



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**MARCA DE AGUA INVISIBLE PARA IMÁGENES
DIGITALES UTILIZANDO UN ÍNDICE ESTRUCTURAL
DE SIMILARIDAD PARA MEDIR LA CALIDAD
PERCEPTUAL DE LA IMAGEN MARCADA**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN TELECOMUNICACIONES

P R E S E N T A :

Edgar Emilio Morales Delgado

DIRECTOR DE TESIS:

Dr. Francisco Javier García Ugalde



MÉXICO, D.F.

2010



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Gracias a mis padres Gloria y Eduardo por facilitarme todo para estudiar, a mi hermano Eduardo por su apoyo y ejemplo de trabajar para conseguir lo que quiero, a mis primos Alejandro y Carolina por divertirme tanto, a mi tía Teresa que es como mi segunda madre, a mis demás familiares, a las condiciones iniciales del Universo por permitirnos existir, a mis amigos de la facultad por compartir conmigo esta bonita etapa de la vida y que me están esperando ya para la fiesta, especialmente a Aura por su colaboración en el seminario y doy gracias a todos los profesores de la facultad que me dieron clases.

Emilio Morales, Ciudad Universitaria a 5 de febrero de 2010.

ÍNDICE

Capítulo 1	Introducción general	1
1.1	<i>Objetivos</i>	2
1.2	<i>Metodología</i>	2
1.3	<i>Hipótesis</i>	3
1.4	<i>Esbozo de la tesis</i>	3
Capítulo 2	Marca de agua digital y su robustez	5
2.1	<i>Introducción</i>	5
2.1.1	<i>Criptografía</i>	6
2.1.2	<i>Esteganografía</i>	6
2.1.3	<i>Marca de agua digital</i>	7
2.2	<i>Características de la marca de agua</i>	8
2.3	<i>Tipos de marca de agua</i>	9
2.4	<i>Distorsiones y ataques</i>	11
2.4.1	<i>Ruido aditivo</i>	12
2.4.2	<i>Filtrado lineal</i>	12
2.4.3	<i>Recorte de la imagen</i>	13
2.4.4	<i>Distorsiones geométricas</i>	14
2.4.5	<i>Compresión</i>	14
2.4.6	<i>Modificaciones del histograma</i>	15
2.5	<i>Aplicaciones del marcado de agua</i>	15
2.5.1	<i>Monitoreo de emisiones de contenido</i>	15
2.5.2	<i>Demostración de propiedad</i>	15
2.5.3	<i>Control de copias</i>	16
2.5.4	<i>Autenticación</i>	16
2.6	<i>Esquema general del marcado y recuperación de la marca</i>	17
2.7	<i>Codificación de la marca de agua</i>	18
2.8	<i>Codificación de canal</i>	19
2.9	<i>Detección y corrección de errores</i>	19
2.10	<i>Códigos convolucionales</i>	20
2.10.1	<i>Diagrama de estados</i>	22
2.10.2	<i>Diagrama de trellis o enrejado</i>	23
2.10.3	<i>Corrección de errores</i>	24
2.10.4	<i>Codificación de la marca de agua</i>	26



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Capítulo 3	Marca de agua en el dominio transformado	29
3.1	<i>Introducción</i>	29
3.2	<i>Transformada Discreta de Fourier</i>	30
3.3	<i>Transformada wavelet</i>	32
3.3.1	<i>Transformada discreta wavelet en 1-D</i>	35
3.3.2	<i>Transformada discreta wavelet en 2-D</i>	37
3.4	<i>Transformada contourlet</i>	39
3.4.1	<i>Pirámide Laplaciana</i>	41
3.4.2	<i>Multiresolución</i>	43
3.4.3	<i>Banco de filtros direccionales</i>	46
3.4.3	<i>Descomposición Direccional y Multiescala</i>	51
3.5	<i>Espectro disperso</i>	53
Capítulo 4	Recuperación de la marca de agua	55
4.1	<i>Introducción</i>	55
4.2	<i>Recuperación de la marca de agua</i>	55
4.3	<i>Decodificación de la marca de agua</i>	58
4.3.1	<i>Decodificador de Viterbi</i>	58
4.4	<i>El sistema visual humano</i>	59
4.5	<i>Índice estructural de similaridad</i>	61
4.6	<i>Algoritmo de inserción y recuperación de la marca de agua</i>	64
4.6.1	<i>Algoritmo 1</i>	64
4.6.1.1	<i>Inserción de la marca de agua</i>	64
4.6.1.2	<i>Recuperación de la marca de agua</i>	66
4.6.2	<i>Algoritmo 2</i>	66
4.6.2.1	<i>Inserción de la marca de agua</i>	66
4.6.2.2	<i>Recuperación de la marca de agua</i>	67
Capítulo 5	Resultados	71
5.1	<i>Introducción</i>	71
5.2	<i>Espectro disperso mejorado sin codificación convolucional</i>	72
5.3	<i>Espectro disperso mejorado con codificación convolucional</i>	74
5.4	<i>Capacidad del sistema de marcado de agua</i>	81
5.5	<i>Prueba a diferentes ataques</i>	83
5.5.1	<i>Compresión JPEG</i>	83
5.5.2	<i>Ruido Gaussiano</i>	92
5.5.3	<i>Ruido salt & pepper “sal y pimienta”</i>	96
5.5.4	<i>Filtrado Gaussiano paso bajas</i>	101

<i>5.5.5 Recorte de la imagen</i>	104
Capítulo 6 Conclusiones	109
<i>6.1 Trabajo futuro</i>	110
Referencias	112
Anexo 1	115
<i>Anexo 1.1 Pseudocódigo del algoritmo 1</i>	115
<i>Anexo 1.2 Pseudocódigo del algoritmo 2</i>	116
<i>Anexo 1.3 Funciones utilizadas en los algoritmos 1 y 2</i>	117

ÍNDICE DE FIGURAS

Figura 2.1 Esquema general del encriptado y desencriptado	6
Figura 2.2 Sistema de marcado de agua	7
Figura 2.3 (a) Imagen original. (b) Marca de agua. (c) Imagen con marca de agua visible. (d) Imagen con marca de agua invisible.	10
Figura 2.4 Ejemplo de filtrado. (a) imagen original. (b) Imagen suavizada con filtro paso bajas Gaussiano de tamaño 5×5 y $\sigma=1$.	13
Figura 2.5 Ejemplo de imagen recortada	13
Figura 2.6 Ejemplo de distorsión geométrica: rotación.	14
Figura 2.7 Esquema general de marcado y recuperación de marca de agua informado.	18
Figura 2.8 Esquema general de marcado y recuperación de marca de agua ciego.	18
Figura 2.9 Codificación	20
Figura 2.10 Ejemplo de codificador convolucional. $V = 2$, $K = 3$. Por cada bit que entra al codificador salen 2 bits codificados.	21
Figura 2.11 Diagrama de estados de código convolucional $V = 2$, $K = 3$.	22
Figura 2.12 Diagrama de <i>trellis</i> de código convolucional $V = 2$, $K = 3$.	23
Figura 2.13 En azul palabras del código, en rojo vectores de S_n .	24
Figura 2.14 En azul palabras del código, en rojo palabras del código luego de transmisión por canal AWGN.	25
Figura 2.15 Codificador de marca de agua binaria. $V = 3$, $K = 5$.	27
Figura 3.1 División del plano tiempo-frecuencia hecha por la STFT. La resolución es la misma en cualquier región del plano tiempo-frecuencia.	32
Figura 3.2 Función de escalamiento Haar.	33
Figura 3.3 “ <i>wavelet madre</i> ” Haar.	34
Figura 3.4 División del plano tiempo-frecuencia hecho por la transformada <i>wavelet</i> .	35
Figura 3.5 Banco de filtros <i>wavelet</i> 1-D	36
Figura 3.6 Respuesta al impulso del banco de filtros pasa banda. Se divide el ancho de banda original BW .	36
Figura 3.7 Síntesis <i>wavelet</i> 1-D	36

Figura 3.8 Análisis <i>wavelet</i> en 2-D. “L” denota banda de paso baja, “H” denota banda de paso alta.	37
Figura 3.9 Análisis <i>wavelet</i> en imágenes	38
Figura 3.10 Descomposición <i>wavelet</i> de la imagen “Bárbara” a nivel 3.	39
Figura 3.11 Aproximación del contorno suave a diferentes resoluciones [16].	40
Figura 3.12 Esquema de la pirámide Laplaciana [17]. H es el filtro pasobajas de análisis y G el de síntesis. $\downarrow M$ representa el submuestreo y $\uparrow M$ el sobremuestreo.	41
Figura 3.13 Se obtiene la aproximación c a partir de la imagen original x filtrada paso bajas por H y submuestreada por M . Sobremuestreando c y luego de un filtro de síntesis se obtiene la predicción p . La diferencia entre la imagen original x y la predicción p nos da los detalles d .	42
Figura 3.14 Primeros 4 niveles de de la pirámide Gaussiana y Laplaciana [18]. Pirámide Gaussiana columna superior, pirámide Laplaciana columna inferior.	43
Figura 3.15 Síntesis Laplaciana. \hat{x} es la imagen reconstruida [17].	43
Figura 3.16 Subespacios multiescala generados por la pirámide Laplaciana [17].	45
Figura 3.17 Ejemplo de imagen remuestreada. (a) Imagen de “Bárbara”. (b) “Bárbara” remuestreada por la matriz $R = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.	47
Figura 3.18 Imagen rotada por matriz de muestreo <i>quincunx</i> Q_0 .	48
Figura 3.19 Respuesta en frecuencia de (a) filtro en forma de diamante (b) filtro en forma de abanico. [17]	48
Figura 3.20 (a) Filtro de abanico. (b) Filtro de cuadrantes. (c) cuatro direcciones de descomposición [17].	49
Figura 3.21 Partición direccional en frecuencia para $l=3$ [17].	49
Figura 3.22 Subespacios multidireccionales generados por el banco de filtros direccionales [17].	50
Figura 3.23 Subespacios generados por la descomposición multiescala multidirección [17].	51

Figura 3.24 Descomposición piramidal direccional. (a) Se realiza la descomposición piramidal; la descomposición direccional es aplicada solamente al canal paso altas. (b) División en frecuencia resultante: el número de direcciones aumenta con la frecuencia.	52
Figura 3.25 Descomposición <i>contourlet</i> de la imagen “Bárbara” con 2 niveles de descomposición. 4 direcciones en el primer nivel y 8 direcciones en el segundo nivel.	52
Figura 4.1 Probabilidades condicionales $p(r/s_1)$ y $p(r/s_2)$.	56
Figura 4.2 Comparación entre decisión dura y decisión suave.	59
Figura 4.3 Ejemplo de la característica número 2 del sistema visual humano. El brillo absoluto de los cuadrados interiores es el mismo.	60
Figura 4.4 Descomposición <i>contourlet</i> empleada en el sistema de marcado de agua.	65
Figura 4.5 Diagrama de bloques de algoritmo de inserción de marca de agua propuesto.	70
Figura 5.1 PSNR contra factor de multiplicación γ sin utilizar codificación convolucional.	73
Figura 5.2 SSIM contra factor de multiplicación γ sin utilizar codificación convolucional.	73
Figura 5.3 Tasa de bits en error en la marca de agua recuperada sin utilizar codificación convolucional.	74
Figura 5.4 PSNR contra factor de multiplicación γ . La marca de agua fue codificada utilizando codificación convolucional previa a su inserción.	75
Figura 5.5 SSIM contra factor de multiplicación γ . La marca de agua fue codificada utilizando codificación convolucional previa a su inserción.	75
Figura 5.6 Tasa de bits en error en la marca de agua recuperada utilizando decisión dura en el decodificador de Viterbi.	76
Figura 5.7 Tasa de bits en error en la marca de agua recuperada utilizando decisión suave en el decodificador de Viterbi.	77
Figura 5.8 Comparación de la tasa de bits en error en la marca de agua recuperada para procedimiento 1.	78
Figura 5.9 Comparación de la tasa de bits en error en la marca de agua recuperada para procedimiento 2.	79

Figura 5.10 Comparación de la tasa de bits en error en la marca de agua recuperada para procedimiento 1.	79
Figura 5.11 Marca de agua "Ingeniería" de tamaño 9×72 pixeles.	80
Figura 5.12 Marca de agua recuperada para espectro disperso mejorado sin codificación convolucional. Porcentaje de compresión JPEG=77%, PSNR= 40, SSIM=.9578, BER=.0586, γ =.2946.	80
Figura 5.13 Marca de agua recuperada para espectro disperso mejorado utilizando codificación convolucional y decisión dura en el decodificador. Porcentaje de compresión JPEG=77%, PSNR= 40, SSIM=.9577, BER=.0478, γ =.0478.	80
Figura 5.14 Marca de agua recuperada para espectro disperso mejorado utilizando codificación convolucional y decisión suave en el decodificador. Porcentaje de compresión JPEG=77%, PSNR= 40, SSIM=.9577, BER=0, γ =.0478.	81
Figura 5.15 PSNR de la imagen marcada contra tamaño de la marca de agua en bits.	82
Figura 5.16 SSIM de la imagen marcada contra tamaño de la marca de agua en bits.	82
Figura 5.17 PSNR de la imagen comprimida para diferentes valores de porcentaje de compresión JPEG.	84
Figura 5.18 SSIM de la imagen comprimida para diferentes valores de porcentaje de compresión JPEG.	84
Figura 5.19 Tasa de bits en error en la marca de agua recuperada para diferentes valores de porcentaje de compresión JPEG. La decodificación se realizó utilizando decisión dura.	85
Figura 5.20 Tasa de bits en error en la marca de agua recuperada para diferentes valores de porcentaje de compresión JPEG. La decodificación se realizó utilizando decisión suave.	85
Figura 5.21 Comparación entre decisión suave y decisión dura. Tasa de bits en error en la marca de agua recuperada para procedimiento 1 contra porcentaje de compresión JPEG.	86
Figura 5.22 Comparación entre decisión suave y decisión dura. Tasa de bits en error en la marca de agua recuperada para procedimiento 2 contra porcentaje de compresión JPEG.	87
Figura 5.23 Comparación entre decisión suave y decisión dura. Tasa de bits en error en la marca de agua recuperada para procedimiento 3 contra porcentaje de compresión JPEG.	87
Figura 5.24 Marca de agua binaria "Ingeniería"	88

Figura 5.25 (a) Imagen de "Lena". (b) Imagen marcada con la marca de agua invisible "Ingeniería" PSNR=33.52dB, SSIM=.8282.	88
Figura 5.26 Histograma de la imagen "Lena" original. Entropía=7.2894 bits	88
Figura 5.27 Histograma de la imagen "Lena" marcada con marca de agua "Ingeniería". Entropía=7.3432 bits	89
Figura 5.28 (a) Imagen marcada comprimida. Porcentaje de compresión JPEG=80%, PSNR=31.2dB, SSIM=.79. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.	89
Figura 5.29 Marca de agua binaria "Puma"	90
Figura 5.30 (a) Imagen "Bárbara". (b) Imagen marcada con la marca de agua "Puma". PSNR=36.24dB, SSIM=.9291.	90
Figura 5.31 Histograma de la imagen "Bárbara" original. Entropía=7.6321 bits.	91
Figura 5.32 Histograma de la imagen "Bárbara" marcada con marca de agua "Puma". Entropía=7.6576 bits.	91
Figura 5.33 (a) Imagen marcada con marca de agua "Puma" y comprimida. Porcentaje de compresión JPEG=80%, PSNR=28.72dB, SSIM=.8466. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.	92
Figura 5.34 PSNR de la imagen "ruidosa" contra potencia del ruido Gaussiano.	93
Figura 5.35 PSNR de la imagen "ruidosa" contra potencia del ruido Gaussiano.	93
Figura 5.36 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido Gaussiano utilizando codificación convolucional con decisión dura.	94
Figura 5.37 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido Gaussiano utilizando codificación convolucional con decisión suave.	94
Figura 5.38 (a) Imagen marcada con ruido Gaussiano de varianza .01. PSNR=19.91dB, SSIM=.3864. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.	95
Figura 5.39 PSNR de la imagen "ruidosa" contra porcentaje de ruido <i>salt & pepper</i> .	96
Figura 5.40 PSNR de la imagen "ruidosa" contra porcentaje de ruido <i>salt & pepper</i> (acercamiento de 0 a 20%)	97

Figura 5.41 SSIM de la imagen “ruidosa” contra porcentaje de ruido <i>salt & pepper</i> .	97
Figura 5.42 SSIM de la imagen “ruidosa” contra porcentaje de ruido <i>salt & pepper</i> (acercamiento de 0 a 20%)	98
Figura 5.43 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido <i>salt & pepper</i> utilizando decisión dura en el decodificador de Viterbi.	98
Figura 5.44 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido <i>salt & pepper</i> (acercamiento de 0 a 20%) utilizando decisión dura en el decodificador de Viterbi.	99
Figura 5.45 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido <i>salt & pepper</i> utilizando decisión suave en el decodificador de Viterbi.	99
Figura 5.46 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido <i>salt & pepper</i> (acercamiento de 0 a 20%) utilizando decisión suave en el decodificador de Viterbi.	100
Figura 5.47 (a) Imagen marcada con 6% de ruido <i>salt & pepper</i> . PSNR=17.33dB, SSIM=.3724. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.	101
Figura 5.48 PSNR de la imagen filtrada contra desviación estándar del <i>kernel</i> Gaussiano.	102
Figura 5.49 SSIM de la imagen filtrada contra desviación estándar del <i>kernel</i> Gaussiano.	102
Figura 5.50 Tasa de bits en error en la marca de agua recuperada de la imagen filtrada utilizando decisión dura en el decodificador de Viterbi.	103
Figura 5.51 Tasa de bits en error en la marca de agua recuperada de la imagen filtrada utilizando decisión suave en el decodificador de Viterbi.	103
Figura 5.52 (a) Imagen marcada filtrada con filtro pasobajas Gaussiano de tamaño 5×5 y varianza $\sigma=1.2$. PSNR=24.4dB, SSIM=.7296. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.	104
Figura 5.53 PSNR de la imagen recortada contra porcentaje recortado de la imagen marcada.	105

Figura 5.54 SSIM de la imagen recortada contra porcentaje recortado de la imagen marcada.	105
Figura 5.55 Tasa de bits en error en la marca de agua recuperada de la imagen recortada utilizando codificación convolucional con decisión dura.	106
Figura 5.56 Tasa de bits en error en la marca de agua recuperada de la imagen recortada utilizando decisión suave en la decodificación.	106
Figura 5.57 (a) Imagen marcada recortada al 35% PSNR=8.9151dB, SSIM=.3755. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.	107

Capítulo 1

Introducción general

En la última década el desarrollo de las redes de telecomunicaciones y la penetración de su uso entre la población mundial permite que cualquier tipo de información pueda ser intercambiada fácilmente. Por lo tanto, existe la posibilidad de compartir ilegalmente contenido digital con derechos de autor. Para evitarlo se han ideado técnicas que permiten añadir información a imágenes, videojuegos, música y video digital referente a su origen y políticas de uso de tal forma que los dispositivos de copiado o reproducción no los copien si no está permitido o no los reproduzcan si se detecta que no es una copia legítima. Sin embargo actualmente ha sido posible alterar la información añadida y violar dicha “seguridad”. Es por ello que se están desarrollando métodos que implican la incrustación de información en todo el trabajo digital cuya modificación implique el dejar inutilizable el trabajo mismo. Tal es el caso de la marca de agua digital.

Nuestro trabajo se enfoca en la marca de agua digital aplicada a imágenes. Como continuación al trabajo desarrollado en [20] y basados en [19] adoptamos una técnica de espectro disperso mejorado para la inserción de la marca de agua digital en el dominio *contourlet* y proponemos el uso de la codificación de la marca de agua mediante un código convolucional con el objetivo de incrementar su robustez. Es decir, se busca que la alteración a la imagen para que la marca de agua sea irreconocible tenga que ser de tal magnitud que deje a la imagen muy alterada visualmente, tan alterada que pierda sentido el pretender distribuirla ilegalmente. Para medir la alteración perceptual de la imagen proponemos la utilización de un índice de similitud estructural como medida de similitud entre la imagen original y la imagen marcada o entre la imagen original y la imagen atacada.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1.1 *Objetivos*

- Implementar un algoritmo de inserción y recuperación de la marca de agua en el dominio *contourlet*.
- Mejorar la robustez de la marca de agua mediante el uso de codificación convolucional.
- Comparar los resultados de la inserción de la marca de agua realizando la codificación convolucional de la misma respecto a la inserción de la marca de agua sin utilizar codificación.
- Para el caso de la codificación de la marca de agua, comparar las diferencias ente la marca de agua recuperada al utilizar decisión dura respecto a la marca de agua recuperada al utilizar decisión suave en el decodificador de Viterbi.
- Basados en [7], evaluar la calidad de la imagen marcada respecto a la imagen original utilizando una medida de similitud estructural.

1.2 *Metodología*

- Se implementará el algoritmo de inserción y recuperación de la marca de agua digital recurriendo a una técnica de espectro disperso en el dominio *contourlet* utilizando el software Matlab®.
- Se implementará la codificación de la marca de agua utilizando codificación convolucional.
- Se implementará el algoritmo que calcula el índice de similitud estructural entre la imagen original y la imagen marcada.
- Se evaluará la robustez de la marca de agua insertada con el algoritmo implementado. Para ello la imagen marcada será sometida a ataques no maliciosos que aparecen comúnmente en un sistema de transmisión de información digital.

1.3 *Hipótesis*

Basados en los objetivos y la metodología, se plantean las siguientes hipótesis cuya veracidad se pretende confirmar durante el desarrollo de este trabajo.

- El uso de codificación convolucional de la marca de agua, previa a su inserción, incrementa la robustez de la misma respecto a la inserción de la marca de agua sin utilizar codificación convolucional.
- Para el caso de codificación convolucional de la marca de agua, la marca de agua recuperada, utilizando decisión suave en el decodificador de Viterbi, tendrá una menor tasa de errores que la marca de agua recuperada utilizando decisión dura en el decodificador.
- La medida de similitud estructural permite evaluar la calidad de la imagen marcada respecto a la imagen original.

1.4 *Esbozo de la Tesis*

En el capítulo 2 se presenta de manera introductoria la marca de agua digital, cuáles son sus características, qué tipos de marca de agua existen y en qué es utilizada. También se habla sobre la necesidad de codificar la marca de agua, la codificación de canal en un sistema de telecomunicaciones, la posibilidad de corregir y detectar errores y se abordan los códigos convolucionales con el propósito de proteger e incrementar la robustez de la marca de agua. El capítulo 3 trata sobre las diversas formas de representar imágenes en un dominio diferente al espacial, se describe la transformada *contourlet* utilizada en este trabajo para la inserción de la marca de agua, se explica en qué consiste el espectro disperso y se describe cómo se incrusta la marca de agua mediante esa técnica. En el capítulo 4 se muestra cómo se realiza la recuperación de la marca de agua y su posterior decodificación; se describe de manera breve el funcionamiento del decodificador de

Viterbi. Se introducen las características principales del sistema visual humano y se presenta una forma de medir la similitud entre dos imágenes utilizando una técnica que considera las propiedades del sistema visual humano [7]. Posteriormente se describe el algoritmo de inserción y recuperación de la marca de agua empleado en este trabajo. En el capítulo 5 se relatan las pruebas realizadas y los resultados obtenidos de las mismas y en el capítulo 6 se presentan las conclusiones.

Capítulo 2

Marca de agua digital y su robustez

2.1 *Introducción*

Actualmente las tecnologías de la información tienen una gran importancia en la vida diaria. El desarrollo de las mismas ha permitido que de manera sencilla sea posible intercambiar información digital perteneciente o no a un propietario. Incurrir en un atentado contra el derecho de autor que junto con la propiedad industrial conforman la propiedad intelectual, es muy factible. Es por ello que se requieren técnicas para proteger la información de usuarios no autorizados o para demostrar quién es el propietario legítimo. Una de esas técnicas es la criptografía la cual permite proteger la información para que sólo la pueda descifrar el destinatario a quien va dirigido, sin embargo una vez descifrada la información pierde su protección. Otra técnica es la marca de agua digital, la cual permite esconder información de manera electrónica en fotografías, videos o música que usualmente son referidos como trabajos y al conjunto de todos ellos como contenido. Tomemos como ejemplo un billete, en el cual la marca de agua permanece oculta y sólo es visible luego de un proceso especial para observarla (como poner el billete a contraluz). En este caso, la marca de agua porta información referente a la autenticidad del contenido en el que se mantiene oculta.

Un trabajo que se encuentra marcado puede ser víctima de diversos tipos de alteraciones intencionales o no intencionales. Para que la marca de agua no pierda su utilidad es necesario que pueda sobrevivir a dichas modificaciones. Una forma de hacerla más resistente es codificándola con un código detector y corrector de errores antes de insertarla en el trabajo.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

2.1.1 Criptografía

El encriptado o cifrado es un proceso para transformar la información, también llamada texto plano, en un criptograma, el cual se caracteriza por ser ininteligible con la finalidad de transmitir un mensaje de forma segura a través de un canal donde la interceptación es posible. Para convertir el criptograma nuevamente en texto plano es necesario realizar el proceso inverso conocido como desencriptado. Para realizar el encriptado y desencriptado se requiere de una clave que debe ser conocida por el transmisor y receptor del mensaje [4].

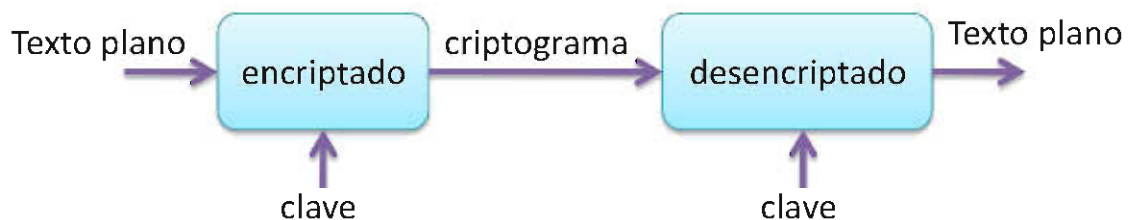


Figura 2.1 Esquema general del encriptado y desencriptado.

La criptografía por sí misma no es suficiente para proteger la información dado que el mensaje ya no permanece seguro luego de haber sido desencriptado.

2.1.2 Esteganografía

Proveniente del griego *steganos* que significa “encubierto” y *graphia* que significa “escrito”, la esteganografía es el arte de la comunicación oculta. En la esteganografía la existencia del mensaje está oculta, excepto para el receptor y el transmisor, por lo que debe de añadirse al trabajo de manera que no haya una considerable alteración perceptual en el mismo. Generalmente la incrustación del mensaje no es tan robusta como la que se pudiese realizar con una marca de agua.

2.1.3 Marca de agua digital

El proceso de incrustación de una marca de agua digital consiste en la inserción de información en una señal portadora la cual puede tratarse de cualquier documento digital. El objetivo principal de insertar información en los contenidos es la protección de la propiedad intelectual. El trabajo marcado puede ser publicado, distribuido a través de una red, comercializado y/o radio emitido; situaciones en las cuales el trabajo puede sufrir alteraciones intencionales o no intencionales. En cualquier momento se puede realizar el proceso de detección y extracción de la marca de agua para demostrar, por ejemplo, la autoría o autenticidad del trabajo. Por lo tanto, de manera general, el sistema de marcado de agua consta de una etapa de inserción y una etapa de recuperación de la marca de agua.



Figura 2.2 Sistema de marcado de agua.

El sistema de marcado de agua digital es similar a un sistema de comunicaciones donde se tiene un transmisor, canal de comunicación y un receptor. En este modelo la inserción de la marca de agua en la señal portadora corresponde a la transmisión de la señal, las modificaciones realizadas por ataques y distorsiones al trabajo marcado corresponden a la transmisión a través del canal de comunicaciones y la recuperación de la información escondida es análoga a la recepción. En los sistemas de cómputo el mensaje a ocultar en la señal portadora es un vector binario de dimensión n de la forma:

$$\bar{b} = (b_1, b_2, \dots, b_n) \quad (2.1)$$

donde $b_k \in \{0,1\}$, $k = \{1,2, \dots, n\}$

Usualmente no se incrusta de manera directa, sino que primero se transforma en una señal \bar{w} con mejores características aleatorias, o de dispersión para ser incrustada.

$$\bar{w} = (w_1, w_2, \dots, w_h) \quad (2.2)$$

donde $w_i \in \{0,1\}$, $i = \{1,2, \dots, h\}$

Al igual que en un sistema de comunicaciones se pueden agregar procesos tales como la codificación de fuente, codificación de canal, espectro disperso, detección y decodificación, como se verá más adelante.

2.2 Características de la marca de agua

- *Robustez*: Es la capacidad que tiene la marca de agua de resistir a las manipulaciones intencionales o no intencionales realizadas a la señal que la porta [1]. El nivel de robustez que debe poseer la marca de agua depende de cada aplicación en particular, sin embargo se pueden considerar cuatro tipos generales de marca de agua de acuerdo a su robustez, los cuales se describirán en el subtema 2.3.
- *Imperceptibilidad*: Es la capacidad que tiene la marca de agua de permanecer oculta en la señal portadora sin alterar ésta última de manera notable.
- *Capacidad*: Cantidad de información en bits que el algoritmo de marcado de agua puede incrustar de tal manera que la marca de agua se pueda recuperar.

Las características de un sistema de marcado de agua digital varían dependiendo de la aplicación. En ocasiones es deseable contar con una marca de agua robusta que sea resistente, en el caso de imágenes, a los ataques como la compresión JPEG, el ruido aditivo, filtrado, recortes o rotaciones. No obstante, al aumentar la robustez de la marca de

agua también aumenta la información de ésta que se añade en el trabajo, produciendo una mayor alteración perceptual.

2.3 Tipos de marca de agua

De acuerdo a su robustez:

- *Marca de agua segura:* Este tipo de marca de agua es resistente a ataques maliciosos y no maliciosos. Los ataques maliciosos son aquellos en los que el atacante conoce el algoritmo de incrustación de la marca de agua e intenta modificarla o extraerla del trabajo. Las manipulaciones no maliciosas son técnicas comunes de procesamiento digital como la compresión con pérdidas, el filtrado y la adición de ruido; en el caso de imágenes las manipulaciones también pueden ser técnicas para aumentar el contraste, modificaciones en el histograma o recortes. Es posible perder la información incrustada sólo después de una significativa alteración de la señal portadora. Esta marca de agua es utilizada en la protección de los derechos de autor.
- *Marca de agua robusta:* Esta marca de agua es resistente sólo a ataques no maliciosos por lo que es usada en aplicaciones donde se espera que nadie intente manipular a la señal portadora con el fin de remover la marca de agua; a pesar de ello también puede ser utilizada en aplicaciones de protección a la propiedad intelectual.
- *Marca de agua semi frágil:* Se utiliza en aplicaciones donde la señal portadora no sufrirá severas alteraciones, sólo algunas ligeras modificaciones como mejoras en la calidad o compresión moderada.

- *Marca de agua frágil:* Es aquella en la que la información incrustada se pierde o altera cuando la señal portadora sufre cualquier tipo de modificación. Es utilizada principalmente en aplicaciones de autenticación.

De acuerdo a su perceptibilidad:

- *Marca de agua visible:* Este tipo demarca de agua es utilizada en imágenes y video. Consiste en la superposición espacial de la imagen portadora y la marca de agua (tipo “logotipo”).
- *Marca de agua invisible:* La marca de agua no es observable en la señal portadora.



(a)



(b)



(c)



(d)

Figura 2.3 (a) Imagen original. (b) Marca de agua. (c) Imagen con marca de agua visible. (d) Imagen con marca de agua invisible.

De acuerdo a su técnica de detección:

- *Algoritmo de marca de agua heurístico:* Es aquel en el que se requiere comparar el trabajo original con el trabajo marcado en el proceso de recuperación de la marca de agua.
- *Algoritmo de marca de agua ciego:* Es aquel que no requiere del trabajo original para extraer la información contenida en el trabajo marcado.

De acuerdo al dominio de inserción

- *Marca de agua en el dominio espacial*

Se trata de una función que modifica directamente los píxeles que componen la imagen de acuerdo a la información que la marca de agua contiene. Usualmente sólo un subconjunto de la imagen es marcado en este dominio.

- *Marca de agua en el dominio transformado*

La marca de agua se inserta en los coeficientes de la transformada de la señal o imagen portadora. Usualmente debido a los otros procesos digitales comunes, se escoge la transformada de Fourier o la transformada discreta coseno para la inserción de la marca de agua en el dominio transformado [1], aunque también se pueden utilizar otras transformaciones tales como la transformada *wavelet*, o la transformada *contourlet*. Comúnmente el dominio transformado proporciona mayor robustez a los ataques.

2.4 Distorsiones y ataques

- *Ataques no maliciosos:* Son los ataques que pueden ocurrir en el uso normal del trabajo.

- *Ataques maliciosos*: Son aquellos cuyo objetivo principal es remover la marca de agua del trabajo o hacerla irrecuperable. Los ataques maliciosos pueden o no explotar el conocimiento del algoritmo de marcado de agua.

2.4.1 *Ruido aditivo*

Algunos procesos realizados en el trabajo pueden tener como efecto la adición de una señal aleatoria considerada ruido aditivo el cual sigue cierta función de densidad de probabilidad.

$$I_N(x, y) = I(x, y) + N(x, y) \quad (2.3)$$

donde I es el trabajo original de tamaño $m \times n$, N es el ruido aditivo, e I_N es el trabajo con ruido.

Este ruido es independiente del trabajo, por lo que las manipulaciones causadas al trabajo por el ruido pueden asumirse como si el trabajo fuese transmitido por un canal con ruido aditivo.

2.4.2 *Filtrado lineal*

En imágenes, se trata de un filtrado espacial que realiza operaciones con los coeficientes de un filtro, también llamado *kernel* del filtro, y los píxeles de una imagen. Es lineal porque la respuesta del filtro está dada por la suma del producto de los coeficientes del filtro y los correspondientes píxeles de la imagen que se enciman con la máscara del filtro, es decir, es una combinación lineal. En el dominio espacial corresponde a una convolución entre los coeficientes del filtro K y la imagen I .

$$I_f(x, y) = I(x, y) * K(i, j) \quad (2.4)$$

donde I es la imagen original de tamaño $m \times n$, I_f es la imagen filtrada, K los coeficientes del filtro de tamaño $I \times J$ y $*$ denota convolución.

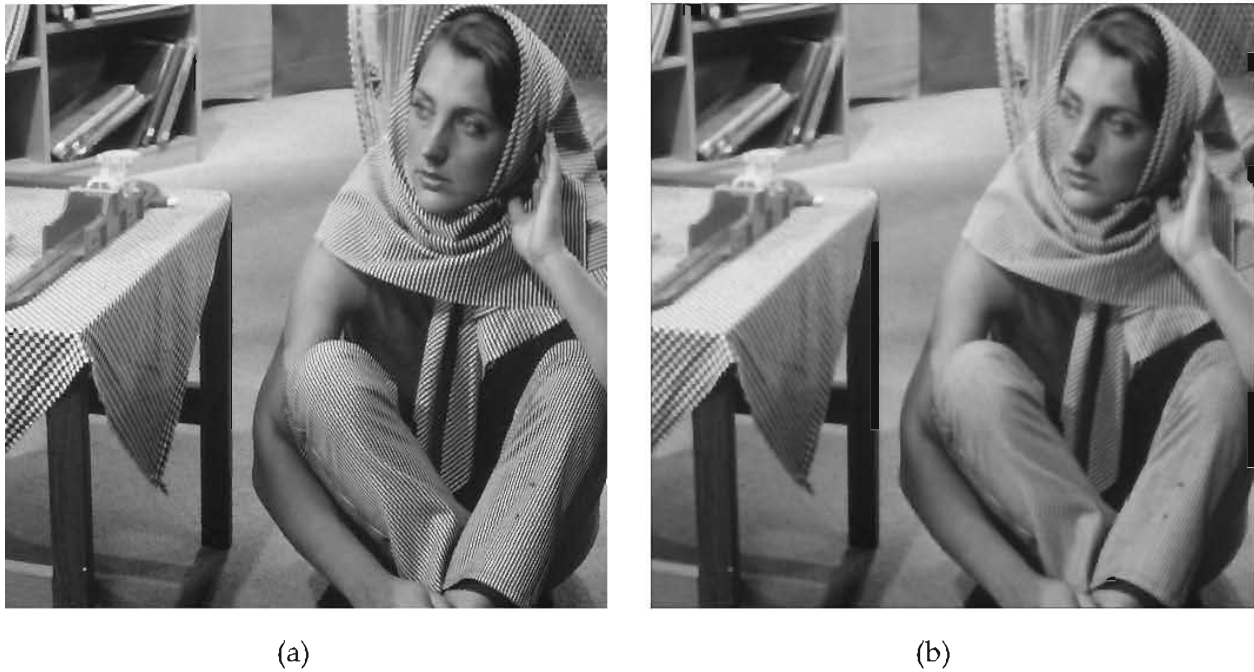


Figura 2.4 Ejemplo de filtrado. (a) imagen original. (b) Imagen suavizada con filtro paso bajas Gaussiano de tamaño 5×5 y $\sigma=1$.

2.4.3 Recorte de la imagen

Este ataque altera el trabajo mediante la remoción de los bordes o alguna parte de la imagen marcada. Se pueden utilizar técnicas de espectro disperso o de dispersión espacial de la marca de agua al momento de incrustarla para hacerla más resistente a este tipo de ataques.



Figura 2.5 Ejemplo de una imagen recortada.

2.4.4 *Distorsiones geométricas*

Estas distorsiones incluyen la rotación, escalamiento espacial, y traslación de la imagen. El atacante puede hacer una distorsión geométrica a la imagen con el propósito de dificultar la extracción de la marca de agua. Al hacer una transformación en las coordenadas de la imagen el algoritmo de detección tendrá dificultad para recuperar la marca de agua a menos que realice una búsqueda exhaustiva con el fin de alinear y sincronizar el trabajo marcado con el patrón de referencia para el cual fue diseñado el detector.



Figura 2.6 Ejemplo de distorsión geométrica: rotación.

2.4.5 *Compresión*

En la compresión con pérdidas el trabajo original no es igual al trabajo descomprimido. En teoría de la compresión, se puede hacer compresión sin considerable degradación debido a que la información que representa la señal electrónica contiene redundancia con respecto a la información necesaria para la percepción humana. Se pueden lograr diferentes niveles de compresión, como en el caso de la compresión de imágenes JPEG, especificando el factor de calidad de la imagen. Cada valor del factor de calidad corresponde a una matriz

de cuantización. Dependiendo de la aplicación, se puede buscar la resistencia de la marca de agua a la compresión JPEG.

2.4.6 Modificaciones del histograma

En algunas aplicaciones será necesario que la marca de agua sea resistente a este tipo de manipulaciones, pues son muy comunes en el mejoramiento de imágenes. Por ejemplo, si una imagen es muy oscura, se buscará aumentar el rango dinámico de los niveles de gris de los píxeles oscuros y disminuir el rango dinámico de los píxeles claros para conseguir un mayor contraste; es decir, se expande el rango de los valores de píxeles oscuros de una imagen y se contrae el de los valores de mayor intensidad. Esto se realiza mediante una operación puntual logarítmica sobre la intensidad de luz de cada píxel que conforma la imagen. También es común obtener el negativo de una imagen para resaltar ciertas figuras presentes en ella. Ambas técnicas realizan modificaciones en el histograma.

2.5 Aplicaciones del marcado de agua

2.5.1 Monitoreo de emisiones de contenido

Existen diferentes organizaciones e individuos interesados en el monitoreo de emisiones de contenido. Las compañías que anuncian sus productos en televisión necesitan asegurar que se emita su publicidad en los espacios “al aire” que rentaron y los propietarios de trabajos con derechos de autor deben asegurar que no se distribuyan ilegalmente sus contenidos. Para ello se puede realizar el monitoreo incrustando una marca de agua que porte información de identificación capaz de ser recuperada por una computadora en la recepción de las emisiones.

2.5.2 Demostración de propiedad

Para demostrar que alguien es el propietario de un trabajo al momento de su creación se puede incrustar una marca de agua que identifique al autor con dicho trabajo. El autor

también puede detectar la distribución de su trabajo en *Internet* mediante un motor de búsqueda que localice su trabajo en la red y así saber si es víctima de fraude.

2.5.3 Control de copias

Para evitar la copia ilegal de contenidos con derechos de autor, como los DVD's, se ha añadido al encabezado MPEG un conjunto de bits llamado bits CGMS (Copy Generation Management System) que le indican a un grabador de DVD's que el disco está protegido contra la copia ilegal abortándose el proceso de copiado. Sin embargo este sistema no ha funcionado pues se puede reproducir el DVD y al mismo tiempo convertir la señal analógica de televisión nuevamente a digital utilizando otro grabador de DVD's creando un disco que ya no posea los bits CGMS, el cual el reproductor de DVD's considerará como disco de libre distribución y todas las copias ilegales de éste podrán ser reproducidas.

Una mejor solución al problema de copias ilegales es migrar a un sistema en el que los bits CGMS también sean incrustados en el video como una marca de agua digital segura. De esta manera si se intenta reproducir un DVD grabado a partir de la señal analógica de video, el reproductor recuperará la marca de agua y al no existir bits CGMS en el encabezado MPEG, el reproductor sabrá que se trata de una copia ilegal [1].

2.5.4 Autenticación

La facilidad de procesar señales de manera digital permite alterar trabajos sin dejar rastros perceptibles de la modificación. Recuperar una marca de agua previamente incrustada en el trabajo original permite saber si el trabajo es el legítimo o se trata de una versión alterada.

2.6 Esquema general del marcado y recuperación de la marca

La fase de inserción de marca de agua digital consiste en una función f que transforma el trabajo original $I(x, y)$ de acuerdo a la marca de agua \bar{w} utilizando una clave K dando como resultado el trabajo marcado $I'(x, y)$. K sirve para dar seguridad al sistema pues aunque se conozca el algoritmo será indispensable emplear la clave correcta para realizar la extracción de la marca de agua. Desde el punto de vista del procesamiento, la marca de agua \bar{w} se trata del vector \bar{b} codificado.

$$I'(x, y) = f(I(x, y), \bar{w}, K) \quad (2.5)$$

donde: $I(x, y)$ es el trabajo original de tamaño $m \times n$, \bar{w} es la marca de agua codificada de dimensión h , y K es la llave.

Como se mencionó anteriormente, existe la detección ciega y la detección informada de la marca de agua. En la detección informada la recuperación se realiza a partir del trabajo marcado, el trabajo original, y la clave, como se muestra en la ecuación 2.6.

$$\bar{w}_r = f(I'(x, y), I(x, y), K) \quad (2.6)$$

donde: \bar{w}_r es la marca de agua recuperada, $I'(x, y)$ es el trabajo marcado, $I(x, y)$ es el trabajo original y K es la clave.

En la detección ciega la recuperación se realiza sin necesidad de utilizar el trabajo original.

$$\bar{w}_r = f(I'(x, y), K) \quad (2.7)$$

donde: \bar{w}_r es la marca de agua recuperada, $I'(x, y)$ es el trabajo marcado y K es la clave. En el sistema de marcado de agua que proponemos la recuperación es ciega, pues no se requiere de la imagen original para hacer la extracción de la marca de agua.

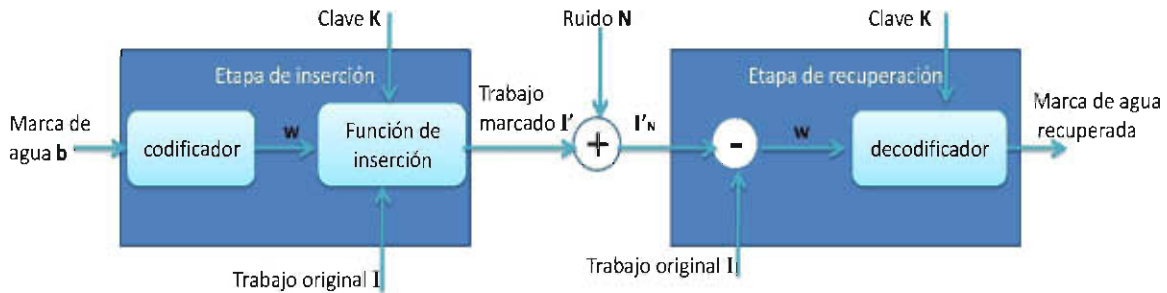


Figura 2.7 Esquema general del sistema de marcado y recuperación de la marca de agua informado.

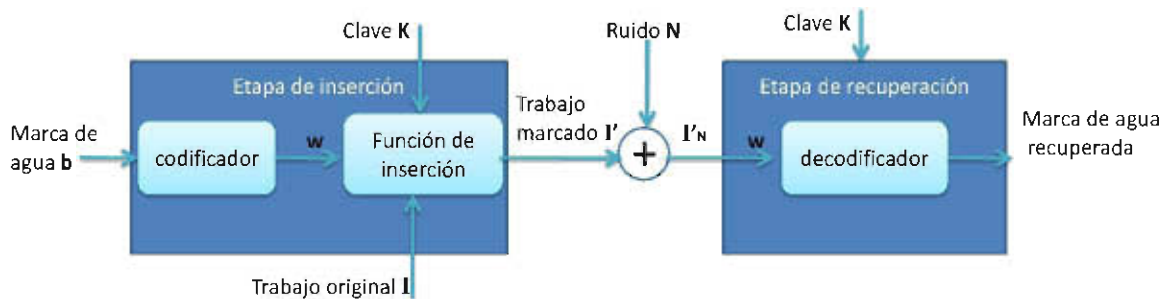


Figura 2.8 Esquema general del sistema de marcado y recuperación de la marca de agua ciego.

Estos esquemas tienen sus variantes como se mostrará más adelante al introducir el diagrama de bloques del algoritmo de inserción y recuperación utilizados en este trabajo.

2.7 Codificación de la marca de agua

Con el propósito de aumentar la robustez del sistema de marcado de agua se puede recurrir a técnicas que impliquen añadir información redundante a la marca de agua que

se desea incrustar. Una de esas técnicas es la codificación de la marca de agua, la cual consiste en transformar el vector binario \bar{b} que contiene los bits de la marca de agua en una versión codificada \bar{w} de mayor longitud. La información redundante adicional permite realizar la detección y corrección de errores en la etapa de extracción de la marca de agua, aunque el costo que se debe pagar por incrustar una mayor cantidad de información es una mayor alteración en el trabajo original.

2.8 Codificación de canal

La codificación de canal consiste en modificar la señal a transmitir de tal forma que sea menos susceptible a sufrir alteraciones en el canal de comunicaciones. Desde un punto de vista de un sistema de transmisión de información, el objetivo es obtener la menor tasa de bits en error posible en el receptor. En el caso de la marca de agua, conseguir una alteración mínima por un posible ataque.

2.9 Detección y corrección de errores

Una forma de disminuir la tasa de bits en error en un sistema de comunicaciones es mediante el uso de códigos que permitan la detección y corrección de errores FEC (*Forward Error Correction*). Algunos de ellos son los códigos de bloque lineales los cuales mapean un bloque de bits de tamaño k en otro de tamaño n , donde $n > k$. Esto se denota comúnmente como código (n, k) , lo cual indica que por cada bloque de k bits que entra al codificador salen n bits codificados. Es decir, se pasa de un espacio vectorial S_k de dimensión k , a otro S_n de dimensión n , mediante el mapeo de cada uno de los 2^k vectores binarios de S_k en su correspondiente vector dentro de S_n . Dado que $n > k$, resulta evidente que el número de vectores posibles (también binarios) dentro de S_n es mayor al de S_k pero en realidad, dado que el mapeo es uno a uno, el número de palabras del código en S_n es igual a 2^k ; por lo

tanto, hay un subespacio vectorial dentro de S_n que contiene las 2^k palabras del código. El resto de los vectores que puede haber en S_n ($2^n - 2^k$ vectores) son vectores que se encuentran dispersos entre las 2^k palabras del código. Esos vectores dispersos hacen que las palabras del código se encuentren más distantes entre sí, lo cual implica que se necesita ruido con mayor potencia para “confundir” palabras del código y es por ello que es posible detectar errores y corregirlos hasta cierto límite. Estos códigos son lineales porque cualquier palabra de código se puede obtener a partir de la combinación lineal de palabras base de código dentro de S_n . Adicionalmente a los códigos de bloque correctores de errores están los códigos convolucionales, también lineales.

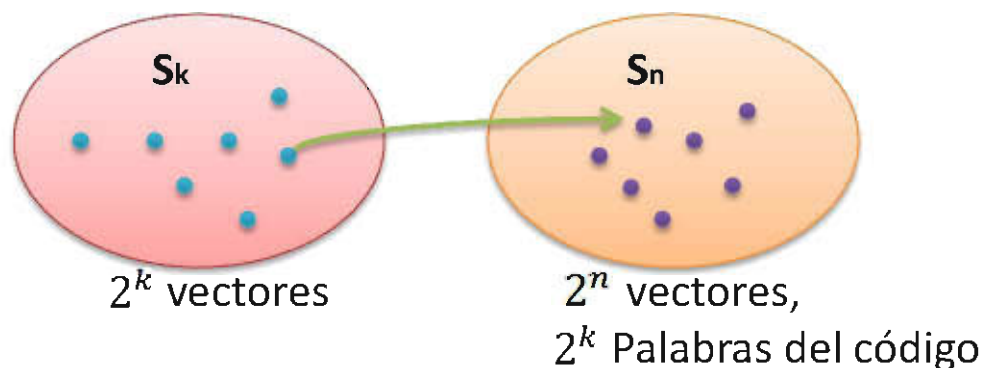


Figura 2.9 Codificación

2.10 Códigos convolucionales

Los códigos convolucionales funcionan con secuencias binarias semi-infinitas tanto a la entrada como a la salida, y por lo general son más rápidos que los códigos de bloque, razón por la cual se prefieren en aplicaciones donde el tiempo de codificación y decodificación es prácticamente en tiempo real respecto a la tasa de transmisión. En este tipo de códigos la codificación se realiza en función del bit que recién acaba de entrar al codificador y algunos bits que han entrado con anterioridad; para ello, el codificador cuenta con registros de corrimiento y sumadores de aritmética módulo-2, los cuales realizan la suma binaria sin acarreo, y producen un efecto de memoria. Cada código

convolucional está definido por su longitud restringida, la tasa del código, y la conexión entre los registros de corrimiento y los sumadores módulo 2. La tasa del código indica el número de bits V que salten del codificador por cada bit que entra: $tasa = \frac{1}{V}$. La longitud restringida K es el número de bits que son considerados para obtener la salida del codificador en un instante de tiempo t , produciendo una memoria de tamaño $K - 1$.

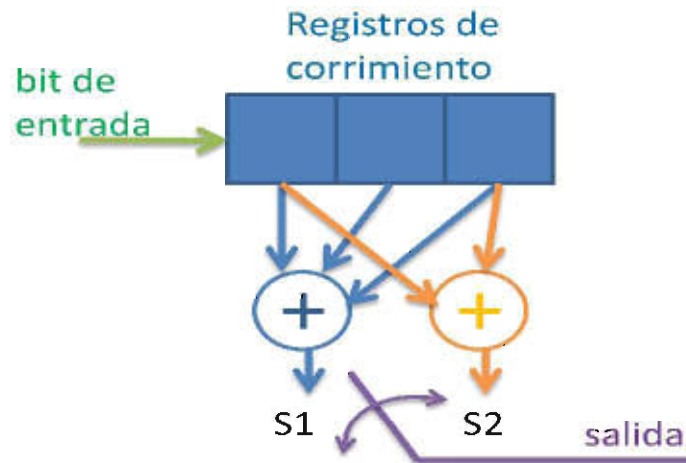


Figura 2.10 Ejemplo de codificador convolucional. $V = 2$, $K = 3$. Por cada bit que entra al codificador salen 2 bits codificados.

La relación entre cada registro de corrimiento y los sumadores está dada por los polinomios generadores del código. Nótese que el número de sumadores del codificador es igual a V . A cada sumador módulo-2 corresponde un polinomio de grado máximo $K - 1$. Para el ejemplo de la Figura 2.10, al sumador $S1$ corresponde el polinomio generador $g_1(x) = 1 + x + x^2$, y al sumador $S2$ el polinomio $g_2(x) = 1 + x^2$. El coeficiente del término de menor orden corresponde a la entrada del registro de corrimiento. También es común expresar los polinomios especificando simplemente sus coeficientes como vectores binarios, es decir, $g_1 = [1\ 1\ 1]$ y $g_2 = [1\ 0\ 1]$.

Es posible representar el codificador mediante diagrama de bloques, diagrama de árbol, diagrama de estados, y enrejado. El diagrama de árbol aunque nos permite seguir la codificación en el tiempo no es muy utilizado por el rápido crecimiento del número de

ramas que se tienen que dibujar para hacerlo. El diagrama de estados es útil para conocer la salida del codificador para cada bit de entrada dependiendo del estado en que se encuentre el registro de corrimientos, pero no nos permite visualizar su evolución en el tiempo. El enrejado es similar al diagrama de estados pero extendido en el tiempo.

2.10.1 Diagrama de estados

Es una representación gráfica de los 2^{K-1} estados en que puede encontrarse los registros de corrimiento del codificador. También muestra la transición de un estado a otro y los bits de salida del codificador dependiendo del bit que entre. Es decir, la entrada de un "1" al codificador se representa con una línea de transición punteada y los bits de salida del codificador se escriben sobre dicha línea; si entra un "0", la línea de transición es continua y se escriben los bits de salida sobre ésta. En la figura 2.11 se muestra el diagrama de estados del codificador convolucional de la figura 2.10.

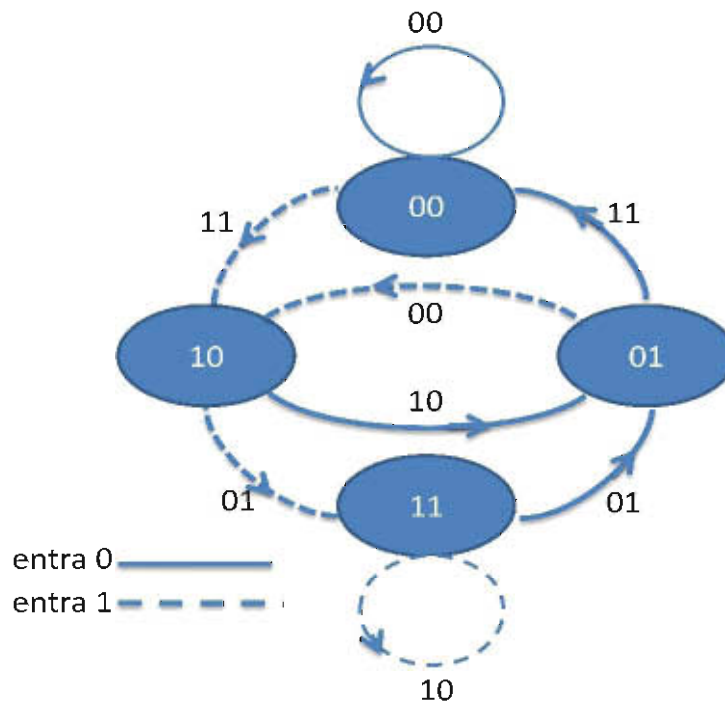


Figura 2.11 Diagrama de estados del código convolucional $V = 2, K = 3$.

2.10.2 Diagrama de *trellis* o enrejado

El diagrama de *trellis*, también conocido como enrejado, es un diagrama que muestra cómo se pasa de un estado a otro en el tiempo dependiendo del bit de entrada que llega al codificador. Al igual que en el diagrama de estados, la entrada de un "0" al codificador se representa con línea continua y la de un "1" con línea puntada. Los estados son representados por un arreglo horizontal de nodos siendo cada renglón de nodos del enrejado correspondiente a un mismo estado. En la figura 2.12 se muestra como ejemplo el enrejado del código convolucional de la figura 2.10.

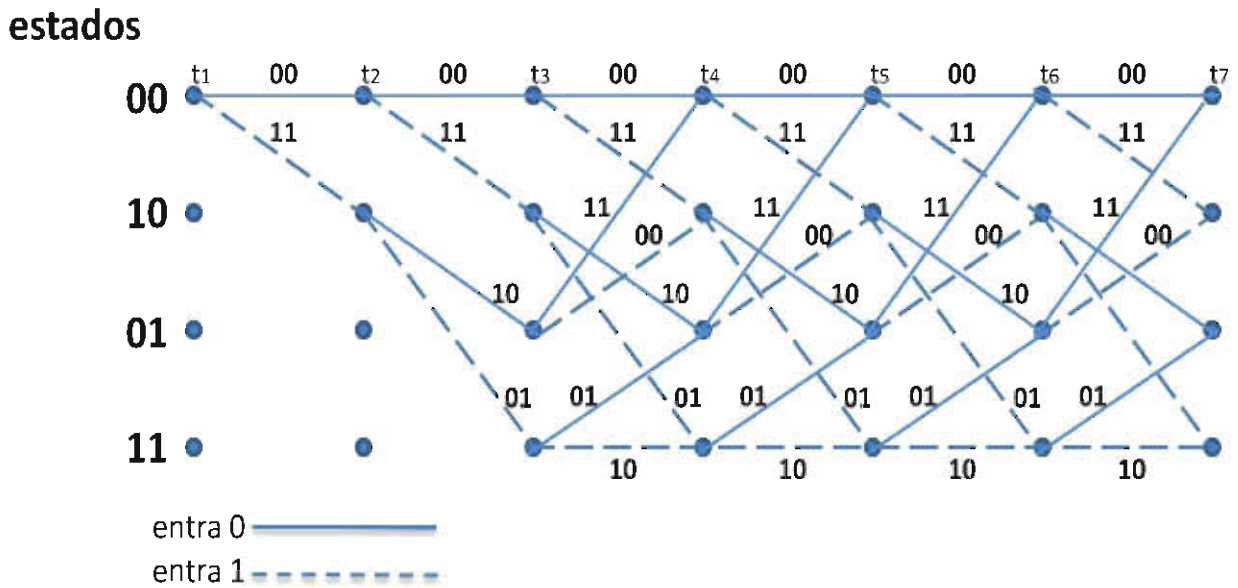


Figura 2.12 Diagrama de *trellis* del código convolucional $V = 2, K = 3$.

Se puede hacer la codificación con el enrejado de la siguiente manera: se parte de un estado inicial en el cual todos los bits del registro de corrimientos del codificador son cero lo cual corresponde al nodo más a la izquierda del primer renglón del enrejado en el tiempo t_1 . Posteriormente se va definiendo un camino dependiendo de los bits de la secuencia que entra al codificador. Los bits sobre las líneas del camino definido corresponden a los bits que salen del codificador.

El enrejado resulta muy útil al realizar la decodificación, como se verá más adelante.

2.10.3 Corrección de errores

La corrección de errores es posible gracias a la redundancia que se introduce al hacer la codificación. En los códigos de bloques, como los códigos Hamming o los códigos BCH, se tiene el concepto de distancia mínima d_{\min} (también conocida como distancia de Hamming) la cual está estrechamente relacionada con la capacidad de corregir errores. Tomemos como ejemplo una codificación mediante un código de bloques (n, k) . Supóngase un conjunto de vectores correspondiente a las palabras del código dentro del espacio S_n . Los vectores se encuentran al centro de esferas de radio t , como se muestra en la figura 2.13. La separación entre dos de los vectores está dada por el número de bits que tienen diferentes, esto es la distancia de Hamming. A mayor distancia de Hamming entre dos vectores, mayor será su separación.

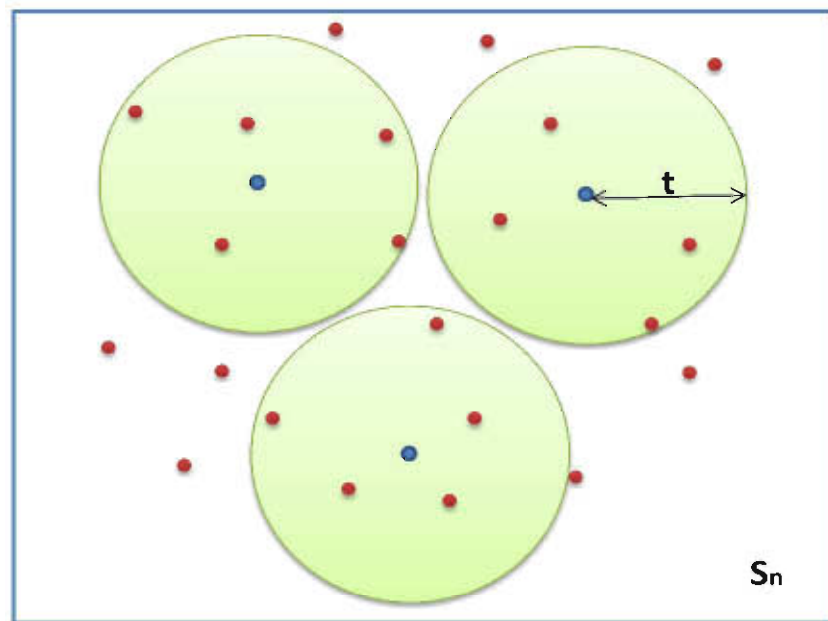


Figura 2.13 En azul palabras del código, en rojo vectores de S_n .

La distancia t máxima del radio de cada esfera sin que se superponga con esferas vecinas se calcula a partir de la distancia mínima que exista entre dos palabras del código. De esta

forma la separación de esferas de igual tamaño en cuyo centro se encuentran las palabras del código está dada por:

$$t = \frac{d_{\min}-1}{2} \quad (2.8)$$

En los códigos de bloques, el valor de t es el número de errores que el decodificador es capaz de corregir en cada bloque.

Si el ruido presente en el canal modifica alguna de las palabras del código, mientras la palabra modificada se encuentre dentro de la esfera de la palabra del código correcta, entonces se podrán corregir esos errores al hacer la decodificación. Para que se tenga error en la decodificación el ruido tendría que sacar una palabra del código fuera de su esfera de pertenencia desplazándola hacia el interior de la esfera correspondiente a otra palabra del código.

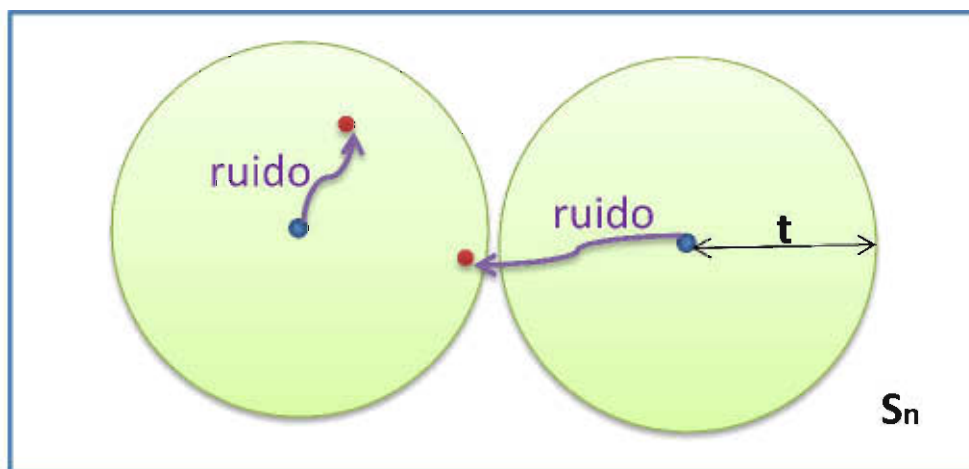


Figura 2.14 En azul palabras del código, en rojo palabras del código luego de transmisión por canal AWGN.

En los códigos convolucionales el equivalente a la distancia mínima se conoce como distancia libre d_{free} y su significado es similar al de la distancia mínima: la separación entre vectores. Como en códigos convolucionales no hay bloques de tamaño fijo definido, se calculan distancias de Hamming acumuladas en el camino del enrejado al hacer la decodificación. La distancia libre se obtiene de la siguiente manera: Se asume que se transmite una secuencia de puros ceros (secuencia nula) y que existe un error en la

secuencia recibida, la distancia libre es la mínima distancia de Hamming acumulada entre el camino de bits codificados con error en el enrejado y el camino correspondiente a la codificación de sólo ceros (secuencia nula). Mientras más grande sea la distancia libre, más robusto será el código convolucional frente al ruido y en nuestro trabajo, más resistente será la marca de agua a los ataques.

Es posible conocer el número de errores que puede corregir un código convolucional de igual forma que se hace con un código de bloque, sólo hay que cambiar d_{min} por d_{free} en la ecuación 2.8.

$$t = \frac{d_{free}-1}{2} \quad (2.9)$$

Sin embargo, en los códigos convolucionales, dado que no se manejan bloques, no se conoce con exactitud cuántos errores se pueden corregir en un bloque de bits determinado. En la práctica, el valor de t es el número máximo de errores que se pueden corregir en una secuencia de tamaño igual a entre 3 y 5 longitudes restringidas [8]. Es por ello que para poder conocer mejor la capacidad de corregir errores y el rendimiento de los códigos convolucionales resulta útil realizar simulaciones y obtener gráficas que relacionan el BER vs SNR.

2.10.4 Codificación de la marca de agua

El código convolucional propuesto para la codificación de la marca de agua fue tomado de [10], en donde se muestran las características y resultados de simulaciones de 27 códigos convolucionales diferentes. Fue escogido por haber presentado las menores tasas de bit en error en las simulaciones [10]. Es un código de tasa $\frac{1}{3}$ y longitud restringida $K = 5$. Los polinomios generadores son $g_1 = [1\ 1\ 1\ 1\ 1]$, $g_2 = [1\ 1\ 0\ 1\ 1]$ y $g_3 = [1\ 0\ 1\ 0\ 1]$. El codificador se muestra en la figura 2.15.

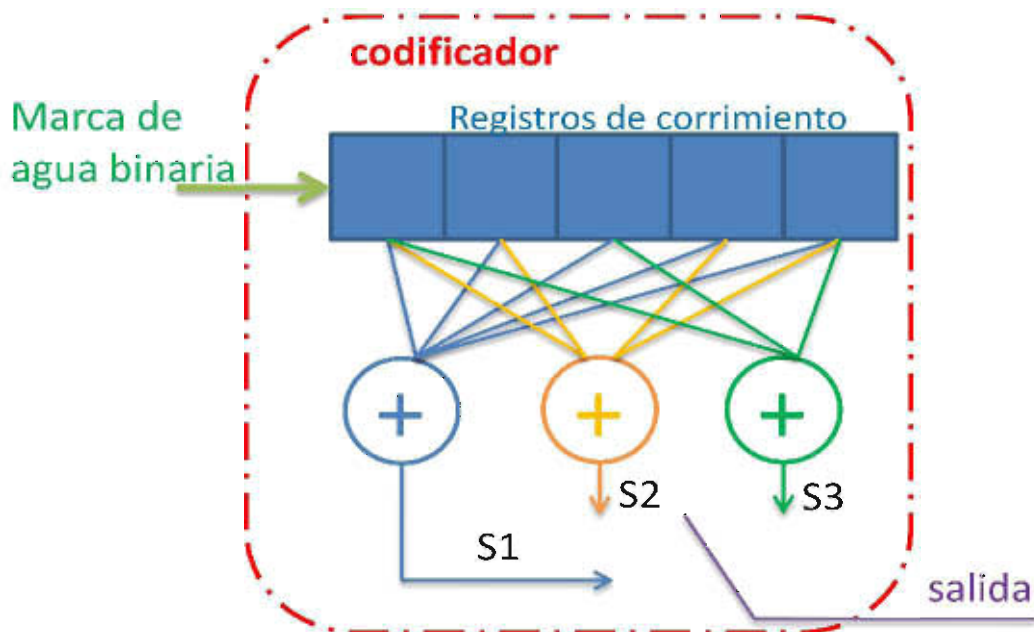


Figura 2.15 Codificador de marca de agua binaria. $V = 3$, $K = 5$.

La distancia libre de este código convolucional es $d_{free} = 12$ [8]. Por lo tanto, el número de errores que puede corregir son 5.

En el proceso de marcado de agua utilizado en esta tesis, la marca de agua binaria es primero codificada utilizando el código convolucional de la figura 2.15 y posteriormente es incrustada utilizando una técnica de espectro disperso, la cual será descrita en el siguiente capítulo.

Capítulo 3

Marca de agua en el dominio transformado

3.1 *Introducción*

Una imagen digital, función bidimensional que asigna a cada par de variables espaciales un valor de intensidad de luz, se puede representar en un dominio distinto al espacial. En otro dominio se tiene una distribución de la energía de la imagen digital diferente a la distribución que se tiene en el dominio espacial. Un aspecto importante de la transformación de imágenes es la posibilidad de realizar el proceso inverso mediante la transformación inversa.

Las transformadas más utilizadas en la representación de imágenes son la transformada de Fourier, la transformada coseno discreta, la transformada *wavelet*, la transformada *contourlet*, entre otras. En el procesamiento digital de imágenes resulta útil poder representar una imagen digital en diferentes dominios. Por ejemplo, la transformada coseno discreta es adoptada por el estándar de compresión con pérdidas JPEG (*Joint Photographic Experts Group*) por su capacidad de representar una señal con pocos coeficientes; es decir, la energía de la señal está compactada en un número pequeño de coeficientes de la transformada. De manera similar, la transformada *wavelet* es empleada en el estándar JPEG-2000 de compresión de imágenes con pérdidas mediante el uso del algoritmo EZW (*Embedded Zerotree Wavelet*).

En este trabajo explotamos la descomposición *contourlet* de la imagen en diferentes bandas de frecuencia y utilizamos técnicas de espectro disperso para la inserción de la marca de agua digital con el propósito de hacerla más robusta. Para expandir su espectro, la marca de agua binaria es multiplicada por una secuencia pseudoaleatoria, generando una señal de mayor banda [12], es decir, tiene más componentes en el dominio de la frecuencia.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Expandir el espectro de una señal aumenta su robustez ante el ruido y en nuestro caso, aumenta la robustez de la marca de agua digital. Una vez incrustada la marca de agua en los coeficientes *contourlet* se realiza la síntesis de la imagen para obtener la imagen marcada.

3.2 Transformada Discreta de Fourier

La transformada de Fourier es una transformada reversible que permite representar una señal como la sumatoria de funciones exponenciales complejas de diferente frecuencia. Para el caso de señales discretas, la transformada discreta de Fourier está dada por:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-i2\pi nk/N} \quad (3.1)$$

$$x(n) = \sum_{k=0}^{N-1} F(k) e^{i2\pi nk/N}$$

donde $n = \{1, 2, \dots, N - 1\}$, $k = \{1, 2, \dots, N - 1\}$.

Dado que una imagen digital es función de dos variables espaciales discretas, se extiende la transformada discreta de Fourier y su transformada inversa al caso de dos dimensiones:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \quad (3.2)$$

para $u = 1, 2, \dots, M - 1$, $v = 1, 2, \dots, N - 1$.

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)}$$

para $x = 1, 2, \dots, M - 1$, $y = 1, 2, \dots, N - 1$.

La desventaja de la transformada de Fourier es que al pasar al dominio de la frecuencia no se puede conocer la información referente al dominio espacial. Podemos conocer la distribución de la energía de la señal en el espectro, pero no sabemos cuál es su

localización en el espacio. Eso no sería problema si la imagen no presentara variaciones de frecuencia al recorrerla en el espacio. Sin embargo la mayoría de las señales, como las imágenes, tienen componentes de diferentes frecuencias en distintas regiones espaciales. Para el caso de señales unidimensionales en función del tiempo, la transformada de Fourier no es una buena representación de éstas si sus componentes frecuenciales cambian en el tiempo; a este tipo de señales sería mejor poder caracterizarlas en el tiempo y la frecuencia. Dicha caracterización se puede lograr mediante la STFT (*Short Time Fourier Transform*). En el caso de señales unidimensionales en función del tiempo, esta transformada consiste en dividir la señal en intervalos de tiempo en los que se asume que es estacionaria; es decir, intervalos de tiempo en los cuales sus componentes en frecuencia permanezcan constantes, y hacer la transformada de Fourier en cada uno de esos intervalos. Para ello se multiplica la función base de la transformada de Fourier por una ventana temporal de longitud finita (su longitud también se conoce como soporte) que se va recorriendo a lo largo de la señal que se va a transformar.

$$STFT(\tau, f) = \int_{-\infty}^{\infty} x(t)g(t - \tau)e^{-i2\pi ft} dt \quad (3.3)$$

Sin embargo se tiene una restricción en la concentración de la energía de la señal en el dominio del tiempo y la frecuencia dada por el principio de incertidumbre de Heisenberg. En mecánica cuántica este principio indica la incertidumbre existente entre el momento lineal de una partícula y su posición. En el caso de la STFT el principio de incertidumbre implica que mientras más estrecha sea la ventana en el tiempo mejor será la localización de la señal en el tiempo pero mayor será la incertidumbre en la frecuencia y viceversa. Es decir, sólo se puede relacionar ciertas bandas de frecuencia con intervalos de tiempo [13].

$$\sigma_t \sigma_\omega \geq \frac{1}{2} \quad (3.4)$$

A mayor resolución en el tiempo menor será la resolución en la frecuencia. La transformada de Fourier es el caso límite en el cual la ventana temporal es infinita y es posible localizar una sola componente frecuencia de la señal pues se tiene la máxima

resolución en la frecuencia. Dennis Gabor introdujo dichas ventajas temporales utilizadas en la STFT y las llamó átomos de Gabor [13].

La STFT divide el plano tiempo-frecuencia en rectángulos del mismo tamaño como se muestra en la figura 3.1. La relación entre la anchura en el tiempo y altura en la frecuencia de cada rectángulo está determinada por el tamaño de la ventana temporal utilizada y el principio de incertidumbre. Ventanas anchas en el tiempo nos dan buena resolución en la frecuencia pero mala resolución en el tiempo; ventanas angostas en el tiempo brindan buena resolución en el tiempo pero mala resolución en la frecuencia.

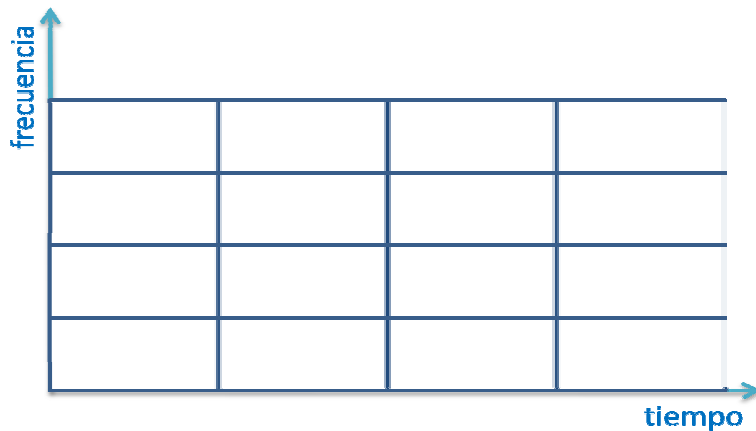


Figura 3.1 División del plano tiempo-frecuencia hecha por la STFT. La resolución es la misma en cualquier región del plano tiempo-frecuencia.

Aunque la STFT nos permite conocer qué componentes espectrales de la señal están presentes en cada intervalo de tiempo existe el problema de escoger el ancho de la ventana temporal que nos brinde la mejor discriminación de las componentes presentes en cada señal.

3.3 Transformada *wavelet*

La transformada *wavelet* permite descomponer las señales en versiones de diferente resolución temporal y frecuencial. Para ello se parte de una función $\varphi(t)$ conocida como

función de escalamiento la cual puede ser trasladada en el tiempo por el parámetro k y escalada en el tiempo por el parámetro j .

$$\varphi_{j,k}(t) = 2^{\frac{j}{2}}\varphi(2^j t - k) \quad (3.5)$$

donde $j, k \in \mathbb{Z}$.

Las funciones de escalamiento pertenecen al espacio vectorial L^2 de funciones cuya integral de su cuadrado es finita. Cada función de escalamiento también pertenece a un subespacio vectorial: $V_j: \{\varphi_{j,k}(t)\}$. Y además:

$$V_j \subset V_{j+1} \subset L^2 \quad (3.6)$$

Lo anterior implica que una función de escala j puede crearse a partir de funciones de escalamiento de mayor resolución $j + n$ mas no es posible lo inverso. Esta es la base del análisis multiresolución.

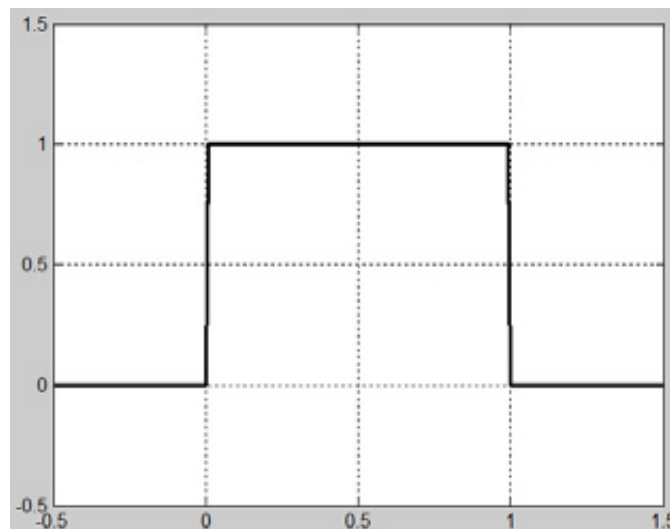


Figura 3.2 Función de escalamiento Haar.

Sin embargo las características de una señal pueden ser descritas mejor mediante un conjunto de funciones que abarquen la diferencia de los espacios vectoriales. Este conjunto de funciones se conocen como *wavelets* [15]. Cada función *wavelet* obtenida a partir de la

combinación lineal de funciones de escalamiento es conocida como “*wavelet madre*” y está dada por:

$$\psi(t) = \sum_n h_\psi(n)\sqrt{2}\varphi(2t - n) \quad (3.7)$$

donde $n \in \mathbb{Z}$.

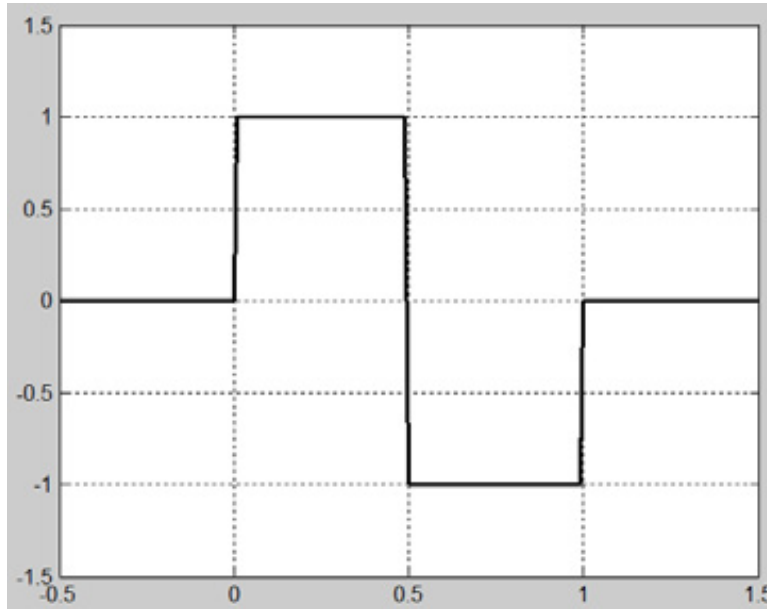


Figura 3.3 “*wavelet madre*” Haar.

Versiones escaladas y trasladadas de la función “*wavelet madre*” pueden ser obtenidas a partir de la siguiente ecuación:

$$\psi_{j,k}(t) = 2^{\frac{j}{2}}\psi(2^j t - k) \quad (3.8)$$

Una vez definido el origen de la función *wavelet*, la transformada *wavelet* continua en una dimensión está dada por [13]:

$$Wf(\tau, s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-\tau}{s} \right) dt \quad (3.9)$$

donde τ es la posición de la *wavelet* en el tiempo y s su escala. La escala es inversa a la frecuencia.

Para cada valor de τ y s se realiza la integral del producto de la función $f(t)$ y la función *wavelet* escalada y trasladada. Componentes de la señal presentes en la escala s y posición en el tiempo τ darán como resultado un valor grande para $Wf(\tau,s)$. La transformada *wavelet* nos permite localizar componentes de la señal original en el tiempo y frecuencia con soportes temporales de diferente tamaño: soportes grandes en el tiempo ofrecen buena resolución en la frecuencia pero mala en el tiempo, y soportes pequeños en el tiempo ofrecen buena resolución en el tiempo pero mala en la frecuencia.

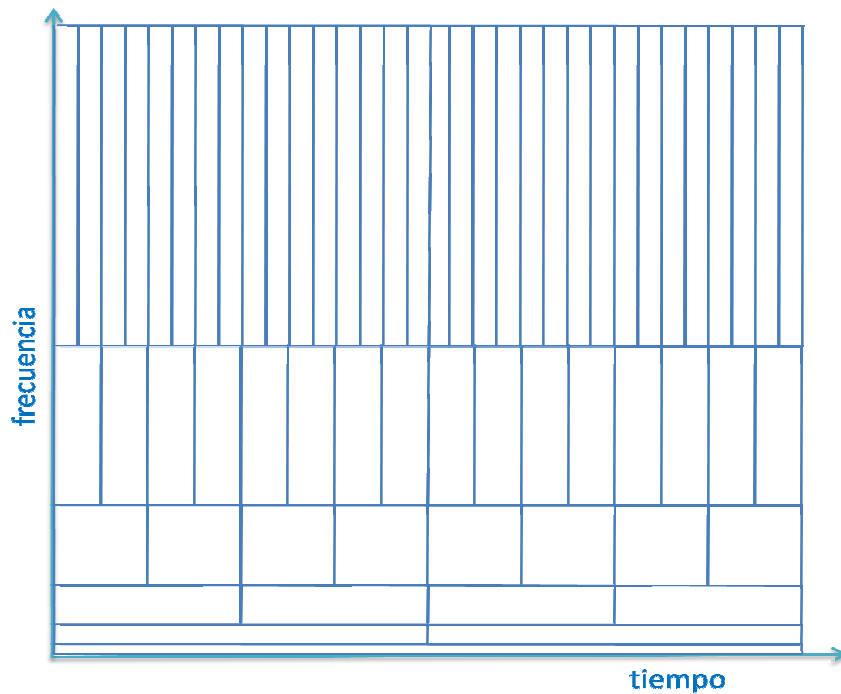


Figura 3.4 División del plano tiempo-frecuencia hecho por la transformada *wavelet*.

3.3.1 Transformada discreta *wavelet* en 1-D

En realidad las computadoras trabajan con un número finito de muestras de señales discretas en el tiempo. En [15] se muestra que la transformada discreta *wavelet* en una dimensión se obtiene a partir de la convolución de la respuesta al impulso de los filtros *wavelet* h_ψ y h_ϕ con la secuencia original $s(n)$, seguidas por un submuestreo. h_ϕ es un filtro pasa bajas y h_ψ un filtro pasa altas. Por lo tanto, la descomposición *wavelet* discreta se puede hacer de manera iterativa utilizando un banco de filtros de análisis *wavelet*.

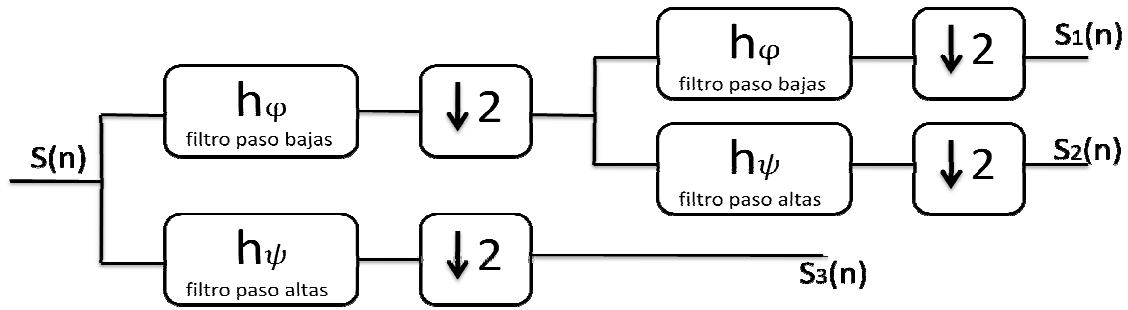


Figura 3.5 Banco de filtros *wavelet* 1-D.

Luego de realizar el filtrado tenemos la mitad del ancho de banda de la señal en cada canal. Es por ello que podemos hacer un submuestreo por 2 sin perder información. En el primer nivel de descomposición, la secuencia original de tamaño M es descompuesta en dos secuencias de tamaño $\frac{M}{2}$ cada una.

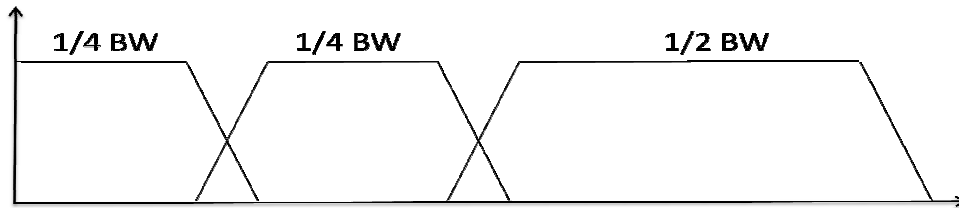


Figura 3.6 Respuesta al impulso del banco de filtros pasa banda. Se divide el ancho de banda original BW .

La síntesis *wavelet* se realiza de manera inversa a la descomposición. Cada canal pasa banda sobremuestreado es filtrado con su respectivo filtro de síntesis. De manera gráfica se muestra el proceso en la figura 3.7.

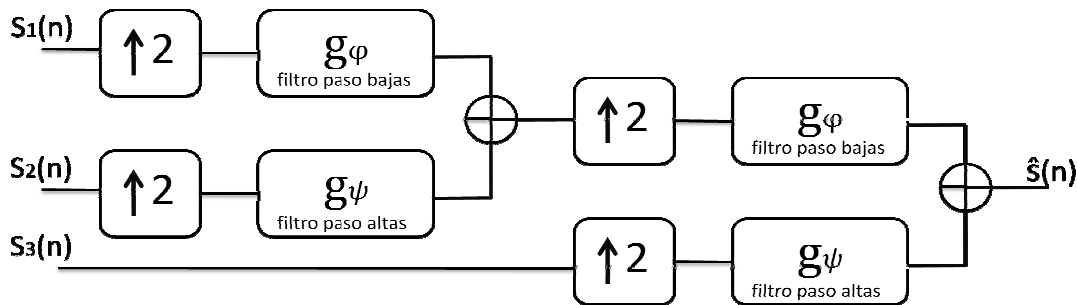


Figura 3.7 Síntesis *wavelet* 1-D.

3.3.2 Transformada discreta wavelet en 2-D

Para el caso de señales bidimensionales, como las imágenes digitales, la transformada discreta wavelet se realiza aplicando el filtro paso bajas h_φ y el paso altas h_ψ en la dirección horizontal x y vertical y de la imagen.

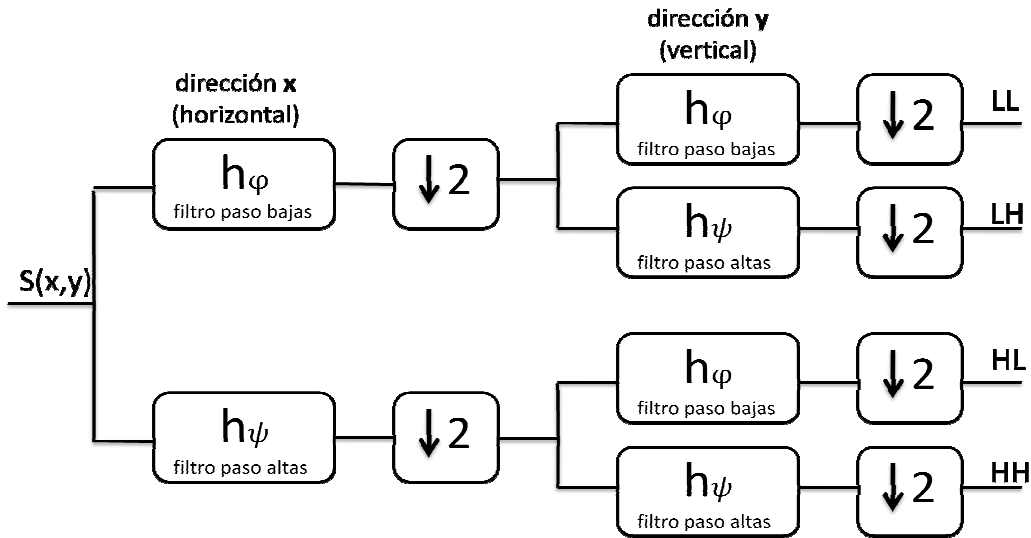


Figura 3.8 Análisis wavelet en 2-D. “L” denota banda de paso baja, “H” denota banda de paso alta.

Al igual que en el caso unidimensional, luego del filtrado, las muertas redundantes hacen posible realizar el decimado por 2 sin perder información.

La DWT aplicada a imágenes en cada nivel de descomposición nos da 4 versiones reducidas en resolución de la imagen original, cada una caracterizada por la combinación de los filtros paso bajas y paso altas. Dicha combinación permite extraer los bordes horizontales, verticales y diagonales de la imagen como se muestra en la tabla 3.1.

LL	baja-baja
LH	bordes horizontales
HL	bordes verticales
HH	bordes diagonales

Tabla 3.1

Como las componentes de baja frecuencias son abundantes en las imágenes, las descomposiciones subsecuentes se realizan usualmente en la versión LL.

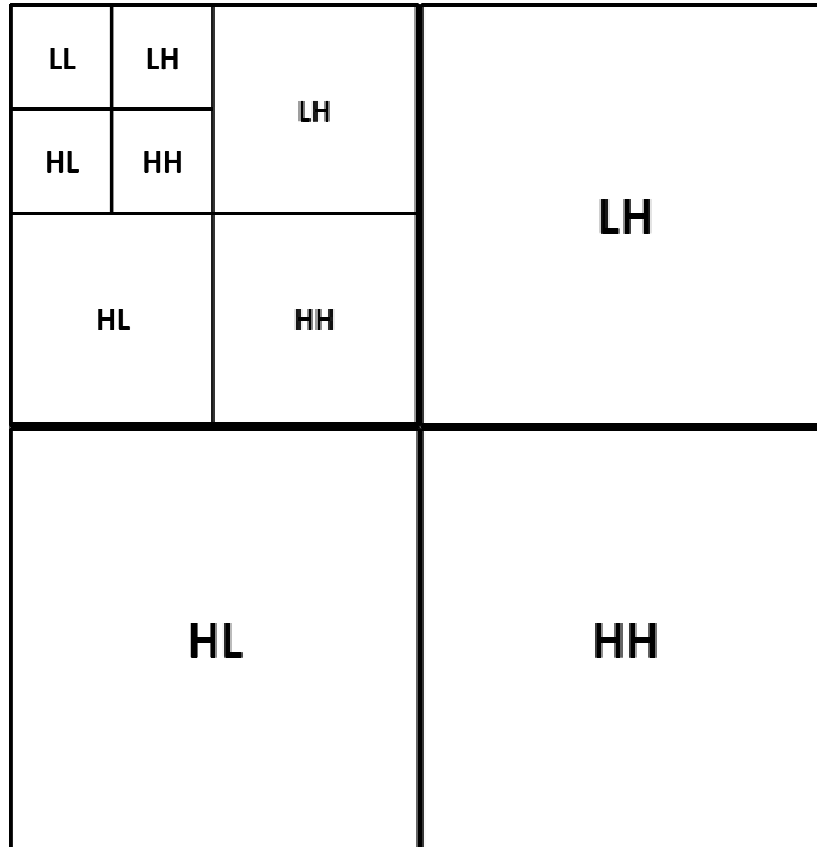


Figura 3.9 Análisis *wavelet* en imágenes.

La transformada *wavelet* permite capturar los detalles horizontales, verticales y diagonales de una imagen, sin embargo no es buena para capturar contornos en otras direcciones.



Figura 3.10 Descomposición *wavelet* de la imagen "Bárbara" a nivel 3.

3.4 Transformada *contourlet*

Supóngase que se desea construir un contorno suave como el que se muestra en la figura 3.11. Se puede hacer a partir de *wavelets* 2-D pero con la limitante de utilizar "formas base" rectangulares. En resoluciones cada vez más finas se observa que se requiere un gran número de rectángulos para capturar el contorno y resulta evidente la deficiencia de las *wavelets* en 2-D. Sin embargo, si se utilizan rectángulos de diferentes tamaños que puedan ser orientados en diferentes direcciones se puede representar dicha curva suave a partir de un pequeño número de rectángulos inclusive en resoluciones finas. Esto es precisamente lo que realiza la Transformada *contourlet* como se verá más adelante.

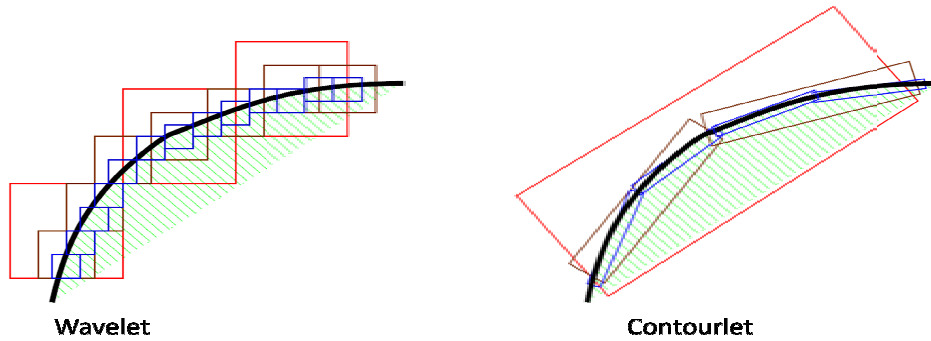


Figura 3.11 Aproximación del contorno suave a diferentes resoluciones [16].

Existe un conjunto de características deseables en la representación de imágenes, las cuales se mencionan enseguida [17]:

Multiresolución: Se debe de poder aproximar las imágenes desde una versión “tosca” hasta una versión de resolución fina.

Localización: Los elementos base de la imagen deben estar bien localizados en el dominio espacial y de la frecuencia.

Muestreo crítico: La representación debe realizarse con la menor redundancia posible.

Direccionalidad: Las funciones base deben de poder ser orientadas en un gran número de direcciones diferentes.

Anisotropía: Las funciones base deben tener diferentes formas y relación de aspecto para poder capturar los contornos.

A diferencia de la representación *wavelet* que sólo cumple con las tres primeras características, la representación *contourlet* cumple con las cinco características.

La transformada *contourlet* realiza una descomposición direccional multiresolución. Utiliza la pirámide Laplaciana para hacer el análisis multiresolución y posteriormente el canal pasa-banda es descompuesto en diferentes direcciones. Enseguida se mostrará más a detalle cómo se realiza la descomposición *contourlet*.

3.4.1 Pirámide Laplaciana

La pirámide Laplaciana [18] es utilizada en la transformada *contourlet* para realizar la descomposición multiresolución de la imagen. Primero la imagen es filtrada con un filtro Gaussiano pasobajas seguida de un submuestreo obteniendo como resultado una versión de menor resolución de la imagen original (aproximación). Es posible dicha reducción en la frecuencia de muestreo pues previamente se ha reducido la frecuencia máxima de la imagen con el filtrado pasobajas. Sobremuestreando la aproximación se obtiene una predicción de la imagen original. A partir de la diferencia entre la imagen original y la predicción se obtiene una versión pasobanda o de error que contiene los detalles de la imagen. Se puede seguir la descomposición de manera iterativa con la versión pasobajas. La pirámide Laplaciana se construye con las versiones pasobanda que se van obteniendo.

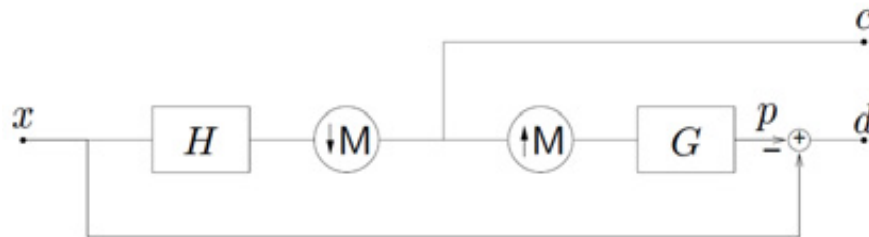


Figura 3.12 Esquema de la pirámide Laplaciana [17]. “ H ” es el filtro pasobajas de análisis y “ G ” el de síntesis. “ $\downarrow M$ ” representa el submuestreo y “ $\uparrow M$ ” el sobremuestreo.

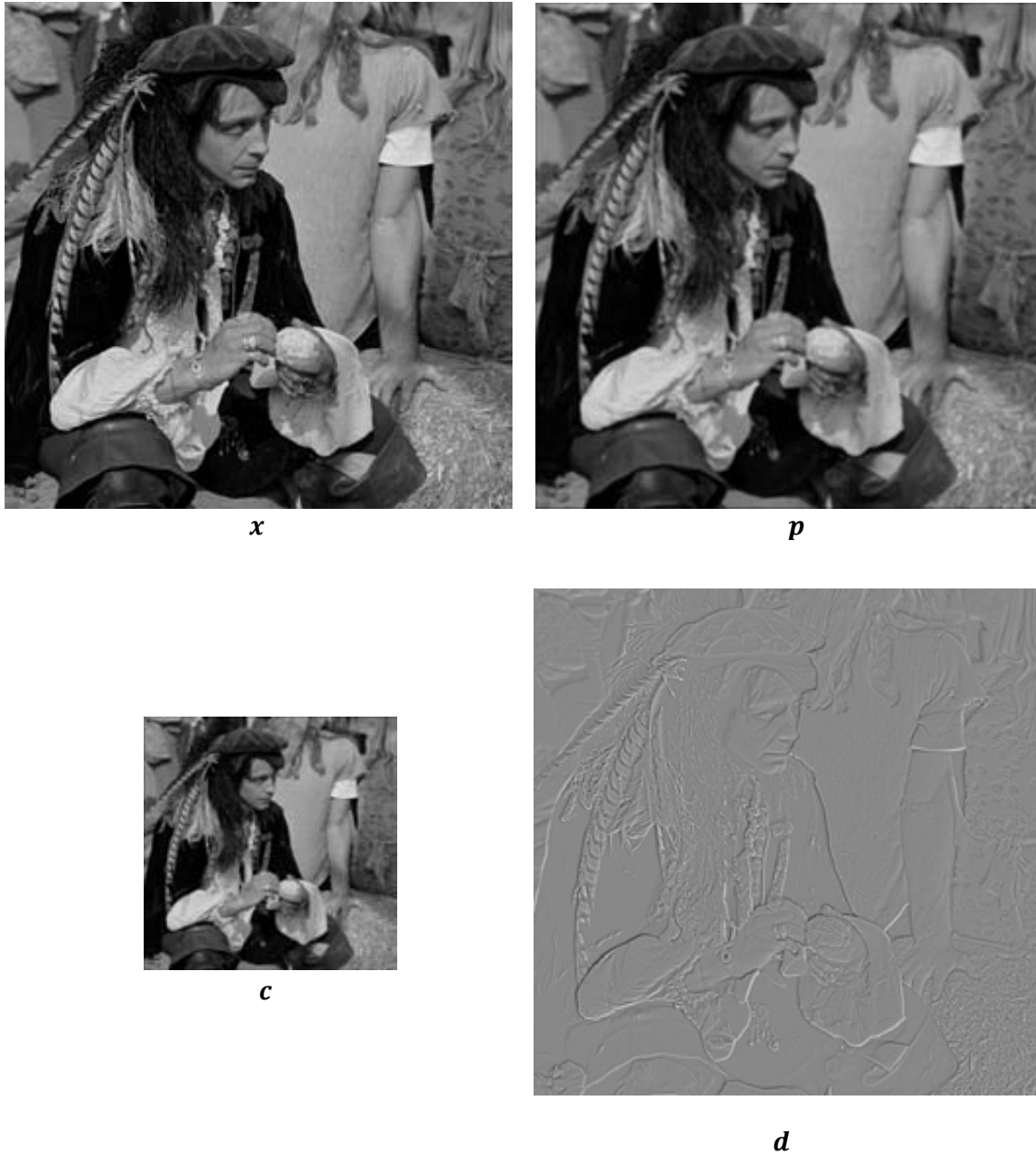


Figura 3.13 Se obtiene la aproximación “ c ” a partir de la imagen original “ x ” filtrada paso bajas por “ H ” y submuestreada por “ M ”. Sobremuestreando “ c ” y luego de un filtro de síntesis se obtiene la predicción “ p ”. La diferencia entre la imagen original “ x ” y la predicción “ p ” nos da los detalles “ d ”.

La Pirámide Laplaciana genera sólo una señal pasobanda en cada nivel de descomposición.

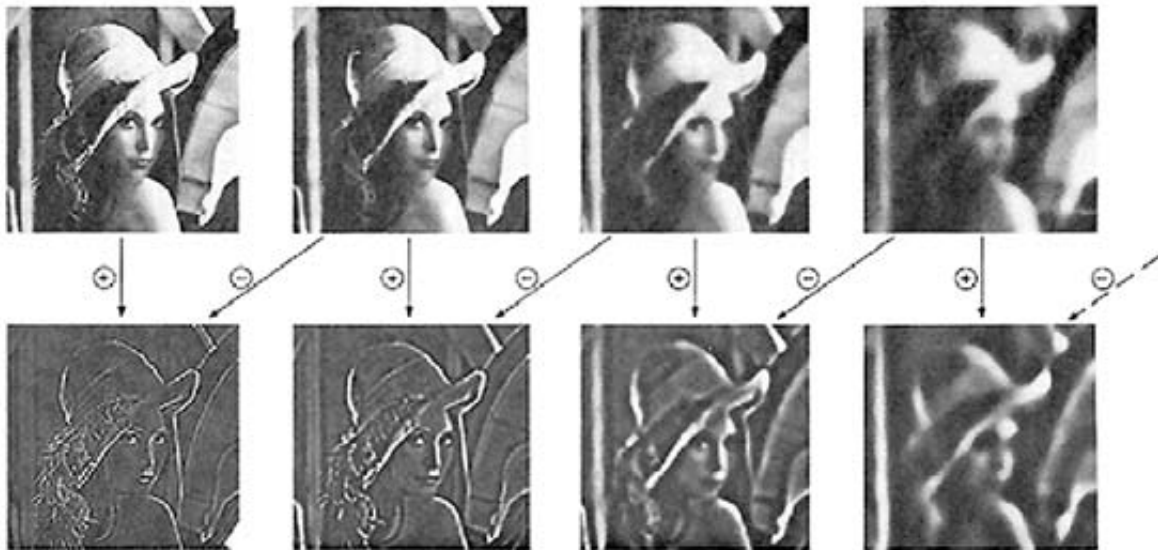


Figura 3.14 Primeros 4 niveles de de la pirámide Gaussiana y Laplaciana [18]. Pirámide Gaussiana columna superior, pirámide Laplaciana columna inferior.

La reconstrucción se realiza a partir de la suma de la aproximación y la versión pasobanda.



Figura 3.15 Síntesis Laplaciana. \hat{x} es la imagen reconstruida [17].

3.4.2 Multiresolución

El análisis realizado con la pirámide Laplaciana se trata de un sistema multiescala. El filtro paso bajas G en cada nivel de la pirámide Laplaciana corresponde con la función de escalamiento $\varphi(t) \in L^2$. La función $\varphi(t)$ puede ser escalada por el parámetro j y “recorrida” por el parámetro n [17].

$$\varphi_{j,n} = 2^{-j}\varphi(2^{-j}t - n) \tag{3.10}$$

donde: $j \in \mathbb{Z}, n \in \mathbb{Z}^2$.

Sea el conjunto de funciones $\{\varphi_{j,n}\}$ una base ortonormal del subespacio V_j . El análisis multiresolución está representado por una secuencia de subespacios vectoriales anidados $\{V_j\}_{j \in \mathbb{Z}}$ cuya intersección es el conjunto vacío y unión es L^2 .

$$\dots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \dots \quad (3.11)$$

$$\bigcup_{j \in \mathbb{Z}} V_j = L^2$$

$$\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$$

Es decir, cada espacio vectorial V_j está contenido dentro de un espacio vectorial de mayor resolución V_{j-n} , lo cual implica que cualquier vector dentro del espacio V_j se puede representar como la combinación lineal de los vectores de un espacio V_{j-n} de resolución superior.

$$\varphi_{0,n} = \varphi(t) = 2 \sum_n g[n] \varphi(2t - n) = \sum_n g[n] \varphi_{-1,n} \quad (3.12)$$

donde: $n \in \mathbb{Z}^2$ es el parámetro de corrimiento y $g[n]$ es el coeficiente asociado a cada n .

Como se muestra en la ecuación 3.12, la función de escalamiento $\varphi_{0,n}$ en el subespacio V_0 corresponde con la suma de versiones ponderadas por $g[n]$ y trasladadas de la función de escalamiento $\varphi_{-1,n} = 2\varphi(2t - n)$ en el subespacio V_{-1} .

Sea W la diferencia entre dos subespacios consecutivos. La unión del subespacio V_j y W_j da como resultado el subespacio de escala $j - 1$:

$$V_{j-1} = V_j \oplus W_j \quad (3.13)$$

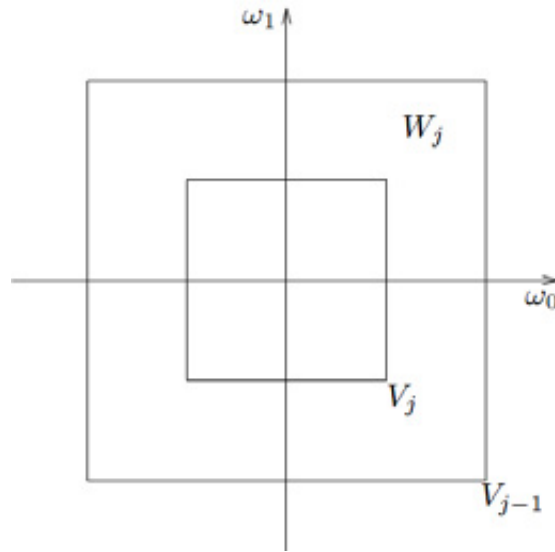


Figura3.16 Subespacios multiescala generados por la pirámide Laplaciana [17].

Por lo tanto, W_j es el complemento ortogonal de V_j . Esto implica que dentro de W_j existen funciones $\psi_{j,n}$ que cubren dicho subespacio diferencia y que se pueden construir a partir de la combinación lineal de funciones de escalamiento de su siguiente subespacio:

$$\psi^{(i)}(t) = 2 \sum_n f_i[n] \varphi(2t - n) \quad (3.14)$$

$$\psi_{j,n}^{(i)} = 2^{-j} \psi^{(i)}(2^{-j}t - n) \quad (3.15)$$

donde: $j \in \mathbb{Z}, n \in \mathbb{Z}^2$.

Sin embargo en este caso los coeficientes $f_i[n]$ por ser el complemento de $g[n]$ corresponden a un filtro paso altas, cada uno asociado a cada función $\psi^{(i)}(t)$.

El desarrollo anterior se relaciona con la pirámide Laplaciana de la siguiente manera. Supóngase que f es una función en el subespacio vectorial V_{j-1} representada a partir de la combinación lineal de las funciones base de V_{j-1} :

$$f(t) = \sum_n c^{(j-1)}[n] \varphi_{j-1,n}(t) \quad (3.16)$$

donde $c^{(j-1)}[n]$ son los coeficientes.

Dado que las funciones de escalamiento $\varphi_{j-1,n}$ son funciones ortogonales base de V_{j-1} los coeficientes se calcula a partir de:

$$c^{(j-1)}[n] = \langle f, \varphi_{j-1,n} \rangle \quad (3.17)$$

El producto interno de f en escala $j - 1$ con funciones base de la siguiente escala j está dado por:

$$c^{(j)}[n] = \langle f, \varphi_{j,n} \rangle = \sum_k c^{(j-1)}[k]g[k - Mn] \quad (3.18)$$

$$d_i^{(j)}[n] = \langle f, \psi_{j,n}^{(i)} \rangle = \sum_k c^{(j-1)}[k]f_i[k - Mn] \quad (3.19)$$

donde $k \in \mathbb{Z}^2$ y M es la matriz diagonal de submuestreo.

Se observa que las ecuaciones 3.18 y 3.19 corresponden a un filtrado multitasa pues la secuencia de entrada $c^{(j-1)}$ se convoluciona con el filtro paso bajas para obtener $c^{(j)}[n]$ y con el filtro paso altas para obtener $d^{(j)}[n]$. El cambio de escala se debió al submuestro M implícito dentro del proceso de filtrado. Por lo tanto, la salida de la pirámide Laplaciana para la secuencia de entrada $c^{(j-1)}$ es $\{c^{(j)}[n], d^{(j)}[n]\}$.

Lo anterior aplicado al análisis de imágenes mediante la pirámide Laplaciana implica que una imagen de tamaño $N \times N$ de escala $j - 1$ introducida en las ecuaciones 3.18 y 3.19 nos dará como resultado dos imágenes c y d de tamaño $\frac{N}{2} \times \frac{N}{2}$ y escala j , siendo c la versión pasobajas y d los detalles, es decir la versión paso altas. Este es el análisis multiresolución que se realiza con la pirámide Laplaciana.

3.4.3 Banco de filtros direccionales

El análisis direccional que realiza la transformada *contourlet* se efectúa mediante un banco de filtros direccionales DFB (*Directional Filter Bank*), proceso que se puede iterar para la descomposición en diferentes niveles.

En sistemas multidimensionales multitasas las operaciones de muestreo se definen sobre matrices. En el caso del banco de filtros direccionales que implementa la transformada *contourlet* se tratan de matrices *quincunx*. Reciben ese nombre por asemejar una estructura geométrica en forma de diamante con cuatro aristas y un quinto punto central. Dichas matrices son utilizadas para realizar un mapeo lineal de las coordenadas de cada pixel que compone la imagen a una nueva posición dando como resultado la rotación de la misma. En la ecuación 3.20 se muestra la forma en que se realizan estas transformaciones lineales. También son conocidas como operaciones de remuestreo pues es posible realizar el submuestreo o sobremuestreo de la imagen a partir de dichas matrices.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.20)$$

donde (x, y) son las coordenadas de cada pixel de la imagen antes de la operación lineal y (x', y') las coordenadas luego de aplicar el operador lineal.

Para poder realizar la descomposición direccional se requiere de matrices que efectúen operaciones de rotación.

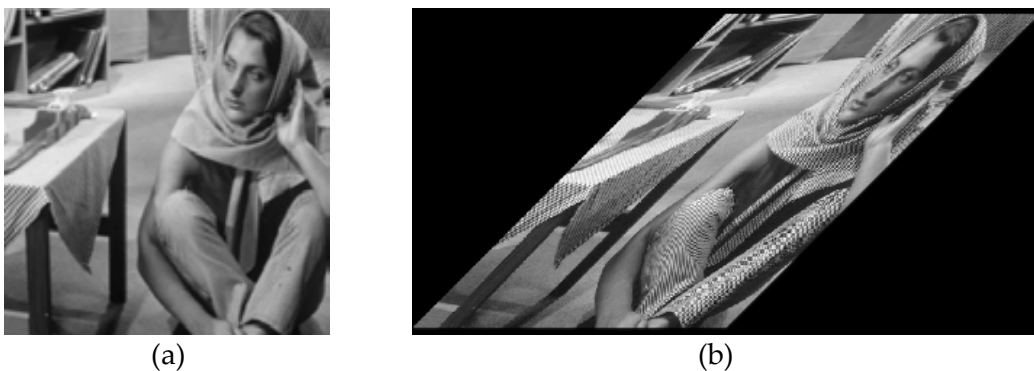


Figura 3.17 Ejemplo de imagen remuestreada. (a) Imagen de "Bárbara". (b) "Bárbara" remuestreada por la matriz $R = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

En realidad la matriz de muestreo *quincunx* Q utilizada en el banco de filtro *quincunx* QFB

(*Quincunx Filter Bank*) puede ser $Q_0 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$ ó $Q_1 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, cada una

correspondiente a la rotación de la imagen de entrada -45° y $+45^\circ$ respectivamente [17], correspondiendo el valor positivo al sentido de giro de las agujas del reloj.

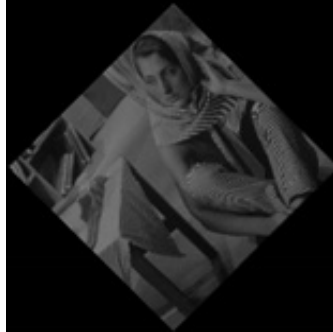


Figura 3.18 Imagen rotada por matriz de muestreo *quincunx* Q_0 .

El banco de filtros *quincunx* se puede utilizar para dividir el espectro en un canal paso altas y otro paso bajas utilizando un par de filtros en forma de diamante o en un canal horizontal y vertical utilizando un par de filtros de abanico, como se muestra en la figura 3.19.

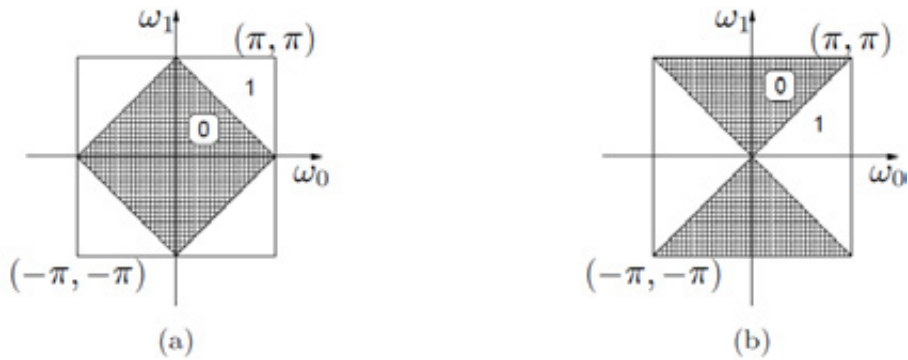


Figura 3.19 Respuesta en frecuencia de (a) filtro en forma de diamante (b) filtro en forma de abanico. [17]

El banco de filtros direccionales propuesto por Minh Do y Martin Vetterli [17] se basa en el uso de un banco de filtros *quincunx* con filtros de abanico combinados con operaciones de rotación hechas mediante remuestreo logrando como resultado un filtro con respuesta en frecuencia dividida en cuadrantes, como se muestra en la figura 3.20 (b). Combinando este filtro con el filtro en forma de abanico se obtiene la partición de frecuencia mostrada en la figura 3.20 (c).

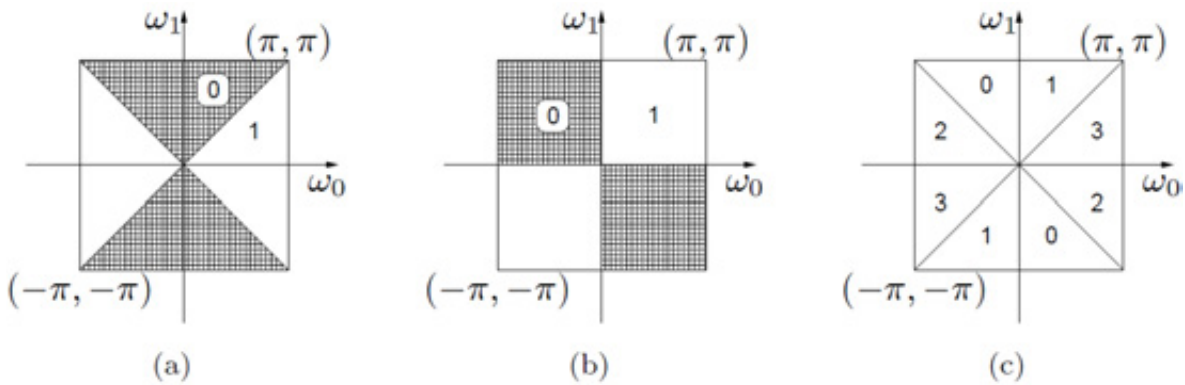


Figura 3.20 (a) Filtro de abanico. (b) Filtro de cuadrantes. (c) cuatro direcciones de descomposición [17].

El resultado es una descomposición direccional en l niveles que genera 2^l subbandas con una partición en frecuencia como la que se muestra en la figura 3.21.

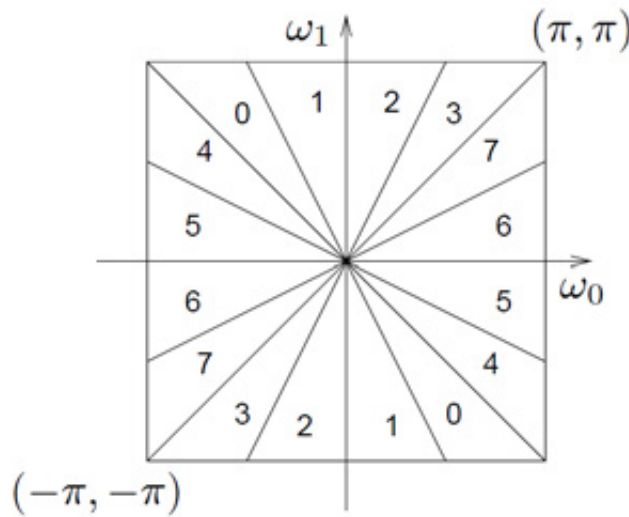


Figura 3.21 Partición direccional en frecuencia para $l=3$ [17].

Las matrices de submuestreo empleadas en la descomposición direccional son matrices diagonales cuyos elementos dependen del nivel de descomposición l .

$$S_k^{(l)} = \begin{cases} \text{diag}(2^{l-1}, 2) & \text{para } 0 \leq k < 2^{l-1} \\ \text{diag}(2, 2^{l-1}) & \text{para } 2^{l-1} \leq k < 2^l \end{cases} \quad (3.21)$$

Supóngase que el DFB utiliza filtros ortogonales con l niveles. La familia de filtros direccionales $g_k^{(l)}[n]$:

$$\{g_k^{(l)}[m - S_k^{(l)}n]\}_{0 \leq k < 2^{l-1}, n \in \mathbb{Z}^2} \quad (3.22)$$

es una base ortonormal del espacio $l^2(\mathbb{Z}^2)$, donde $k = 0, \dots, 2^{l-1} - 1$ corresponde con las direcciones dentro de $[-45^\circ, +45^\circ]$; $k = 2^{l-1}, \dots, 2^l - 1$ corresponde con las direcciones comprendidas dentro del intervalo $[+45^\circ, +135^\circ]$.

Aplicando el banco de filtros direccionales a los subespacios multiresolución V_j :

$$\theta_{j,k,n}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} g_k^{(l)}[m - S_k^{(l)}n] \varphi_{j,m}(t) \quad (3.23)$$

La familia $\{\theta_{j,k,n}^{(l)}(t)\}_{n \in \mathbb{Z}^2}$ es una base ortonormal del subespacio direccional $V_{j,k}^{(l)}$ para $k = 0, \dots, 2^l - 1$. Estos subespacios son ortogonales y además:

$$V_{j,k}^{(l)} = V_{j,2k}^{(l+1)} \oplus V_{j,2k+1}^{(l+1)} \quad (3.24)$$

$$V_j = \bigoplus_{k=0}^{2^l-1} V_{j,k}^{(l)} \quad (3.25)$$

Es decir, el subespacio multiresolución V_j se divide en los subespacios direccionales $V_{j,k}^{(l)}$.

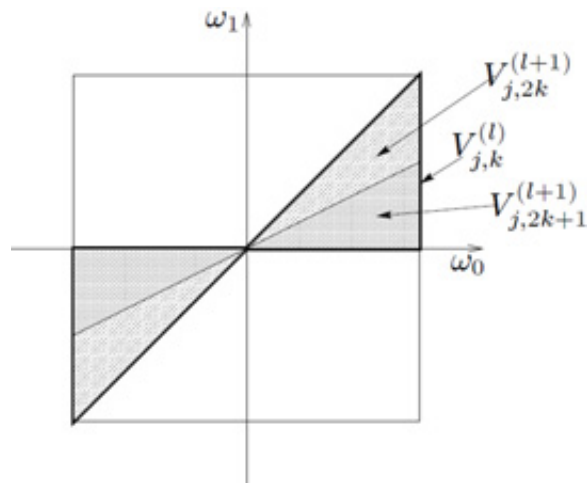


Figura 3.22 Subespacios multidireccionales generados por el banco de filtros direccionales [17].

Aplicando la descomposición de la familia de filtros direccionales $g_k^{(l)}[n]$ al subespacio W_{j+1} definido previamente, sea:

$$\mu_{j,2n+k_i}(t) = \psi_{j+1,n}^{(i)} \quad (3.26)$$

$$\rho_{j,k,n}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} g_k^{(l)}[m - S_k^{(l)}n] \mu_{j,m}(t) \quad (3.27)$$

La familia $\{\rho_{j,k,n}^{(l)}\}_{n \in \mathbb{Z}^2}$ es un marco ajustado del subespacio $W_{j+1,k}^{(l)}$ es decir, es una base de dicho subespacio. Además:

$$W_{j+1,k}^{(l)} = W_{j,2k}^{(l+1)} \oplus W_{j,2k+1}^{(l+1)} \quad (3.28)$$

$$W_{j+1} = \bigoplus_{k=0}^{2^l - 1} W_{j+1,k}^{(l)} \quad (3.29)$$

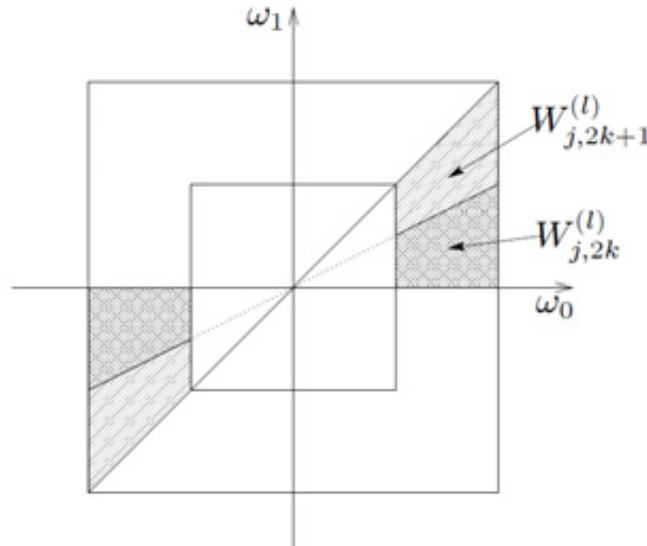


Figura 3.23 Subespacios generados por la descomposición multiescala multidirección [17].

3.4.3 Descomposición Direccional y Multiescala

Realizando el análisis direccional al canal paso banda de cada nivel de la pirámide Laplaciana se obtiene la descomposición direccional piramidal a través de bancos de filtros

direccionales piramidales PDFB (*Piramidal Directional Filter Bank*). Se realiza el análisis direccional al canal paso banda por ser este el que contiene las componentes de alta frecuencia que representan direccionalidad en las imágenes. El resultado es una descomposición direccional en múltiples escalas.

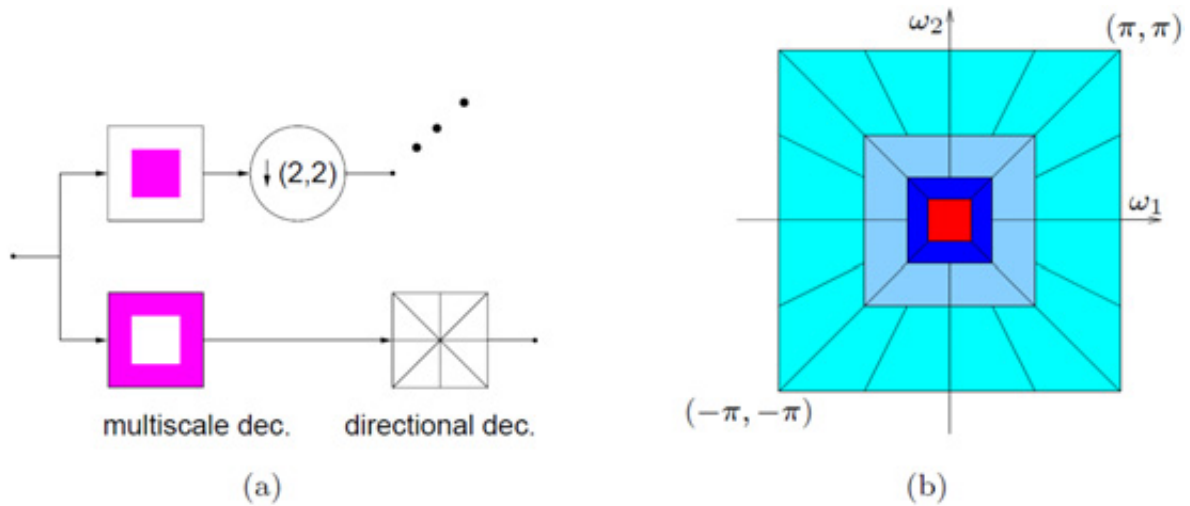


Figura 3.24 Descomposición piramidal direccional. (a) Se realiza la descomposición piramidal; la descomposición direccional es aplicada solamente al canal paso altas. (b) División en frecuencia resultante: el número de direcciones aumenta con la frecuencia.

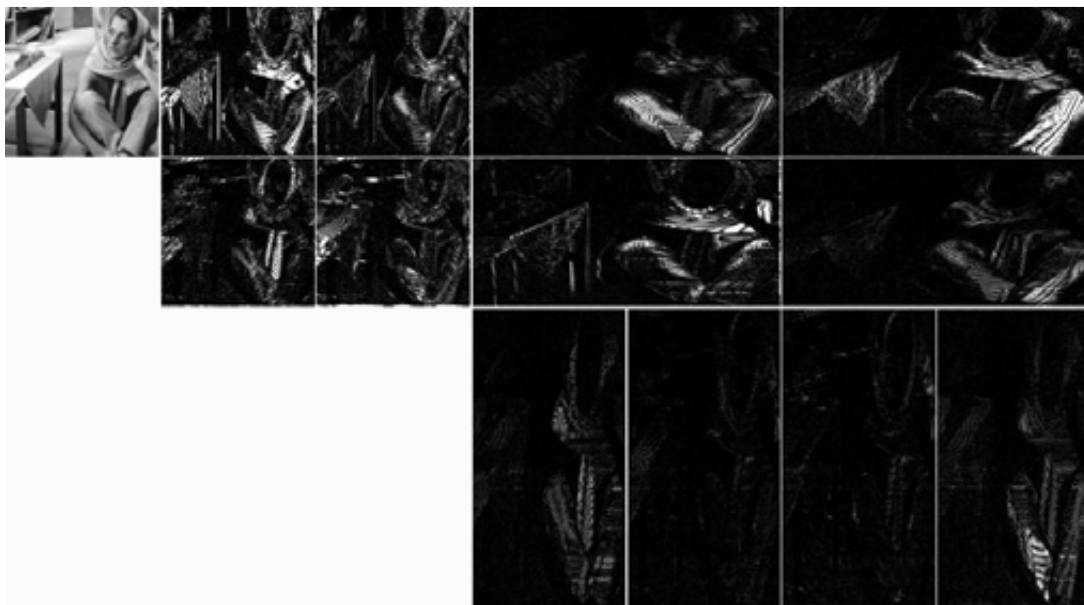


Figura 3.25 Descomposición *contourlet* de la imagen "Bárbara" con 2 niveles de descomposición. 4 direcciones en el primer nivel y 8 direcciones en el segundo nivel.

Como se observa en la figura 3.25 la transformada *contourlet* realiza una descomposición multiresolución y direccional. Los coeficientes de mayor valor son los que se muestran en blanco. La transformada *contourlet* captura las componentes de alta frecuencia en diferentes direcciones y permite identificar texturas presentes en la imagen por lo que es adecuada para capturar contornos.

3.5 Espectro disperso

Basados en [18] y [1], adoptamos una técnica aditiva de espectro disperso para la inserción de la marca de agua en el dominio *contourlet*. Cada bit de la marca de agua codificada (ver 2.10.4) es multiplicado por un código binario pseudoaleatorio arreglado en forma matricial y añadido en alguna de las bandas resultantes a partir del análisis *contourlet*.

$$Y' = Y + \gamma \sum_{i=1}^h w_i P_i \quad (3.30)$$

donde $\bar{w} = (w_1, w_2, \dots, w_h)$ es la marca de agua codificada de tamaño h , nótese que los “0s” se han cambiado por -1, $w_i \in \{-1, 1\}$, $i = \{1, 2, \dots, h\}$, Y es una matriz de tamaño $m \times n$ cuyos elementos contienen los coeficientes *contourlet* de la banda en donde se incrusta la marca de agua, Y' es la matriz de tamaño $n \times m$ con los coeficientes de la banda marcada, $\{P_1, P_2, \dots, P_h\}$ es el conjunto de matrices (idealmente no correlacionadas entre sí) que contienen los “chips” de las secuencias pseudoaleatorias binarias, cada matriz es de tamaño $n \times m$, es decir, del tamaño de la banda en donde se incrusta la marca de agua y γ es un parámetro de escala que controla la intensidad de la marca de agua. Las secuencias pseudoaleatorias son creadas a partir de una clave que sirve como semilla para generarlas, por lo que en la recuperación se requiere de la clave para poder generar las secuencias correctas para su detección. De esta forma se logra expandir cada bit de la marca de agua codificada en toda la banda donde se desea incrustarla. Finalmente se obtiene la imagen marcada a partir de la síntesis *contourlet*.

La técnica de espectro disperso es utilizada por brindar a la marca de agua una mayor robustez, pues al esparcir la señal en toda una banda de la descomposición *contourlet* el ruido tendría que estar presente en toda la banda para poder producir bits en error en la detección. Sin embargo, no sólo el ruido producido por ataques afecta a la marca de agua, también la misma banda *contourlet* interfiere con la marca de agua aditiva. Existe una técnica llamada *espectro disperso mejorado* que reduce la interferencia entre la banda y la marca de agua, como se describirá más adelante. Explotando la descomposición multiescala y multidirección de la transformada *contourlet* junto con el conocimiento del sistema visual humano, como se verá en el siguiente capítulo, se puede conseguir mayor robustez y menor perceptibilidad de la marca de agua.

Capítulo 4

Recuperación de la marca de agua

4.1 *Introducción*

La marca de agua incrustada en las bandas *contourlet* de la imagen derivadas de la descomposición multiescala y multiresolución mediante la técnica de espectro disperso debe ser recuperada luego de que la imagen ha sufrido posibles ataques.

Desde el punto de vista de un sistema de transmisión de información, la recuperación de la marca de agua consiste en la detección de señales binarias en canales con ruido aditivo Gaussiano. Lo anterior se asume considerando el teorema del límite central y la diversidad de ataques independientes a los cuales se puede someter la imagen marcada.

4.2 *Recuperación de la marca de agua*

Sea $R = Y' + N$, la banda Y' de la imagen marcada luego de haber sufrido algún ataque. Realizando el producto interno de R con cada uno de los códigos pseudoaleatorios conocidos a priori $\langle R, P_i \rangle$, para cada uno de los bits incrustados se obtiene:

$$r_i = \frac{\langle R, P_i \rangle}{\langle P_i, P_i \rangle} \quad (4.1)$$

donde r_i es la i -ésima salida del correlador.

Es decir, la imagen marcada y atacada se descompone en sus coeficientes *contourlet* y se realiza la correlación entre los bits de la banda marcada y el código pseudoaleatorio correspondiente al i -ésimo bit de la marca de agua codificada. Debido a la ortogonalidad entre los códigos, al hacer la correlación el resto de los bits esparcidos por los demás códigos son rechazados.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dado que la correlación es una operación lineal y asumiendo que la distribución de los coeficientes *contourlet* sigue una distribución normal con media nula, a la salida del correlador el ruido es una variable aleatoria independiente Gaussiana con media cero [8]. Por lo tanto, la salida del correlador es una variable aleatoria Gaussiana continua cuya media es +1 ó -1 dependiendo de si se incrustó un “uno” binario o un “cero” binario respectivamente y se puede reescribir de la siguiente forma para cada bit recuperado i :

$$r_i = w_{i,s} + n_i \quad (4.2)$$

donde $s = \{1, -1\}$ $i = \{1, 2, \dots, h\}$, h es el número de bits que conforman la marca de agua codificada y n es el ruido.

Para cada bit recuperado, sea la hipótesis H_1 : se incrustó un “+1” y la hipótesis H_2 : se incrustó un “-1” correspondiente a la incrustación de un símbolo s_1 “uno binario” y un símbolo s_2 “cero binario” respectivamente. Para la variable aleatoria r , sean las probabilidades condicionales $p(r/s_1)$ y $p(r/s_2)$ cuyas funciones de densidad de probabilidad se muestran en la figura 4.1 y están dadas por:

$$p(r/s_1) = \frac{1}{\sigma_r \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{r-a_1}{\sigma_r} \right)^2} \quad (4.3)$$

$$p(r/s_2) = \frac{1}{\sigma_r \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{r-a_2}{\sigma_r} \right)^2} \quad (4.4)$$

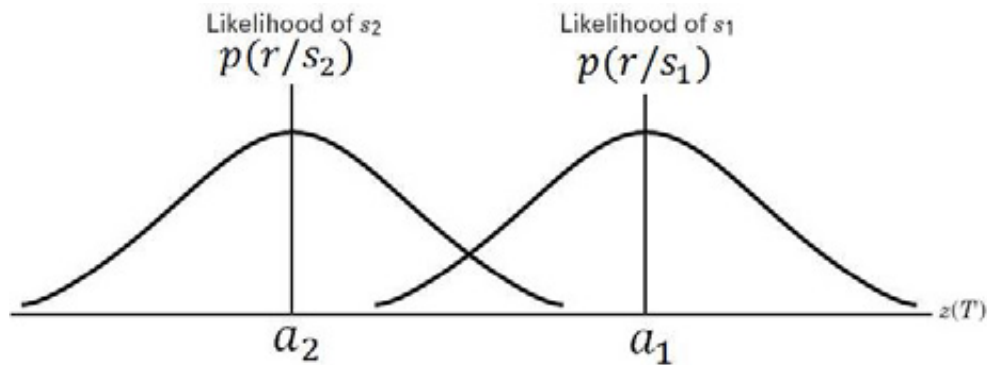


Figura 4.1 Probabilidades condicionales $p(r/s_1)$ y $p(r/s_2)$.

Sean las probabilidades a posteriori $P(s_1/r)$ y $P(s_2/r)$, y el criterio de decisión en el receptor es:

$$P(s_1/r) \underset{H_2}{\overset{H_1}{\geq}} P(s_2/r) \quad (4.5)$$

Es decir, si la probabilidad de que se haya incrustado un s_1 dado que se obtuvo r es mayor a la probabilidad de que se haya incrustado un s_2 luego de obtener r , se escoge la hipótesis H_1 , sino la hipótesis H_2 . Empleado el teorema de Bayes se obtiene la razón de máxima verosimilitud, también conocida como el criterio del mínimo error pues se minimizan las decisiones incorrectas.

$$L(r) = \frac{p(r/s_1)}{p(r/s_2)} \underset{H_2}{\overset{H_1}{\geq}} \frac{P(s_2)}{P(s_1)} \quad (4.6)$$

Sustituyendo $p(r/s_1)$ y $p(r/s_2)$ se obtiene

$$L(r) = \frac{\frac{1}{\sigma_r \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{r-a_1}{\sigma_r}\right)^2}}{\frac{1}{\sigma_r \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{r-a_2}{\sigma_r}\right)^2}} = \frac{e^{-\frac{1}{2} \left(\frac{r-a_1}{\sigma_r}\right)^2}}{e^{-\frac{1}{2} \left(\frac{r-a_2}{\sigma_r}\right)^2}} \underset{H_2}{\overset{H_1}{\geq}} \frac{P(s_2)}{P(s_1)} \quad (4.7)$$

Obteniendo el logaritmo natural de la función $L(r)$:

$$\ln(L(r)) = l(r) \underset{H_2}{\overset{H_1}{\geq}} \ln \frac{P(s_2)}{P(s_1)} \quad (4.8)$$

En el caso equiprobable $P(s_1) = P(s_2) = .5$

$$\ln \frac{P(s_2)}{P(s_1)} = 0 \quad (4.9)$$

Despejando r :

$$r \underset{H_2}{\overset{H_1}{\geq}} \frac{a_1 + a_2}{2} = \gamma_0 \quad (4.10)$$

Es decir, para un valor de r mayor al valor del umbral γ_0 se decide que se incrustó un “uno binario” y para r inferior al valor del umbral se decide que se incrustó un “cero binario”. Dado que la hipótesis H_1 : se incrustó un “ $a_1 = +1$ ” y la hipótesis H_2 : se incrustó un “ $a_2 = -1$ ”, el umbral es $\gamma_0 = 0$; por lo tanto cada bit w estimado se obtiene a partir de:

$$w = \text{sgn}(r) \quad (4.11)$$

Sin embargo, basados en [19], adoptamos una variante de la técnica de incrustación de la marca de agua llamada *espectro disperso mejorado*, como se describe en el subtema 4.6.

4.3 Decodificación de la marca de agua

Luego de la recuperación de los bits incrustados en los coeficientes *contourlet* es necesario realizar la decodificación de la marca de agua previamente codificada utilizando códigos convolucionales.

4.3.1 Decodificador de Viterbi

La decodificación se efectúa mediante el uso del algoritmo de Viterbi [8]. Cada rama del diagrama de *trellis* de decodificación se etiqueta con la distancia de Hamming entre la secuencia recibida y la correspondiente palabra en cada rama del diagrama de *trellis* del codificador. Las palabras de las ramas del diagrama de *trellis* son los símbolos que se esperaba recibir de la salida del codificador como resultado de todas las transiciones de estados. El algoritmo de decodificación utiliza como métrica la distancia de Hamming para buscar el camino de menor distancia acumulada en el enrejado. Al llegar dos ramas al mismo nodo se escoge el trayecto con la menor distancia de Hamming acumulada y en caso de empate se elige al azar alguna de las dos.

Es posible configurar el decodificador de Viterbi en modo de decisión suave y decisión dura. En decisión dura al decodificador se introducen los bits recibidos y se realiza el cálculo de una medida de similitud o distancia entre la señal recibida y todos los caminos

que conducen a todos los nodos en un mismo tiempo. En decisión suave el decodificador se alimenta con una cadena de bits que contiene información de confianza sobre el valor de “uno” o de “cero” de cada bit que se recibe y la métrica que se utiliza es la distancia euclidiana. En la siguiente imagen se ilustra la diferencia entre decisión dura y decisión suave

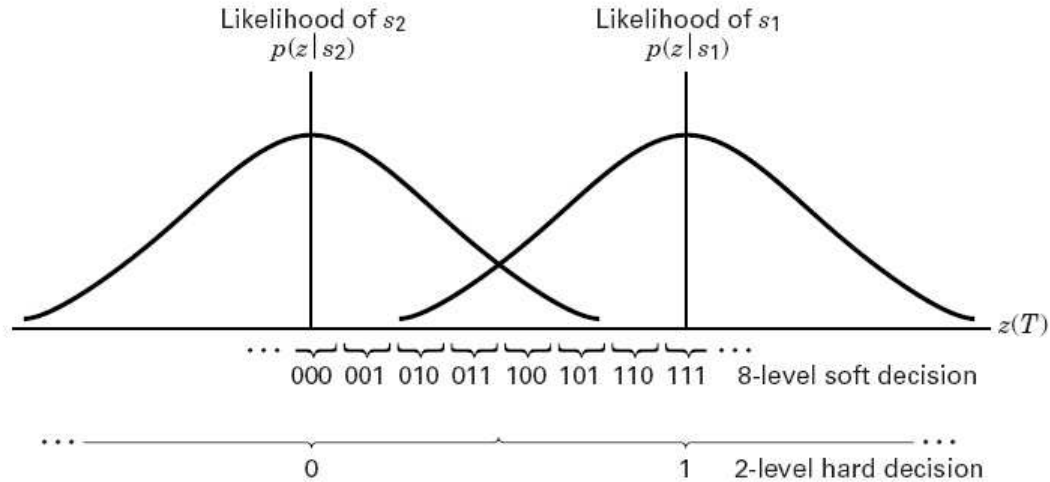


Figura 4.2 Comparación entre decisión dura y decisión suave.

En la práctica el beneficio de realizar la decodificación con decisión suave con 8 niveles de decisión es el incremento en 2dB de la razón señal a ruido a la salida del decodificador [8]. En la sección de resultados se muestra la diferencia entre utilizar decisión suave o decisión dura en la decodificación de la marca de agua y su repercusión en la calidad de la marca de agua recuperada.

4.4 El sistema visual humano

Para sacar provecho de la descomposición multidireccional que brinda la transformada *contourlet* y poder explotarla para mejorar la robustez de la marca de agua, resulta imprescindible conocer las siguientes características principales del sistema visual humano:

1. El sistema visual humano funciona como un filtro pasobajas pues tiene un número limitado de células receptoras de luz en el área de la fovea (área de mayor densidad de fotorreceptores).
2. El brillo aparente no depende de la luminancia absoluta, también depende del brillo en las regiones vecinas [21].
3. De acuerdo con [6], el ojo humano es menos sensible al ruido en bandas cuya orientación es de 45° .
4. El ojo es menos sensible al ruido en áreas donde el brillo en la imagen es muy alto o muy bajo.
5. El sistema visual humano es menos sensible al ruido en áreas de gran textura, pero es más sensible en los bordes de dichas áreas.



Figura 4.3 Ejemplo de la característica número 2 del sistema visual humano. El brillo absoluto de los cuadrados interiores es el mismo.

En este trabajo se explotan las características número 3 y 5. Considerando la característica número 3, las bandas *contourlet* donde se inserta la marca de agua deben contener bandas con direcciones diagonales. En cuanto a la propiedad número 5, las regiones de gran textura corresponden con las altas frecuencias y la descomposición direccional se efectúa sobre la versión pasobanda en la descomposición piramidal Laplaciana. Es decir, utilizando la transformada *contourlet* se puede incrustar la marca de agua en una banda de alta frecuencia.

4.5 Índice estructural de similitud

Para conocer la calidad de la imagen marcada es necesario contar con una medida que nos indique la similitud entre la imagen original y la imagen marcada. Existen diferentes medidas para ello. Una de ellas es la razón señal pico a ruido PSNR *Peak Signal to Noise Ratio*, la cual se obtiene a partir del error cuadrático medio y el valor máximo de intensidad que puede tomar un pixel en la imagen. Usualmente es expresada en decibeles. Para unas imágenes I_1 e I_2 de tamaño $M \times N$:

$$MSE = \sum_{m=1}^M \sum_{n=1}^N \frac{[I_1(m,n) - I_2(m,n)]^2}{M \times N} \quad (4.12)$$

$$PSNR = 10 \log_{10} \left(\frac{I_{max}^2}{MSE} \right) \quad (4.13)$$

donde I_{max} es el máximo valor de intensidad que puede tomar cada pixel de la imagen.

Basados en [7] adoptamos una novedosa medida de medición de las diferencias entre una imagen distorsionada y una imagen de referencia basada en la comparación de la información estructural de las imágenes asumiendo que el sistema visual humano está adaptado para extraer información estructural de una escena [7]. Dicha medida es llamada índice de similitud estructural SSI *Structural Similarity Index* y logra penalizar los errores de acuerdo a su visibilidad.

Concretamente el SSI contempla la medición de la similitud entre dos imágenes considerando tres parámetros: luminancia, contraste y estructura de la imagen. La información estructural de una imagen es la que representa el arreglo de los objetos presentes en la escena y es independiente de la iluminación y contraste.

Sean dos imágenes \mathbf{X} e \mathbf{Y} , considerando que una de ellas es la imagen de referencia, la medida de la luminancia de cada imagen es estimada como el valor medio de la intensidad:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (4.14)$$

La función de luminancia $l(\mathbf{X}, \mathbf{Y})$ depende de μ_x y μ_y y está dada por:

$$l(\mathbf{X}, \mathbf{Y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2\mu_y^2 + C_1} \quad (4.15)$$

El contraste de cada imagen se estima a partir de la desviación estándar pues considera la diferencia entre el nivel de intensidad de cada pixel de la imagen y el nivel de intensidad promedio:

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}} \quad (4.16)$$

La comparación de contraste de las imágenes $c(\mathbf{X}, \mathbf{Y})$ es una función de sus desviaciones estándar y se obtiene por la siguiente expresión:

$$c(\mathbf{X}, \mathbf{Y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2\sigma_y^2 + C_2} \quad (4.17)$$

La función de similitud estructural está dada por:

$$s(\mathbf{X}, \mathbf{Y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (4.18)$$

Las funciones de luminancia contraste y estructura son combinadas para derivar la medida de similitud $SSI(\mathbf{X}, \mathbf{Y})$ entre dos imágenes:

$$SSI(\mathbf{X}, \mathbf{Y}) = f(l(\mathbf{X}, \mathbf{Y}), c(\mathbf{X}, \mathbf{Y}), s(\mathbf{X}, \mathbf{Y})) \quad (4.19)$$

El índice SSI cumple con las siguientes condiciones:

$$SSI(\mathbf{X}, \mathbf{Y}) = SSI(\mathbf{Y}, \mathbf{X})$$

$$SSI(\mathbf{X}, \mathbf{Y}) \leq 1$$

$$SSI(\mathbf{X}, \mathbf{Y}) = 1 \text{ si } \mathbf{X} = \mathbf{Y}$$

El índice de similitud estructural total se calcula a partir del producto de las 3 funciones cuya contribución al índice estructural total se controla con los parámetros α , β y γ .

$$SSI(\mathbf{X}, \mathbf{Y}) = [l(\mathbf{X}, \mathbf{Y})]^\alpha \cdot [c(\mathbf{X}, \mathbf{Y})]^\beta \cdot [s(\mathbf{X}, \mathbf{Y})]^\gamma \quad (4.20)$$

Los parámetros C_1 , C_2 y C_3 que se introducen al cociente de las funciones l , c y s , respectivamente, tienen el propósito de dar estabilidad cuando el denominador es cercano a cero, siendo $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$, L es el rango dinámico de cada pixel de la imagen ($L = 255$ representando cada pixel con 8 bits), $K_1 \ll 1$, $K_2 \ll 1$ y $C_3 = \frac{C_2}{2}$. De [7] adoptamos los valores $\alpha = \beta = \gamma = 1$, $K_1 = .01$ y $K_2 = .03$, obteniendo la siguiente expresión para el cálculo del SSI:

$$SSI(\mathbf{X}, \mathbf{Y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4.21)$$

Para la medición de la calidad de la imagen marcada es mejor calcular el índice SSI localmente en vez de calcularlo en toda la imagen porque los parámetros estadísticos de la imagen y las distorsiones presentes en la misma varían en el espacio. Basados en [7], realizamos el cómputo del SSI sobre ventanas de tamaño 11×11 recorridas pixel por pixel en toda la imagen. Para dar mayor peso al valor central de cada ventana se multiplica por una máscara Gaussiana w de desviación estándar 1.5. Así se tiene que:

$$\mu_x = \sum_{i=1}^N w_i x_i \quad (4.22)$$

$$\sigma_x = \left(\sum_{i=1}^N w_i (x_i - \mu_x)^2 \right)^{\frac{1}{2}} \quad (4.23)$$

$$\sigma_{xy} = \sum_{i=1}^N w_i (x_i - \mu_x)(y_i - \mu_y) \quad (4.24)$$

El índice MSSI *Mean Structural Similarity Index* total de la imagen es el promedio de los índices calculados de manera local:

$$SSIM = \frac{1}{M} \sum_{j=1}^M SSI_j \quad (4.25)$$

En este trabajo se realizó la implementación del algoritmo que realiza el cálculo del SSIM y se utilizó tanto la PSNR como el SSIM para medir la calidad de la imagen marcada con respecto a la imagen original.

4.6 Algoritmo de inserción y recuperación de la marca de agua

Basados en los algoritmos de espectro disperso simple y de espectro disperso mejorado en el dominio *contourlet* publicados en [20], implementamos el algoritmo de espectro disperso mejorado utilizando la codificación y decodificación mediante código convolucional de la marca de agua binaria. Enseguida se describen los algoritmos de inserción y recuperación de marca de agua utilizados en esta tesis, los cuales pueden tener variantes menores para realizar comparaciones al realizar las pruebas de los mismos. Dichas variantes son la selección de diferentes bandas *contourlet* o la configuración del decodificador en modo de decisión suave o decisión dura.

4.6.1 Algoritmo 1

4.6.1.1 Inserción de la marca de agua

La marca de agua binaria $\bar{b} = (b_1, b_2, \dots, b_n)$ es codificada con el código convolucional de tasa $\frac{1}{3}$ y longitud restringida $K = 5$ cuyos polinomios generadores son $g_1 = [1 \ 1 \ 1 \ 1 \ 1]$, $g_2 = [1 \ 1 \ 0 \ 1 \ 1]$ y $g_3 = [1 \ 0 \ 1 \ 0 \ 1]$. A la salida del codificador, mostrado en la figura 2.14, se obtiene la marca de agua binaria codificada $\bar{w} = (w_1, w_2, \dots, w_h)$.

Posteriormente, se realiza la descomposición *contourlet* de la imagen original. Las bandas resultantes se muestran en la figura 4.4.

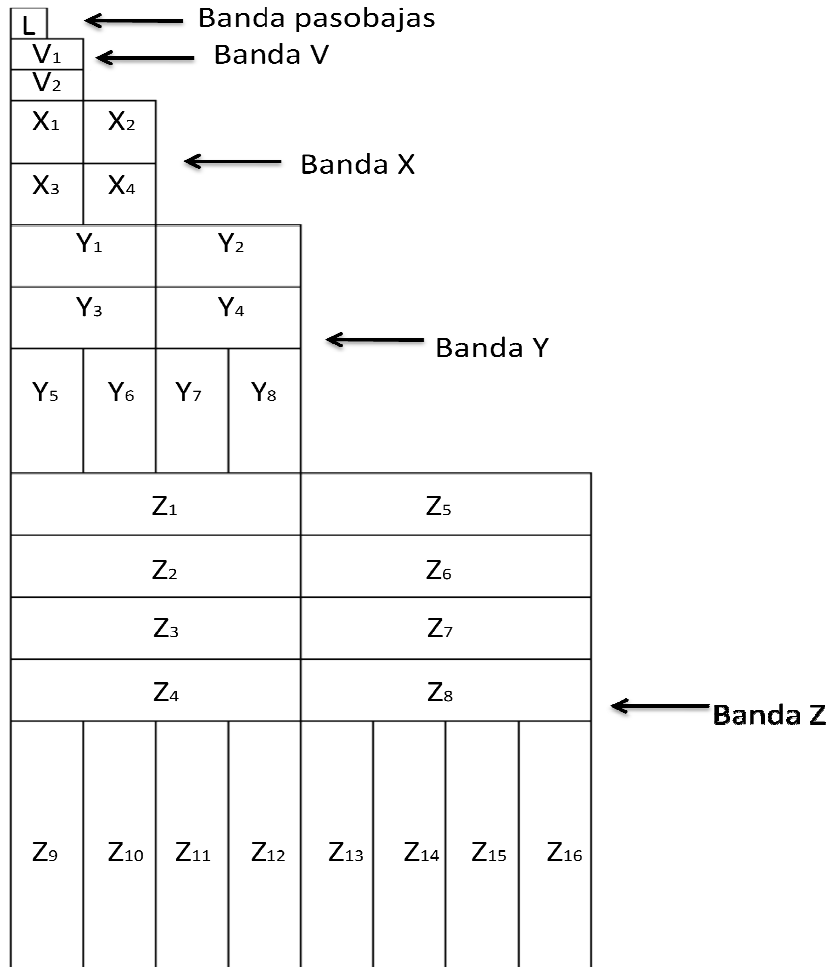


Figura 4.4 Descomposición *contourlet* empleada en el sistema de marcado de agua.

Luego de la descomposición *contourlet* se selecciona la banda en la que se desee incrustar la marca de agua y los coeficientes presentes en ella se arreglan en una matriz del tamaño de dicha banda. En la técnica más simple de espectro disperso descrita en 3.5, cada bit de la marca de agua es modulado por la secuencia pseudoaleatoria e incrustada en los coeficientes de la banda *contourlet* seleccionada. Esto es:

$$Y' = Y + \gamma \sum_{i=1}^h w_i P_i \tag{4.26}$$

donde Y es la banda *contourlet* seleccionada para la inserción de la marca de agua, γ es el parámetro de intensidad de los bits insertados y $\{P_1, P_2, \dots, P_h\}$ es el conjunto de matrices que contienen los “chips” de las secuencias pseudoaleatorias binarias (para más detalles se

puede consultar el subtema 3.5). Una vez esparcida la marca de agua codificada en la banda, se realiza la síntesis *contourlet* para obtener la imagen marcada.

4.6.1.2 Recuperación de la marca de agua

La imagen marcada, luego de sufrir diferentes procesamientos, es descompuesta en los mismos niveles de descomposición *contourlet* en que se descompuso en el proceso de inserción de la marca de agua. Sea $R = Y' + N$ la banda *contourlet* seleccionada luego de que la imagen sufrió posibles ataques. Realizando el producto interno $\langle R, P_i \rangle$ se obtiene un valor estimado del bit de la marca de agua como se muestra en el subtema 4.2.

Los bits recuperados son decodificados mediante el algoritmo de Viterbi y se obtiene la marca de agua recuperada \hat{b} .

4.6.2 Algoritmo 2

4.6.2.1 Inserción de la marca de agua

La marca de agua binaria \bar{b} es codificada con el mismo código convolucional utilizado en el algoritmo 1 obteniendo a la salida del codificador la marca de agua codificada \bar{w} .

Dado que la banda *contourlet* en la que se incrusta la marca de agua es en sí misma una fuente de interferencia que puede inclusive ser mucho más intensa que el ruido introducido por los ataques, basados en [19] utilizamos una técnica llamada espectro disperso mejorado ISS (*Improved Spread Spectrum*) cuyo propósito es incrementar la robustez de la marca de agua reduciendo la interferencia propia de la banda en la que se incrusta. Esto se logra modulando la energía de cada bit de la marca de agua mediante la variación de la amplitud de la secuencia pseudoaleatoria por una función dependiente del bit a incrustar y de la proyección de la banda sobre cada secuencia pseudoaleatoria. Es decir, por cada bit w de la marca de agua codificada \bar{w} que se inserta se calcula:

$$Y' = Y + (\gamma w - \lambda y)P \quad (4.27)$$

donde Y es la banda *contourlet* en la que se incrusta la marca de agua, Y' es la banda con la marca de agua incrustada, γ es un parámetro de intensidad del bit w insertado también llamado factor de multiplicación, $y \triangleq \frac{\langle Y,P \rangle}{\langle P,P \rangle}$ es la proyección de la banda Y sobre la secuencia P , y λ es el factor que controla la remoción de la interferencia introducida por la misma banda Y . La ecuación anterior indica que para un valor óptimo de λ la distorsión de la banda es removida de la proyección de Y sobre P de tal manera que al hacer la correlación en la recuperación, en ausencia de ruido o ataques, la proyección de Y' sobre cada P dé como resultado γw , es decir, el bit w insertado ponderado por γ . Luego de haber insertado la marca de agua en la banda, se obtiene la imagen marcada a partir de la síntesis de los coeficientes *contourlet*.

4.6.2.2 Recuperación de la marca de agua

Asumiendo que los ataques a la imagen introducen ruido AWGN, en el detector se recibe:

$$R = Y' + N = Y + (\gamma w - \lambda y)P + N \quad (4.28)$$

donde R es la banda *contourlet* en el detector. El criterio que se utiliza para saber si se insertó un “uno” o un “cero” es el siguiente:

$$r = \frac{\langle R,P \rangle}{\langle P,P \rangle} = \frac{\langle Y+(\gamma w-\lambda y)P+N,P \rangle}{\langle P,P \rangle} = \gamma w + y(1 - \lambda) + n \quad (4.29)$$

donde, $n = \frac{\langle N,P \rangle}{\langle P,P \rangle}$.

En la ecuación anterior se observa que mientras más cercano a 1 sea λ menor es la distorsión propia de la banda *contourlet* y en el caso de ausencia de ruido $r = \gamma w$, por lo que la recuperación de cada bit en el caso de espectro disperso mejorado se puede realizar a partir de $r = \frac{\langle R,P \rangle}{\langle P,P \rangle}$, es decir, como en el caso de espectro disperso simple.

De acuerdo a [19] es posible obtener un valor $\lambda = \lambda_{\text{optimo}}$ que minimiza la probabilidad de error en la etapa de decisión de la recuperación de la marca de agua. Para ello se procede de la siguiente forma:

La distorsión esperada al insertar un bit de la marca de agua es:

$$\begin{aligned} E[D] &= E[|Y' - Y|] = E[|(\gamma w - \lambda y)P|] = E[(\gamma w - \lambda y)P, (\gamma w - \lambda y)P] \quad (4.30) \\ &= E[(\gamma w - \lambda y)^2 \sigma_P^2] = \left(\gamma^2 + \frac{\lambda^2 \sigma_y^2}{N \sigma_P^2} \right) \sigma_P^2 \end{aligned}$$

Donde N es el número de elementos de P , que es igual al número de coeficientes de la banda *contourlet*. Haciendo que la distorsión sea igual a la introducida por el método de espectro disperso simple, se fuerza que $E[D] = \sigma_P^2$

$$\left(\gamma^2 + \frac{\lambda^2 \sigma_y^2}{N \sigma_P^2} \right) \sigma_P^2 = \sigma_P^2, \therefore \gamma = \frac{N \sigma_P^2 - \lambda^2 \sigma_y^2}{N \sigma_P^2} \quad (4.31)$$

En el detector sea la hipótesis H_1 : se incrustó un “+1” y la hipótesis H_2 : se incrustó un “-1” correspondiente a la inserción de un símbolo s_1 y un símbolo s_2 , respectivamente. Para la variable aleatoria r , sean las probabilidades condicionales $p(r/s_1)$ y $p(r/s_2)$ cuyas funciones de densidad de probabilidad se muestran en la figura 4.1 y están dadas por las ecuaciones (4.3) y (4.4).

La probabilidad de cometer un error en la etapa de decisión es:

$$Pe = P(H_2/s_1)P(s_1) + P(H_1/s_2)P(s_2) \quad (4.32)$$

Asumiendo que se tiene el caso equiprobable: $P(s_1) = P(s_2) = .5$

$$Pe = P(H_2/s_1) = P(H_1/s_2) \quad (4.33)$$

$$P(H_2/s_1) = \int_{-\infty}^{\gamma_0} p(r/s_1) dr \quad (4.34)$$

$$P(H_1/s_2) = \int_{\gamma_0}^{\infty} p(r/s_2) dr \quad (4.35)$$

Como $P(s_1) = P(s_2) = .5$, el umbral γ_0 está dado por: $\gamma_0 = \frac{\gamma b_1 + \gamma b_2}{2}$, donde $b_1 = +1$, $b_2 = -1$

$$Pe = \int_{\gamma_0}^{\infty} \frac{1}{\sigma_r \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{r - \gamma b_2}{\sigma_r} \right)^2} dr \quad (4.36)$$

Haciendo el cambio de variable $u = \frac{r-\gamma b_2}{\sigma_r}$, $dr = \sigma_r du$

$$Pe = \int_{u_0 = \frac{\gamma b_1 - \gamma b_2}{2\sigma_r}}^{u=\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2} du = Q(u_0) = Q\left(\frac{\gamma b_1 - \gamma b_2}{2\sigma_r}\right) \quad (4.37)$$

Como $b_1 = +1$, $b_2 = -1$,

$$Pe = Q\left(\frac{\gamma}{\sigma_r}\right) \quad (4.38)$$

donde $\gamma = \frac{N\sigma_P^2 - \lambda^2 \sigma_Y^2}{N\sigma_P^2}$ y $\sigma_r^2 = \frac{\sigma_n^2 + (1-\lambda)^2 \sigma_Y^2}{N\sigma_P^2}$.

Lo que se busca es minimizar la probabilidad de error lo cual ocurre para el valor máximo del argumento de la función $Q(x)$.

Obteniendo el valor de λ que maximiza $\frac{\gamma}{\sigma_r}$, $\frac{\partial(\frac{\gamma(\lambda)}{\sigma_r(\lambda)})}{\partial \lambda} = 0$ se obtiene [19]

$$\lambda_{\text{ópt}} = \frac{1}{2} \left(\left(1 + \frac{\sigma_n^2}{\sigma_Y^2} + \frac{N\sigma_P^2}{\sigma_Y^2} \right) - \sqrt{\left(1 + \frac{\sigma_n^2}{\sigma_Y^2} + \frac{N\sigma_P^2}{\sigma_Y^2} \right)^2 - 4 \frac{N\sigma_P^2}{\sigma_Y^2}} \right) \quad (4.39)$$

$\lambda_{\text{ópt}}$ es el valor de λ que minimiza la probabilidad de cometer un error en la decisión, y que reduce la interferencia de la banda *contourlet* en la que se incrusta la marca de agua.

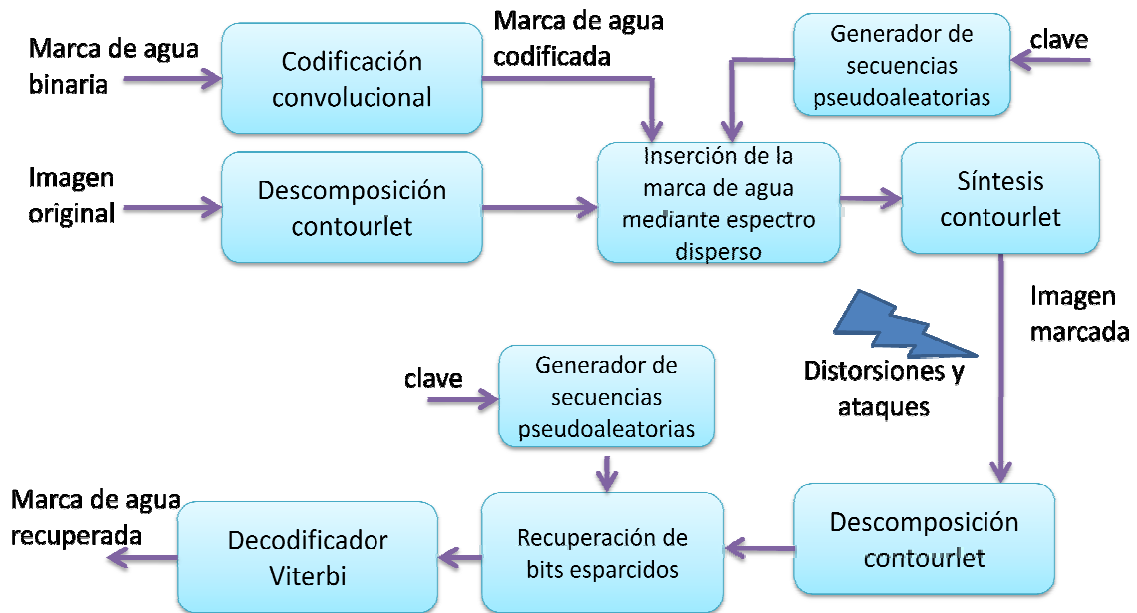


Figura 4.5 Diagrama de bloques del algoritmo de inserción de marca de agua propuesto.

Capítulo 5

Resultados

5.1 *Introducción*

Realizamos la implementación de los algoritmos 1 y 2 de inserción y recuperación de la marca de agua digital en el dominio *contourlet*, definidos en la sección 4.6, utilizando el software Matlab®. Para realizar la prueba de los algoritmos, empleamos las imágenes “Lena” y “Bárbara”, las cuales son frecuentemente utilizadas en la literatura relacionada con el procesamiento digital de imágenes. Probamos con 3 procedimientos diferentes, cada uno de ellos correspondiente a la inserción de la marca de agua en diferentes subbandas de la descomposición *contourlet*. Las bandas se muestran en la figura 4.4 y los procedimientos se enlistan enseguida:

Procedimiento 1: Toda la banda Y. Contiene direcciones diagonales en la imagen y se trata de una banda con componentes de alta frecuencia. (Considera características de percepción ligera del sistema visual humano respecto a alteraciones en direcciones diagonales y no realiza alteraciones en las bajas frecuencias de la imagen.)

Procedimiento 2: Subbandas Z3, Z4, Z7 y Z8. Se incrusta la marca de agua en subbandas de alta frecuencia.

Procedimiento 3: Subbandas Y1, Y4, Y5, Y8, Z9 y Z16. La selección de estas subbandas se efectuó considerando la menor percepción del sistema visual humano respecto a cambios de intensidad en direcciones diagonales de la imagen y a modificaciones en altas frecuencias.

Los resultados mostrados en las secciones 5.2 y 5.3 corresponden a las simulaciones utilizando la técnica de espectro disperso sin codificación convolucional y con codificación convolucional respectivamente, de la marca de agua con el objetivo de compararlas entre



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

sí. Posteriormente utilizando solamente el algoritmo propuesto, es decir, el que implica la codificación de la marca de agua, se prueban los 3 procedimientos ante diferentes tipos de ataques. En algunos de los ejemplos de inserción y recuperación de la marca de agua que se describen en la sección 5.4 se muestra el histograma y entropía de cada imagen. El histograma de una imagen es una gráfica que muestra la frecuencia de ocurrencia de cada nivel de gris presente en la imagen. Normalizando el histograma para que el área bajo la curva sea igual a 1, podemos considerarlo como la función de densidad de probabilidad de la variable aleatoria que representa el nivel de gris de los píxeles de la imagen. Sobre dicha variable aleatoria se puede calcular la entropía de la imagen. La entropía $H(X)$ es una medida de la cantidad de información promedio de la variable aleatoria X . Es el número promedio de bits que se requieren para describir la variable aleatoria, está dada por:

$$H(X) = -\sum_x p(x)\log_2 p(x) \quad (5.1)$$

5.2 *Espectro disperso mejorado sin codificación convolucional*

Utilizando el algoritmo de espectro disperso mejorado sin codificación convolucional se incrustó una marca de agua pseudoaleatoria de 2048 bits de tamaño en la imagen “Bárbara”. Se varió el factor de multiplicación γ para los 3 procedimientos y se obtuvo la PSNR y SSIM entre la imagen marcada y la original. Sin realizar ataque alguno a la imagen marcada se calculó el BER (*Bit Error Rate*) de la marca de agua recuperada, obteniendo las siguientes gráficas:

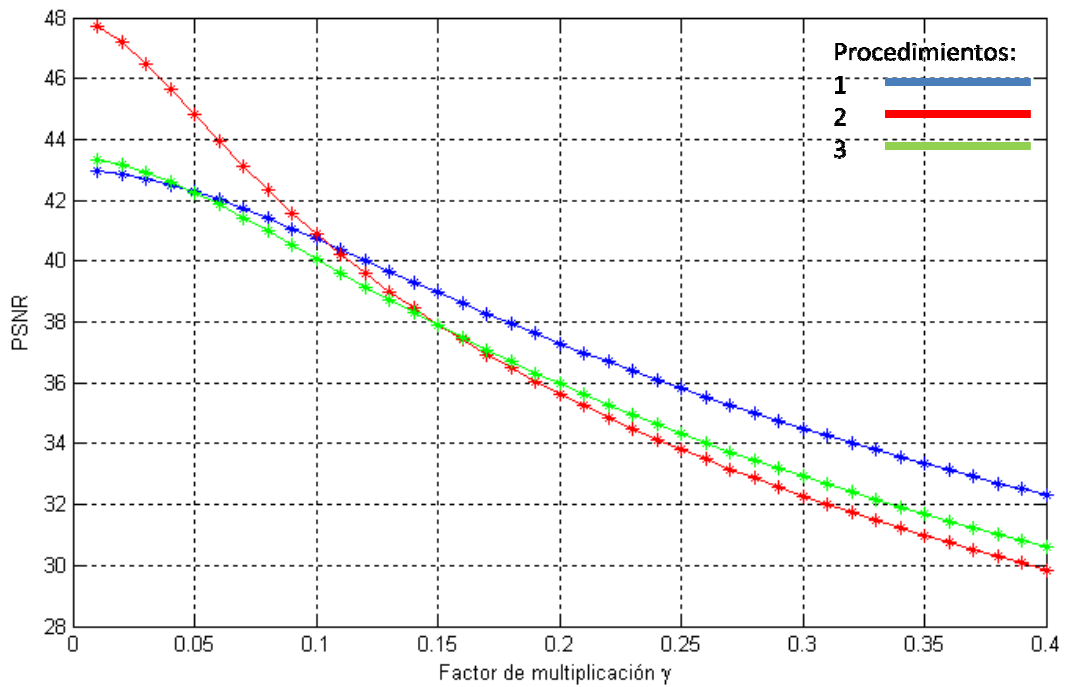


Figura 5.1 PSNR contra factor de multiplicación γ sin utilizar codificación convolucional.

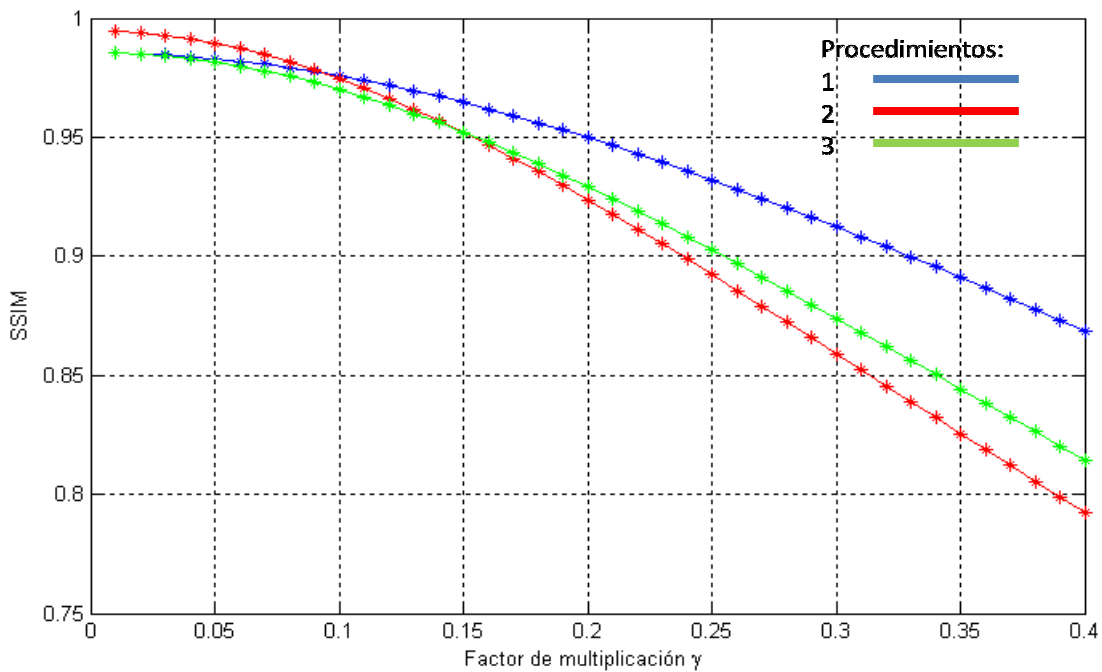


Figura 5.2 SSIM contra factor de multiplicación γ sin utilizar codificación convolucional.

De las figuras 5.1 y 5.2 se observa que para valores pequeños de γ se obtiene una menor alteración perceptual de la imagen marcada. Sin embargo al aumentar la robustez de la marca de agua (correspondiente a incrementar el valor de γ), de los tres procedimientos definidos, aquel que conserva la mayor similitud entre la imagen original y la imagen marcada es el número 1.

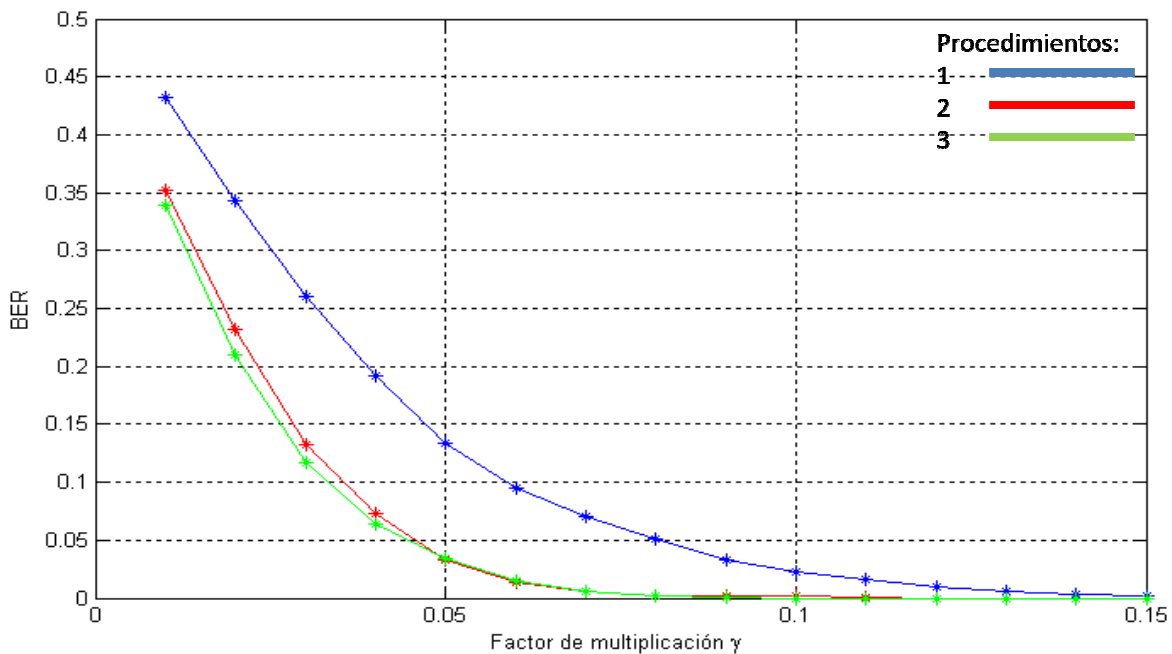


Figura 5.3 Tasa de bits en error en la marca de agua recuperada sin utilizar codificación convolucional.

La figura 5.3 muestra que los procedimientos 2 y 3 tienen una menor tasa de bits en error respecto al procedimiento 1 para factores de multiplicación menores a .15.

5.3 Espectro disperso mejorado con codificación convolucional

Utilizando el algoritmo de espectro disperso mejorado con codificación convolucional y una marca de agua pseudoaleatoria de 2048 bits de tamaño se varió el factor de multiplicación γ para los 3 procedimientos y se obtuvieron la PSNR y SSIM entre la

imagen marcada y la original. Sin realizar ataque alguno a la imagen marcada se calculó el BER de la marca de agua recuperada. La recuperación de la marca de agua se realizó empleando decisión dura y decisión suave con 8 regiones de decisión. Las gráficas obtenidas son las siguientes:

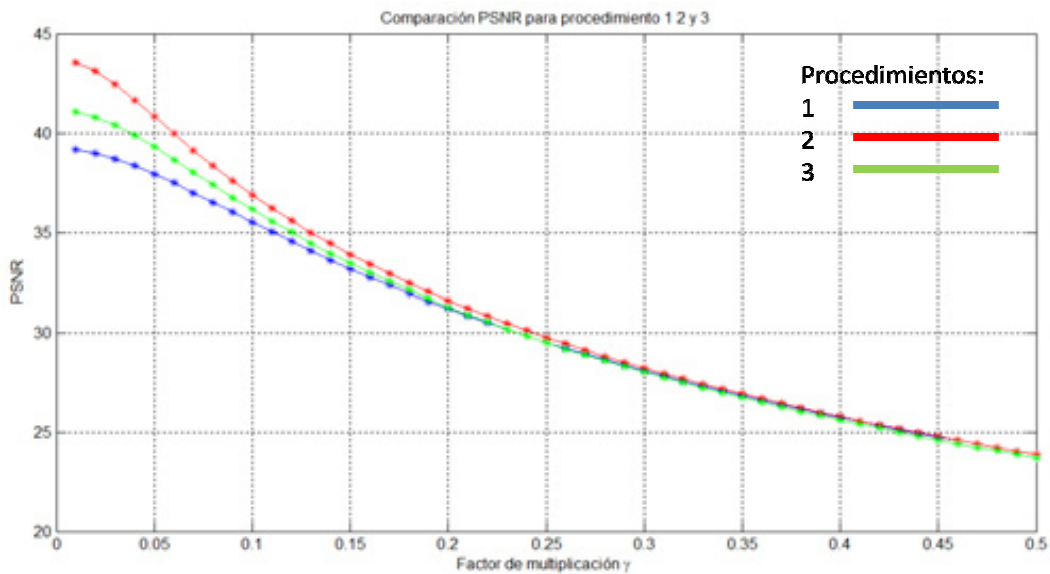


Figura 5.4 PSNR contra factor de multiplicación γ . La marca de agua fue codificada utilizando codificación convolucional previa a su inserción.

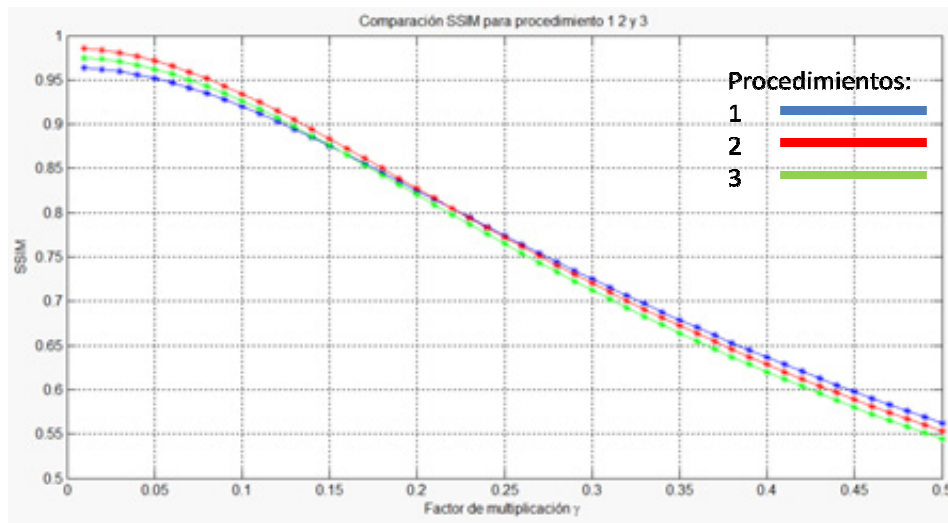


Figura 5.5 SSIM contra factor de multiplicación γ . La marca de agua fue codificada utilizando codificación convolucional previa a su inserción.

Las figuras 5.4 y 5.5 muestran que para valores del factor de multiplicación mayores a .25 se mantiene una mayor calidad en la imagen marcada utilizando el procedimiento 1. Las figuras 5.6 y 5.7 muestran la diferencia en la tasa de bits en error en la marca de agua recuperada al emplear decisión dura y decisión suave en el decodificador de Viterbi.

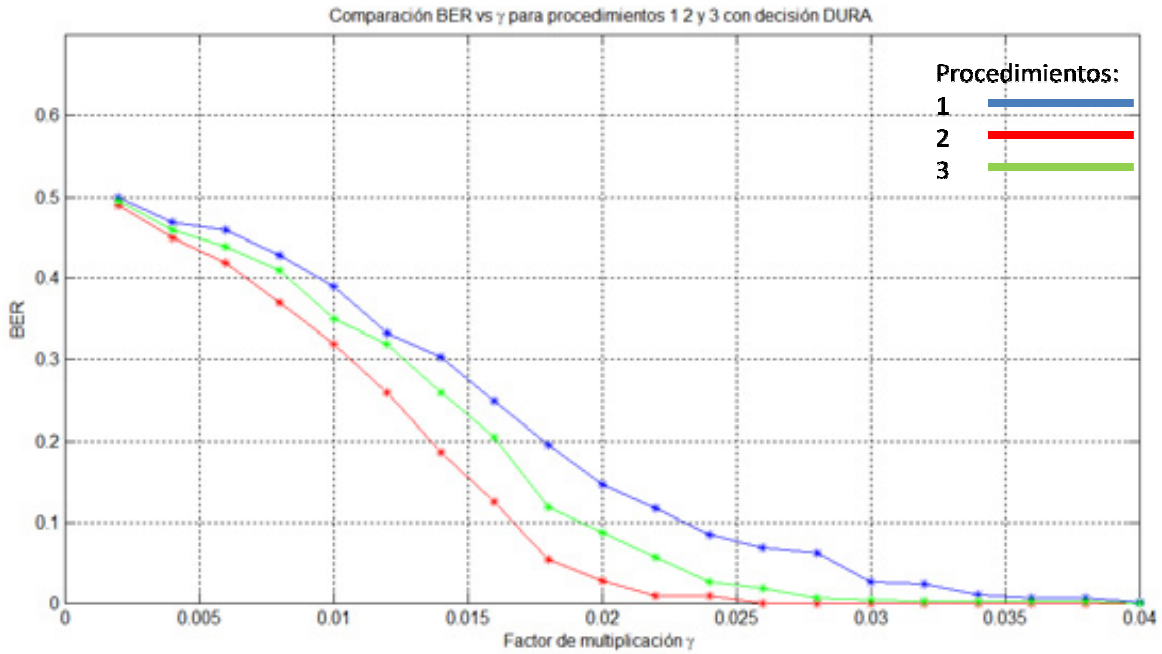


Figura 5.6 Tasa de bits en error en la marca de agua recuperada utilizando decisión dura en el decodificador de Viterbi.

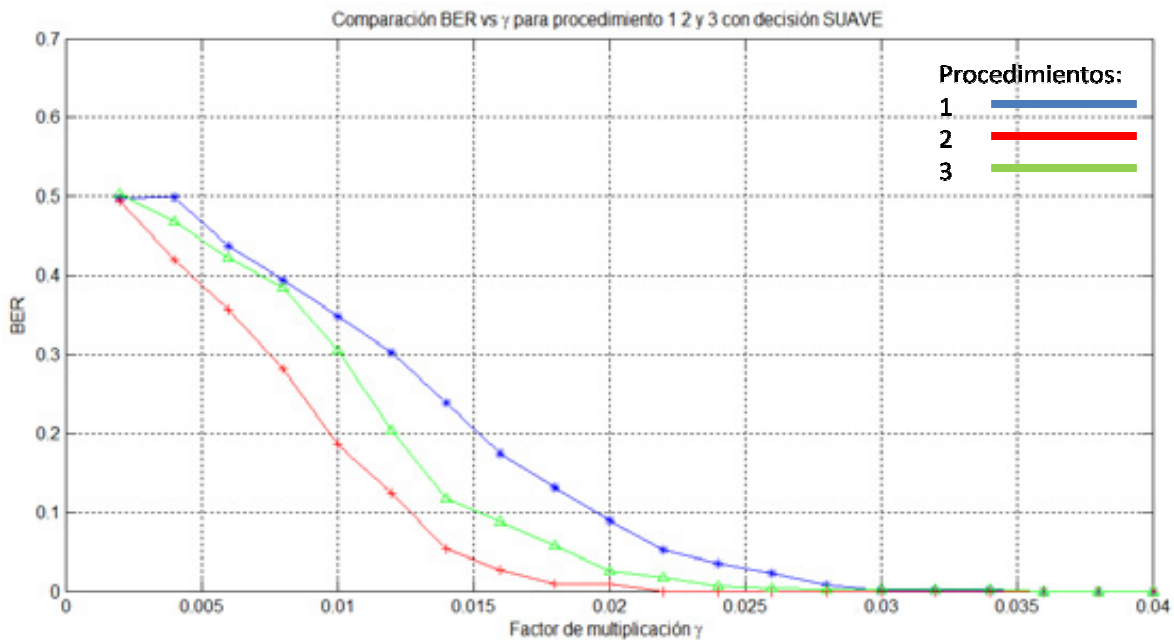


Figura 5.7 Tasa de bits en error en la marca de agua recuperada utilizando decisión suave en el decodificador de Viterbi.

El hecho de que los procedimientos 2 y 3 tengan una menor tasa de bits en error que el procedimiento 1 para un mismo valor del factor de multiplicación tiene relación con lo mostrado por las gráficas 5.4 y 5.5 de PSNR contra γ y SSIM contra γ . Dado que para un mismo valor de gama la alteración perceptual de la imagen marcada es mayor en los procedimientos 2 y 3 que en el procedimiento 1, inversamente se tiene más “fuerza” en la marca de agua incrustada, es por ello que procedimientos 2 y 3 presentan una menor tasa de bits en error.

Comparando las gráficas 5.1 con 5.4 y 5.2 con 5.5 de PSNR y SSIM para el algoritmo de espectro disperso sin codificación convolucional y con codificación convolucional se observa que para el caso de codificación convolucional las curvas de PSNR comienzan 4dB abajo respecto al caso de espectro disperso mejorado sin codificación convolucional. Esto se debe a que el hecho de incrustar una marca de agua codificada con un código convolucional de tasa 1/3 implica incrustar 3 veces más bits en la marca de agua, lo que se traduce en una mayor alteración perceptual de la imagen.

Enseguida se compara la tasa de bits en error en la marca de agua recuperada de tamaño 2048 bits para el caso de espectro disperso mejorado sin codificación convolucional y con codificación convolucional para cada uno de los procedimientos.

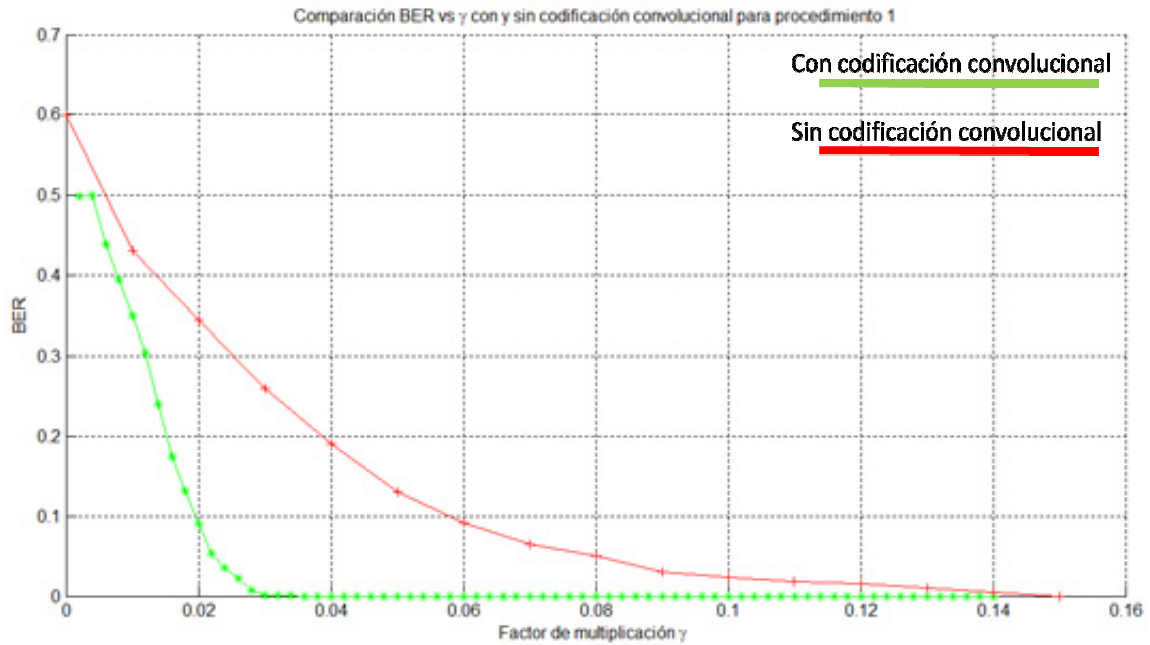


Figura 5.8 Comparación de la tasa de bits en error en la marca de agua recuperada para procedimiento 1.

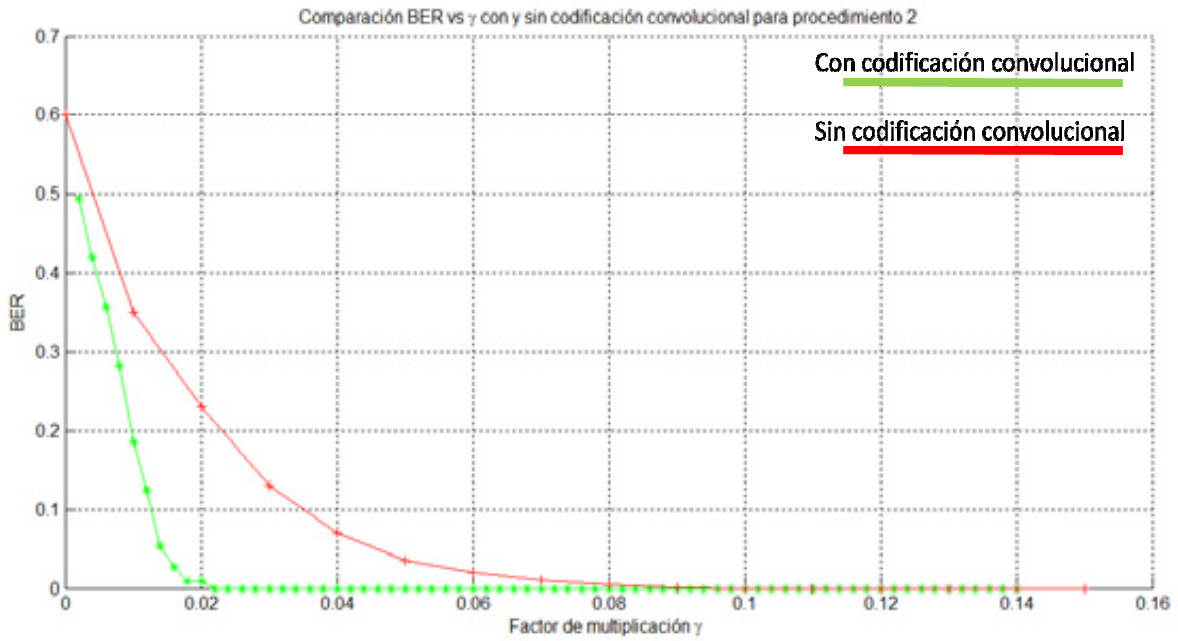


Figura 5.9 Comparación de la tasa de bits en error en la marca de agua recuperada para procedimiento 2.

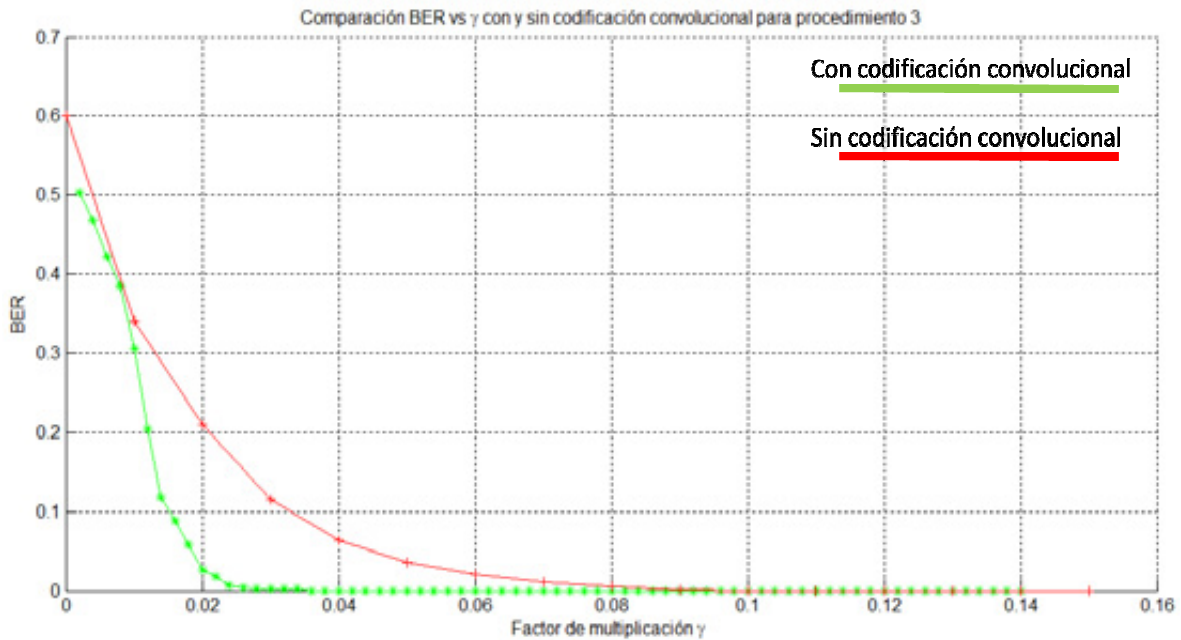


Figura 5.10 Comparación de la tasa de bits en error en la marca de agua recuperada para procedimiento 1.

Las figuras 5.8, 5.9 y 5.10 muestran que para los tres procedimientos el uso de codificación convolucional reduce la tasa de bits en error en la marca de agua recuperada para un mismo valor de γ . Al usar codificación convolucional, para un mismo factor de multiplicación, se tiene una menor tasa de errores en la marca de agua recuperada, respecto al caso donde no se usa la codificación.

Para poder realizar una mejor comparación entre la inserción de la marca de agua con codificación convolucional y sin codificación convolucional es necesario analizar la tasa de bits en error para una misma alteración perceptual de la imagen marcada; es decir para un mismo valor de PSNR o de SSIM, se debe comparar la tasa de bits erróneos en la marca de agua recuperada usando, y sin usar codificación convolucional.

Utilizando la marca de agua “Ingeniería” de tamaño 9×72 pixeles y el procedimiento 1, variamos el valor del factor de multiplicación γ para el caso del algoritmo que utiliza codificación convolucional, así como el que no la utiliza, hasta conseguir un mismo valor de PSNR y SSIM en ambos casos. Posteriormente la imagen fue atacada realizando una compresión JPEG. Los resultados correspondientes son los siguientes:



Figura 5.11 Marca de agua “Ingeniería” de tamaño 9×72 pixeles.



Figura 5.12 Marca de agua recuperada para espectro disperso mejorado sin codificación convolucional. Porcentaje de compresión JPEG=77%, PSNR= 40, SSIM=.9578, BER=.0586, $\gamma=.2946$.



Figura 5.13 Marca de agua recuperada para espectro disperso mejorado utilizando codificación convolucional y decisión dura en el decodificador. Porcentaje de compresión JPEG=77%, PSNR= 40, SSIM=.9577, BER=.0478, $\gamma=.0478$.



Figura 5.14 Marca de agua recuperada para espectro disperso mejorado utilizando codificación convolucional y decisión suave en el decodificador. Porcentaje de compresión JPEG=77%, PSNR= 40, SSIM=.9577, BER=0, γ =.0478.

Comparando la figura 5.12 con la 5.14 se observa la clara ventaja del algoritmo propuesto sobre el algoritmo que no realiza la codificación convolucional para una misma alteración de la imagen marcada.

5.4 *Capacidad del sistema de marcado de agua*

La capacidad del sistema de marcado de agua puede estar limitada por la alteración que realiza la marca de agua en la imagen que la porta. De acuerdo a las pruebas a los ataques descritas en el subtema 5.5, para un valor de $\gamma \in [.1, .2]$ podemos considerar que tenemos una marca de agua robusta. La siguiente gráfica muestra cómo afecta el número de bits que se incrustan en la imagen para 3 valores diferentes del factor de multiplicación γ : $\gamma = .1$, $\gamma = .15$ y $\gamma = .2$.

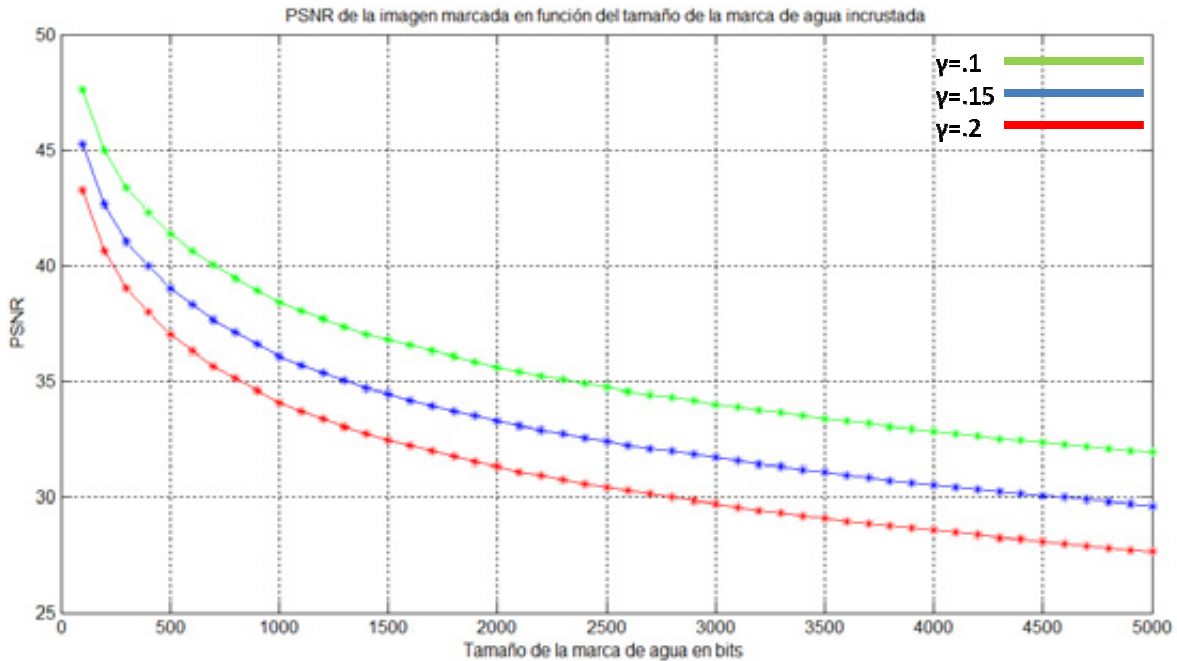


Figura 5.15 PSNR de la imagen marcada contra tamaño de la marca de agua en bits.

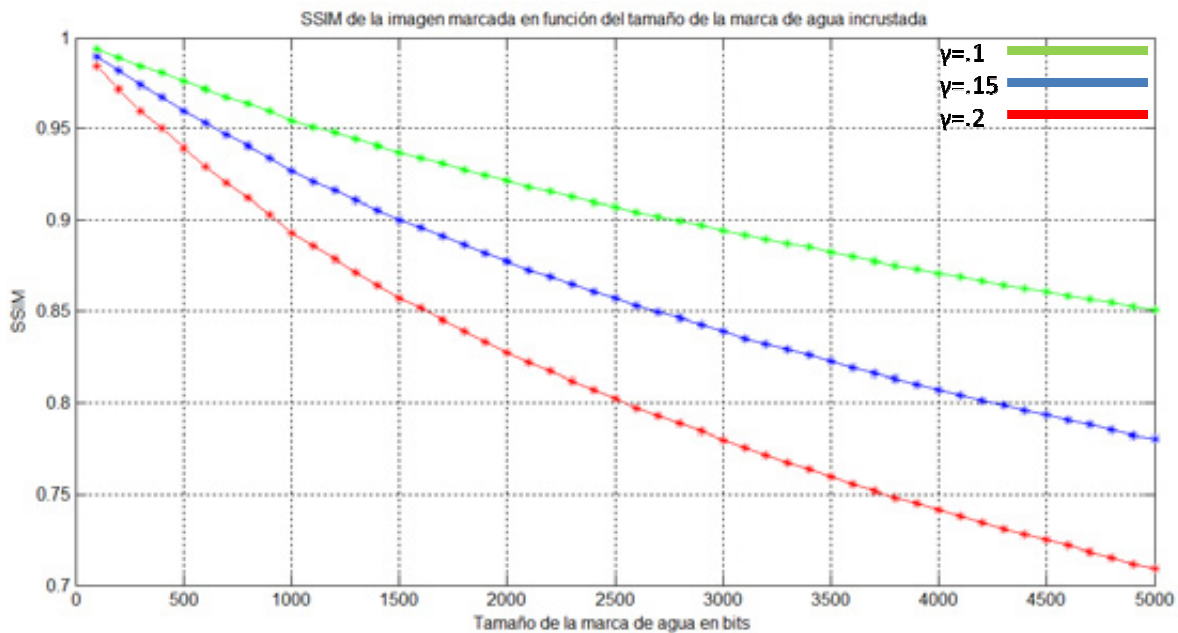


Figura 5.16 SSIM de la imagen marcada contra tamaño de la marca de agua en bits.

A partir de pruebas subjetivas realizadas podemos decir que para valores de $SSIM > 0.8$ y de $PSNR > 32\text{dB}$ tenemos una “buena” calidad perceptual de la imagen marcada. Las gráficas anteriores revelan que para un valor $\gamma=0.15$ podemos insertar una marca de agua de máximo 3Kbits antes de poder percibir notorias alteraciones en la imagen marcada.

1.5 Prueba a diferentes ataques

Si bien el estudio realizado en esta tesis no contempla un análisis exhaustivo de los diferentes ataques conocidos en la literatura, sí se consideró aquellos no maliciosos que aparecen típicamente en un sistema de transmisión de información digital.

5.5.1 Compresión JPEG

Utilizando el algoritmo de espectro disperso mejorado con codificación convolucional se insertó una marca de agua pseudoaleatoria de 2048 bits en la imagen “Bárbara”. El valor del factor de multiplicación empleado fue $\gamma = .15$. El tamaño de la marca de agua fue seleccionado de tal manera que para el valor $\gamma = .15$ no se tenga una gran alteración perceptual de la imagen como lo muestran las gráficas 5.11 y 5.12. Además, para aplicaciones de protección a la propiedad intelectual, consideramos que una marca de 2048bits es suficiente. Por ejemplo, si se codificaran en ASCII caracteres representado cada uno con 8 bits, la marca de agua de 2048bits permite incrustar hasta 256 caracteres en la imagen, suficientes para contener información referente al propietario de la misma.

Se varió el porcentaje de compresión JPEG y se obtuvo la PSNR y SSIM para los 3 procedimientos utilizados en este trabajo, entre la imagen marcada y la original, así como el BER de la marca de agua recuperada. La recuperación de la marca de agua se realizó empleando decisión dura y decisión suave con 8 regiones en la decodificación. Enseguida se muestran los resultados obtenidos.

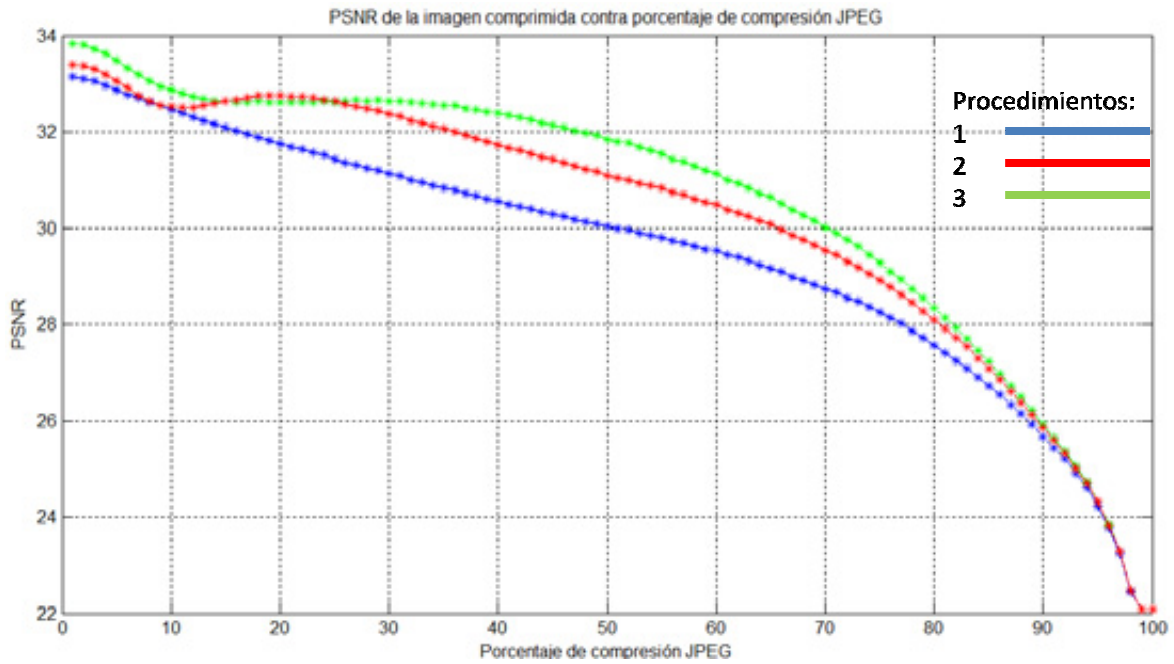


Figura 5.17 PSNR de la imagen comprimida para diferentes valores de porcentaje de compresión JPEG.

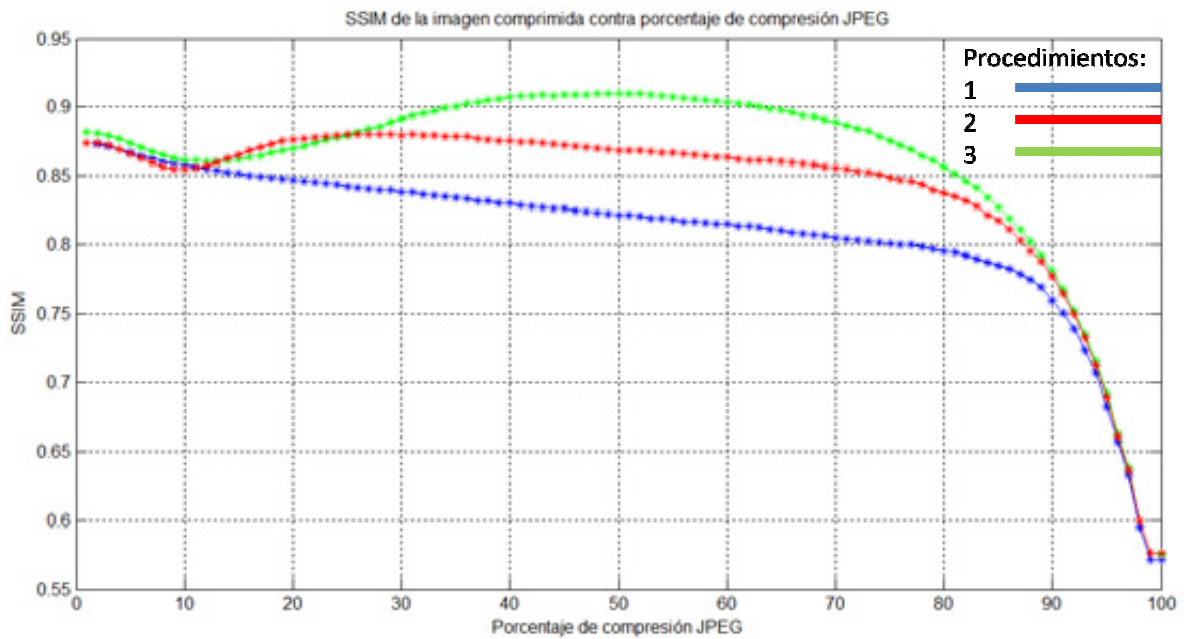


Figura 5.18 SSIM de la imagen comprimida para diferentes valores de porcentaje de compresión JPEG.

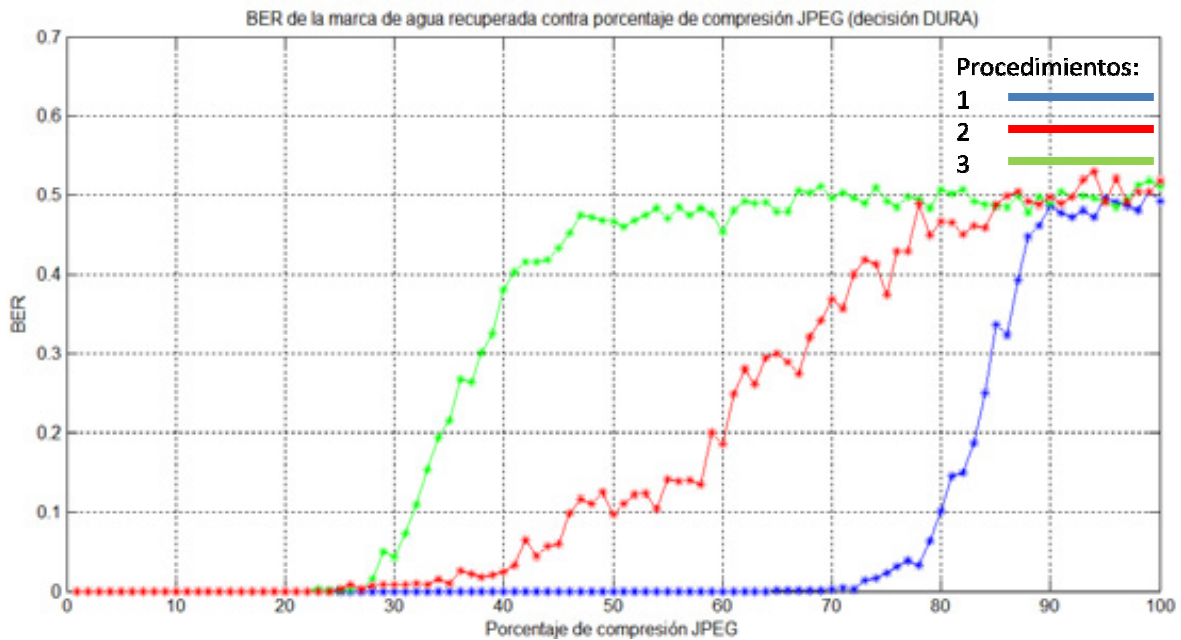


Figura 5.19 Tasa de bits en error en la marca de agua recuperada para diferentes valores de porcentaje de compresión JPEG. La decodificación se realizó utilizando decisión dura.

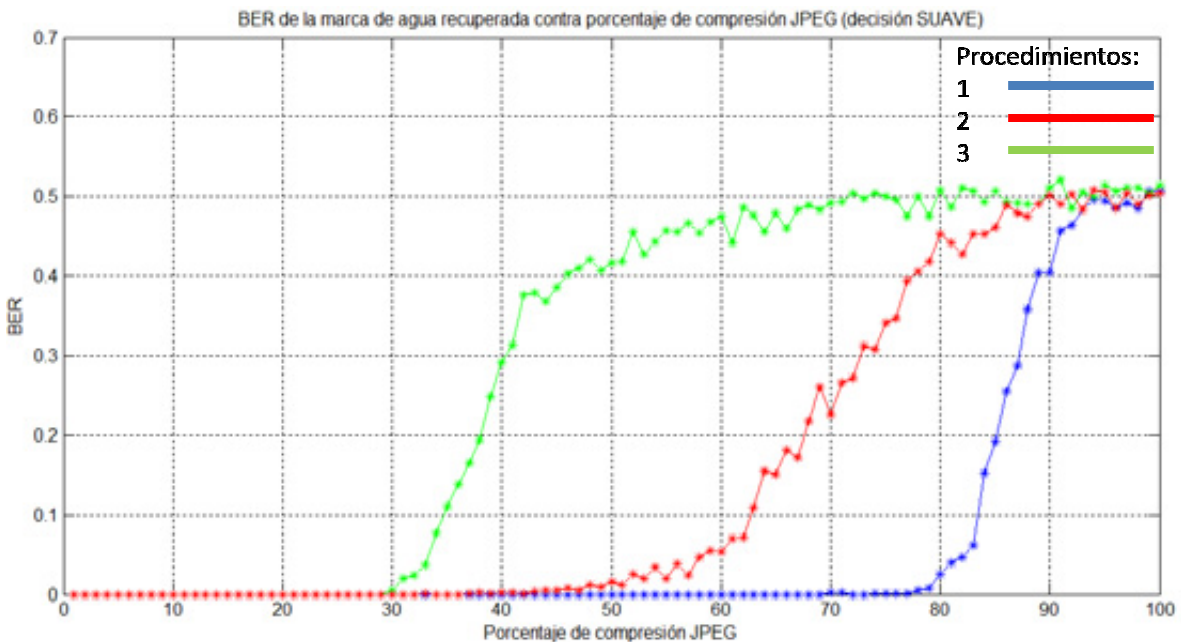


Figura 5.20 Tasa de bits en error en la marca de agua recuperada para diferentes valores de porcentaje de compresión JPEG. La decodificación se realizó utilizando decisión suave.

Las figuras 5.17, 5.18, 5.19 y 5.20 revelan que los procedimientos 1 y 2 son los que presentan la mayor robustez frente a la compresión JPEG pues mantienen la menor tasa de bits en error en la marca de agua recuperada. Ahora bien, haciendo la comparación entre decisión dura y decisión suave en la decodificación, se observa que se obtiene una menor tasa de errores en la marca de agua recuperada al utilizar decisión suave. En las siguientes gráficas se muestra la comparación del BER para decisión suave y decisión dura utilizando los 3 procedimientos.

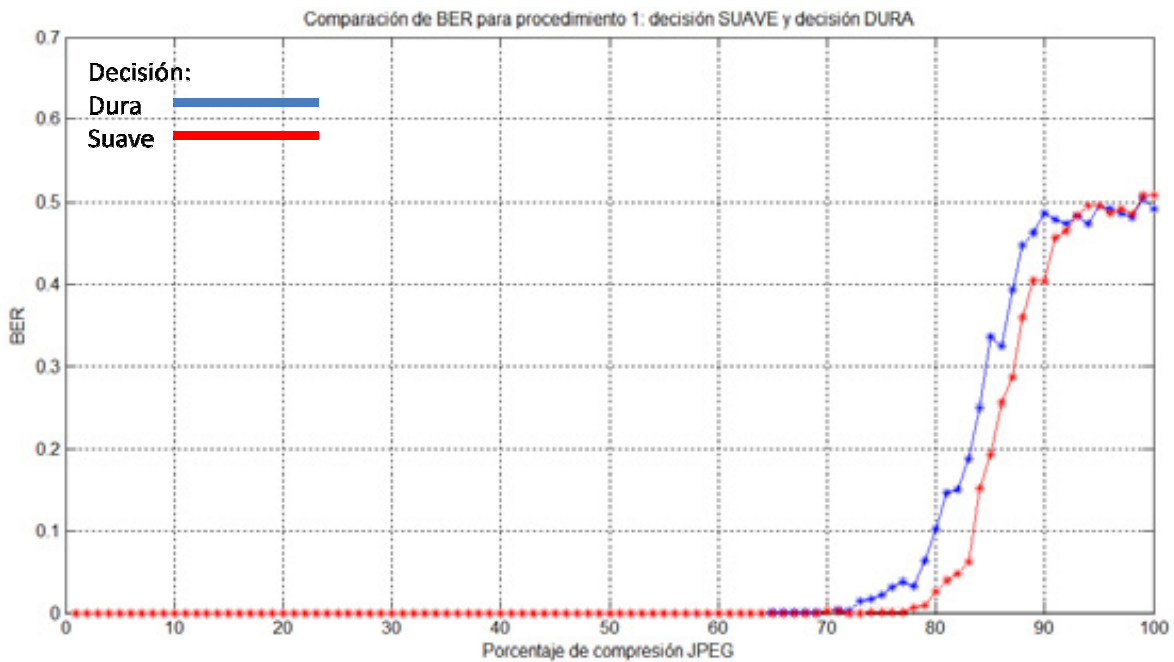


Figura 5.21 Comparación entre decisión suave y decisión dura. Tasa de bits en error en la marca de agua recuperada para procedimiento 1 contra porcentaje de compresión JPEG.

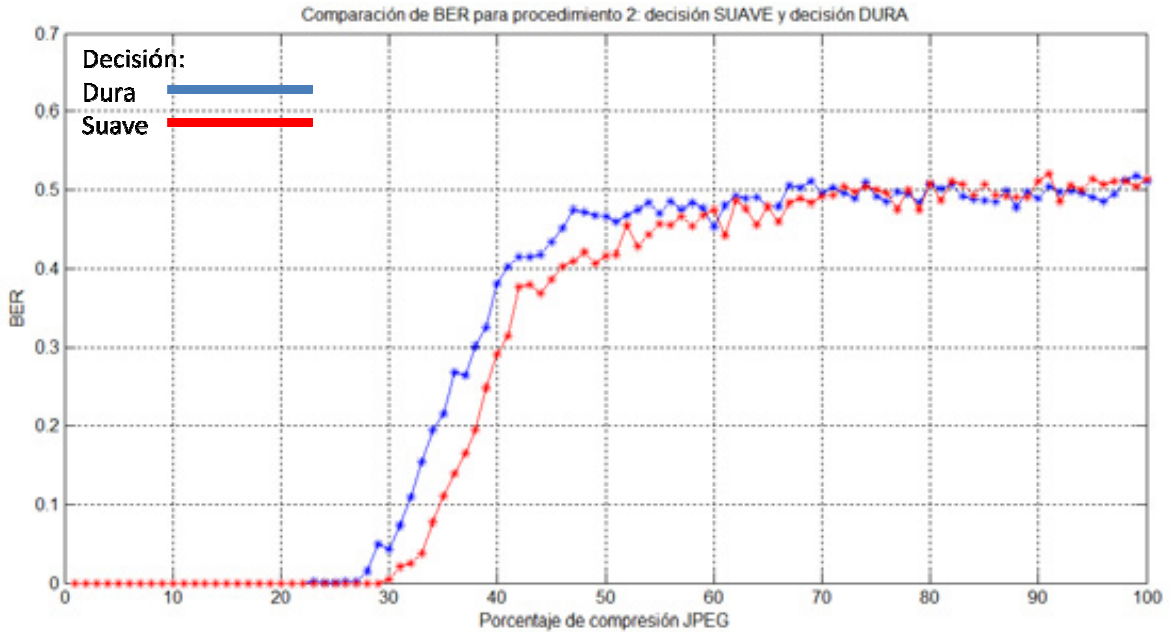


Figura 5.22 Comparación entre decisión suave y decisión dura. Tasa de bits en error en la marca de agua recuperada para procedimiento 2 contra porcentaje de compresión JPEG.

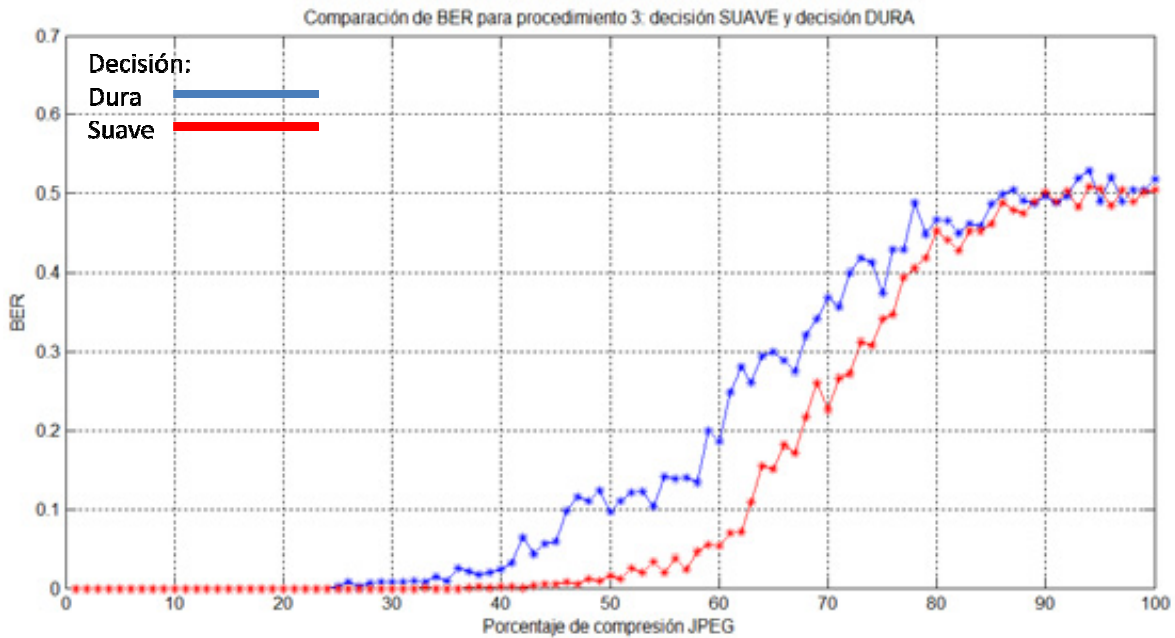


Figura 5.23 Comparación entre decisión suave y decisión dura. Tasa de bits en error en la marca de agua recuperada para procedimiento 3 contra porcentaje de compresión JPEG.

Las figuras 5.21, 5.22 Y 5.23 muestran que se tiene una menor tasa de errores utilizando decisión suave en el decodificador de Viterbi.

Utilizando el procedimiento 1, el cual presentó la menor tasa de bits en error, se insertó la marca de agua binaria "Ingeniería" de tamaño 18×125 bits mostrada en la figura 5.24 en la imagen "Lena" empleando un valor del factor de multiplicación $\gamma = .15$.

INGENIERÍA

Figura 5.24 Marca de agua binaria "Ingeniería"



Figura 5.25 (a) Imagen de "Lena". (b) Imagen marcada con la marca de agua invisible "Ingeniería" PSNR=33.52dB, SSIM=.8282.

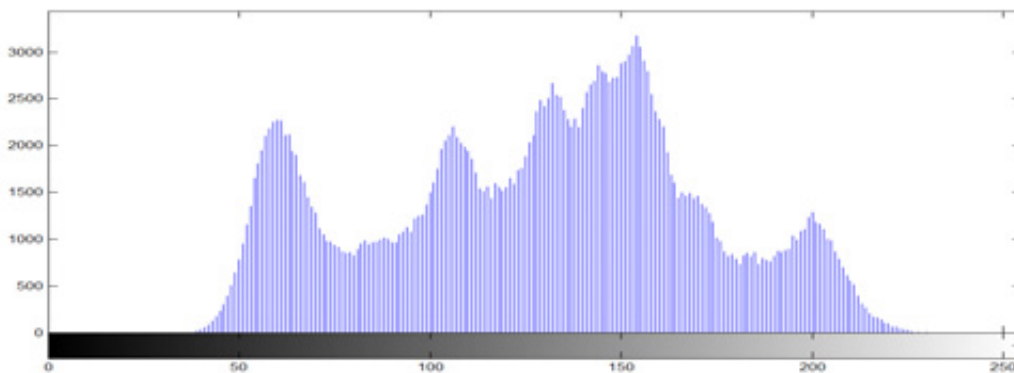


Figura 5.26 Histograma de la imagen "Lena" original. Entropía=7.2894 bits

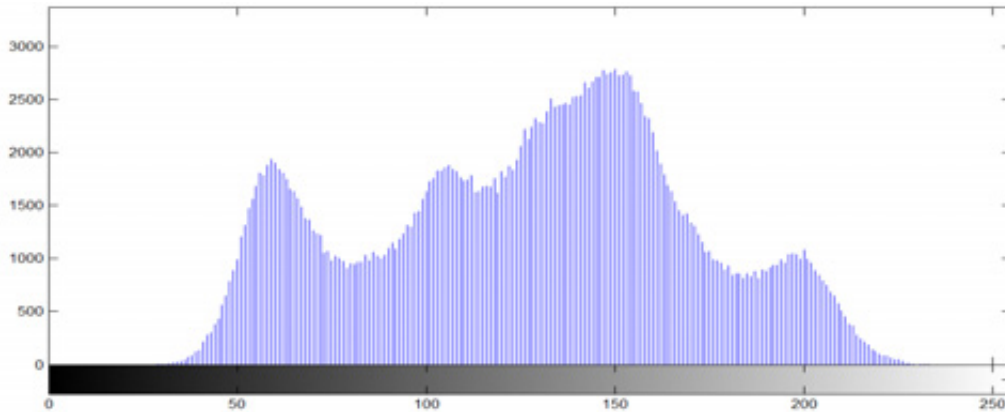


Figura 5.27 Histograma de la imagen "Lena" marcada con marca de agua "Ingeniería".

Entropía=7.3432 bits

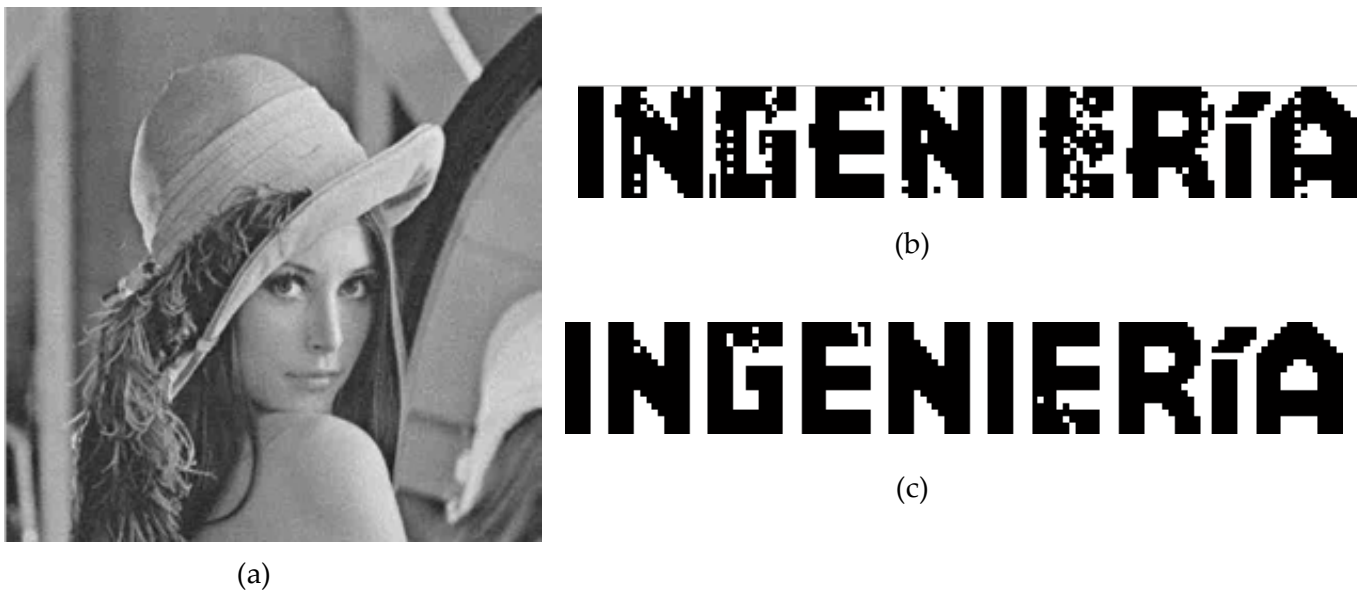


Figura 5.28 (a) Imagen marcada comprimida. Porcentaje de compresión JPEG=80%, PSNR=31.2dB, SSIM=.79. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.

Posteriormente con el procedimiento 1 se insertó la marca de agua binaria "Puma" de tamaño 31×32 bits mostrada en la figura 5.29 en la imagen "Bárbara" empleando un valor del factor de multiplicación $\gamma = .15$.



Figura 5.29 Marca de agua binaria "Puma"



(a)



(b)

Figura 5.30 (a) Imagen "Bárbara". (b) Imagen marcada con la marca de agua "Puma".
PSNR=36.24dB, SSIM=.9291.

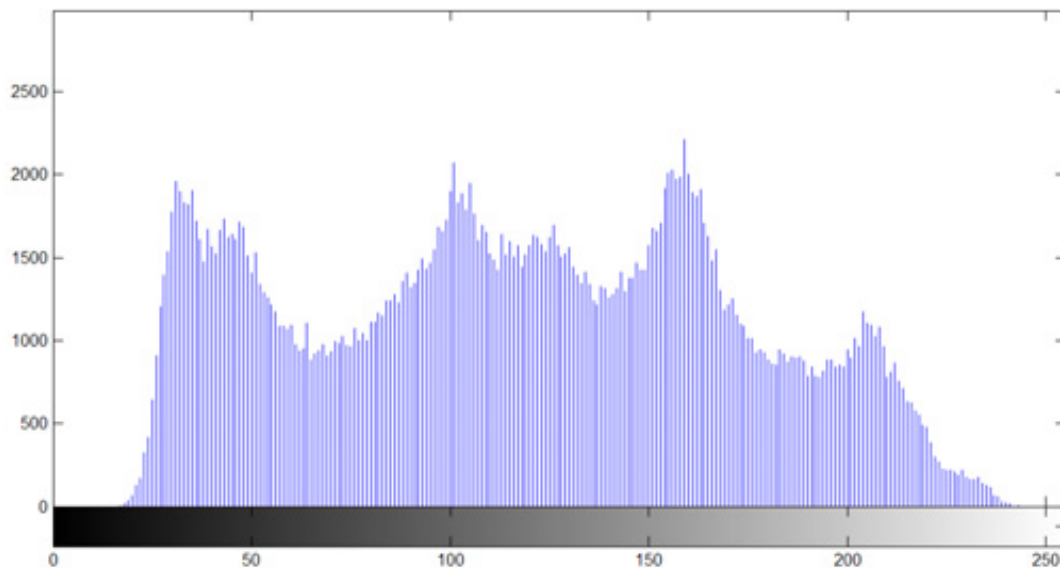


Figura 5.31 Histograma de la imagen "Bárbara" original. Entropía=7.6321 bits.

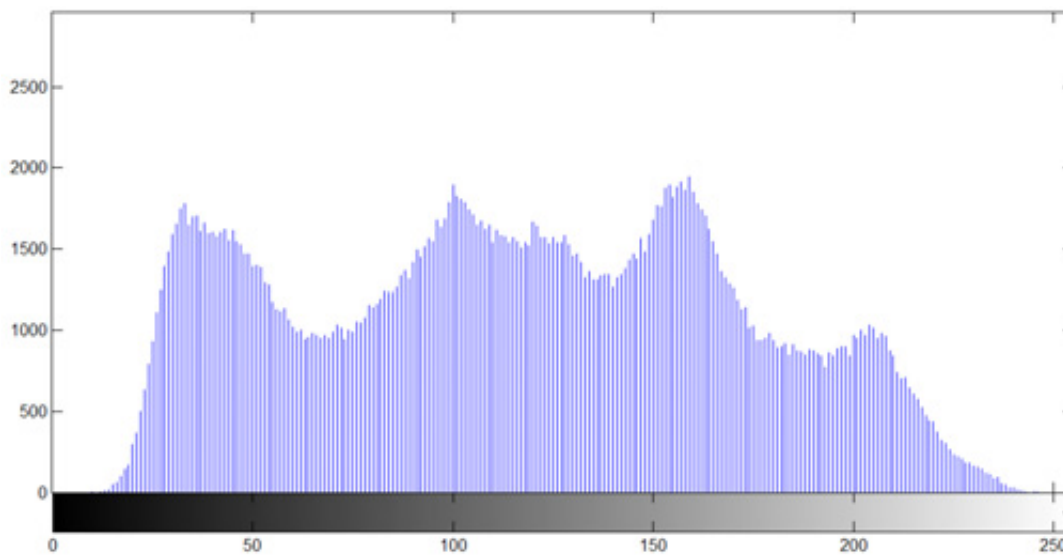


Figura 5.32 Histograma de la imagen "Bárbara" marcada con marca de agua "Puma".
Entropía=7.6576 bits.

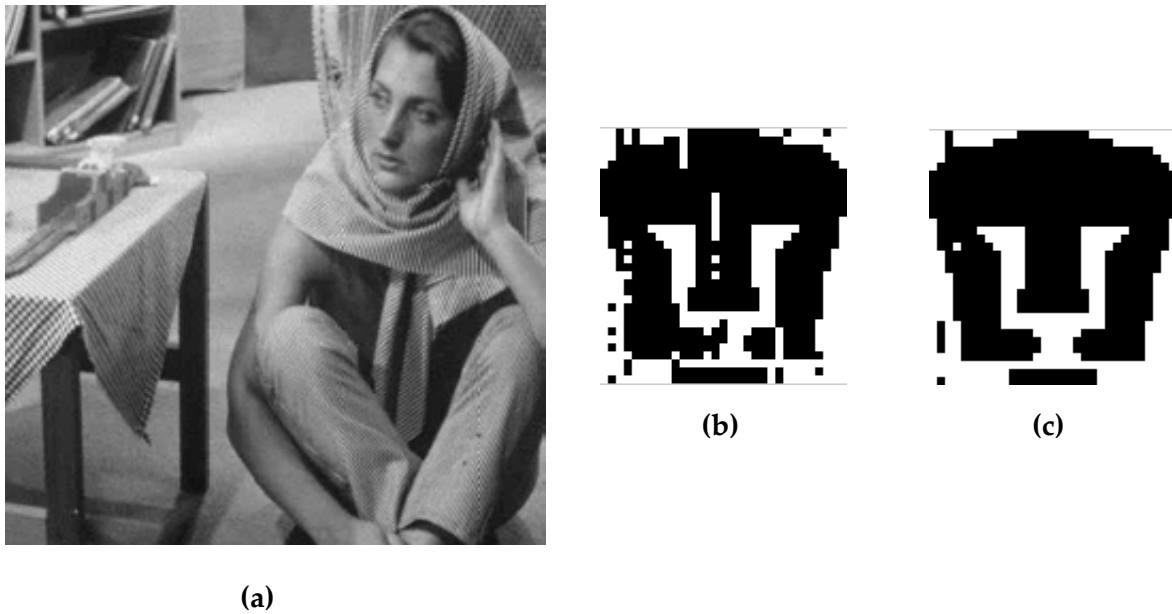


Figura 5.33 (a) Imagen marcada con marca de agua “Puma” y comprimida. Porcentaje de compresión JPEG=80%, PSNR=28.72dB, SSIM=.8466. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.

5.5.2 Ruido Gaussiano

Utilizando el algoritmo de espectro disperso mejorado con codificación convolucional se insertó una marca de agua pseudoaleatoria de 2048 bits en la imagen “Bárbara”. El valor del factor de multiplicación empleado fue $\gamma = .15$. Se atacó la imagen marcada con ruido aditivo Gaussiano con media cero y moviendo la varianza σ , se midieron la PSNR y SSIM entre la imagen marcada y la original para los 3 procedimientos, así como el BER de la marca de agua recuperada. La recuperación de la marca de agua se realizó empleando decisión dura y decisión suave con 8 regiones. Enseguida se muestran los resultados obtenidos al variar la varianza del ruido Gaussiano.

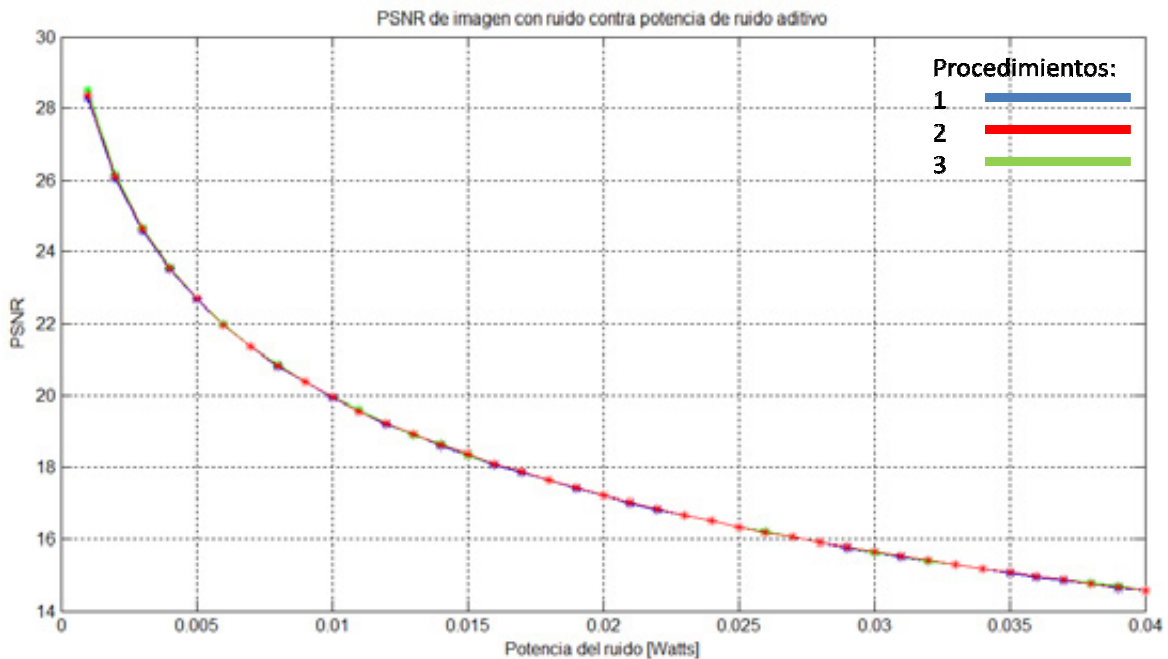


Figura 5.34 PSNR de la imagen “ruidosa” contra potencia del ruido Gaussiano.

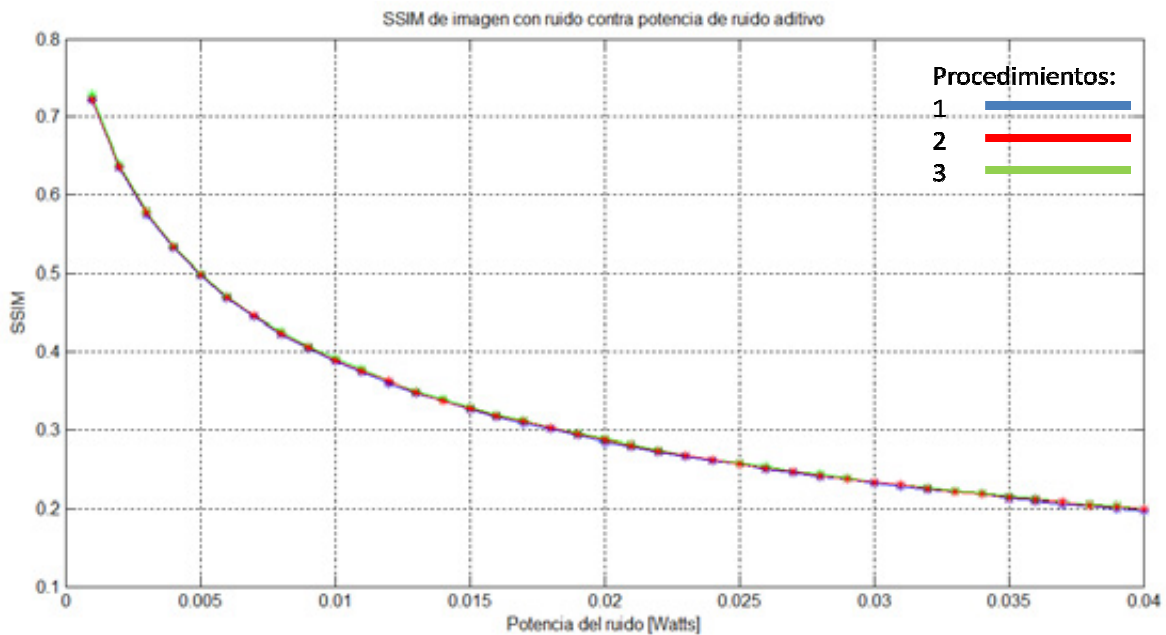


Figura 5.35 PSNR de la imagen “ruidosa” contra potencia del ruido Gaussiano.

En las figuras 5.34 y 5.35 se observa cómo se va degradando la calidad visual de la imagen marcada luego de añadir el ruido Gaussiano. Al incrementar la potencia del ruido se disminuye la semejanza entre la imagen atacada y la imagen original. Las gráficas para los 3 procedimientos están prácticamente encimadas porque el ruido afecta de la misma

manera a la imagen en los 3 procedimientos y además, de acuerdo a las gráficas de las figuras 5.4 y 5.5, para $\gamma=15$, las curvas prácticamente se intersectan en dicho valor de γ , es por ello que en las dos gráficas anteriores aparecen tan cercanas entre sí.

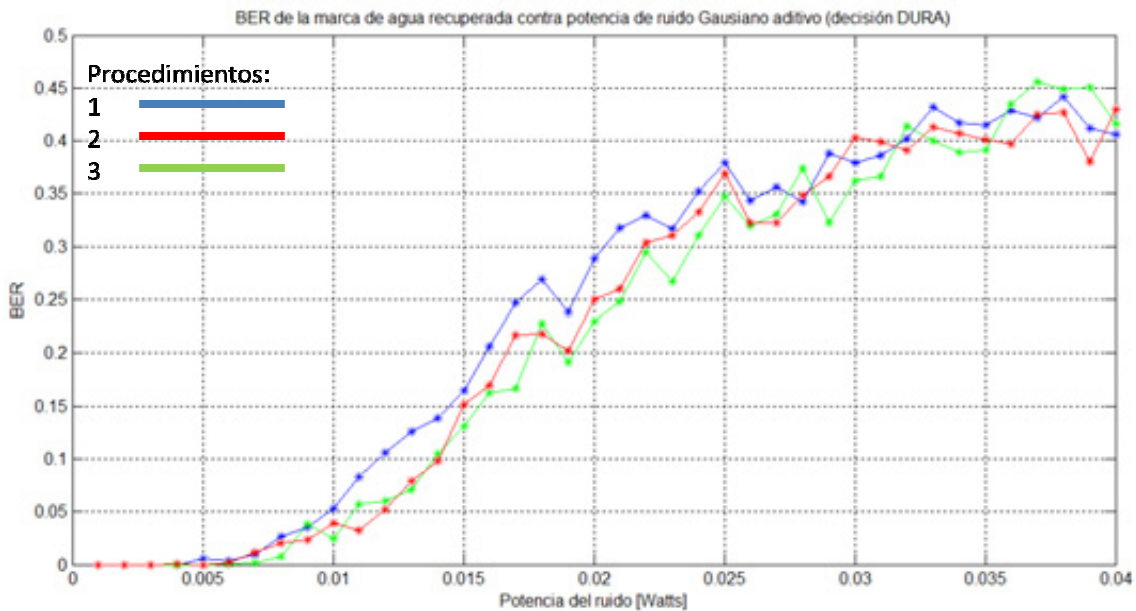


Figura 5.36 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido Gaussiano utilizando codificación convolucional con decisión dura.

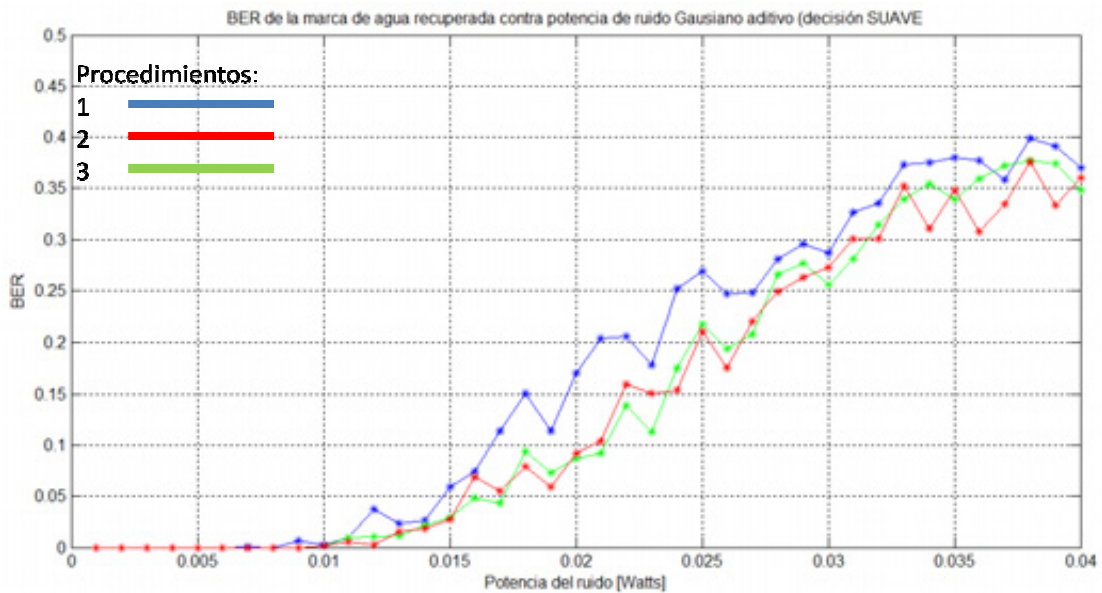


Figura 5.37 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido Gaussiano utilizando codificación convolucional con decisión suave.

Las figuras 5.36 y 5.37 muestran que existe un valor de umbral de potencia para que se comiencen a producir errores en la marca de agua recuperada. Comparando ambas gráficas se puede observar que para el caso de decisión suave se puede ganar aproximadamente 2.5dB en la PSNR, respecto a la decodificación utilizando decisión dura. Es decir, para tener una misma tasa de bits en error se puede decodificar con decisión suave con una PSNR en la imagen ruidosa, 2.5dB por debajo de la PSNR que se necesita tener utilizando una decisión dura.

Luego, con el procedimiento 2, se insertó la marca de agua binaria "Ingeniería" de tamaño 18×125 bits, mostrada en la figura 5.24, en la imagen "Bárbara" empleando un valor del factor de multiplicación $\gamma = .15$.

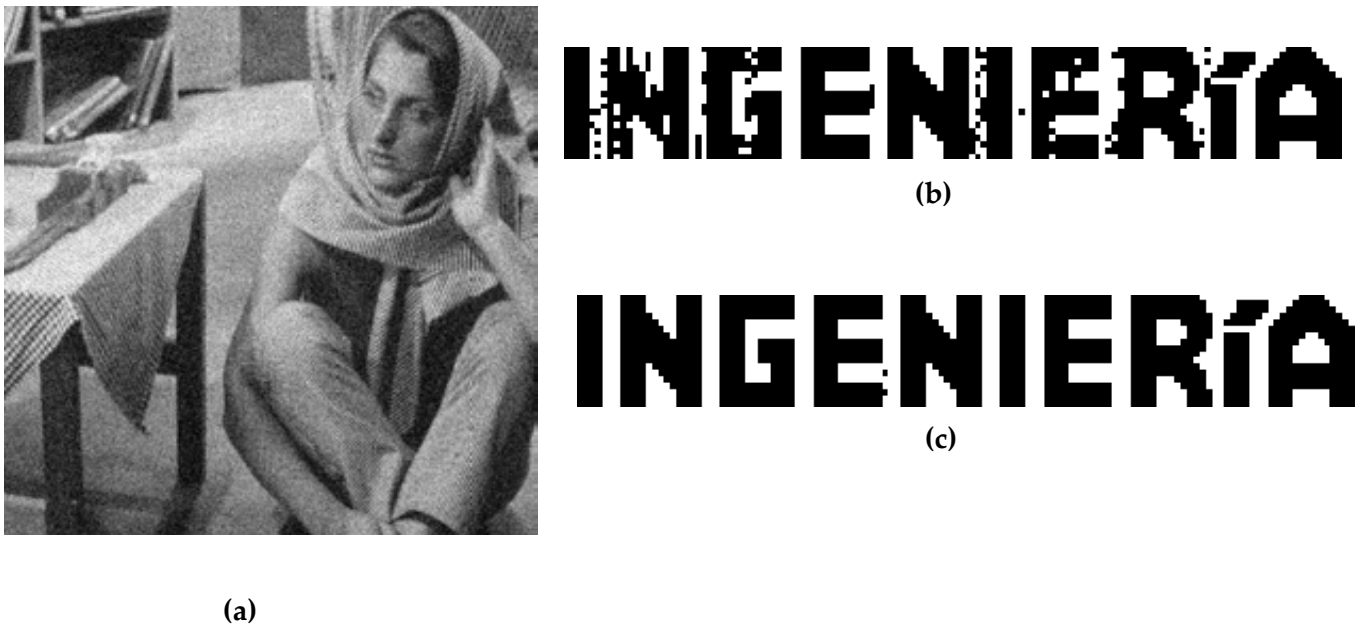


Figura 5.38 (a) Imagen marcada con ruido Gaussiano de varianza .01. PSNR=19.91dB, SSIM=.3864. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.

5.5.3 Ruido salt & pepper “sal y pimienta”

El ruido *salt & pepper* es aquel que puede tomar 2 valores en el pixel que altera: el valor de máxima o mínima intensidad. En las imágenes que estamos utilizando el número de bits utilizados para representar el nivel de intensidad de cada pixel es 8 bits, por lo que el valor de mínima intensidad es 0 y el de máxima 255. Generalmente el “nivel de ruido” *salt & pepper* se expresa en porcentaje de pixeles que son afectados por el mismo.

Utilizando el algoritmo de espectro disperso mejorado con codificación convolucional se insertó una marca de agua pseudoaleatoria de 2048 bits en la imagen “Bárbara”. El valor del factor de multiplicación empleado fue $\gamma = .15$. Se atacó la imagen marcada con ruido *salt & pepper* y se obtuvieron la PSNR y SSIM para los 3 procedimientos entre la imagen marcada y la original, así como el BER de la marca de agua recuperada. La recuperación de la marca de agua se realizó empleando en el decodificador de Viterbi decisión dura y decisión suave con 8 regiones. Enseguida se muestran los resultados obtenidos al variar el “porcentaje de ruido” *salt & pepper*.

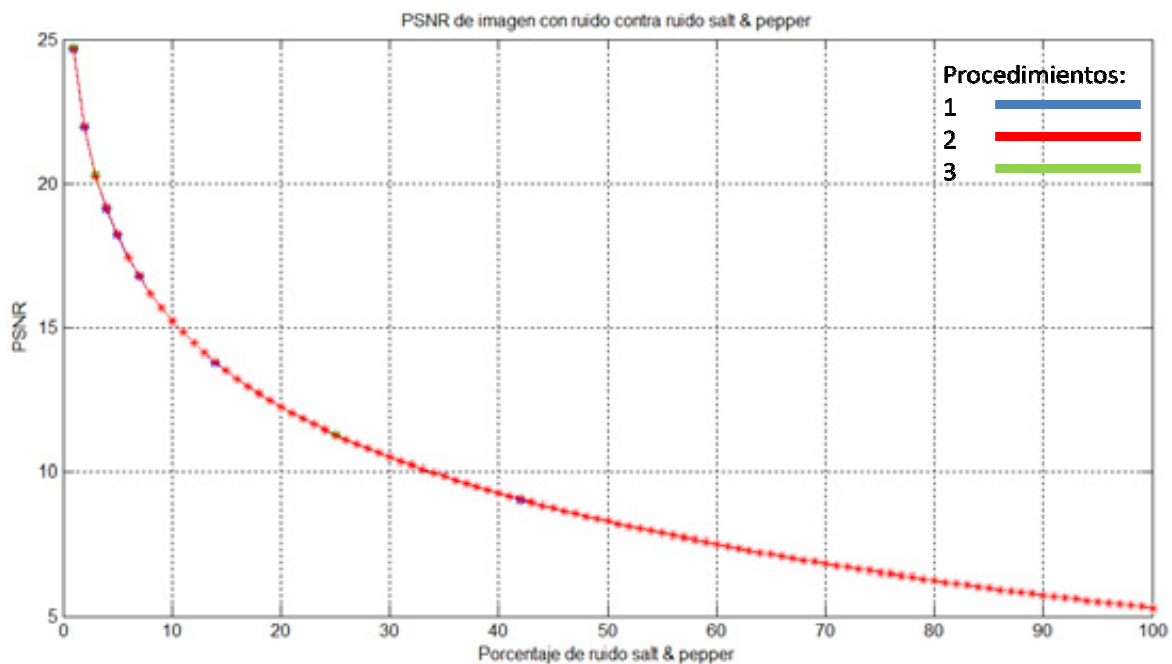


Figura 5.39 PSNR de la imagen “ruidosa” contra porcentaje de ruido *salt & pepper*.

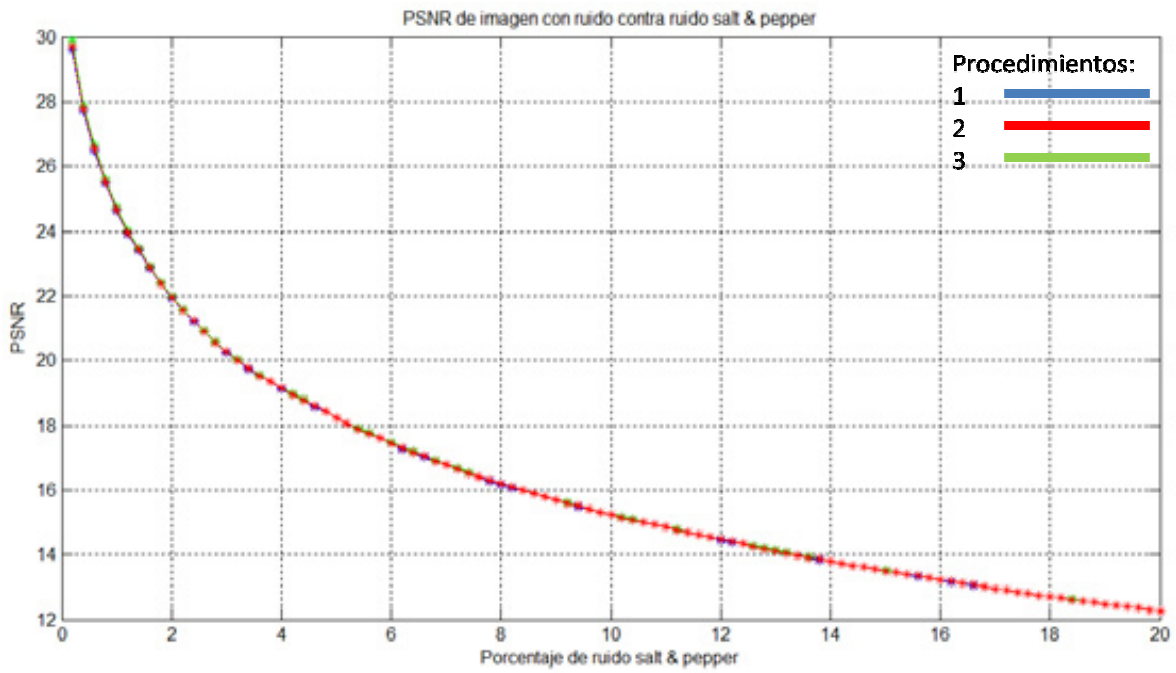


Figura 5.40 PSNR de la imagen "ruidosa" contra porcentaje de ruido *salt & pepper* (acercamiento de 0 a 20%)

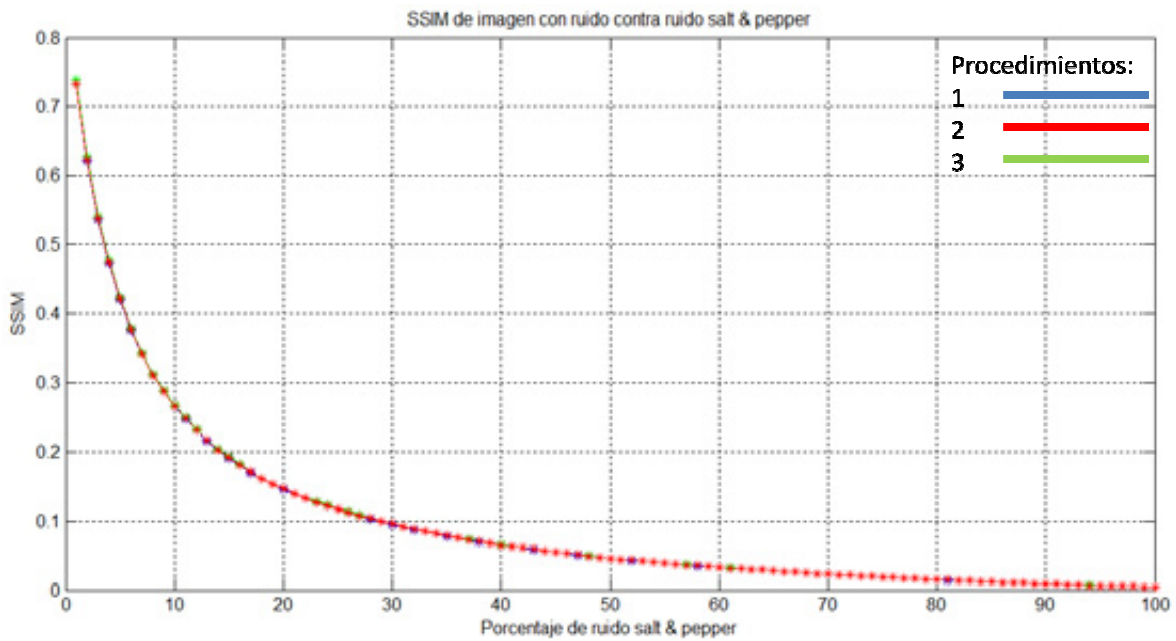


Figura 5.41 SSIM de la imagen "ruidosa" contra porcentaje de ruido *salt & pepper*.

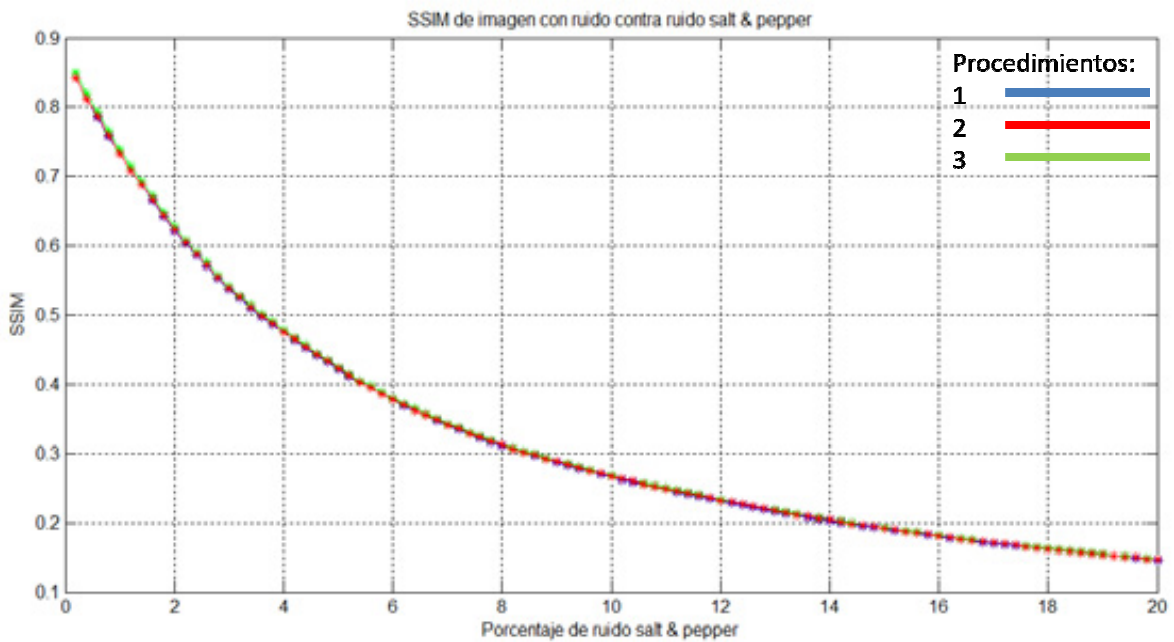


Figura 5.42 SSIM de la imagen “ruidosa” contra porcentaje de ruido salt & pepper (acercamiento de 0 a 20%)

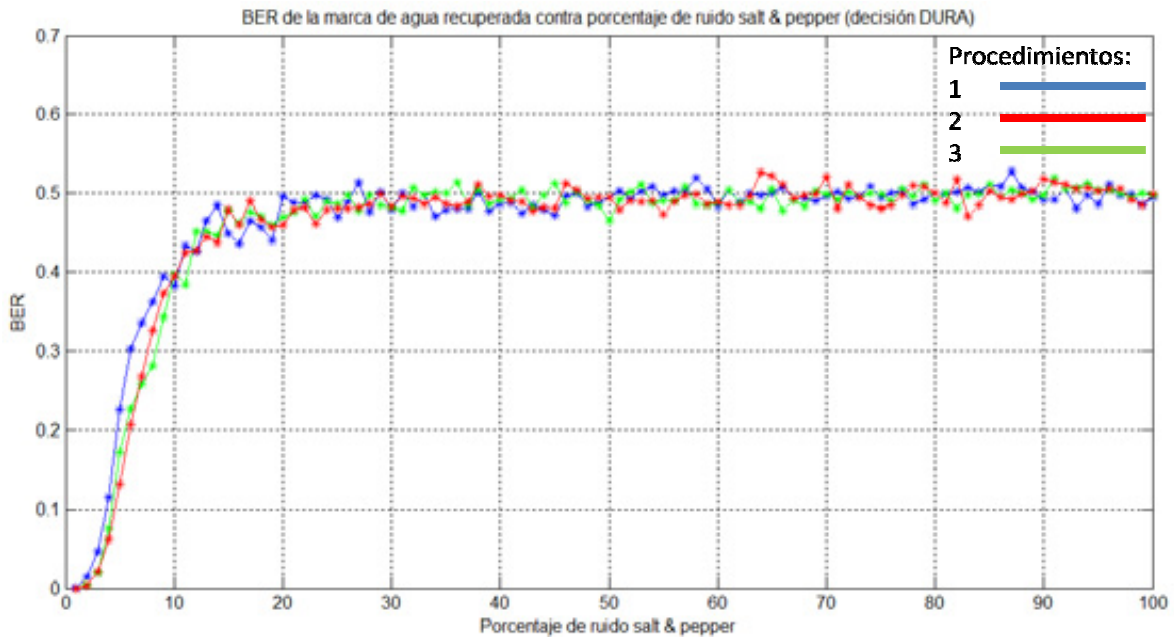


Figura 5.43 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido salt & pepper utilizando decisión dura en el decodificador de Viterbi.

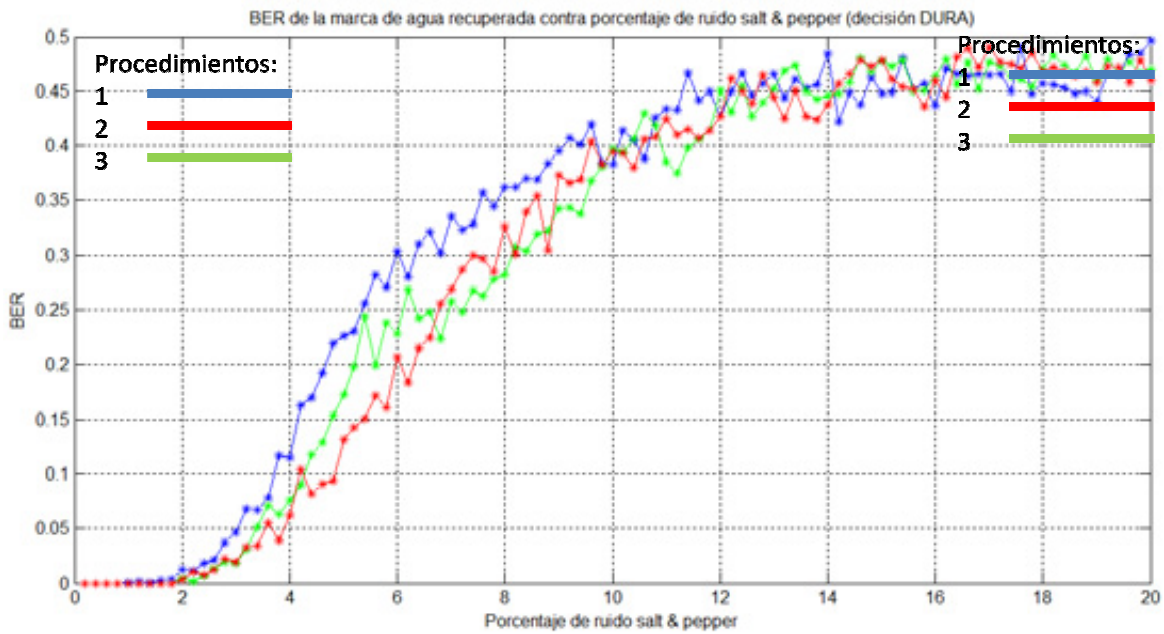


Figura 5.44 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido *salt & pepper* (acercamiento de 0 a 20%) utilizando decisión dura en el decodificador de Viterbi.

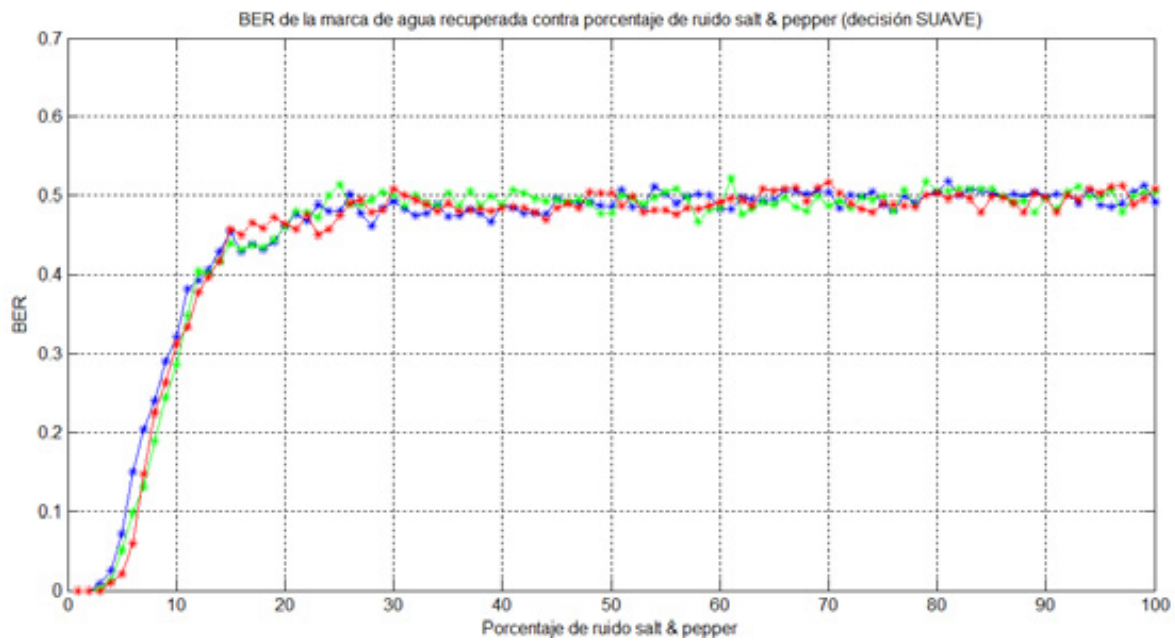


Figura 5.45 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido *salt & pepper* utilizando decisión suave en el decodificador de Viterbi.

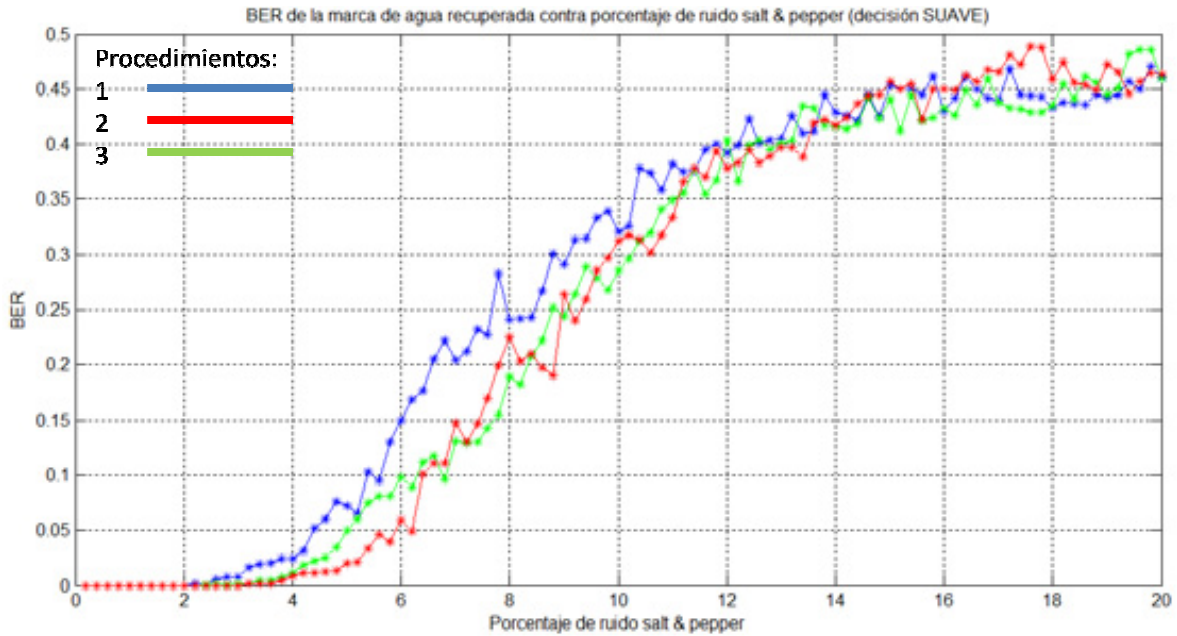


Figura 5.46 Tasa de bits en error en la marca de agua recuperada de la imagen con ruido *salt & pepper* (acercamiento de 0 a 20%) utilizando decisión suave en el decodificador de Viterbi.

Nuevamente se observa que el utilizar decisión suave en el decodificador disminuye la tasa de bits en error en la marca de agua recuperada.

Utilizando el procedimiento 1 se insertó la marca de agua binaria "Ingeniería" de tamaño 18×125 bits mostrada en la figura 5.24 en la imagen "Bárbara" empleando un valor del factor de multiplicación $\gamma = .15$.

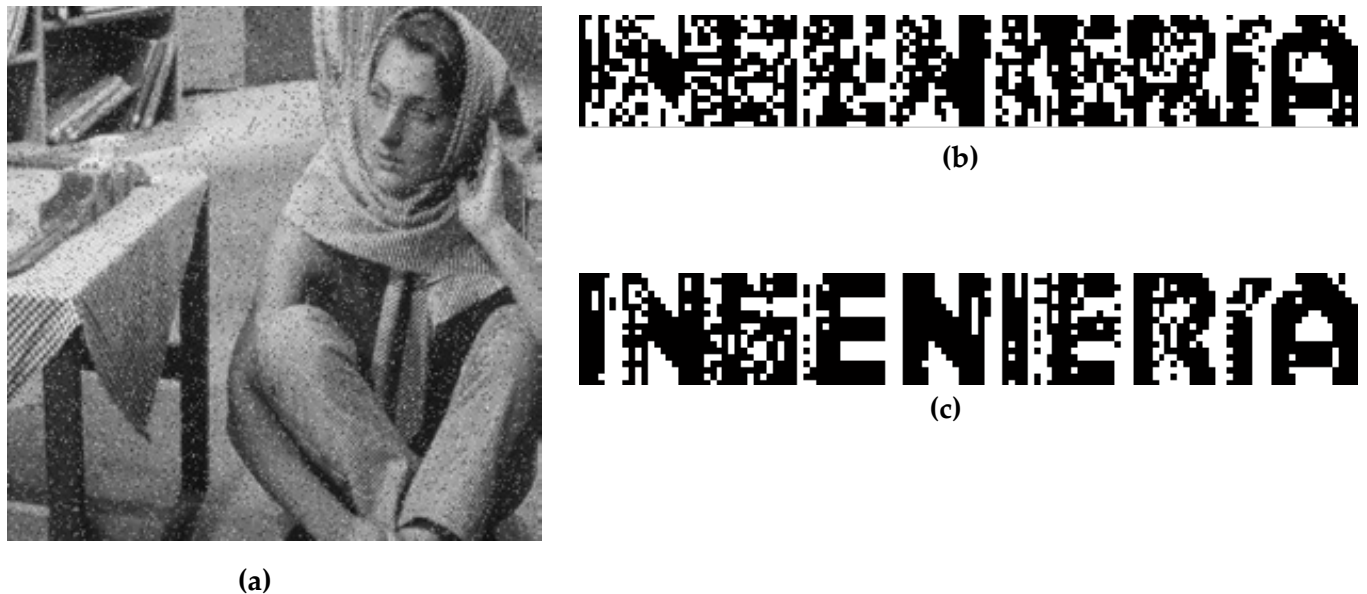


Figura 5.47 (a) Imagen marcada con 6% de ruido *salt & pepper*. PSNR=17.33dB, SSIM=.3724. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.

5.5.4 Filtrado Gaussiano paso bajas

Utilizando el algoritmo de espectro disperso mejorado con codificación convolucional se insertó una marca de agua pseudoaleatoria de 2048 bits en la imagen “Bárbara”. El valor del factor de multiplicación empleado fue $\gamma = .15$. Se atacó la imagen marcada con un filtrado pasobajas con *kernel* Gaussiano de tamaño 5×5 bits. Se obtuvieron la PSNR y SSIM, para los 3 procedimientos, entre la imagen marcada y la original, así como el BER de la marca de agua recuperada. Igualmente, la recuperación de la marca de agua se realizó empleando decisión dura y decisión suave con 8 regiones. Enseguida se muestran los resultados obtenidos al variar la desviación estándar del filtro Gaussiano.

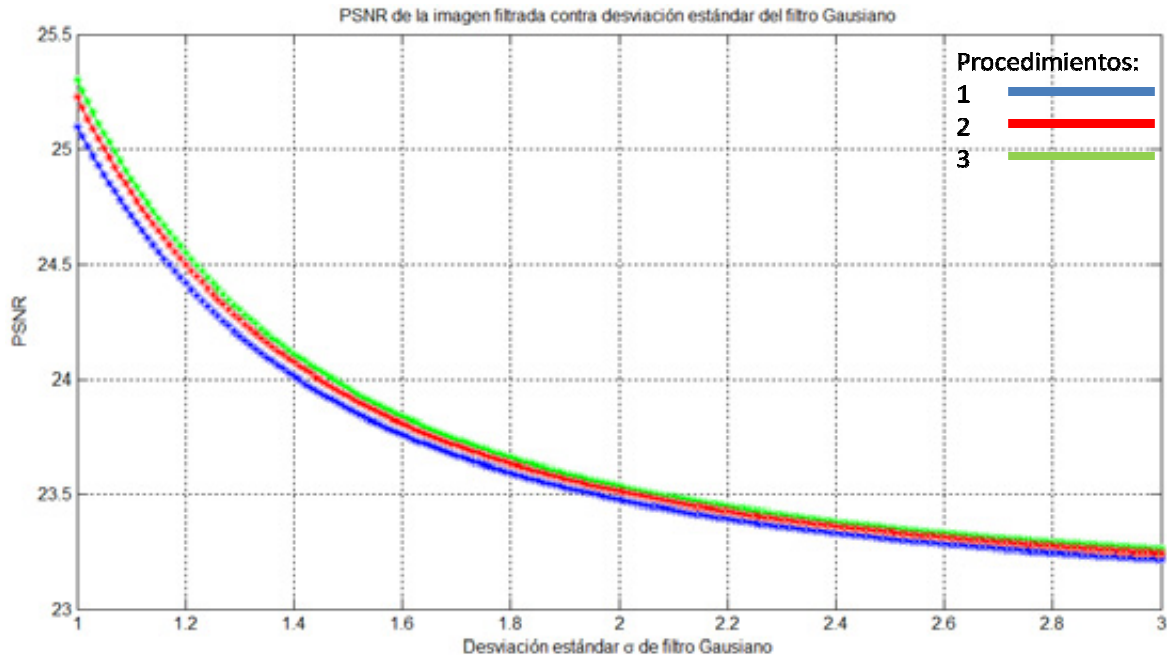


Figura 5.48 PSNR de la imagen filtrada contra desviación estándar del *kernel* Gaussiano.

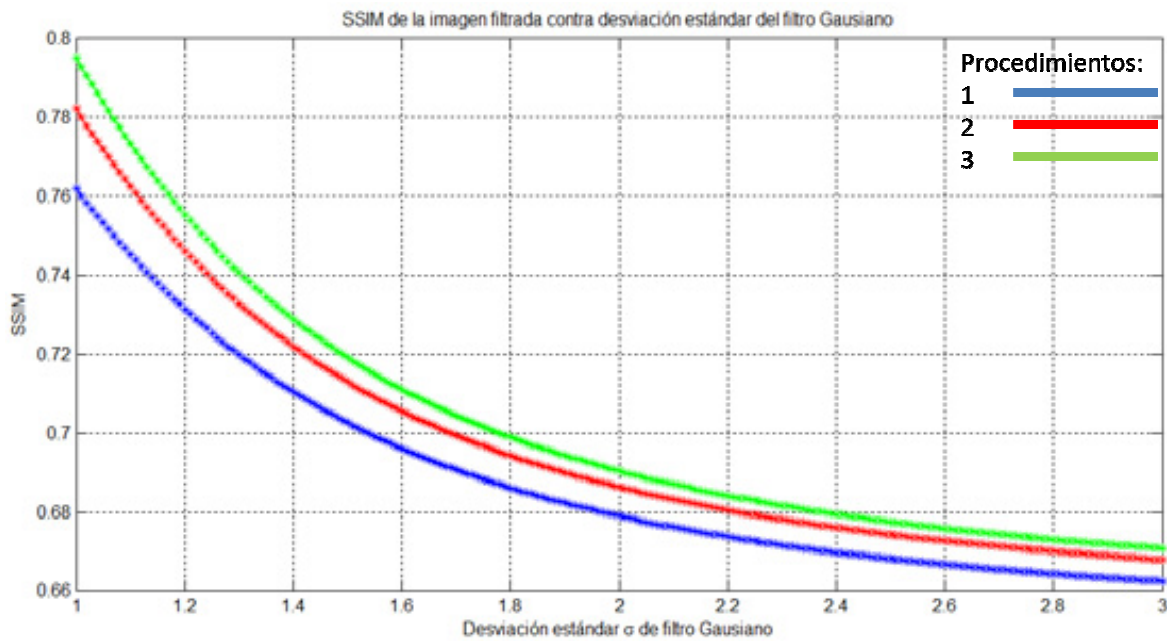


Figura 5.49 SSIM de la imagen filtrada contra desviación estándar del *kernel* Gaussiano.

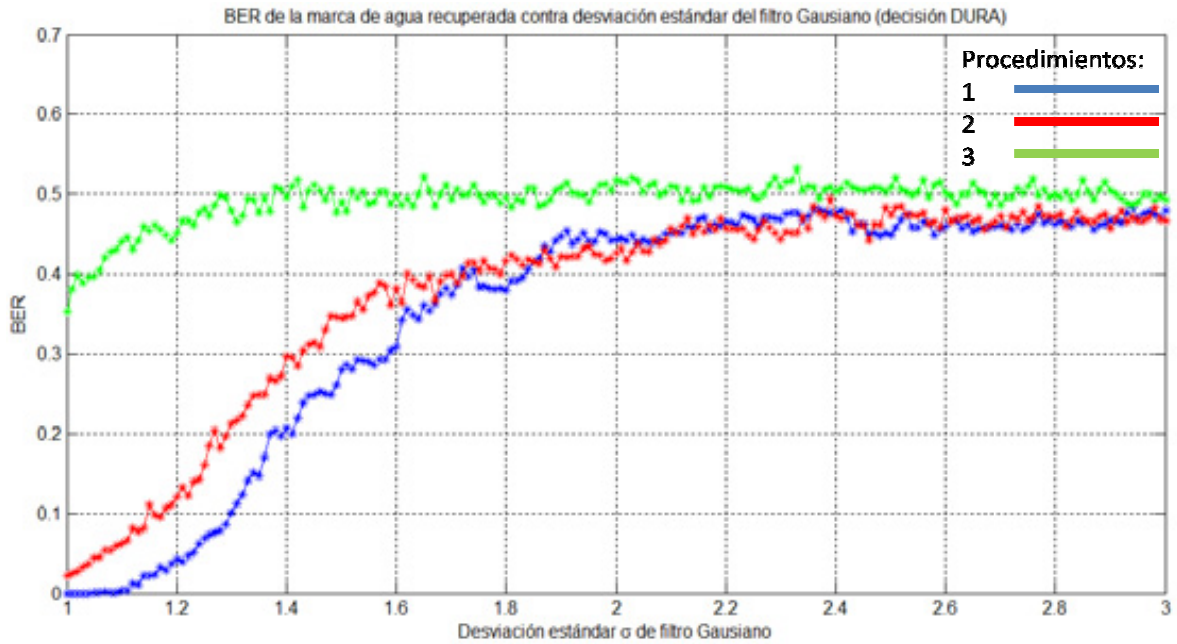


Figura 5.50 Tasa de bits en error en la marca de agua recuperada de la imagen filtrada utilizando decisión dura en el decodificador de Viterbi.

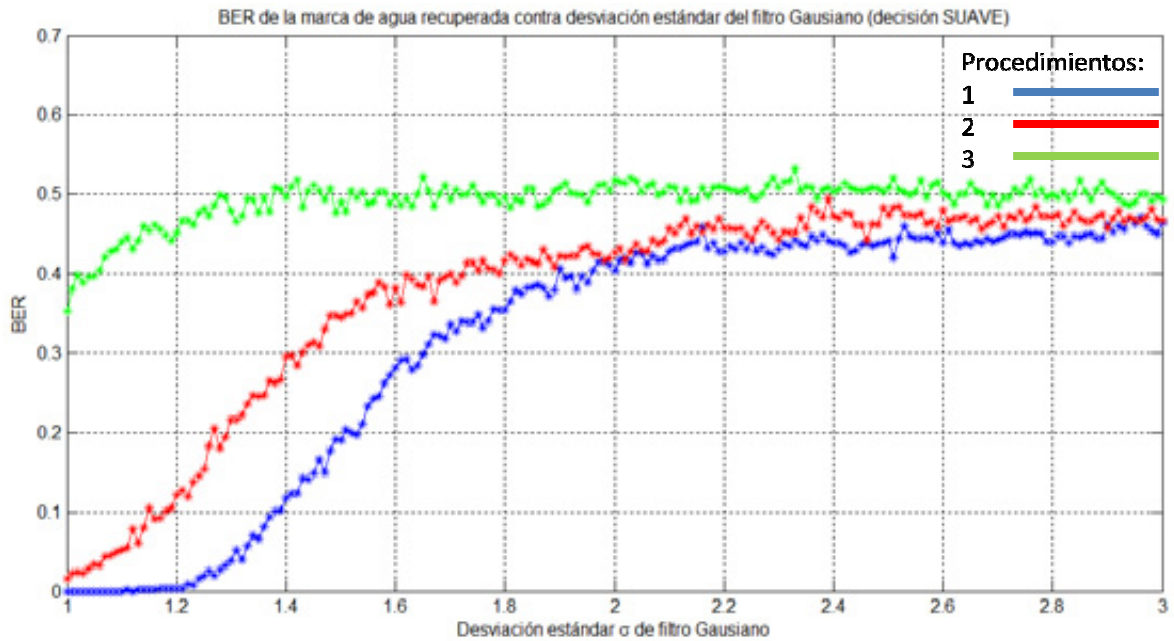


Figura 5.51 Tasa de bits en error en la marca de agua recuperada de la imagen filtrada utilizando decisión suave en el decodificador de Viterbi.

Utilizando el procedimiento 1 se insertó la marca de agua binaria “Ingeniería” de tamaño 18×125 bits mostrada en la figura 5.24 en la imagen “Bárbara” empleando un valor del factor de multiplicación $\gamma = .15$.

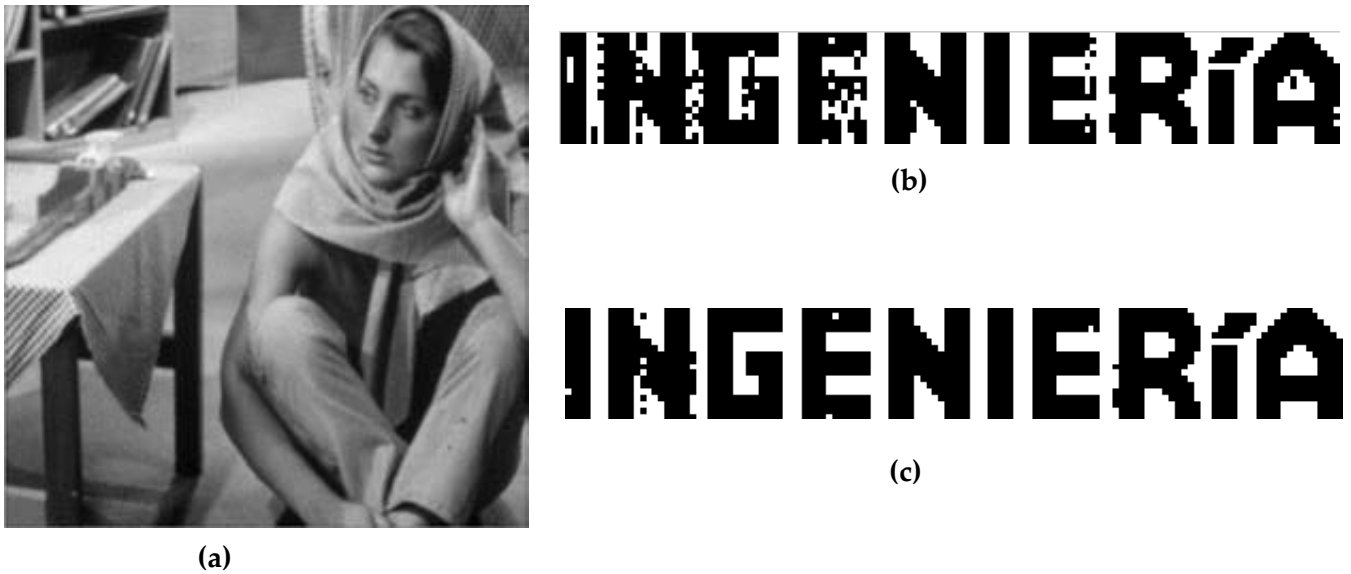


Figura 5.52 (a) Imagen marcada filtrada con filtro pasobajas Gaussiano de tamaño 5×5 y varianza $\sigma=1.2$. PSNR=24.4dB, SSIM=.7296. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.

1.5.5 Recorte de la imagen

Utilizando el algoritmo de espectro disperso mejorado con codificación convolucional se insertó una marca de agua pseudoaleatoria de 2048 bits en la imagen “Bárbara”. El valor del factor de multiplicación empleado fue $\gamma = .15$. Se atacó la imagen marcada realizando recortes a la misma y se obtuvieron la PSNR y SSIM, para los 3 procedimientos, entre la imagen marcada y la original, así como el BER de la marca de agua recuperada. La recuperación de la marca de agua se realizó empleando decisión dura y decisión suave con 8 regiones. Enseguida se muestran los resultados obtenidos al variar el porcentaje de pixeles recortados de la imagen marcada.

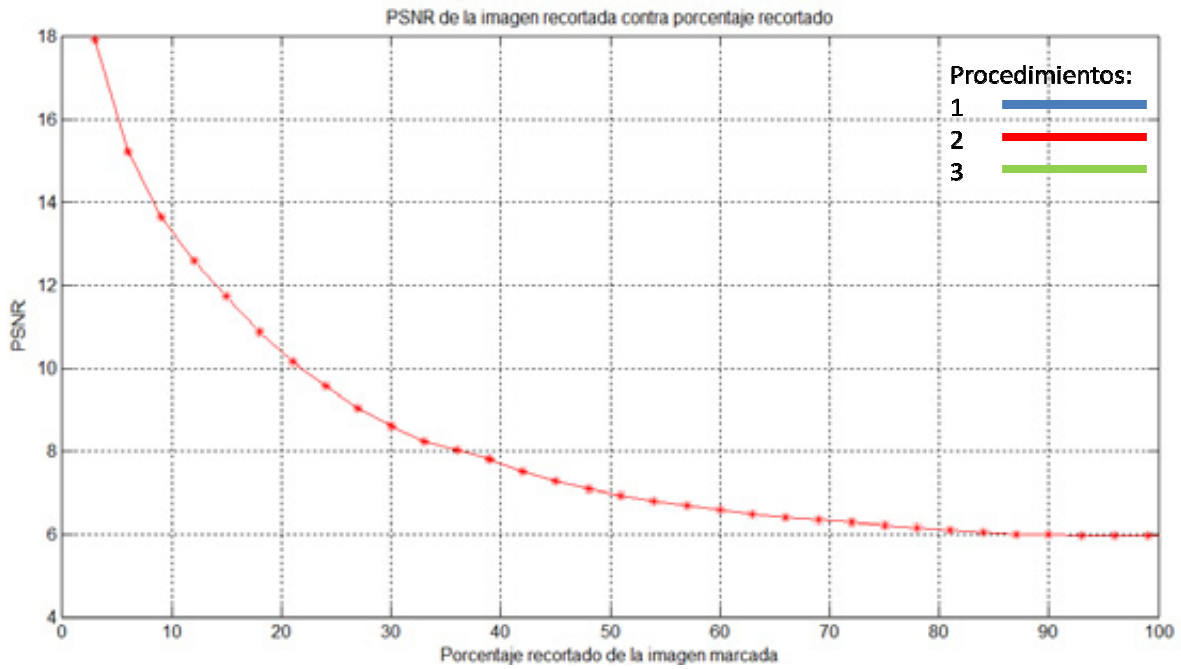


Figura 5.53 PSNR de la imagen recortada contra porcentaje recortado de la imagen marcada.

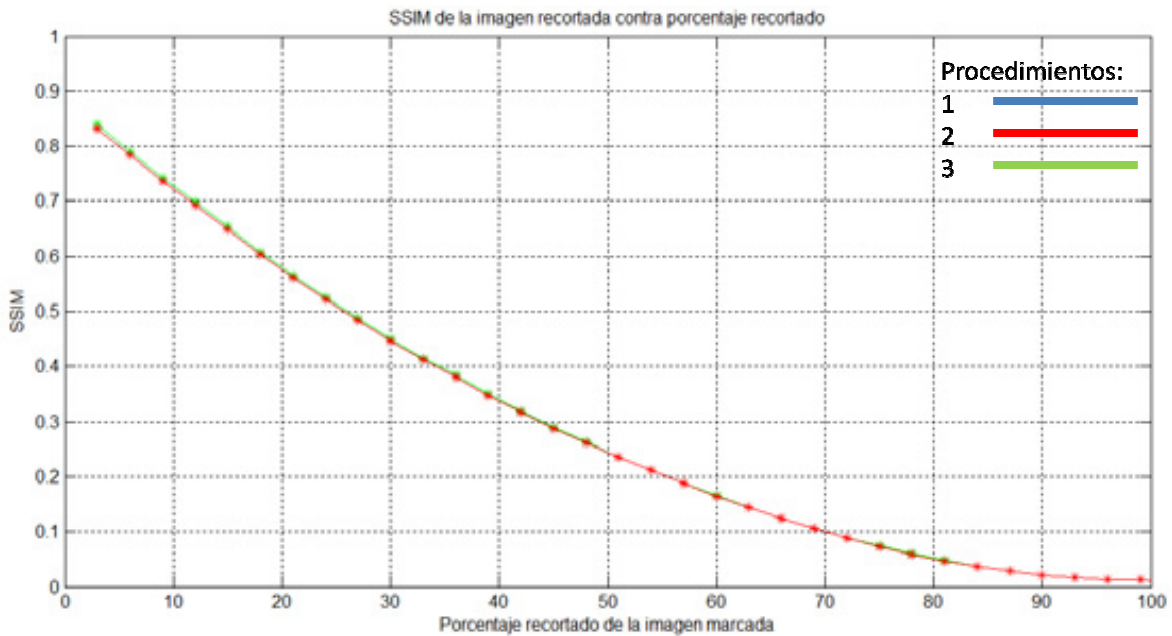


Figura 5.54 SSIM de la imagen recortada contra porcentaje recortado de la imagen marcada.

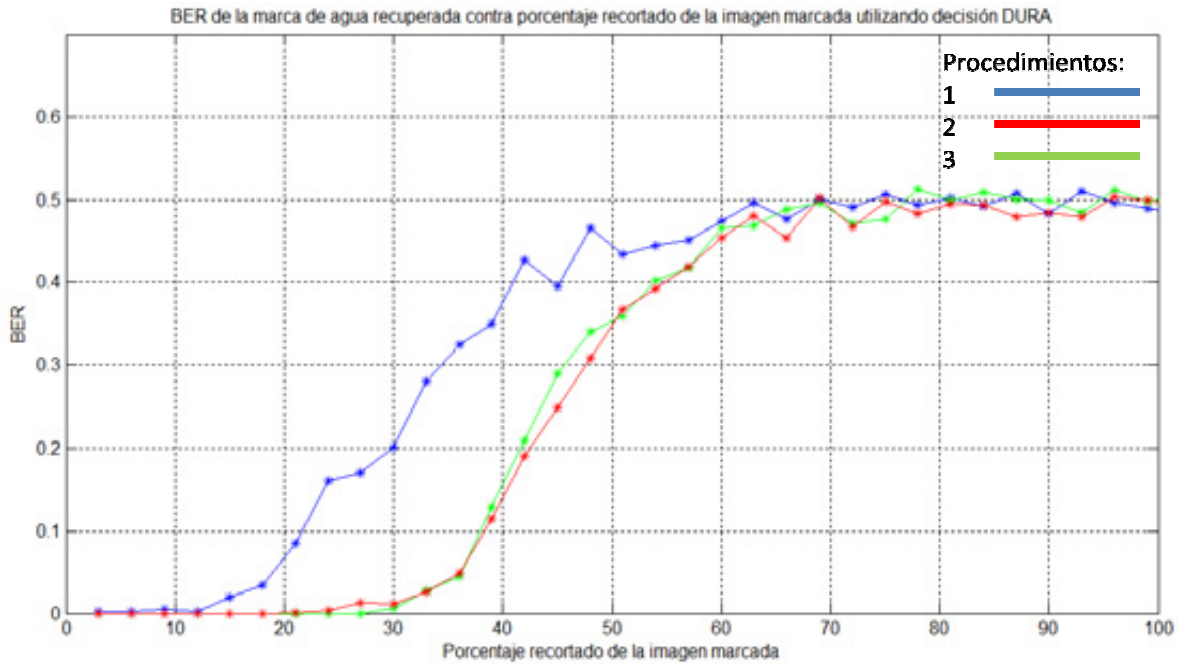


Figura 5.55 Tasa de bits en error en la marca de agua recuperada de la imagen recortada utilizando codificación convolucional con decisión dura.

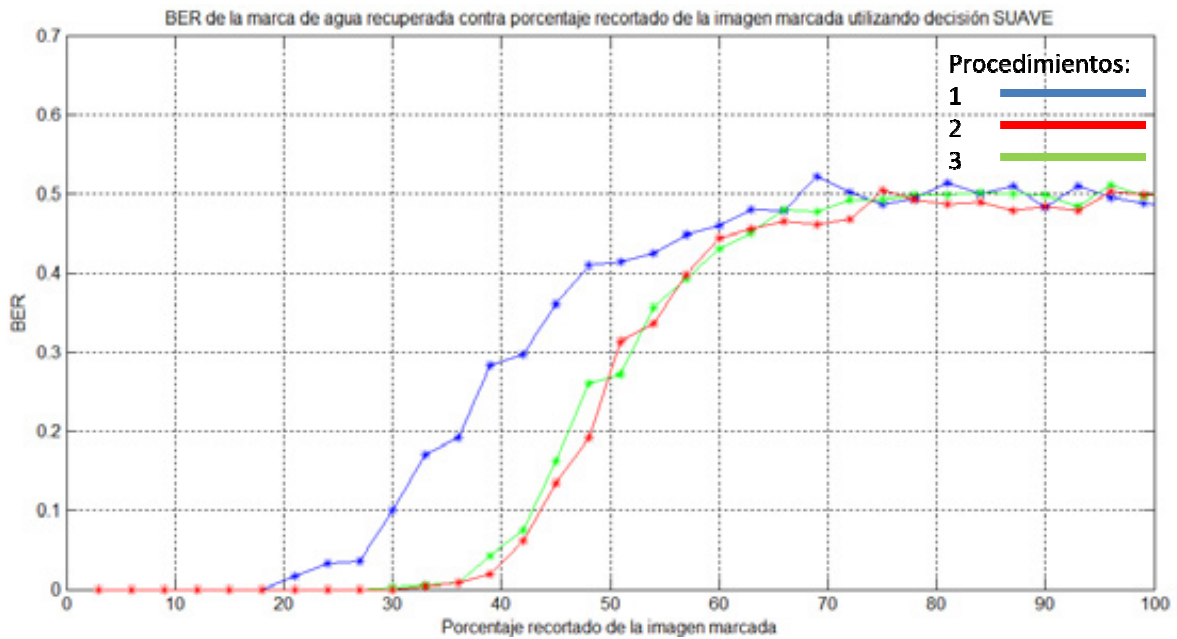


Figura 5.56 Tasa de bits en error en la marca de agua recuperada de la imagen recortada utilizando decisión suave en la decodificación.

Luego, utilizando el procedimiento 2 se insertó la marca de agua binaria “Ingeniería” de tamaño 18×125 bits, mostrada en la figura 5.62, en la imagen “Bárbara” empleando un valor del factor de multiplicación $\gamma = .15$.

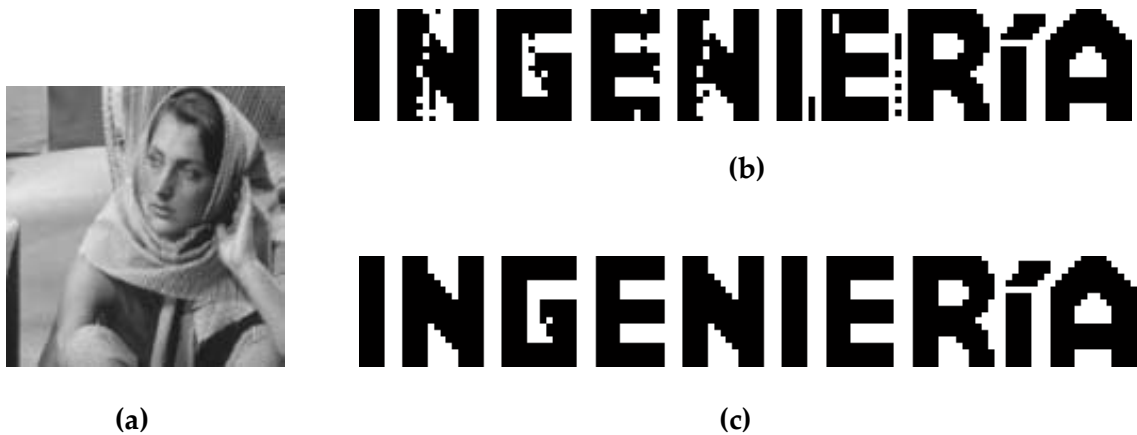


Figura 5.57 (a) Imagen marcada recortada al 35% PSNR=8.9151dB, SSIM=.3755. (b) Marca de agua recuperada utilizando decisión dura. (c) Marca de agua recuperada utilizando decisión suave.

Capítulo 6

Conclusiones

A partir de los resultados de las simulaciones podemos concluir que la utilización de codificación convolucional sí incrementó la robustez de la marca de agua. El hecho de codificar la marca de agua incrementa el número de bits que se insertan en las bandas *contourlet*. Dado que utilizamos un código convolucional de tasa 1/3, se insertan 3 veces el número de bits que se insertarían al no usar la codificación. Sin embargo, realizando la comparación entre el sistema propuesto que contempla la codificación convolucional con el algoritmo que no la utiliza, resulta que la codificación incrementa la robustez de la marca de agua para un mismo valor de SSIM, o de PSNR. Es decir, para la misma alteración perceptual de la imagen marcada se incrementa la robustez usando codificación convolucional.

Para el sistema de marcado de agua propuesto se obtuvo una menor tasa de errores en la marca de agua recuperada utilizando decisión suave en el decodificador de Viterbi, respecto al caso de decodificación con decisión dura. Utilizando decisión suave fue posible recuperar la marca de agua en imágenes con PSNR hasta 2.5dB mayores, que la PSNR de una imagen utilizando decisión dura para una misma tasa de errores. Esto implica que utilizando decisión suave se puede alcanzar una misma robustez alterando menos perceptualmente la imagen, respecto al caso de la decisión dura.

De acuerdo a los resultados de las simulaciones, el conocimiento de las características del sistema visual humano ayudó a reducir la alteración perceptual de la imagen marcada mediante una adecuada selección de subbandas *contourlet*. En las simulaciones donde cada procedimiento correspondía a una selección de subbandas, no fue el mismo procedimiento el que obtuvo la menor tasa de bits en error para todos los ataques en la marca de agua recuperada. Para el caso de ataque mediante compresión JPEG y filtrado pasobajas el



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

procedimiento más robusto fue el número 1 mientras que para ruido Gaussiano, ruido *salt & pepper* y recortes, los procedimientos más robustos fueron los 2 y 3.

Comparando la entropía de la imagen marcada y la imagen original se observa que el hecho de incrustar la marca de agua aumenta la entropía de la imagen; es decir, añadir la información de la marca de agua incrementa la información promedio de la variable aleatoria “nivel de gris” de la imagen. La diferencia en la entropía está relacionada con la existencia de diferencias entre los histogramas de la imagen sin marcar, y el histograma de la imagen que porta la marca de agua.

La capacidad de nuestro sistema de marcado de agua está limitada por la distorsión que se introduce en la imagen al añadir los bits en las bandas *contourlet*. Conforme aumenta el tamaño de la marca de agua decrece la calidad de la imagen marcada. Es por ello que se debe hacer un balance entre robustez, capacidad y alteración perceptual de la imagen. Dependiendo del tipo de ataque esperado para la imagen podemos realizar la selección de subbandas *contourlet* que hagan a la marca de agua más robusta a ese tipo de ataque. De acuerdo a las pruebas que realizamos, para una marca de agua de 2048 bits de capacidad incrustada utilizando un factor $\gamma > .15$ y considerando a los tipos de marca de agua descritos en el capítulo 2, podemos clasificar nuestro procedimiento de inserción de marca de agua como robusto, pues es resistente a ataques no maliciosos, comunes en el procesamiento digital de imágenes tales como la compresión JPEG, el ruido Gaussiano, ruido *salt & pepper*, filtrado pasobajas y recorte de la imagen.

6.1 Trabajo futuro

Los algoritmos implementados de inserción y recuperación de marca de agua constituyen un sistema de marcado de agua robusto. Sin embargo para aplicaciones en las que se requiere de una marca de agua segura, la alteración perceptual de la imagen, utilizando los algoritmos presentados en este trabajo, degradan considerablemente la imagen marcada.

En este trabajo, la inserción de la marca de agua se realizó en el dominio *contourlet*. En futuros trabajos se podría probar la inserción de la marca de agua utilizando técnicas híbridas que inserten los bits de la marca de agua tanto en el dominio espacial como en el dominio transformado explotando las ventajas de cada una de ellas. Por ejemplo, del dominio transformado, explotar la robustez que brinda incrustar la marca de agua utilizando espectro disperso, y del dominio espacial se pueden aprovechar diversas características del sistema visual humano para disminuir la alteración perceptual de la imagen. Una de ellas es el hecho de que las perturbaciones son menos visibles en áreas oscuras o muy brillantes de la imagen. Por lo tanto, se pueden conjuntar técnicas de espectro disperso con técnicas de inserción de la marca de agua en los píxeles de mayor y de menor valor en el dominio espacial.

En la presente tesis se demostró que la codificación convolucional de la marca de agua aumenta la robustez de la misma para un mismo valor del factor de multiplicación y para una misma alteración perceptual de la imagen marcada respecto al algoritmo que no realiza la codificación. En un trabajo futuro se podría buscar aumentar la robustez utilizando otros códigos correctores de errores como los códigos Reed-Solomon o los turbocódigos, aunque con el correspondiente aumento de complejidad.

Referencias:

- [1] M. Barni and F. Bartolini, *Watermarking Systems Engineering* (Signal Processing and Communications Series, Marcel Dekker, Inc., USA 2004).
- [2] I. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking* (Morgan Kaufmann Publishers, San Francisco CA, USA, 2002).
- [3] J.-S. Pan, H.-C. Huang and L. C. Jain, *Intelligent Watermarking Techniques* (Series of Innovative Intelligence, Vol 7, World Scientific, Sigapore, 2004).
- [4] D. E. Robling, *Cryptography and Data Security* (Addison-Wesley Publishing Company, Menlo Park California).
- [5] B. Schneier, *Applied Cryptography* (Second Edition, John Wiley & Sons, Inc.).
- [6] M. Barni and F. Bartolini. "Improved *wavelet*-based watermarking through pixel wise masking" *IEEE Transactions on Image Processing*, Vol. 10, No. 5, May 2001.
- [7] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. "Image quality assessment: from error visibility to structural similarity" *IEEE Transactions on Image Processing*. Vol. 13, No. 4, April 2004.
- [8] Bernard Sklar, *Digital Communications: Fundamentals and Applications* (Second Edition, System View)
- [9] John G. Proakis *Digital Communications* (3rd Edition, McGrall-Hill)
- [10] NASA Technical Report, *Simulation Results for the Viterbi Decoding Algorithm*, November 1972.
- [11] Rafael C. Gonzalez & Richard E.Woods, *Digital Image Processing* (Second Edition, Prentice Hall)
- [12] Jorge Lira Chavez, *Introducción al Tratamiento Digital de Imágenes*.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

- [13] S. Mallat, *A Wavelet Tour of Signal Processing, Second Edition, (Wavelet Analysis and Its Applications)* Academic Press, 1999.
- [14] Mischa Schwartz, *Mobile Wireless Communications* (Cambridge University Press 2005)
- [15] C. Sidney Burrus, Ramesh A. Gopinath & Haitao Guo, *Introduction to wavelets and wavelet transforms A Primer* (Rice University DSP Group)
- [16] Minh N. Do and Martin Vetterli. "The *contourlet* Transform: An Efficient Directional Multiresolution Image Representation" *IEEE Transactions on Image Processing*.
- [17] Minh N. Do, *Directional Multiresolution Image Representations*, Ph.D. Thesis, October 23, 2001.
- [18] P. J. Burt and E. H. Adelson. "The Laplacian pyramid as a compact image code" *IEEE Trans. Communications*, pp. 532-540, April 1983.
- [19] J. Kovacevic and M. Vetterli. "Nonseparable multidimensional perfect reconstruction filter banks and *wavelet* bases for R^n " *IEEE Transactions on Information Theory, Special Issue on Wavelet Transform and Multiresolution Signal Analysis*, 38(2):533-555, March 1992
- [18] M. Jayalakshmi, S. Merchant, and U. B. Desai, "Blind Watermarking in *contourlet* Domain with Improved Detection" In *IIH-MSP*, pp. 449-452 (IEEE Computer Society, 2006).
- [19] H. Malvar and D. Florêncio, "Improved spread spectrum: a new modulation technique for robust watermarking," *IEEE Transactions on Image Processing* 51, 898-905 (2003).
- [20] Mazay Jiménez Salinas, *Marca de Agua Invisible en el Dominio de la Transformada Contourlet, Utilizando una Técnica de Espectro Esparcido para Inserción de la Marca y un Detector Correlacionado para Detección de la Marca*, Tesis de licenciatura, Julio 2008.
- [21] Aper Koz, *Digital Watermarking Based on Human Visual System*, Ph.D. Thesis, September 2002.

Anexo 1

Pseudocódigo de los algoritmos 1 y 2 definidos en la sección 4.6 y pseudocódigo del algoritmo que realiza el cómputo del SSIM. Las variables se muestran en color rojo y las funciones en color azul.

Anexo 1.1 Pseudocódigo del algoritmo 1

Inicio

```

procedimiento // Definir el procedimiento de inserción a utilizar
ataque // Definir el ataque a realizar así como sus características
clave // Definir la clave para la generación de la secuencia pseudoaleatoria
wm // Leer la marca de agua binaria, convertir la marca de agua binaria de matriz a vector
im // Leer imagen original
gama // Definir el factor de multiplicación
BC = descContourlet(im) // Realizar la descomposición contourlet de la imagen im
BCWM = Incrustarwm(BC, procedimiento, gama, clave) // Incrustar la marca de agua
im2 = sintesisContourlet(BCWM) // Reconstruir la imagen a partir de los coeficientes contourlet
marcados
PSNR = psnr(im, im2) //Calcular la PSNR entre la imagen original y la imagen marcada
SSIM = ssim(im, im2) //Calcular el SSIM entre la imagen original y la imagen marcada
Si ataque = JPEG, entonces
    Realizar compresión JPEG de la imagen marcada
Si ataque = ruido "sal y pimienta", entonces
    Atacar imagen con ruido "sal y pimienta"
Si ataque = ruido Gaussiano, entonces
    Atacar imagen con ruido Gaussiano
Si ataque = recorte, entonces
    Recotar imagen
Si ataque = filtrado
    Realizar filtrado paso bajas de la imagen
Si ataque = 0
    No realizar ataque alguno
wmrec = recwm(im2, procedimiento, clave) // Recuperar la marca de agua de la imagen marcada y
reacomodar la marca de agua recuperada de vector a matriz
Mostrar la marca de agua recuperada
BER=biterr(wm, wmrec) //Calcular la tasa de bits en error de la marca de agua recuperada respecto a
la original
Fin

```



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Anexo 1.2 Pseudocódigo del algoritmo 2

Inicio

```

procedimiento // Definir el procedimiento de inserción a utilizar
ataque // Definir el ataque a realizar así como sus características particulares
clave // Definir la clave para la generación de la secuencia pseudoaleatoria
wm // leer la marca de agua binaria
decisión // Definir el tipo de decisión a utilizar en la decodificación de la marca de agua
convertir la marca de agua binaria de matriz a vector
im // leer la imagen original
gama // Definir el factor de multiplicación
wmc = codifConv(wm) //Realizar la codificación convolucional de la marca de agua
BC = descContourlet(im) //Realizar la descomposición contourlet de la imagen original
BCWM = Incrustarwm(BC, procedimiento, gama, clave) Incrustar la marca de agua codificada
im2 = sintesisContourlet(BCWM) //Reconstruir imagen a partir de los coeficientes contourlet
marcados
PSNR = psnr(im, im2) //Calcular la PSNR entre la imagen original y la imagen marcada
SSIM = ssim(im, im2) //Calcular el SSIM entre la imagen original y la imagen marcada
Si ataque = JPEG, entonces
    Realizar compresión JPEG de la imagen marcada
Si ataque = ruido "sal y pimienta", entonces
    Atacar imagen con ruido sal y pimienta
Si ataque = ruido Gaussiano, entonces
    Atacar imagen con ruido Gaussiano
Si ataque = recorte, entonces
    Recotar imagen
Si ataque = filtrado
    Realizar filtrado paso bajas de la imagen
Si ataque = 0
    No realizar ataque alguno
wmrec = recwm(im2, procedimiento, clave) //Recuperar la marca de agua codificada de la imagen
marcada
wmrec = decodeViterbi(wmrec, decisión) //Realiza la decodificación de la marca de agua
recuperda
Reacomodar la marca de agua recuperada en forma de matriz
Mostrar marca de agua recuperada
BER=biterr(wm, wmrec) // Calcular tasa de bits en error de marca de agua recuperada respecto a la
original
Fin

```


Anexo 1.3 Funciones utilizadas en los algoritmos 1 y 2

función descContourlet(im)

Realiza la descomposición contourlet de la imagen **im**

Regresa **BC**, conjunto de matrices que contienen las bandas derivadas de la descomposición contourlet

función Incrustarwm(BC, procedimiento, gama, clave)

si **procedimiento** =1, entonces

Incrustar la marca de agua mediante la técnica de espectro disperso mejorado en la banda Y, utilizando el valor gama y la secuencia pseudoaleatoria generadas por la clave.

si **procedimiento** =1, entonces

Incrustar la marca de agua mediante la técnica de espectro disperso mejorado en las subbandas Z3, Z4, Z7 y Z8, utilizando el valor gama y la secuencia pseudoaleatoria generadas por la clave.

si **procedimiento** =3, entonces

Incrustar la marca de agua mediante la técnica de espectro disperso mejorado en las subbandas Y1, Y4, Y5, Y8, Z9, Z16, utilizando el valor gama y la secuencia pseudoaleatoria generadas por la clave.

Regresa **BCWM**, conjunto de matrices que contienen las bandas contourlet marcadas con la marca de agua

función sintesisContourlet(BCWM)

Realiza la reconstrucción de la imagen original a partir de las bandas contourlet marcadas regresa **im2**, imagen marcada

función recwm(im2, procedimiento, clave)

Realiza la recuperación de la marca de agua incrustada en las bandas contourlet definidas por procedimiento de la imagen **im2**. Para hacer la recuperación reproduce la secuencia pseudoaleatoria generada a partir de **clave**.

Regresa **wmrec**, vector binario que contiene los bits incrustados en las bandas contourlet de la imagen.

función codifConv(wm)

Realizar la codificación convolucional de la marca de agua regresa **wmc**, la marca de agua codificada

función `decodeViterbi(wmrec, decisión)`

Si `decisión` = dura, entonces

Realiza la decodificación de la marca de agua utilizando decisión dura

Si `decisión` = suave, entonces

Realiza la decodificación de la marca de agua utilizando decisión suave con 8 niveles de decisión

regresa `wmrec`, la marca de agua decodificada

función `biterr(wm, wmrec)`

Calcular tasa de bits en error de marca de agua recuperada respecto a la original

regresa `BER`, la tasa de bits en error en la marca de agua recuperada

función `psnr(im, im2)`

Calcula la PSNR entre la imagen original y la imagen marcada

Regresa `PSNR`.

función `SSIM(x, y)`

Computar el promedio de intensidad sobre ventanas de tamaño 11×11 para la imagen `x` y la imagen `y`

Computar la desviación estándar sobre ventanas de tamaño 11×11 para la imagen `x` y la imagen `y`.

Computar la covarianza entre la imagen `x` y la imagen `y`

Evaluar el índice `SSI` de similaridad estructural para cada una de las ventanas

Hacer un promedio del índice `SSI` de cada una de las ventanas de tamaño 11×11 píxeles. El promedio es `SSIM`

Regresa `SSIM`