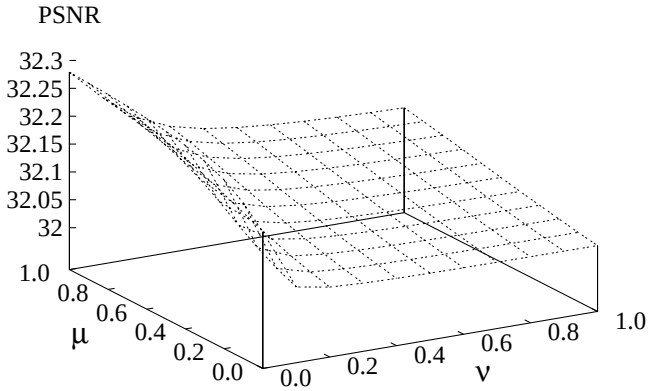


(b)

Figura 5.12: Evolución del $PSNR$ en función de los valores de μ y ν para la imagen $couple$ comprimida a 0.30bpp. (a) usando los parámetros estimados en el codificador cuantificados a 3 bits ($m_c^{cod} = 32^{-1}$, $m_r^{cod} = 0.1^{-1}$ y $n^{cod} = 64^{-1}$). (b) usando los parámetros estimados en el codificador cuantificados a 6 bits ($m_c^{cod} = 20^{-1}$, $m_r^{cod} = 16^{-1}$ y $n^{cod} = 80^{-1}$).



(a)



(b)

Figura 5.13: Evolución del $PSNR$ en función de los valores de μ y ν para la imagen $lena$ comprimida a 0.30bpp. (a) usando los parámetros estimados en el codificador cuantificados a 3 bits ($m_c^{cod} = 160^{-1}$, $m_r^{cod} = 32^{-1}$ y $n^{cod} = 0.1^{-1}$). (b) usando los parámetros estimados en el codificador cuantificados a 6 bits ($m_c^{cod} = 176^{-1}$, $m_r^{cod} = 44^{-1}$ y $n^{cod} = 4^{-1}$).

Capítulo 6

Reconstrucción de imágenes en color

6.1 Introducción

La mayoría de los algoritmos desarrollados para la reducción de los artificios de los bloques tratan solamente con imágenes en escala de grises y, si se aplican a imágenes en color, sólo procesan la banda de luminosidad de la imagen pero no modifican las bandas de información cromática.

En este capítulo queremos dar un primer paso, que sirva como base de trabajos posteriores, para la reconstrucción de imágenes en color. Sólo abordaremos el problema de la reconstrucción en el decodificador, proponiendo la extensión del método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido, descrito en la sección 4.3, para la reconstrucción de imágenes en color. Utilizaremos este método al ser el que mejores resultados nos ha dado para la estimación de la imagen y de los hiperparámetros en el decodificador.

Como ya indicamos en el capítulo 1, en las especificaciones de JPEG para imágenes en color, se describe que el algoritmo de compresión es independiente del espacio de color, aunque se suele usar la codificación que separe la información de luminosidad de la información cromática de la imagen, como la YCbCr (ver [34]). Según estas especificaciones cada banda de la imagen se codifica independientemente, es decir, se realiza el mismo tipo de codificación para la banda de luminosidad y para las bandas de cromatismo, aunque usando diferentes matrices de cuantificación. En JPEG, además, se suele aplicar un muestreo a las bandas de cromatismo, usualmente notada como 4:2:2 o 4:1:1. El formato 4:2:2 especifica que, para ca-

da cuatro muestras de la banda de información Y, hay dos muestras de las bandas Cb y Cr mientras que, en el formato 4:1:1, para cada cuatro muestras de Y, hay una muestra de Cb y Cr (ver [12]).

En principio parece apropiado extender el algoritmo descrito en la sección 4.3 simplemente procesando cada banda independientemente y esto es lo que se hará en este capítulo. Es obvio que una posible extensión es tener en cuenta la correlación entre las bandas y usar técnicas de reconstrucción multicanal como las utilizadas en [85, 86, 87] y [21, 22, 26] o utilizar técnicas de interpolación sobre los pixels de las bandas cromáticas, en lugar de, simplemente, duplicar los valores de los pixels al reconstruir una banda a la que se le ha aplicado un muestreo 4:1:1. Sin embargo todo esto se deja para trabajos posteriores.

El resto del capítulo se divide como sigue: En la sección 6.2 se presenta la notación necesaria, en la sección 6.3 se extienden los modelos de imagen y ruido para ajustarse a imágenes en color, en la sección 6.4 se propone un algoritmo para reconstruir las imágenes y en la sección 6.5 se muestran resultados experimentales sobre imágenes reales usando este método y se extraen conclusiones sobre los mismos.

6.2 Notación

Puesto que una imagen en color está formada por tres bandas diferentes en esta sección definiremos la notación necesaria para caracterizar los pixels en la frontera de los bloques para cada una de las bandas, generalizando, de esta forma, la notación descrita en la sección 3.2 y usada a lo largo de toda esta memoria.

Sea $\mathbf{f}$ una imagen en color con tres bandas, cada una de las cuales se denotará por $\mathbf{f}^I$, $I \in \{Y, Cb, Cr\}$, y sea $\mathbf{g}$ la imagen comprimida, cuyas bandas se denotarán por $\mathbf{g}^I$, $I \in \{Y, Cb, Cr\}$. Asumamos que usamos bloques $k \times k$ para codificar una imagen en la que el tamaño original de cada una de las bandas es $U \times V$, U y V múltiplos de k.

Para un muestreo 4:1:1 notaremos que, para la banda de luminosidad Y, se procesará una imagen de tamaño $M^Y \times N^Y$, con $M^Y = U$ y $N^Y = V$, mientras para que para las bandas de cromatismo las imágenes serán de tamaño $M^I \times N^I$ donde $M^I = U/2$ y $N^I = V/2$ con $I \in \{Cb, Cr\}$. No obstante, en ambos casos, el tamaño de los bloques usados es $k \times k$.

Extendamos ahora la notación utilizada en temas anteriores para su uso con imágenes en color.

Para $I \in \{Y, Cb, Cr\}$, sean $\mathbf{f}_{cl}^I$ y $\mathbf{g}_{cl}^I$ vectores columna definidos por el apilamiento de todos los elementos de $\mathbf{f}^I$ y $\mathbf{g}^I$, respectivamente, que están a la izquierda de una frontera vertical de bloque, cl, pero no en la intersección de cuatro bloques, es decir,

$$\begin{aligned}\mathbf{f}_{cl}^I &= \{\mathbf{f}^I(u)\},\ \mathbf{g}_{cl}^I = \{\mathbf{g}^I(u)\},\ u = (x, y), x = k * i + l,\ y = k * j,\\ &\quad i = 0, 1, 2, \ldots, M^I/k - 1,\ j = 1, 2, \ldots, N^I/k - 1,\ l = 2, 3, \ldots, k - 1\}.\end{aligned}$$

Por tanto, el vector $\mathbf{f}_{cl}^I$ tendrá tamaño $p_I \times 1$, con $p_I = (k-2)(M^I/k) \times (N^I/k - 1)$

Análogamente definimos $\mathbf{f}_{cr}^I$ y $\mathbf{g}_{cr}^I$, apilando todos los elementos de $\mathbf{f}^I$ y $\mathbf{g}^I$, respectivamente, que están a la derecha de una frontera vertical de bloque, cr, pero no en la intersección de cuatro bloque.

De forma similar definiremos $\mathbf{f}_{ra}^I$, $\mathbf{g}_{ra}^I$, $\mathbf{f}_{rb}^I$ y $\mathbf{g}_{rb}^I$, los vectores columna, $q_I \times 1$, con $q_I = (k-2)(N^I/k) \times (M^I/k - 1)$, que representan las filas de encima y debajo de una frontera horizontal de bloque de $\mathbf{f}^I$ y $\mathbf{g}^I$, respectivamente.

Además apilamos los elementos de $\mathbf{f}^I$ que están sobre una frontera horizontal y a la izquierda de una frontera vertical en una intersección de cuatro bloques, indicada por al, en el vector $\mathbf{f}_{al}^I$ definido como

$$\mathbf{f}_{al}^I = \{\mathbf{f}^I(u) \mid u = (x, y)\},$$

con $x = k * i$, $y = k * j$, $i = 1, 2, \ldots, M^I/k - 1$, $j = 1, 2, \ldots, N^I/k - 1$.

De forma semejante definimos los vectores $\mathbf{f}_{ar}^I$, formado por los elementos de $\mathbf{f}^I$ situados sobre una frontera horizontal y a la derecha de una frontera vertical, $\mathbf{f}_{br}^I$, bajo una frontera horizontal y a la derecha de una frontera vertical y, $\mathbf{f}_{bl}^I$, bajo una frontera horizontal y a la izquierda de una frontera vertical en una intersección de cuatro bloques. Siguiendo el mismo procedimiento, definimos los vectores de la observación para esos pixels en la intersección de cuatro bloques, $\mathbf{g}_{al}^I$, $\mathbf{g}_{ar}^I$, $\mathbf{g}_{br}^I$ y $\mathbf{g}_{bl}^I$. Todos estos vectores tiene tamaño $m_I \times 1$, con $m_I = (M^I/k - 1) \times (N^I/k - 1)$.

Una vez definidos los elementos que necesitamos para realizar la reconstrucción y la notación que usaremos, estudiemos los elementos necesarios para realizar la reconstrucción según el paradigma bayesiano jerárquico, es decir, los modelos de imagen y ruido ya que, puesto que en este capítulo sólo abordaremos la reconstrucción de la imagen en el decodificador, el modelo sobre los hiperparámetros será una distribución uniforme, como la definida en la Ec. (3.32).

6.3 Modelos de imagen y ruido

El modelo global que usaremos en este capítulo es el modelo adaptativo con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido aunque extendido para cumplir su función para una imagen con tres bandas de color.

Para captar las propiedades locales de la imagen definimos, de nuevo, una serie de matrices diagonales de pesos que ponderarán cada uno de los pixels en una frontera de bloque en función de las propiedades de la imagen en esa zona. Así, para captar las propiedades locales verticales de la imagen, definiremos la matriz diagonal $p_I \times p_I$, $\mathbf{W}_c^I$, como

$$\mathbf{W}_c^I = \begin{bmatrix} \omega_c^I(1) & 0 & 0 & \cdots & 0 \\ 0 & \omega_c^I(2) & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \omega_c^I(p_I) \end{bmatrix},$$

donde los $\omega_c^I(i)$, $i = 1, 2, \ldots, p_I$, ponderarán cada diferencia entre los vectores que representan las columnas. Nótese que los pixels en una intersección de cuatro bloques no se incluyen en esta matriz.

De forma análoga podemos definir la matriz diagonal, $q_I \times q_I$, $\mathbf{W}_r^I$ para captar las propiedades locales horizontales de la imagen, sin incluir los pixels en la intersección de cuatro bloques. y las matrices $\mathbf{W}_{x1}^I, \mathbf{W}_{x2}^I, \mathbf{W}_{x3}^I$ y $\mathbf{W}_{x4}^I$ para captar las propiedades locales de la imagen en las intersecciones de cuatro bloques, todas ellas de tamaño $m_I \times m_I$, como se describió en la sección 3.2.

Usando las definiciones anteriores, el conocimiento a priori sobre la suavidad en las fronteras de los bloques para cada banda $I \in \{Y, Cb, Cr\}$, de la imagen tiene la forma

$$\mathrm{p}(\mathbf{f}^I \mid \alpha_c^I, \alpha_r^I) = {\alpha_c^I}^{\frac{p_I}{2}} {\alpha_r^I}^{\frac{q_I}{2}} F^I(\alpha_c^I, \alpha_r^I) \exp\left\{-P^I(\mathbf{f}^I \mid \alpha_c^I, \alpha_r^I)\right\} \tag{6.1}$$

con

$$\begin{aligned} P^I(\mathbf{f}^I \mid \alpha_c^I, \alpha_r^I) &= \alpha_c^I \parallel \mathbf{W}_c^I(\mathbf{f}_{cl}^I - \mathbf{f}_{cr}^I) \parallel^2 + \alpha_r^I \parallel \mathbf{W}_r^I(\mathbf{f}_{ra}^I - \mathbf{f}_{rb}^I) \parallel^2 \\ &+ \frac{1}{2}\Big\{\alpha_c^I \parallel \mathbf{W}_{x1}^I(\mathbf{f}_{al}^I - \mathbf{f}_{ar}^I) \parallel^2 + \alpha_r^I \parallel \mathbf{W}_{x2}^I(\mathbf{f}_{ar}^I - \mathbf{f}_{br}^I) \parallel^2 \\ &+ \alpha_c^I \parallel \mathbf{W}_{x3}^I(\mathbf{f}_{br}^I - \mathbf{f}_{bl}^I) \parallel^2 + \alpha_r^I \parallel \mathbf{W}_{x4}^I(\mathbf{f}_{bl}^I - \mathbf{f}_{al}^I) \parallel^2\Big\}, \end{aligned} \tag{6.2}$$

y

$$F^I(\alpha_c^I, \alpha_r^I) = \prod_{i=1}^{m_I} \left[\alpha_c^I \alpha_r^I \left(\alpha_c^I \left(\frac{1}{{\omega_{x2}^I}^2(i)} + \frac{1}{{\omega_{x4}^I}^2(i)} \right) + \alpha_r^I \left(\frac{1}{{\omega_{x1}^I}^2(i)} + \frac{1}{{\omega_{x3}^I}^2(i)} \right) \right) \right]^{\frac{1}{2}}, \quad (6.3)$$

La fidelidad a los datos en las fronteras de los bloques para cada banda $I \in \{Y, Cb, Cr\}$ viene expresada por

$$\mathrm{p}(\mathbf{g}^I \mid \mathbf{f}^I, \beta^I) \propto {\beta^I}^{(p_I + q_I + 2m_I)} \exp\left\{ -N^I(\mathbf{g}^I \mid \mathbf{f}^I, \beta^I) \right\}, \quad (6.4)$$

donde

$$\begin{aligned} N^I(\mathbf{g}^I \mid \mathbf{f}^I, \beta^I) &= \frac{1}{2}\beta \Big\{ \| \mathbf{g}_{cl}^I - \mathbf{f}_{cl}^I \|^2 + \| \mathbf{g}_{cr}^I - \mathbf{f}_{cr}^I \|^2 + \| \mathbf{g}_{ra}^I - \mathbf{f}_{ra}^I \|^2 + \| \mathbf{g}_{rb}^I - \mathbf{f}_{rb}^I \|^2 \\ &+ \| \mathbf{g}_{al}^I - \mathbf{f}_{al}^I \|^2 + \| \mathbf{g}_{ar}^I - \mathbf{f}_{ar}^I \|^2 + \| \mathbf{g}_{br}^I - \mathbf{f}_{br}^I \|^2 + \| \mathbf{g}_{bl}^I - \mathbf{f}_{bl}^I \|^2 \Big\}, \quad (6.5) \end{aligned}$$

Por tanto el modelo global que vamos a usar se define como

$$\mathrm{p}(\mathbf{f}, \mathbf{g} \mid \alpha_c, \alpha_r, \beta) = \prod_{I \in \{Y, Cb, Cr\}} \mathrm{p}(\mathbf{f}^I, \mathbf{g}^I \mid \alpha_c^I, \alpha_r^I, \beta^I)$$

donde

$$\begin{aligned} \mathrm{p}(\mathbf{f}^Y, \mathbf{g}^Y \mid \alpha_c^Y, \alpha_r^Y, \beta^Y) &\propto \mathrm{p}(\mathbf{f}^Y \mid \alpha_c^Y, \alpha_r^Y)\mathrm{p}(\mathbf{g}^Y \mid \mathbf{f}^Y, \beta^Y), \\ \mathrm{p}(\mathbf{f}^{Cb}, \mathbf{g}^{Cb} \mid \alpha_c^{Cb}, \alpha_r^{Cb}, \beta^{Cb}) &\propto \mathrm{p}(\mathbf{f}^{Cb} \mid \alpha_c^{Cb}, \alpha_r^{Cb})\mathrm{p}(\mathbf{g}^{Cb} \mid \mathbf{f}^{Cb}, \beta^{Cb}), \\ \mathrm{p}(\mathbf{f}^{Cr}, \mathbf{g}^{Cr} \mid \alpha_c^{Cr}, \alpha_r^{Cr}, \beta^{Cr}) &\propto \mathrm{p}(\mathbf{f}^{Cr} \mid \alpha_c^{Cr}, \alpha_r^{Cr})\mathrm{p}(\mathbf{g}^{Cr} \mid \mathbf{f}^{Cr}.\beta^{Cr}), \end{aligned}$$

Nótese que este modelo define de forma independiente nuestro conocimiento sobre cada una de las bandas aunque la información sobre las bandas cromáticas se puede usar para reconstruir la banda de luminosidad (ver [125]) y sería posible aplicar técnicas de reconstrucción multicanal. Obsérvese además que la reconstrucción se hace sobre la imagen en el espacio de color YCbCr pero también sería posible realizarla sobre cualquier otro espacio de color

6.4 Reconstrucción en el decodificador

Una vez definido el modelo que vamos a usar, procedemos a aplicar el paradigma bayesiano jerárquico para la reconstrucción de la imagen en el decodificador usando una distribución plana para los hiperparámetros.

Puesto que, como dijimos al final de la sección anterior, el modelo para cada banda es independiente de las otras bandas, podemos realizar la reconstrucción de cada banda de forma separada mediante la aproximación de la evidencia descrita en la sección 4.3. Esto implica aplicar el algoritmo 4.2 a cada una de las bandas. Por tanto, se puede usar el siguiente algoritmo para reconstruir una imagen en color:

Algoritmo 6.1 Reconstrucción en el decodificador adaptativa con diferentes parámetros para las filas y las columnas en el modelo de imagen y un único parámetro en el modelo de ruido.

1. *Dividir la imagen* $\mathbf{g}$ *en tres imágenes* $\mathbf{g}^Y$, $\mathbf{g}^{Cb}$ *y* $\mathbf{g}^{Cr}$, *una para representar cada una de las tres bandas de la imagen.*

2. *Para* $I \in \{Y, Cb, Cr\}$, *ejecutar el algoritmo 4.2 usando* $\mathbf{g}^I$ *como observación.*

3. *La imagen reconstruida* $\mathbf{f}$, *es la imagen resultado de componer las tres bandas de la imagen,* $\mathbf{f}^Y$, $\mathbf{f}^{Cb}$ *y* $\mathbf{f}^{Cr}$.

6.5 Resultados experimentales

Habiendo propuesto un algoritmo para reconstruir imágenes en color, en esta sección vamos a realizar experimentos que determinen la bondad de la reconstrucción de este tipo de imágenes. Para ello compararemos los resultados con el algoritmo 6.1 propuesto en este capítulo con los resultados obtenidos de aplicar el algoritmo 4.2 solamente sobre la banda de luminosidad de las imágenes en color.

6.5.1 Imágenes utilizadas y criterios de bondad

Dada la dificultad de encontrar las mismas imágenes usadas como banco de pruebas en los capítulos anteriores en su versión en color, las pruebas se realizarán sobre un conjunto de tres imágenes en color con semejantes características a las empleadas como imagen de muestra en los capítulos anteriores. La figura 6.4 muestras estas tres imágenes. De nuevo tenemos una imagen con un gran nivel de detalle (*baboon*), una imagen con un nivel de detalle medio (*airplane*) y una imagen en la que predominan las zonas suaves (*lena*). Todas estas imágenes se encuentran codificadas a 24 bits por pixel.

Para determinar la bondad de los resultados obtenidos también será necesario en esta sección usar una medida objetiva de la distancia entre la imagen original y la imagen reconstruida. Dado que no hay constancia una medida única para medir esta distancia en imágenes en color nuestra propuesta será usar el pico de la relación señal-ruido ($PSNR$) para cada una de las bandas. De esta forma, el $PSNR$ para cada banda de dos imágenes $\mathbf{f}$ y $\mathbf{g}$ de tamaño $M \times N$ con un rango $[0, 255]$, se define como

$$PSNR^I = 10 \log_{10} \left[\frac{M \times N \times 255^2}{\| \mathbf{g}^I - \mathbf{f}^I \|^2} \right], \tag{6.6}$$

con $I \in \{Y, Cb, Cr\}$.

También se usará, como medida de la bondad subjetiva, la calidad visual de la imagen.

6.5.2 Niveles de compresión utilizados

Cada una de las imágenes del conjunto de prueba se comprimen mediante la implementación de JPEG llevada a cabo por *Independent JPEG Group, IJG*, a diferentes razones de compresión. Como muestra en este estudio hemos utilizado tres razones de compresión diferentes: una razón alta, en la que la imagen presenta una gran prominencia del efecto de bloques, una razón de compresión media, y una razón de compresión baja que prácticamente no produce este efecto bloque. Como medida de la compresión se usará el número de bits por pixel. Usaremos, para los experimentos, imágenes comprimidas a, aproximadamente, 0.30bpp, 0.40bpp y 0.60bpp o, equivalentemente, comprimidas a, aproximadamente, 83:1, 55:1 y 40:1, correspondiéndose estas razones a una compresión alta, media y baja compresión, respectivamente.

6.5.3 Matriz de pesos y criterio de parada

La forma de calcular las matrices de pesos $\mathbf{W}^I$, $I \in \{Y, Cb, Cr\}$, necesarias en la Ec. (6.2), que ponderan cada diferencia de pixels en el modelo de imagen, será también muy semejante a la utilizada en los capítulos anteriores. El valor de cada uno de los elementos de la matriz, $\omega^I(i)$, se define como

$$\omega^I(i) = \log \left(1 + \frac{\sqrt{\mu^I(i)}}{1 + \sigma^I(i)} \right),$$

tal como se describe en la Ec. (B.1) del Apéndice B, para $I \in \{Y, Cb, Cr\}$.

A cada una de las imágenes del conjunto de prueba se les aplicó el algoritmo 6.1. El criterio de parada fue, en todos los casos, que la diferencia de las reconstrucciones de cada banda (en

norma) entre dos iteraciones consecutivas fuese menor que 0.1, es decir,

$$\| \mathbf{f}^{I(\alpha_c^{k-1},\alpha_r^{k-1},\beta^{k-1})} - \mathbf{f}^{I(\alpha_c^{k},\alpha_r^{k},\beta^{k})} \| < 0.1, \tag{6.7}$$

con $I \in \{Y, Cb, Cr\}$.

6.5.4 Análisis visual

Las figuras 6.5, 6.7 y 6.9 muestran la zona central 256×256 de cada imagen original junto al resultado de la compresión mediante la implementación de JPEG llevada a cabo por *Independent JPEG Group, IJG* para las razones de compresión alta, media y baja, y las figuras 6.6, 6.8 y 6.10 muestran la zona central de la reconstrucción de las imágenes con el algoritmo 4.2, aplicado solamente a la banda Y, y usando el algoritmo 6.1, que también procesa las bandas de cromatismo. Los resultados para todas las razones de compresión se resumen en las figuras 6.2, 6.1 y 6.3 donde se muestra, para cada razón de compresión de la imagen (alta, media y baja), el pico de la razón señal-ruido de la imagen comprimida y de la reconstrucción para cada una de las bandas.

Como se puede observar, en ambos casos, es decir, tanto procesando sólo la banda Y de la imagen como procesando también las bandas de cromatismo, se aumenta sensiblemente la calidad de las imágenes mediante el proceso de reconstrucción.

Sin embargo, no se aprecian diferencias claras en la reducción del artificio de los bloques al reconstruir también las bandas de cromatismo. Esto es debido a que el efecto visual de bloques está causado principalmente por la banda de luminosidad ya que ésta es la responsable de los saltos de intensidad entre las fronteras de los bloques.

Por otra parte, sí es posible observar, al menos en su representación en una pantalla, que los colores de la imagen reconstruida mediante el algoritmo 6.1 presentan una mejora en cuanto a su suavidad en las transiciones. Este efecto es debido a que, puesto que las bandas de cromatismo sólo afectan a la información de color de la imagen, suavizar los bordes de los bloques en estas bandas sólo afecta a como los colores que se ven en la imagen no produciendo prácticamente ningún efecto sobre la percepción de los artificios en los bloques.

6.5.5 Análisis del $PSNR$

En cuanto a los resultados respecto al pico de la relación señal-ruido, en la totalidad de las reconstrucciones se aumenta el valor de esta medida para cada una de las bandas. Sin embargo,

esta mejora no se traduce, para las bandas de información cromática, Cb y Cr, en una mejora visual clara de la imagen por las razones anteriormente expuestas.

A la vista de los resultados podemos concluir que, si bien el uso de este sencillo modelo de imagen aplicado a cada una de las bandas mejora tanto la calidad visual como el $PSNR$ de la imagen, no es de gran ayuda a la hora de reducir el artificio de bloques en imágenes en color.

Esto es debido, en parte, a la escasa correlación entre las bandas que hace que la información sobre los artificios presente en las bandas de cromatismo no se use para la reconstrucción de la banda de luminosidad, principal responsable de la visibilidad de dichos artificios.

Posibles soluciones a este problema se podrían basar en el uso de otro espacio de color o en el uso de un modelo de imagen que tuviera en cuenta la información de las bandas cromáticas a la hora de reconstruir la banda de luminosidad.

Además, el procesamiento realizado sobre las bandas de cromatismo es claramente insuficiente siendo necesario, posiblemente, realizar una calibración de las medias de los bloques o la estimación de los primeros coeficientes de la transformada coseno para evitar saltos bruscos de color en la imagen.

No obstante, tanto el estudio de modelos de imagen que tenga en cuenta la información cromática, o la elección de un espacio de color en el que se realice una correcta combinación de las informaciones luminosas y cromáticas, como un tratamiento más exhaustivo de las bandas de cromatismo quedan fuera del ámbito de esta memoria siendo propuestas para un posterior estudio.

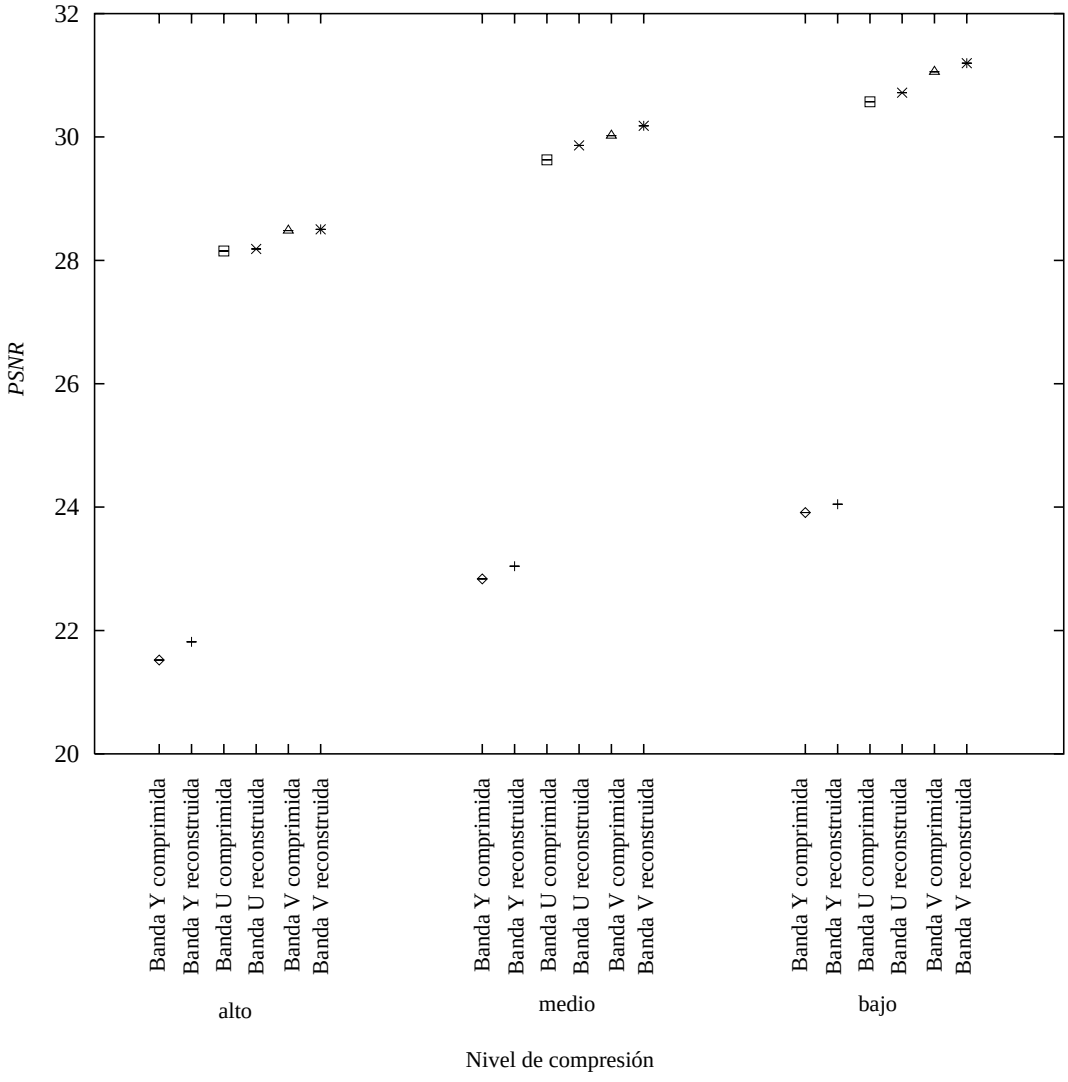


Figura 6.1: Resultados de la reconstrucción con el algoritmo 6.1, expresados en $PSNR$, para la imagen $baboon$ a diferentes razones de compresión.

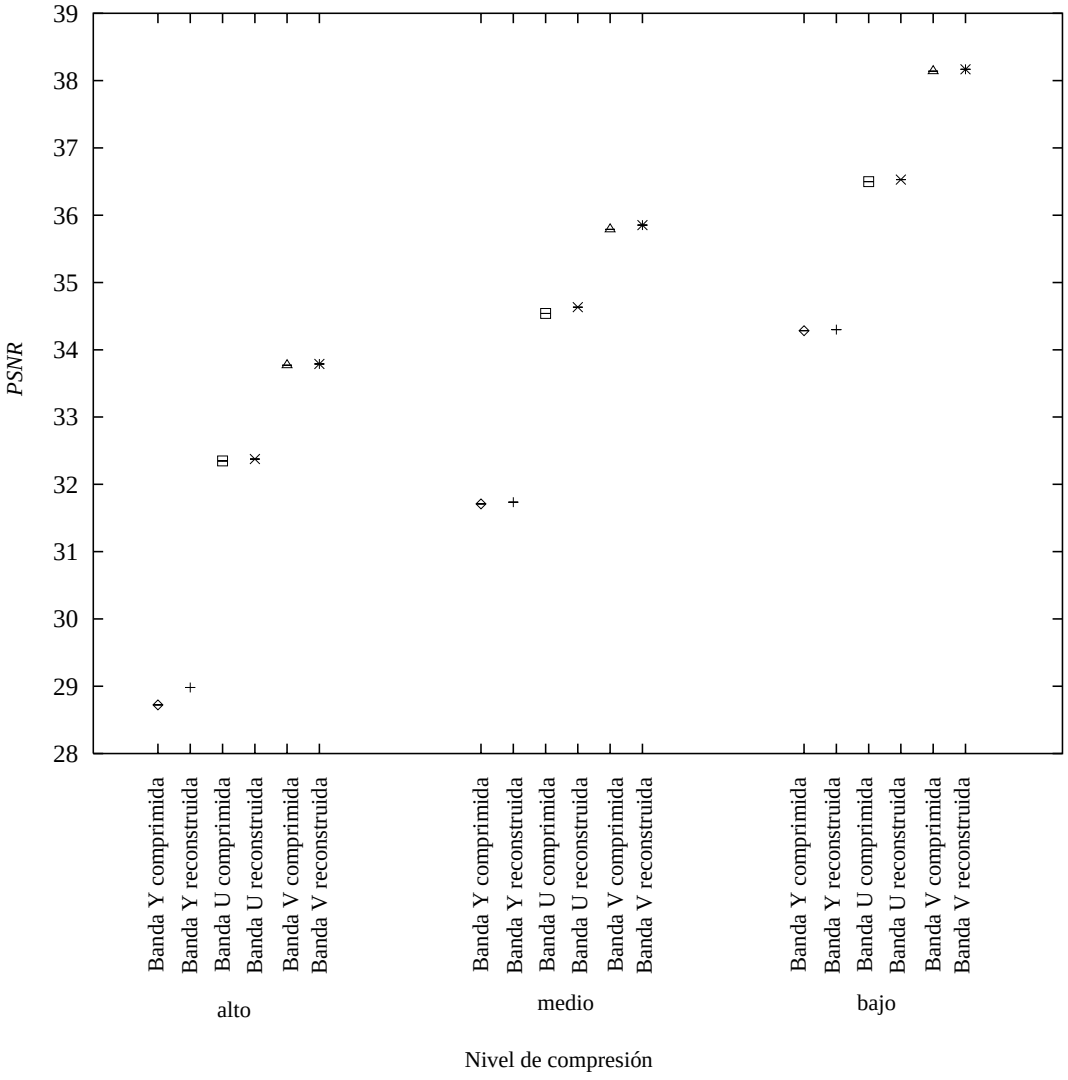


Figura 6.2: Resultados de la reconstrucción con el algoritmo 6.1, expresados en $PSNR$, para la imagen $airplane$ a diferentes razones de compresión.

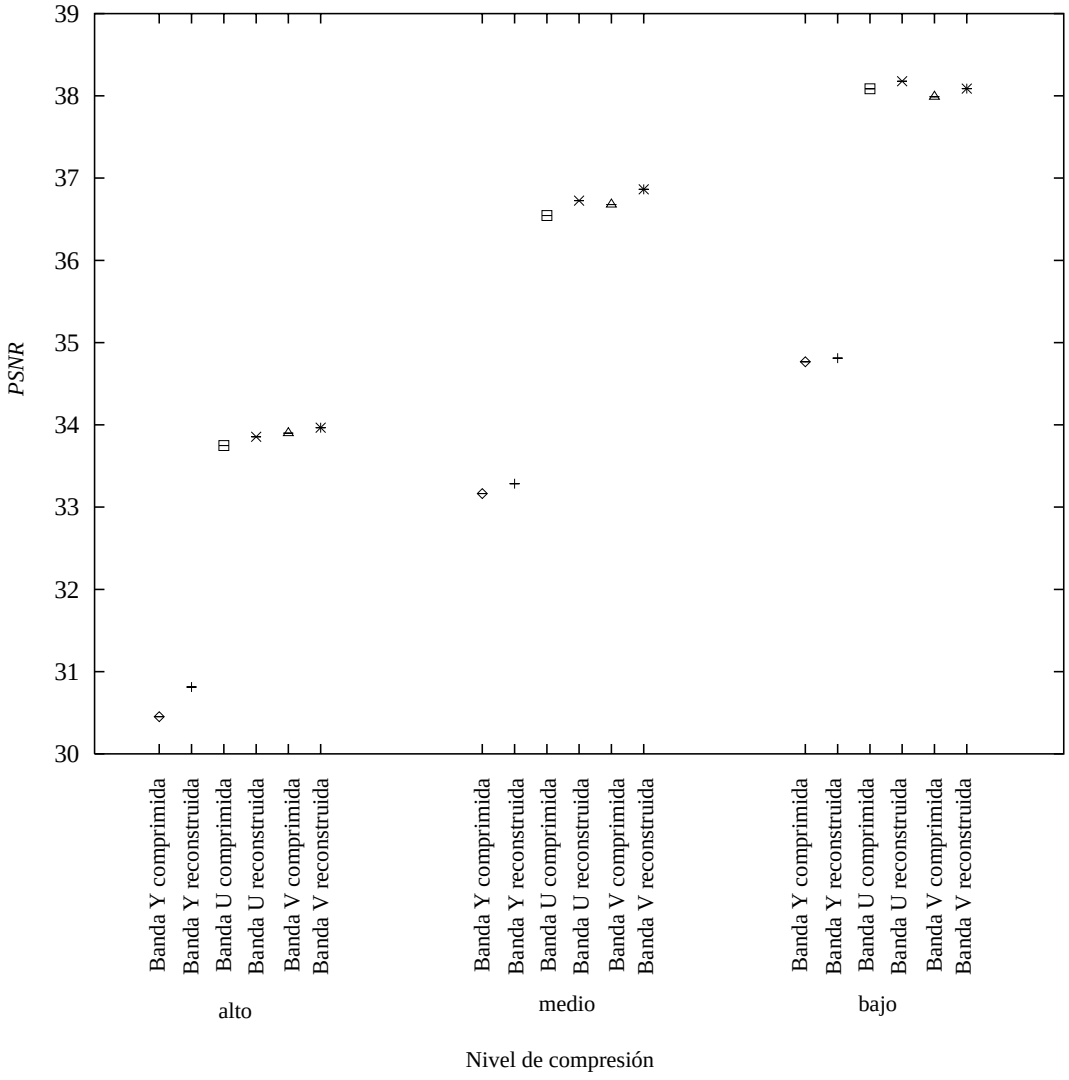


Figura 6.3: Resultados de la reconstrucción con el algoritmo 6.1, expresados en $PSNR$, para la imagen $lena$ a diferentes razones de compresión.

Figura 6.4: Conjunto de imágenes de prueba. Arriba: imagen con alta actividad espacial, $baboon$. Abajo izquierda: imagen con actividad espacial media, $airplane$. Abajo derecha: imagen con actividad espacial baja, $lena$.

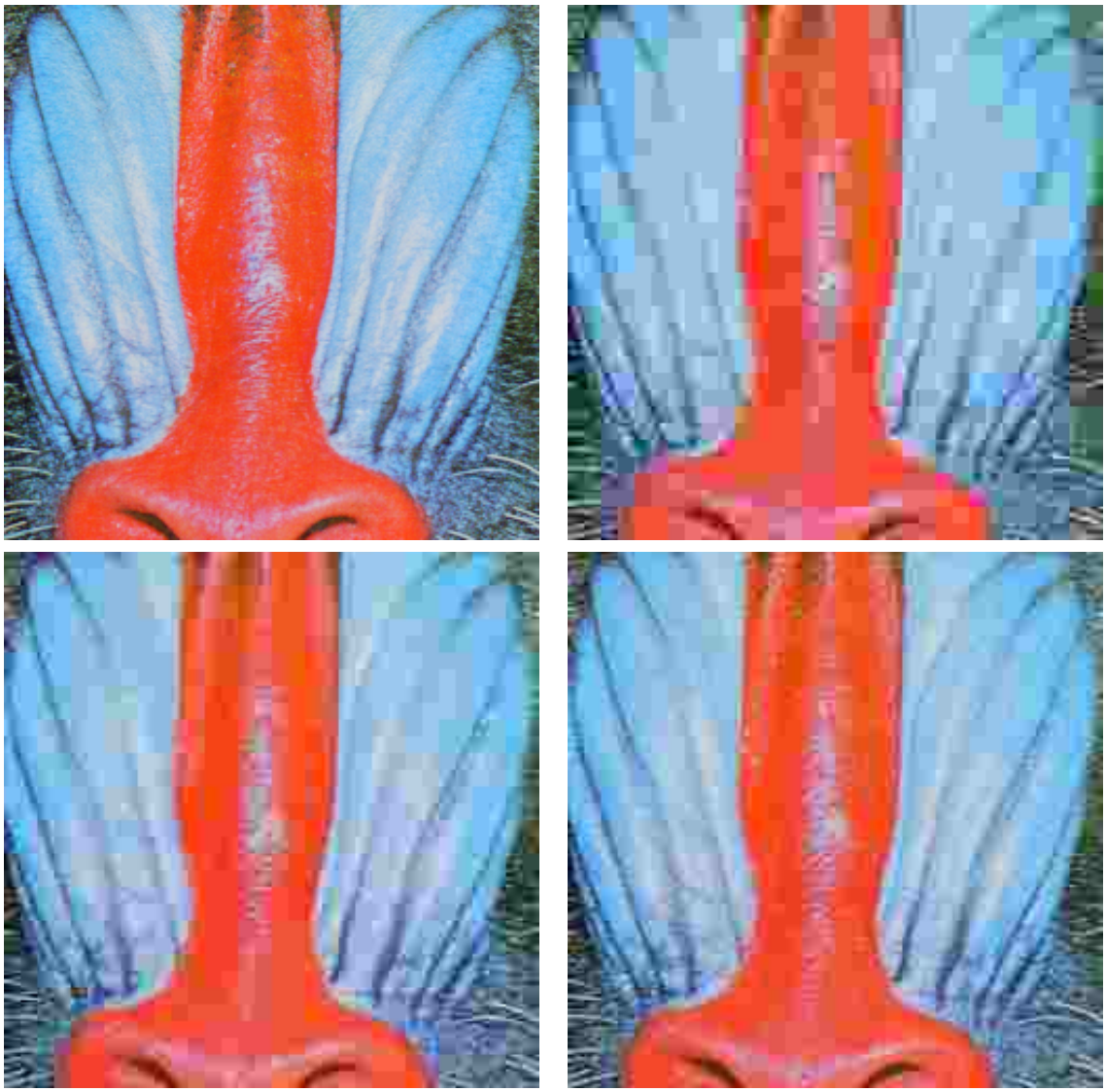


Figura 6.5: De izquierda a derecha y de arriba a abajo: Imagen $baboon$ original. Imagen comprimida a 0.30bpp. Imagen comprimida a 0.40bpp. Imagen comprimida a 0.60bpp.

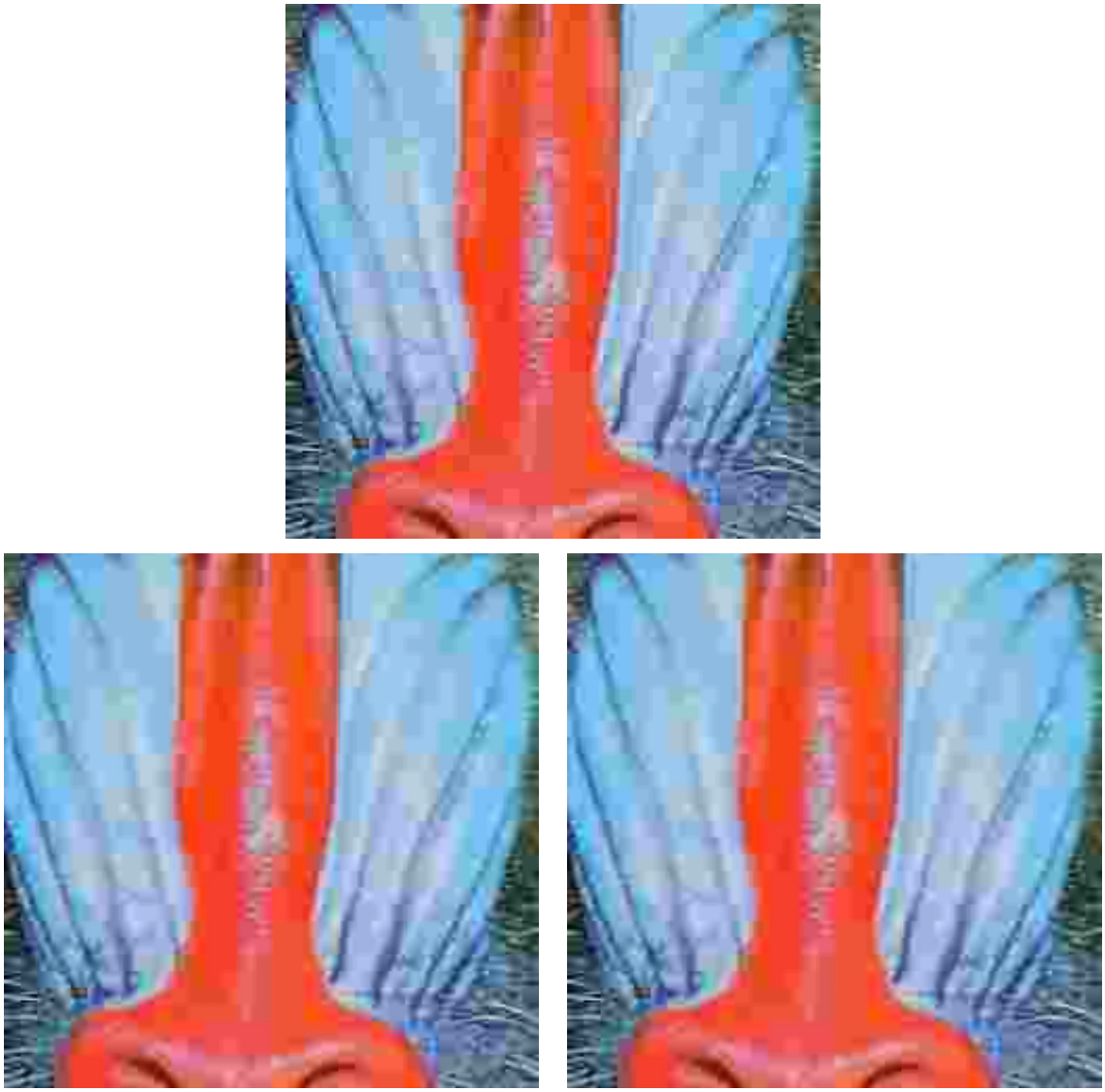


Figura 6.6: Arriba: Imagen $baboon$ comprimida a 0.40bpp. Abajo izquierda: Reconstrucción con el Algoritmo 4.2 aplicado a la banda Y. Abajo derecha: Reconstrucción con el Algoritmo 6.1.

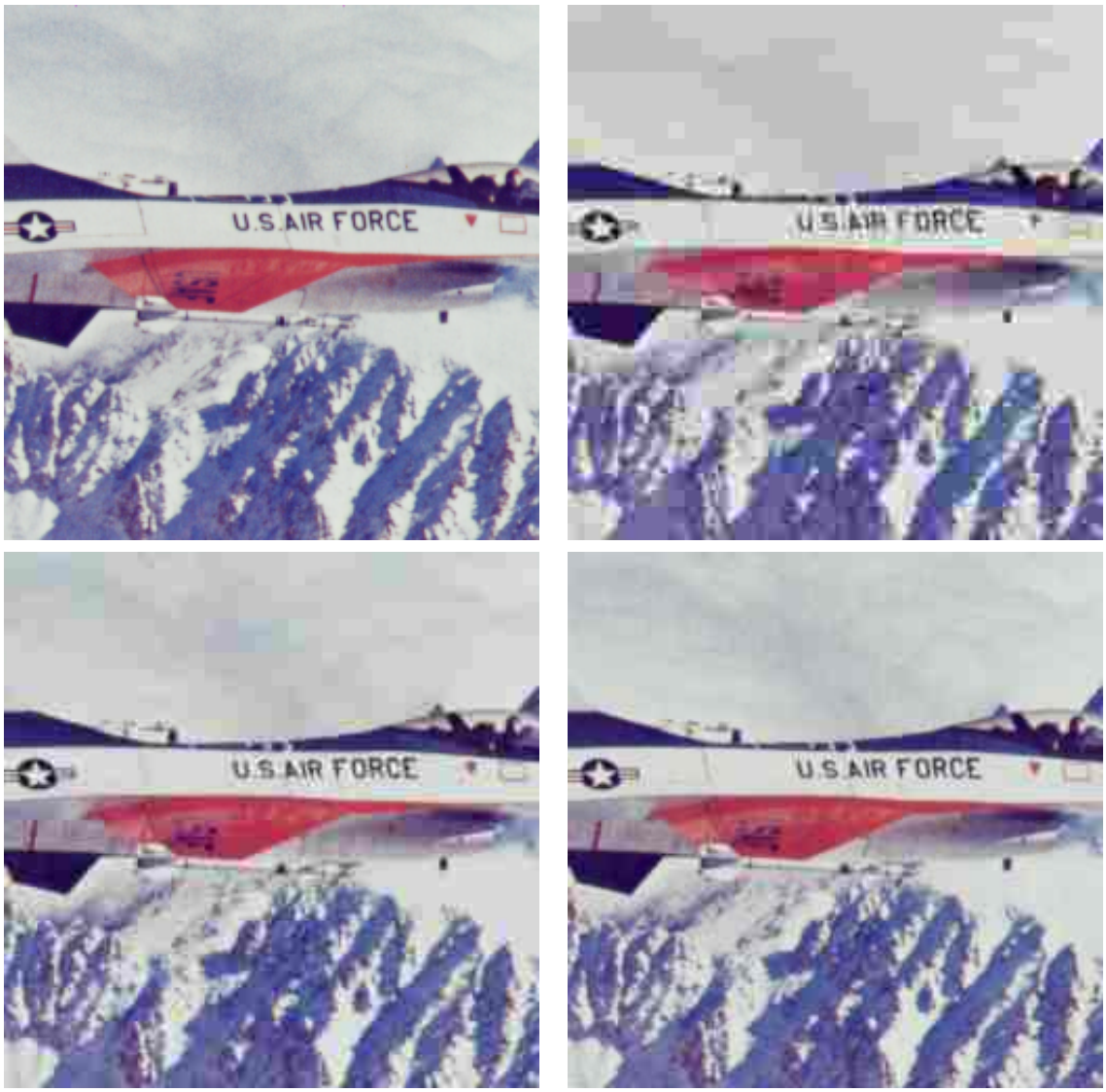


Figura 6.7: De izquierda a derecha y de arriba a abajo: Imagen $airplane$ original. Imagen comprimida a 0.30bpp. Imagen comprimida a 0.40bpp. Imagen comprimida a 0.60bpp.

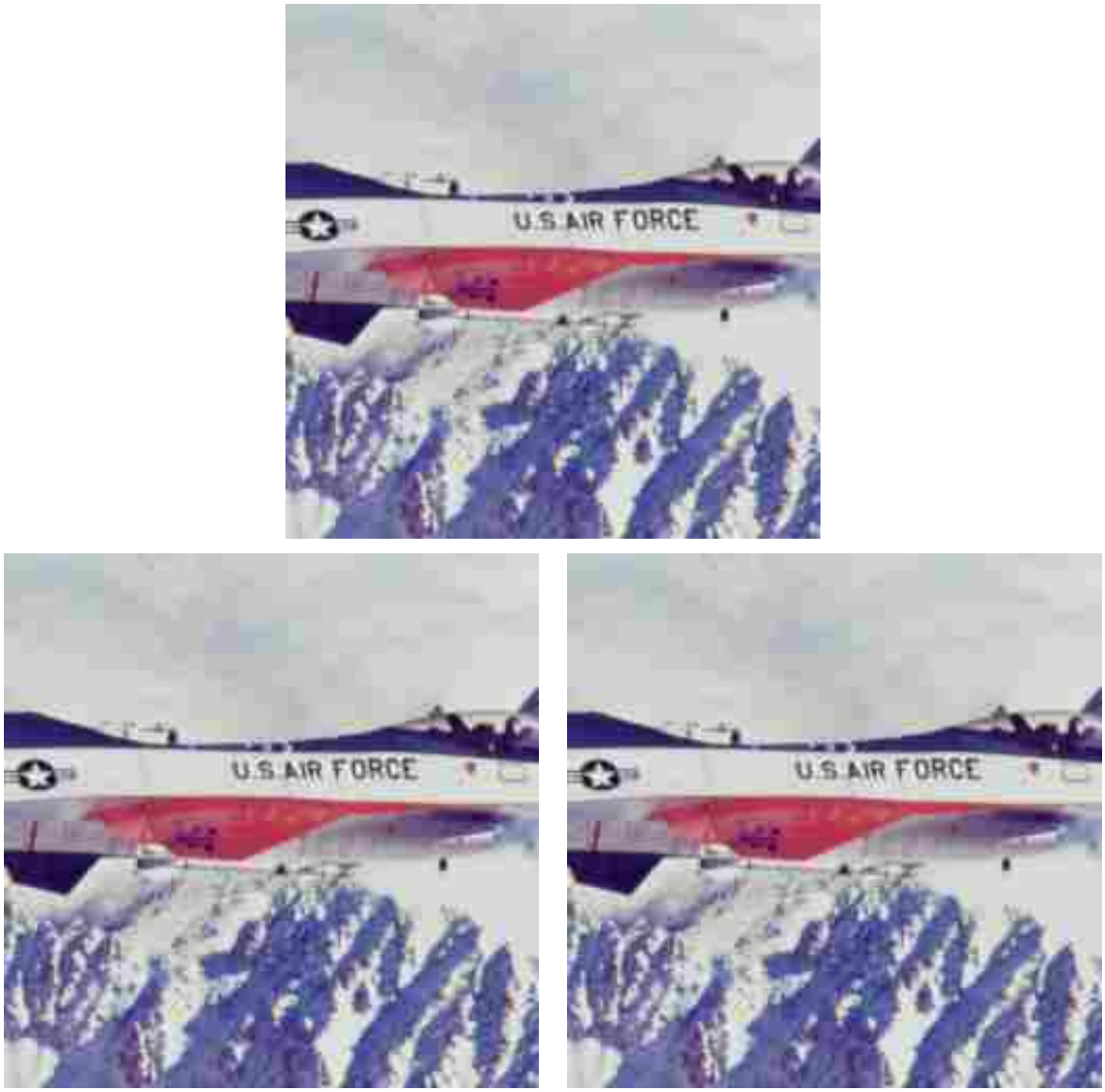


Figura 6.8: Arriba: Imagen $airplane$ comprimida a 0.40bpp. Abajo izquierda: Reconstrucción con el Algoritmo 4.2 aplicado a la banda Y. Abajo derecha: Reconstrucción con el Algoritmo 6.1.

Figura 6.9: De izquierda a derecha y de arriba a abajo: Imagen $lena$ original. Imagen comprimida a 0.30bpp. Imagen comprimida a 0.40bpp. Imagen comprimida a 0.60bpp.

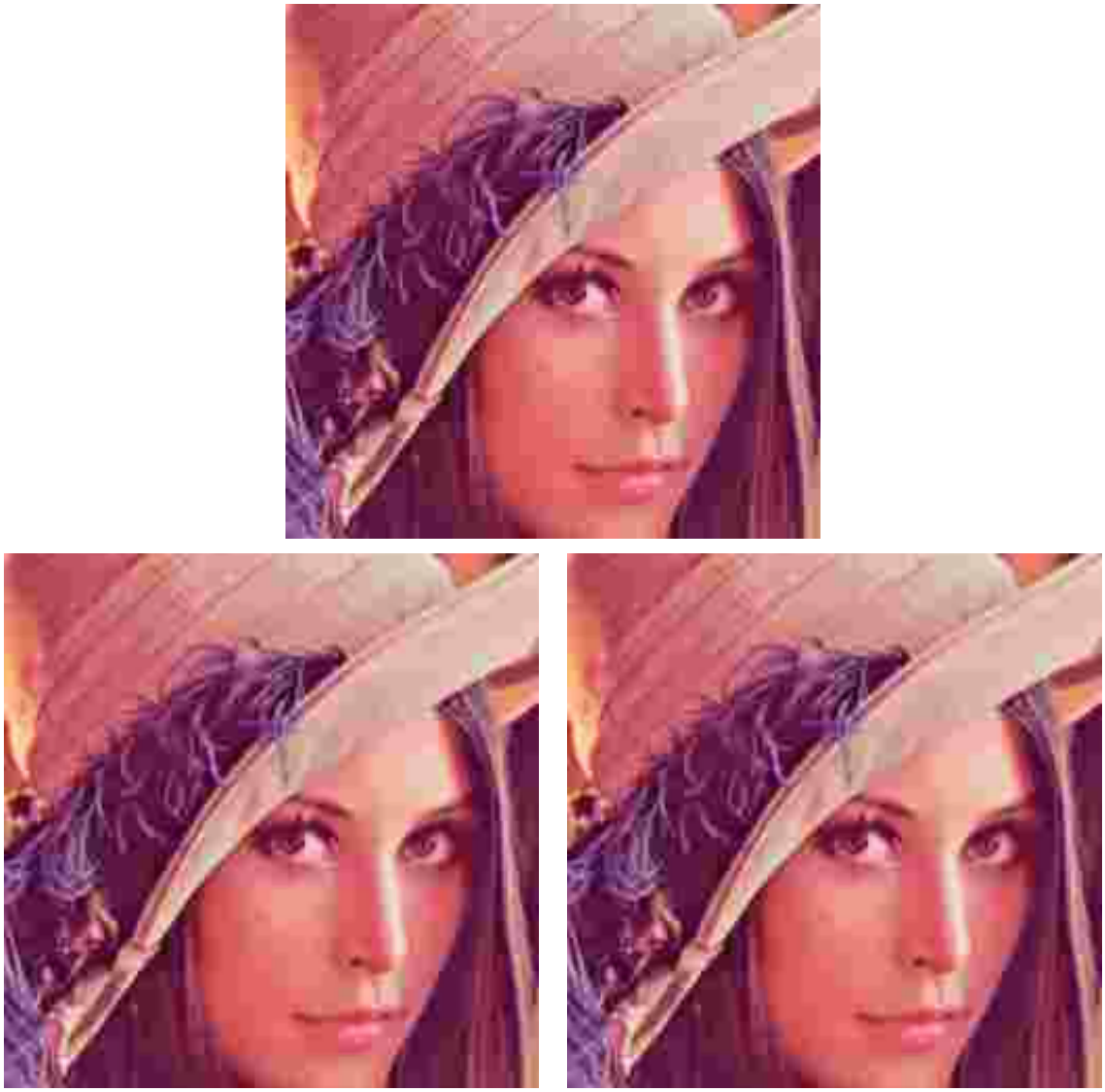


Figura 6.10: Arriba: Imagen $lena$ comprimida a 0.30bpp. Abajo izquierda: Reconstrucción con el Algoritmo 4.2 aplicado a la banda Y. Abajo derecha: Reconstrucción con el Algoritmo 6.1.

Conclusiones y trabajos futuros

Conclusiones

- Se han presentado diferentes métodos para la reconstrucción en el decodificador de imágenes. Todos estos métodos son automáticos en el sentido de que no necesitan del ajuste de ningún parámetro por parte del usuario.

- Se ha mostrado experimentalmente que el método de reconstrucción adaptativa con parámetros diferentes para filas y columnas en modelos de imagen y el mismo parámetro para filas y columnas en el modelo de ruido, produce los mejores resultados tanto a nivel de $PSNR$ como de calidad visual.

- Se ha presentado un método automático para obtener información sobre los parámetros en el codificador y como, usando esta información, reconstruir la imagen en el decodificador.

- Se ha mostrado como, usando una teoría bien fundamentada, se puede combinar la información sobre los parámetros proveniente del codificador con la información obtenida de la imagen comprimida en el decodificador para realizar métodos de reconstrucción que incorporen ambos conocimientos.

- Se ha mostrado que esta combinación de información produce mejores reconstrucciones, tanto en calidad visual como en $PSNR$, que los métodos que sólo usan la información del decodificador.

- Se ha presentado un primer paso en la reconstrucción de imágenes en color, abriendo una línea de investigación futura con un gran campo de trabajo aún sin explorar.

Trabajos futuros

- Estudio de modelos de imagen que tengan en cuenta la información de las bandas cromáticas para reconstruir la banda de luminosidad en imágenes en color.
- Estudio de espacios de color en los que se pueda realizar la combinación de la información de las bandas cromáticas y de luminosidad.
- Estudio de métodos automáticos de reconstrucción para las bandas de cromatismo que realicen una calibración de las medias de los bloques, o bien estimen los primeros coeficientes de la DCT para cada bloque.
- Aplicación de técnicas multicanal que tengan en cuenta la correlación presente entre las bandas de una imagen para obtener métodos de reconstrucción de imágenes en color.
- Aplicación de técnicas de interpolación sobre los pixels de las bandas de cromatismo en lugar de, simplemente, duplicar los valores de los pixels al reconstruir bandas a las que se ha aplicado un muestreo 4:1:1.
- Estudio de modelos perceptuales y métodos que nos permitan el cálculo de los pesos asociados el modelo de imagen de forma que capten las propiedades del sistema visual humano y tengan en cuenta las fronteras naturales de la imagen.
- Extensión de los métodos de reconstrucción de imágenes en escala de grises propuestos para la reconstrucción de secuencias de imágenes comprimidas mediante métodos basados en transformada coseno discreta, como los estándares MPEG, H.261 y H.263.
- Extensión de los métodos de reconstrucción de imágenes en color para la reconstrucción de secuencias de imágenes comprimidas mediante métodos basados en transformada coseno discreta.

Apéndice A

Distribución normal singular

Empecemos dando la definición de la distribución normal singular [69].

La función de densidad de probabilidad de $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ implica a $\boldsymbol{\Sigma}^{-1}$. Sin embargo, si rango$(\boldsymbol{\Sigma}) = k < p$, definimos la densidad singular de $\mathbf{x}$ como

$$\frac{2\pi^{-k/2}}{[\lambda_1 \lambda_2 \ldots \lambda_k]^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-}(\mathbf{x}-\boldsymbol{\mu})\right\},$$

donde

1. $\mathbf{x}$ satisface $N^t(\mathbf{x}-\boldsymbol{\mu}) = 0$, donde N es una matriz $p \times (p-k)$ tal que

$$N^t \boldsymbol{\Sigma} = 0, \quad N^t N = I_{p-k}.$$

2. $\boldsymbol{\Sigma}^{-}$ es la g-inversa de $\boldsymbol{\Sigma}$ y $\lambda_1, \lambda_2, \ldots, \lambda_k$ son los autovalores no nulos de $\boldsymbol{\Sigma}$.

Veamos como se aplica esto a nuestro problema de reconstrucción. Para cada elemento de $\mathbf{f}_c$, $\mathbf{f}_c(i)$, $i = 1, \ldots, p$, podemos escribir su distribución de probabilidad para el modelo de imagen descrito en la Ec. 3.14 como

$$\begin{aligned} \mathrm{p}(\mathbf{f}_c(i) \mid \alpha) &\propto F(\alpha) \times \\ &\times \exp\left[-\frac{1}{2}(\mathbf{f}_{cl}(i), \mathbf{f}_{cr}(i)) \begin{pmatrix} 2\alpha & -2\alpha \\ -2\alpha & 2\alpha \end{pmatrix} \begin{pmatrix} \mathbf{f}_{cl}(i) \\ \mathbf{f}_{cr}(i) \end{pmatrix}\right], \end{aligned}$$

donde

$$\Sigma^{-} = \begin{pmatrix} 2\alpha & -2\alpha \\ -2\alpha & 2\alpha \end{pmatrix}.$$

Puesto que el único autovalor no nulo de Σ^- es $\lambda = 4\alpha$, el único autovalor no nulo de Σ será $\lambda = 1/4\alpha$ y, por tanto,

$$F(\alpha) = \left(\prod_{i=1}^{k} \lambda_i\right)^{-\frac{1}{2}} = (4\alpha)^{\frac{1}{2}}.$$

Luego, el modelo de imagen en la Ec. 3.14, se definirá como

$$\mathrm{p}(\mathbf{f}_c \mid \alpha) = \prod_{i=1}^{p} \mathrm{p}(\mathbf{f}_c(i) \mid \alpha) \propto \alpha^{\frac{p}{2}} \exp\{-\alpha \parallel \mathbf{f}_{cl} - \mathbf{f}_{cr} \parallel^2\}.$$

Este mismo razonamiento puede aplicarse al modelo de imagen para las filas, $\mathrm{p}(\mathbf{f}_r \mid \alpha)$ y para los modelos adaptativos de imagen, incluso cuando aparecen diferentes parámetros para las filas y las columnas.

Para obtener el valor del parámetro de normalización para la intersección de cuatro bloques, hacemos el cambio de variable $H(\mathbf{f}_x(i)) = \mathbf{x}$, definido por

$$\begin{aligned} x_1 &= \omega_{x1}(i)(\mathbf{f}_{al}(i) - \mathbf{f}_{ar}(i)), \\ x_2 &= \omega_{x2}(i)(\mathbf{f}_{ar}(i) - \mathbf{f}_{br}(i)), \\ x_3 &= \omega_{x3}(i)(\mathbf{f}_{br}(i) - \mathbf{f}_{bl}(i)), \\ x_4 &= \omega_{x4}(i)\mathbf{f}_{bl}(i), \end{aligned}$$

obteniendo que

$$\begin{aligned} \mathrm{p}(\mathbf{x} \mid \alpha_c, \alpha_r) &= \mathrm{p}(H^{-1}(\mathbf{x}) \mid \alpha_c, \alpha_r) \mid J \mid \\ &\propto F(\alpha_c, \alpha_r) \exp\left\{-\frac{1}{2}\left\{\alpha_c \parallel x_1 \parallel^2 + \alpha_r \parallel x_2 \parallel^2 + \alpha_c \parallel x_3 \parallel^2 + \right.\right. \\ &\qquad \left.\left. \alpha_r \parallel \omega_{x4}\left(\frac{x_1}{\omega_{x1}} + \frac{x_2}{\omega_{x2}} + \frac{x_3}{\omega_{x3}}\right) \parallel^2\right\}\right\} \mid J \mid, \end{aligned}$$

donde $|\cdot|$ significa el valor absoluto,

$$J = \det \frac{\partial H^{-1}(\mathbf{x})}{\partial \mathbf{x}} = \left[\det \frac{\partial H(\mathbf{f}_x(i))}{\partial \mathbf{f}_x(i)}\right]^{-1} = \omega_{x1}\omega_{x2}\omega_{x3}\omega_{x4},$$

y, por tanto,

$$F(\alpha_c, \alpha_r) = \left[\alpha_c\alpha_r\left(\alpha_c\left(\frac{1}{\omega_{x2}^2} + \frac{1}{\omega_{x4}^2}\right) + \alpha_r\left(\frac{1}{\omega_{x1}^2} + \frac{1}{\omega_{x3}^2}\right)\right)\right]^{\frac{m}{2}}.$$

En el caso de que el modelo sea no adaptativo, $\mathbf{W} = \mathbf{I}$, tendremos que

$$F(\alpha_c, \alpha_r) = \left[2\alpha_c\alpha_r\left(\alpha_c + \alpha_r\right)\right]^{\frac{m}{2}},$$

y en el caso en que tengamos un único parámetro para controlar la suavidad en las filas y las columnas, $\alpha_c = \alpha_r = \alpha$, nuestro parámetro de normalización, $F(\alpha)$, será

$$F(\alpha) = 4\alpha^{\frac{3m}{2}}.$$

Apéndice B

Elección de los pesos del modelo adaptativo de imagen

De la discusión mantenida en la sección 3.5 está claro que los valores de los pesos $\omega_c(i)$ y $\omega_r(i)$ usados en el modelo adaptativo de imagen deberían estar basados en las propiedades del sistema visual humano y en características estadísticas locales que sean fáciles de modelizar en nuestro sistema.

Para obtener los pesos $\omega_c(i)$ y $\omega_r(i)$, el pixel en la posición i se trata como una variable aleatoria con media $\mu(i)$ y varianza $\sigma^2(i)$. La media nos servirá como una medida del brillo de la imagen en ese punto y la varianza como una medida de la actividad espacial local, o nivel de detalle local, en el pixel en la posición i.

A partir de experimentos psicofísicos sobre el *límite de la visibilidad* del ruido o de diferentes estímulos en imágenes digitales (ver capítulo 4 de [91]) se puede afirmar que la visibilidad del ruido (los artificios en nuestro problema) decrece en las cercanías de zonas con alto nivel de detalle o espacialmente muy activas. Esto nos lleva a considerar una función para los pesos que sea decreciente en función de $\sigma(i)$, es decir, que trate de suavizar más las zonas con poca actividad espacial preservando los detalles en las zonas activas espacialmente. Un ejemplo de esta función es [132]

$$\omega(i) = \frac{1}{1 + \sigma(i)},$$

a la que se le añade un 1 al denominador para solventar dificultades matemáticas cuando $\sigma(i) = 0$.

Los estudios sobre psicofísica de la visión también afirman que la intensidad de la luz del

fondo de una zona también influye en la visibilidad de los artificios en esa zona, siendo más visibles en zonas claras que en zonas oscuras. Esta propiedad también se puede añadir a nuestra función de pesos definiéndola, por ejemplo, como

$$\omega(i) = \frac{\sqrt{\mu(i)}}{1+\sigma(i)}.$$

Por último, estos estudios indican que hay una transformación no lineal de intensidad en las primeras etapas del sistema visual humano y que esta transformación es aproximadamente logarítmica en varios órdenes de magnitud de intensidad, por lo que definiremos la función de pesos como [132]

$$\omega(i) = \log\left(1 + \frac{\sqrt{\mu(i)}}{1+\sigma(i)}\right), \tag{B.1}$$

una función de rango comprimido que decrece en función de la actividad espacial alrededor del pixel estudiado y que capta los cambios de intensidad en la zona.

Estas funciones son sólo ejemplos de las que se podrían usar e ilustran como se pueden incorporar las características del sistema visual humano al modelo de imagen. Sin embargo, un estudio con detenimiento de las funciones de visibilidad de los artificios de los bloques está fuera del ámbito de esta memoria. Además, la forma de estas funciones depende también del medio en que las imágenes se presentan al usuario, es decir, monitores CRT, copia impresa, monitores de plasma líquido, etc. y de otros muchos factores como la relación entre el tamaño de la imagen y la distancia del observador a la misma o la iluminación ambiental.

En nuestros ejemplos, el cálculo de $\omega_c(i)$ y $\omega_r(i)$ se realizará a partir de la Ec. (B.1) usando los datos disponibles en el receptor, en el formato de la transformada coseno discreta. Puesto que tenemos que calcular los pesos para las direcciones horizontal y vertical tenemos que calcular dos medias $\mu_c(i)$ y $\mu_r(i)$ y dos varianzas $\sigma_c^2(i)$ y $\sigma_r^2(i)$ que se corresponden con $\omega_c(i)$ y $\omega_r(i)$, respectivamente. Para el cálculo de los pesos hemos supuesto bloques de tamaño fijo $k \times k$.

Puesto que los coeficientes DC representan la media (salvo una constante) de los pixels dentro de cada bloque, la media $\mu_c(i)$ se calcula como sigue: considérese un pixel en una frontera vertical, l_c, y sean DC_l y DC_r los coeficientes DC de los bloques a su izquierda y derecha, respectivamente. Entonces la media $\mu_c(l_c)$ usada para el cálculo de $\mathbf{W}_c$ viene dada por

$$\hat{\mu}_c(l_c) = \frac{DC_l + DC_r}{2 \times k^2}.$$

Una fórmula similar se usa para estimar la media $\mu_r(l_r)$ usada en el cálculo de los pesos $\mathbf{W}_r$ teniendo en cuenta, en este caso, los coeficientes DC de los bloques sobre y bajo a esta frontera horizontal.

La varianza $\sigma_c^2(l_c)$ en una frontera vertical de un bloque l_c se estima como

$$\hat{\sigma}_c^2(l_c) = \frac{V\,AC_l + V\,AC_r}{2 \times k^2},$$

donde $V\,AC_l$ y $V\,AC_r$ son la suma de los cuadrados de los coeficientes de la primera columna de los bloques a izquierda y derecha de la frontera l_c, respectivamente. De forma similar, $\sigma_r^2(l_r)$ se estima usando la suma de los cuadrados de coeficientes AC de los bloques de superior e inferior de esta frontera horizontal.

Bibliografía

[1] Ahumada, A. J. y Horng, R., "De-blocking DCT compressed images", en *Human Vision, Visual Processing and Digital Display V, SPIE Proc.*, B. Rogowitz y J. Allbach, eds., págs. 109–116, 1994.

[2] Ahumada, A. J. y Horng, R., "Smoothing DCT compression artifacts", en *1994 SID International Symposium Digest of Technical Papers*, págs. 708–711, jun. 1994.

[3] Archer, G. y Titterington, D., "On some Bayesian/regularization methods for image restoration", *IEEE Trans. on Image Processing*, tomo 4, págs. 989–995, 1995.

[4] Baskurt, A., Prost, R. y Goutte, R., "Iterative constrained restoration of DCT-compressed images", *Signal Processing*, tomo 17, págs. 201– 211, 1989.

[5] Berger, J. O., *Statistical decision theory and Bayesian analysis*, Springer Verlag, New York, 1985.

[6] Bradley, J., Brislawn, C. y Hopper, T., "The FBI wavelet/scalar quantization of fingerprint image compression standard", Inf. Téc. LA-UR-94-1409, Los Alamos National Laboratory, 1994.

[7] Brailean, J. C., Özcelik, T. y Katsaggelos, A. K., "A video coding algorithm based on recovery techniques using mean field annealing", en *Proceedings of the Visual Communication and Image Processing 95, SPIE Proc.*, págs. 284–294, 1995.

[8] Buntine, W., *A theory of learning classification rules*, Tesis Doctoral, University of Technology, Sidney, Australia, 1991.

[9] Buntine, W., "Theory refinement on Bayesian networks", en *Seventh Conference on Uncertainty in Artificial Intelligence*, págs. 52–60, 1991.

[10] Buntine, W. y Weigund, A., "Bayesian back-propagation", *Complex Systems*, tomo 5, págs. 603–643, 1991.

[11] Castagno, R. y Villaroel, J. A., "A spline based adaptive filter for the removal of blocking artifacts in image sequences coded at very low bit rate", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 44–48, 1996.

[12] CCIR Rec. 601-2, *Encoding parameters of digital television for studios*, tomo XI, Parte 1, Broadcasting Service (Television), págs. 95–104, International Telecomminication Union, 1990.

[13] Chan, Y.-H. y Siu, W.-C., "An approach to subband DCT image coding", *Journal of Visual Communication and Image Representation*, tomo 5, nº 1, págs. 95–106, mar. 1994.

[14] Choy, S. S. O., Chan, Y.-H. y Siu, W.-C., "Reduction of block-transform image coding artifacts by using local statistics of transform coefficients", *IEEE Signal Processing Letters*, tomo 4, nº 1, págs. 5–7, ene. 1997.

[15] Cooper, G. y Herkovsits, E., "A Bayesian method for the induction of probabilistic networks from data", *Machine Learning*, tomo 9, págs. 309–347, 1992.

[16] Crouse, M. y Ramchandran, K., "Nonlinear constrained least squares estimation to reduce artifacts in block transform-coded images", en *Proceedings of the International Conference on Image Processing ICIP 95*, tomo 1, págs. 462–465, 1995.

[17] da Silva, E. y Mendonca, G., "Image coding by the mesh block transform", en *1991 IEEE International Symposium on Circuits and Systems*, tomo 1, págs. 304–307, 1991.

[18] Derviaux, C., Coudoux, F. X., Gazalet, M. G. y Corlay, P., "Blocking artifact reduction of DCT coded image sequences using a visually adaptive postprocessing", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 5–8, 1996.

[19] Fan, Z. y Li, F., "Reducing artifacts in JPEG decompression by segmentation and smoothing", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 17–20, 1996.

[20] Fok, Y.-H., Au, O. C. y Chang, C., "Bitrate and blocking artifact reduction by iterative pre-distortion", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 13–16, 1996.

[21] Galatsanos, N. P. y Chin, R. T., "Digital restoration of multichannel images", *IEEE Trans. Acoust., Speech, Signal Processing*, tomo 37, nº 3, págs. 415–421, 1989.

[22] Galatsanos, N. P., Katsaggelos, A. K., Chin, R. T. y Hillery, A. D., "Least squares restoration of multichannel images", *IEEE Trans. on Signal Processing*, tomo 39, nº 10, págs. 2222–2236, 1991.

[23] Galatsanos, N. P., Mesarovic, V. Z., Molina, R. y Katsaggelos, A. K., "Hierarchical bayesian image restoration for partially-known blurs", Enviado a IEEE Trans. on Image Processing, 1998.

[24] Gersho, A. y Gray, R. M., *Vector quantization and signal compression*, Kluwer Academic Press, 1992.

[25] Gull, S., "Developments in maximum entropy data analysis", en *Maximum Entropy and Bayesian Methods*, J. Skilling, ed., págs. 53–71, Kluwer, 1989.

[26] Guo, Y. P., Lee, H. P. y Teo, C. L., "Multichannel image restorations using iterative algorithm in space domain", *Image and Video Computing*, tomo 14, págs. 389–400, 1996.

[27] Hilton, M., Jawerth, B. y Sengupta, A., "Compressing still and moving images with wavelets", *Multimedia systems*, tomo 2, nº 3, 1994.

[28] Ho, P.-S. y Kim, M.-H., "Perceptually-adaptive image compression based on block DCT", *Journal of KISS[A] [Computer Systems and Theory]*, tomo 22, nº 10, págs. 1405–1415, 1995.

[29] Hong, S. W., Chan, Y. H. y Siu, W. C., "A practical real time postprocessing technique for block effect elimination", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 21–24, 1996.

[30] Horng, R. y Ahumada, A. J., "A fast DCT block smoothing algorithm", en *Proceedings of the Visual Communication and Image Processing 95, SPIE Proc.*, tomo 2501, págs. 28–39, mayo 1995.

[31] Hsung, T.-C., Lun, D. P.-K. y Siu, W.-C., "A deblocking technique for JPEG decoded image using wavelets transform modulus maxima representation", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 561–564, 1996.

[32] International Telecommunications Union, *ITU-T Recomendadion H.263. Video coding for low rate communication*, 1996.

[33] International Telegraph and Telephone Consultive Committee Recommendation H.261 CDM XV-R 37-E, *Video codec for audiovisual services at $p \times 64$ Kbits/s*, 1990, también ITU-T Recommendation H.261, 1993.

[34] ISO/IEC JTC 1 / SC 29 International Standard 10918-1, *Digital compression and coding of continuous-tone still images, part 1, requirements and guidelines*, 1994.

[35] ISO/IEC JTC 1 / SC 29 International Standard 10918-2, *Digital compression and coding of continuous-tone still images, part 2, compliance testing*, 1995.

[36] ISO/IEC JTC 1 / SC 29 International Standard 10918-2, *Digital compression and coding of continuous-tone still images, part 3, extensions*, 1995.

[37] ISO/IEC JTC1 / SC 29 International Standard 13818-2, *Information technology – Generic coding of moving pictures and associated audio information: Video*, 1996.

[38] ISO/IEC JTC1 Draft International Standard ISO/IEC 11172-2, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video*, 1993.

[39] Itoh, Y., "Detail preserving noise filtering for compressed image", *IEICE Trans. on Communications*, tomo E79-B, nº 10, págs. 1459–1466, oct. 1996.

[40] Itoh, Y., "Detail-preserving noise filtering using binary index", en *Proceedings of the Image and Video Processing IV Conf., SPIE Proc.*, tomo 2666, págs. 119–130, 1996.

[41] Jacovitti, G. y Neri, A., "Bayesian removal of coding block artifacts in the harmonic angular filters features domain", en *Proceedings of the Nonlinear Image Processing VI Conf., SPIE Proc.*, tomo 2424, págs. 139–150, 1995.

[42] Jain, A., *Fundaments of digital image processing*, Prentice Hall, 1989.

[43] Jeon, B., Jeong, J. y Jo, J. M., "Blocking artifacts reduction in image coding based on minimum block boundary discontinuity", en *Proceedings of the Visual Communications and Image Processing 95*, págs. 198–209, 1995.

[44] Jeong, J. y Jeon, B., "Use of a class of two-dimensional functions for blocking artifacts reduction in image coding", en *Proceedings of the International Conference on Image Processing ICIP 95*, tomo 1, págs. 478–481, 1995.

[45] Joung, H., Chong, U. y Kim, S. P., "Block artifact reduction by optimization utilizing subband decomposition", en *Proceedings of the 7th KSEA Northeast Regional Conference*, 1996.

[46] Jové, F. J., Santamaría, M. E., Tarrés, F. y Trabado, J., "Minimization of the mosaic effect in hierarchical multirate vector quantization of image coding", en *Proceedings of the Visual Communication and Image Processing 95, SPIE Proc.*, tomo 2501, págs. 555–561, mayo 1995.

[47] Kang, M. G. y Katsaggelos, A. K., "Simultaneous iterative image restoration and evaluation of the regularization parameter", *IEEE Trans. on Signal Processing*, tomo 40, págs. 2320–2334, 1992.

[48] Kang, M. G. y Katsaggelos, A. K., "General choice of the regularization functional in regularized image restoration", *IEEE Trans. Image Processing*, tomo 4, nº 5, págs. 594–602, 1995.

[49] Karlsson, G. y Vetterli, M., "Packed video and its integration into the network architecture", *IEEE Journal of Selected Areas in Communications*, tomo 7, nº 5, págs. 739–751, 1989.

[50] Katsaggelos, A. K. (ed.), *Digital image restoration*, tomo 23 de *Springer Series in Information Sciences*, Springer-Verlag, 1991.

[51] Katsaggelos, A. K. y Kang, M. G., "Iterative evaluation of the regularization parameter in regularized image restoration", *Journal of Visual Communication and Image Representation*, tomo 3, págs. 446–455, 1992.

[52] Kim, H. C. y Park, H. W., "Signal adaptive postprocessing for blocking effects reduction in JPEG image", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 41–44, 1996.

[53] Kuo, C. y Hsieh, R., "Adaptive postprocessor for block encoded images", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 5, nº 4, págs. 298–304, ago. 1995.

[54] Kwak, K. Y. y Haddad, R., "Projection-based eigenvector decomposition for reduction of blocking artifacts of DCT coded image", en *Proceedings of the International Conference on Image Processing ICIP 95*, tomo 2, págs. 527–530, 1995.

[55] Lagendijk, R. L. y Biemond, J., *Iterative identification and restoration of images*, Kluwer Academic Publishers, 1991.

[56] Lai, Y.-K., Li, J. y Kuo, C.-C. J., "Image enhancement for low bit-rate JPEG and MPEG coding via postprocessing", en *Proceedings of the Visual Communications and Image Processing '96, SPIE Proc.*, tomo 2727, págs. 1484–1494, 1996.

[57] Lai, Y.-K., Li, J. y Kuo, C.-C. J., "Removal of blocking artifacts of DCT transform by classified space-frequency filtering", en *Conference Record of The Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, tomo 2, págs. 1457–1461, 1996.

[58] Lay, K. T. y Katsaggelos, A. K., "Image identification and restoration based on the expectation-maximization algorithm", *Optical Engineering*, tomo 29, págs. 436–445, 1990.

[59] Linares, I., Mersereau, R. y Smith, M., "JPEG estimated spectrum adaptive postfiltering using image adaptive Q-tables and Canny edge detectors", en *Proceedings of the 1996 IEEE International Symposium on Circuits and Systems*, tomo 2, págs. 722–725, 1996.

[60] Liu, T.-S. y Jayant, N., "Adaptive postprocessing algorithms for low bit rate video signals", *IEEE Trans. on Image Processing*, tomo 4, nº 7, págs. 1032–1035, jul. 1995.

[61] Luo, J., Chen, C. W., Parker, K. J. y Huang, T. S., "A new method for block effect removal in low bit-rate image compression", en *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP 94*, tomo 5, págs. 341–344, 1994.

[62] Luo, J., Chen, C. W., Parker, K. J. y Huang, T. S., "Artifact reduction in low bit rate DCT-based image compression", *IEEE Trans. on Image Processing*, tomo 5, nº 9, págs. 1363–1368, sep. 1996.

[63] Lynch, W. E., Reibman, A. R. y Liu, B., "Post processing transform coded images using edges", en *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing*, tomo 4, 1995.

[64] MacKay, D. J. C., "Bayesian interpolation", *Neural Computation*, tomo 4, págs. 415–447, 1992.

[65] MacKay, D. J. C., "A practical Bayesian framework for backprop networks", *Neural Computation*, tomo 4, págs. 448–472, 1992.

[66] MacKay, D. J. C., "Comparison of Approximate Methods for Handling Hyperparameters", Enviado a *Neural Computation*, 1996.

[67] Mallat, S. y Zhong, S., "Characterization of signals from multiscale edges", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, tomo 14, nº 7, págs. 710–732, jul. 1992.

[68] Malvar, H. y Staelin, D. H., "The LOT: transform coding without blocking effects", *IEEE Trans. on Acoustic, Speech and Signal Processing*, tomo 37, págs. 553–559, abr. 1989.

[69] Mardia, K. V., Kent, J. T. y Bibly, J. M., *Multivariate analysis*, Academic Press, 1979.

[70] Mateos, J., Ilia, C., Jiménez, B., Molina, R. y Katsaggelos, A. K., "Reduction of blocking artifacts in block transformed compressed color images", en *Proceedings of the International Conference on Image Processing ICIP 98*, 1998, pendiente de celebración.

[71] Mateos, J., Katsaggelos, A. K. y Molina, R., "Parameter estimation in regularized reconstruction of BDCT compressed images for reducing blocking artifacts", en *Proceedings of the Conference on Digital Compression Technologies & Video Communications, SPIE Proc.*, tomo 2952, págs. 70–81, 1996.

[72] Mateos, J., Katsaggelos, A. K. y Molina, R., "Bayesian reconstruction of BDCT compressed images", en *Preprints of the VII National Symposium on Pattern Recognition and Image Analysis*, tomo I, págs. 341–346, 1997.

[73] Mateos, J., Katsaggelos, A. K. y Molina, R., "A Bayesian approach for the estimation and transmission of regularization parameters for reducing blocking artifacts", Enviado a *IEEE Trans. on Image Processing*, 1998.

[74] Mateos, J., Molina, R. y Katsaggelos, A. K., "Estimating and transmitting regularization parameters for reducing blocking artifacts", en *Proceedings of the 13th International Conference on Digital Signal Processing*, págs. 209–212, 1997.

[75] McDonnell, J. D., Shorten, R. y Fagan, A. D., "An edge classification based approach to the post-processing of transform coded images", en *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP 94*, tomo 5, págs. 329–332, 1994.

[76] Minami, S. y Zakhor, A., "An optimization approach for removing blocking effects in transform coding", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 5, nº 2, págs. 74–81, abr. 1995.

[77] Molina, R., "On the hierarchical Bayesian approach to image restoration. Application to astronomical images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, tomo 9, nº 6, págs. 721–742, 1994.

[78] Molina, R. y Katsaggelos, A., "On the hierarchical Bayesian approach to image restoration and the iterative evaluation of the regularization parameter", en *Proc. of the Visual Communication and Image Processing'94, Chicago*, págs. 255–251, 1994.

[79] Molina, R., Katsaggelos, A. K. y Mateos, J., "Bayesian and regularization methods for hyperparameter estimation in image restoration", Aceptado para su publicación en *IEEE Trans. on Image Processing*, 1998.

[80] Molina, R., Katsaggelos, A. K. y Mateos, J., "Removing Blocking Artifacts in BDCT compressed Images. A survey and New Results for Grey Level and Colour Images", en *Recovery Techniques for Image and Video Compression*, A. K. Katsaggelos y N. P. Galatsanos, eds., cap. 1, Kluwer Academic Publishers, 1998, por aparecer.

[81] Molina, R., Katsaggelos, A. K., Mateos, J. y Abad, J., "Compound Gauss-Markov random fields for astronomical image restoration", *Vistas in Astronomy. Special Issue on Vision Modelling and Information Coding*, tomo 40, nº 4, págs. 539–546, 1996.

[82] Molina, R., Katsaggelos, A. K., Mateos, J. y Abad, J., "Restoration of severely blurred high range images using compound models", en *Proceedings of International Conference on Image Processing ICIP 96*, 1996.

[83] Molina, R., Katsaggelos, A. K., Mateos, J. y Hermoso, A., "Restoration of severely blurred high range images using stochastic and deterministic relaxation algorithms in compound Gauss Markov random fields", en *Energy Minimization Methods in Computer Vision and Pattern Recognition*, M. Pelillo y E. R. Hancock, eds., tomo 1223 de *Lecture Notes in Computer Science*, págs. 117–132, Springer-Verlag, Berlin, 1997.

[84] Molina, R., Katsaggelos, A. K., Mateos, J. y Hermoso, A., "Simulated annealing for compound Gauss Markov random fields in the presence of blurring", Enviado a *Pattern Recognition*, 1997.

[85] Molina, R. y Mateos, J., "Image models and their role in image processing. Some multichannel results", en *Actas de la Conferencia Final de la Red Cientifica de la European Science Foundation, "Converging Computing Methodologies in Astronomy"*, 1997, proceedings disponibles en el Web del CDS/Strasbourg Observatory.

[86] Molina, R. y Mateos, J., "Image models and their role in image processing. Some multichannel results", en *Proceedings of Sonthofen Conference "Advanced Techniques and Methods for Astronomical Information Handling"*, M. Maccarone, F. Murtagh, M. Kurtz y A. Bijaoui, eds., págs. 21–31, 1997.

[87] Molina, R. y Mateos, J., "Mutichannel image restoration in Astronomy", Por aparecer en *Vistas in Astronomy*, 1998.

[88] Nakajima, Y., Hori, H. y Kanoh, T., "A pel adaptive reduction of coding artifacts for MPEG video signals", en *Proceedings International Conference on Image Processing ICIP 94*, tomo 2, págs. 928–932, nov. 1994.

[89] Nasrabadi, N. M. y King, R. A., "Image coding using vector quantization: A review", *IEEE Trans. on Communications*, tomo 36, págs. 957–971, ago. 1988.

[90] Neal, R., *Bayesian learning for neural networks*, Tesis Doctoral, University of Toronto, Department of Computer Science, 1995.

[91] Netravali, A. y Haskell, B. G., *Digital pictures: representation, compression and standards*, Plenum Press, 2ª edón., 1994.

[92] O'Rourke, T. P. y Stevenson, R. L., "Improved image decompression for reduced transform coding artifacts", en *Proceedings of the Image and Video Processing II Conf., SPIE Proc.*, tomo 2182, págs. 90–101, 1994.

[93] O'Rourke, T. P. y Stevenson, R. L., "Improved image decompression for reduced transform coding artifacts", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 5, nº 6, págs. 490–499, dic. 1995.

[94] Özcelik, T., Brailean, J. C. y Katsaggelos, A. K., "Image and video compression algorithms based on recovery techniques using mean field annealing", *Proceedings of the IEEE*, tomo 83, nº 2, págs. 304–316, feb. 1995.

[95] Paek, H. y Lee, S.-U., "A projection-based post-processing technique to reduce blocking artifact using *a prior* information on DCT coefficients of adjacent blocks", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 53–56, 1996.

[96] Paek, H., Park, J.-W. y Lee, S.-U., "Non-iterative post-processing technique for transform coded image sequence", en *Proceedings of the International Conference on Image Processing ICIP 95*, tomo 3, págs. 208–211, 1995.

[97] Park, S. H., Kim, D. S. y Lee, J. S., "On the error concealment techniques for DCT based image coding", en *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP94*, tomo 3, págs. 293–296, 1994.

[98] Park, S. H., Kim, D. S. y Lee, J. S., "Projection onto narrow vector quantization constraint set for postprocessing of vector quantized images", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 57–60, 1996.

[99] Pennebaker, W. B. y Mitchell, J. L., *JPEG still image compression standard*, Van Nostrand Reinhold, 1992.

[100] Peterson, H. A., Ahumada, A. y Watson, A., "The visibility of DCT quantization noise", en *Society for Information Display. Digest of technical papers*, págs. 942–945, 1993.

[101] Peterson, H. A., Ahumada, A. y Watson, A., "Visibility of DCT quantization noise: spatial frequency summation", en *Proceedings of the 1994 SID International Symposium Digest of Technical Papers*, págs. 704–7, 1994.

[102] Prost, R., Ding, Y. y Baskurt, A., "JPEG dequantization array for regularized decompression", *IEEE Trans. on Image Processing*, tomo 6, nº 6, págs. 883–888, jun. 1997.

[103] Raiffa, H. y Schlaifer, R., *Applied statistical decision theory*, Division of Research, Graduate School of Business, Administration, Harvard University, Boston, 1961.

[104] Ramamurthi, G. y Gersho, A., "Nonlinear space-variant postprocessing of block coded images", *IEEE Trans. on Acoustic, Speech and Signal Processing*, tomo 34, nº 5, págs. 1258–1269, oct. 1986.

[105] Reeves, H. C. y Lim, J. S., "Reduction of blocking effects in image coding", *Optical Engineering*, tomo 23, nº 1, págs. 34–37, ene. 1984.

[106] Reeves, S. J. y Eddins, S. L., "Comments on "Iterative procedures for reduction of blocking effects in transform image coding"", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 3, nº 6, págs. 439–440, dic. 1993.

[107] Sampson, D. G., Papadimitriou, D. V. y Chamzas, G., "Post-processing of block-coded images at low bitrates", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 1–4, 1996.

[108] Sauer, K., "Enhancement of low bit-rate coded images using edge detection and estimation", *CVGIP: Graphical Models and Image Processing*, tomo 53, nº 1, págs. 52–62, ene. 1991.

[109] Sikora, T. y Li, H., "Optimal block-overlapping synthesis transforms for coding images and video at very low bitrates", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 6, nº 2, págs. 157–67, abr. 1996.

[110] Silverstein, D. y Klein, S., "Restoration of compressed images", en *Proceedings of the Image and Video Compression Conf., SPIE Proc.*, tomo 2186, págs. 56–64, 1994.

[111] Sonka, M., Hlavac, V. y Boyle, R., *Image processing, analysis and machine vision*, Chapman & Hall, 1993.

[112] Spiegelhalter, D. J. y Lauritzen, S., "Sequential updating of conditional probabilities on directed graphical structures", *Networks*, tomo 20, págs. 579–605, 1990.

[113] Stevenson, R. L., "Reduction of coding artifacts in transform image coding", en *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP 93*, tomo 5, págs. 401–404, 1993.

[114] Stevenson, R. L., "Reduction of coding artifacts in low-bit-rate video coding", en *Proceedings of the 38th Midwest Symposium on Circuits and Systems.*, págs. 854–857, ago. 1995.

[115] Strauss, C., Wolpert, D. y Wolf, D., "Alpha, evidence and the entropic prior", en *Maximum Entropy and Bayesian Methods, Paris*, A. Mohammed-Djafari, ed., págs. 53–71, Kluwer, 1992.

[116] Sultan, A. y Latchman, H., "Adaptive quantization scheme for MPEG video coders based on HVS (human visual system)", en *Proceedings of the Digital Video Compression: Algorithms and Technologies 1996, SPIE Proc.*, tomo 2668, págs. 181–188, 1996.

[117] Suthaharan, S. y Wu, H., "Adaptive-neighbourhood image filtering for MPEG-1 coded images", en *Proceedings of the Fourth International Symposium on Signal Processing and its Applications. ISSPA 96*, tomo 1, págs. 166–167, 1996.

[118] Suthaharan, S., Wu, H. y Yuen, M., "A new post-filtering technique for block-based transform coded images", en *Proceedings of the 1996 IEEE TENCON Digital Signal Processing Applications*, tomo 2, págs. 702–705, 1996.

[119] Tan, S. H., Pang, K. K. y Ngan, K., "Classified perceptual coding with adaptive quantization", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 6, nº 4, págs. 375–388, ago. 1996.

[120] Temerinac, M. y Edler, B., "Overlapping block transform: window design, fast algorithm, an image coding experiment", *IEEE Trans. on Communications*, tomo 43, nº 9, págs. 2417–2425, 1995.

[121] Tzou, K.-H., "Post-filtering of transform-coded images", en *Applications of the digital image processing XI, SPIE Proc.*, tomo 974, págs. 121–126, 1988.

[122] Uz, K. M., Vetterli, M. y LeGall, D., "Interpolative multiresolution coding of advanced television with compatible subchannels", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 1, nº 1, págs. 86–99, 1991.

[123] Wallace, G. K., "Overview of the JPEG (ISO/CCITT) still image compression standard", en *Proceedings of the Image Processing Algorithms and Techniques, SPIE Proc.*, tomo 1244, págs. 220–233, feb. 1990.

[124] Wang, Y., Zhu, Q.-F. y Shaw, L., "Maximally smooth image recovery in transform coding", *IEEE Trans. on Communications*, tomo 41, nº 10, págs. 1544–1551, oct. 1993.

[125] Webb, J. L., "Post-Processing to reduce blocking artifacts for low bit-rate video coding using chrominance information", en *Proceedings of the International Conference on Image Processing ICIP 96*, tomo 2, págs. 9–12, 1996.

[126] Wolpert, D., "On the use of evidence in neural networks", en *Advances in Neural Information Processing Systems 5, San Mateo, California*, C. Giles, S. Hanson y J. Cowan, eds., págs. 539–546, Morgan Kaufmann, 1993.

[127] Xiong, Z., Orchard, M. T. y Zhang, Y.-Q., "A deblocking algorithm for JPEG compressed images using overcomplete wavelet representation", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 7, nº 4, págs. 433–437, abr. 1997.

[128] Yan, L., "Adaptive spatial-temporal postprocessing for low bit-rate coded image sequence", en *Proceedings of the image and video processing II conf., SPIE Proc.*, tomo 2182, págs. 102–109, 1994.

[129] Yan, L., "A nonlinear algorithm for enhancing low bit-rate coded motion video sequence", en *Proceedings of the International Conference on Image Processing ICIP 94*, tomo 2, págs. 923–927, 1994.

[130] Yang, Y. y Galatsanos, N. P., "Edge-preserving reconstruction of compressed images using projections and a divide-and-conquer strategy", en *Proceedings of the International Conference on Image Processing ICIP 94*, tomo 2, págs. 535–539, nov. 1994.

[131] Yang, Y., Galatsanos, N. P. y Katsaggelos, A. K., "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 3, nº 6, págs. 421–432, dic. 1993.

[132] Yang, Y., Galatsanos, N. P. y Katsaggelos, A. K., "Projection-based spatially-adaptive reconstruction of block-transform compressed images", *IEEE Trans. on Image Processing.*, tomo 4, nº 7, págs. 896–908, jul. 1995.

[133] Yuen, M. y Wu, H. R., "Reconstruction artifacts in digital video compression", en *Proceedings of the Digital Video Compression: Algorithms and Technologies 1995 Conf., SPIE Proc.*, tomo 2419, págs. 455–465, 1995.

[134] Zakhor, A., "Iterative procedures for reduction of blocking effects in transform image coding", *IEEE Trans. on Circuits and Systems for Video Technology*, tomo 2, nº 1, págs. 91–95, mar. 1992.