



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

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

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN TELECOMUNICACIONES

PRESENTA:
MAZAY OSWALDO JIMÉNEZ SALINAS

DIRECTOR DE TESIS:
DR. FRANCISCO JAVIER GARCÍA UGALDE



MÉXICO,D.F.

JULIO 2008

JURADO ASIGNADO

Dr. Boris Escalante Ramírez, Presidente

Dr. Francisco J. García Ugalde, Vocal

Dr. Miguel Moctezuma Flores, Secretario

Dr. Víctor García Garduño, 1^{er} Suplente

M.I. Federico Vargas Sandoval, 1^{do} Suplente

RESUMEN

En este trabajo se presentan dos algoritmos de marca de agua en el dominio de la transformada *contourlet*. El primer algoritmo [1] consiste en utilizar una técnica de espectro disperso aditiva para insertar la marca y un detector basado en correlación es utilizado para detectar las secuencias pseudoaleatorias incrustadas. El segundo algoritmo(propuesto) consiste en utilizar un técnica de espectro disperso mejorado aditiva para insertar la marca y la detección se lleva acabo con un detector basado en correlación, pero su umbral de detección es mejorado bajo el criterio de Neyman-Pearson. Los logos recuperados proveen la autenticidad de la imagen. La similitud del logo insertado y el original, se mide mediante una técnica de correlación. La robustez de los métodos propuestos se mide contra diferentes tipos de ataques realizados por el programa StirMark 4.

AGRADECIMIENTOS

A mi maestro, quien siempre ha estado a mi lado y aunque por momentos me aleje de él, él nunca se ha separado de mí.

A la Universidad Nacional Autónoma de México, que a través de la Facultad de Ingeniería me ha dado una formación impecable.

Al Dr. Francisco Javier García Ugalde, a quien le agradezco por su valioso tiempo, aportaciones a este trabajo y por siempre alentarme a continuar con este trabajo.

A mis sinodales Dr. Boris Ramírez Escalante, Dr. Miguel Moctezuma Flores, Dr. Víctor García Garduño y M.I. Federico Vargas Sandoval, por su tiempo, correcciones y valiosas aportaciones a este trabajo.

Al Ing. Víctor Sánchez Esquivel, por sus valiosos consejos y alentarme a hacer las cosas siempre con un "poco de interés".

A Hector, Dulce y Jaime por brindarme siempre una amistad sincera.

Al Ing. Leobardo Salinas Ferrer, por sus valiosos consejos y por siempre alentarme a alcanzar mis metas.

Finalmente quiero dedicar este trabajo a mis padres, quienes siempre han estado a mi lado y su apoyo y consejos son los que me han traído hasta aquí.

Índice general

Tabla de Contenidos	IX
Lista de Figuras	XIII
Lista de Tablas	XIX
INTRODUCCIÓN	XXI
1. Antecedentes	1
1.1. Introducción	1
1.2. Criptografía, Steganografía y Marca de Agua	2
1.2.1. Criptografía	2
1.2.2. Steganografía	3
1.2.3. Marca de Agua	3
1.3. Llaves	4
1.4. Tipos de marca de agua	4
1.5. Sistemas de marca de agua	4
1.6. Distorsiones y Ataques	5
1.6.1. Ruido Aditivo	5
1.6.2. Filtrado	5
1.6.3. Recorte	6
1.6.4. Compresión	6
1.6.5. Rotación y Escalamiento	6
1.6.6. Promedio Estadístico	7
1.6.7. Múltiples Marcas de Agua	7
1.7. Aplicaciones	7
1.7.1. Marcas de agua como firmas	7
1.7.2. Marcas de agua para autenticación	7
1.7.3. Marcas de agua para monitoreo de transmisiones	8
1.7.4. Marcas de agua para huellas digitales	8
1.7.5. Marcas de agua para control de copias	8
1.8. Esbozo de la Tesis	8
2. Marca de Agua Digital	11
2.1. Introducción	11
2.2. Técnicas básicas en la marca de agua	12

2.2.1.	Marca de agua visible	12
2.2.2.	Marca de agua invisible	12
2.2.3.	Marca de agua en el dominio espacial	15
2.2.4.	Marca de agua en el dominio de la transformada	15
2.3.	Sistema de marca de agua	16
2.4.	Propiedades de la marca de agua	19
3.	Herramientas para la representación de imágenes	21
3.1.	Introducción	21
3.2.	Transformada de Fourier	22
3.3.	STFT Short Time Fourier Transform	23
3.4.	Wavelets	24
3.4.1.	Transformada Continua Wavelet CWT	24
3.4.2.	Transformada Discreta Wavelet DWT	24
3.4.3.	Transformada Discreta Wavelet en dos Dimensiones DWT2	26
3.5.	Transformada Curvelet	31
3.6.	Transformada Contourlet	32
3.6.1.	Descomposición Multiescala	33
3.6.2.	Descomposición Direccional	35
3.6.3.	Descomposición Multiescala y Direccional	38
3.6.4.	Análisis Multiresolución	39
3.7.	Descripción de algoritmos de inserción de la marca de agua	50
3.7.1.	Algoritmo 1	50
3.7.2.	Algoritmo 2(Propuesto)	53
4.	Resultados	59
4.1.	Introducción	59
4.2.	Parámetros Importantes	60
4.2.1.	Histograma	60
4.2.2.	Varianza	60
4.2.3.	Relación señal pico a ruido	60
4.3.	Resultados de las simulaciones para la imagen "lena"	61
4.3.1.	Histograma	61
4.3.2.	Variación de la PSNR	66
4.3.3.	Prueba de compresión JPEG	74
4.3.4.	Prueba de Ruido	82
4.3.5.	Prueba de Filtro de Mediana	84
4.3.6.	Prueba de Recorte	87
4.3.7.	Prueba de Rotación y escalamiento	92
4.4.	Resultados de las simulaciones para la imagen "barbara"	94
5.	Conclusiones	107
5.1.	Trabajo a futuro	108
	Bibliografía	111

A. Código de los algoritmos implementados	115
A.1. Código de Algoritmo 1	115
A.2. Código de Algoritmo 2	123
Índice	130

Índice de figuras

1.1.	Sistema de marcado de agua genérico	2
1.2.	Criptografía para comunicaciones seguras	3
2.1.	Ejemplo de marca de agua visible.	13
2.2.	Ejemplo de marca de agua invisible.	14
2.3.	Sistema de marca de agua informado	18
2.4.	Sistema de marca de agua ciego	18
3.1.	Ejemplo de compresión de imágenes, de izquierda a derecha; Lena original, Lena comprimida con un radio 43:1 con estándar JPEG, Lena comprimida con un radio 43:1 con estándar JPEG2000.	22
3.2.	Principio de incertidumbre de Heisenberg, representación del esparcimiento de la energía de dos átomos de Gabor [2]	23
3.3.	Proceso de descomposición	25
3.4.	Proceso de reconstrucción	26
3.5.	Ejemplo codificación de los coeficiente Wavelet por subbandas	27
3.6.	Ejemplo descomposición Wavelet en un nivel.	28
3.7.	Ejemplo descomposición Wavelet en tres niveles.	29
3.8.	Las wavelets fallan en la captura de regularidades geométricas en imágenes [3].	29
3.9.	Las wavelets no detectan la diferencia entre estas dos imágenes [3].	30
3.10.	Relación de escalamiento anisotrópico para curvas. El soporte rectangular de las funciones base que se ajustan a la curva exhibe una relación cuadrática $ancho \propto longitud^2$. [3]	32
3.11.	Dos esquemas de descomposición contourlet. (a)Esquema original basado en la transformada de bloque Ridgelet que es aplicada a cada una de las imágenes subbanda. (b) Esquema de banco de filtro propuesto, cada imagen es descompuesta por una estructura doble de banco de filtros [3].	33
3.12.	Ejemplo: Descomposición y reconstrucción de la pirámide laplaciana [3].	34
3.13.	Ejemplo: Desorden de frecuencias debido al sobremuestreo del canal pasoaltas. <i>Arriba</i> Espectro después de filtrado pasoaltas. <i>Abajo</i> Espectro después de sobremuestreo. Las áreas rellenas indican que las altas frecuencias se doblan dentro de las bajas frecuencias [4].	34
3.14.	Pirámide Laplaciana. (a) Análisis: la salidas son, una versión reducida en resolución c y una diferencia d entre la imagen original de señal y la predicción. El proceso puede ser iterado descomponiendo la versión reducida repetidamente. (b) Síntesis Usual. (c) Esquema de reconstrucción propuesto para la Pirámide Laplaciana [4].	35

3.15. Partición frecuencial del banco de filtros direccional donde $l = 3$ y hay $2^3 = 8$ bandas frecuenciales [4].	36
3.16. Dos posible configuraciones para los filtros en los QFB. (a) Filtros en forma de diamante. (b) Filtros en forma de ventilador [4].	36
3.17. Partición del espectro en dos dimensiones usando el banco de filtro quincunxes en forma de ventilador. Las regiones negras representan la respuesta ideal de cada filtro [3].	37
3.18. Ejemplo de una operación de remuestreo que es usado efectivamente como una opera- ción de rotación para la descomposición del DFB. (a) Imagen "cameraman".(b) Imagen "cameraman"después del remuestreo [4].	37
3.19. Respuesta al impulso de 32 filtros equivalentes para la primera mitad de los canales de un DFB de 6 niveles, los cuales usan filtro Haar. Los cuadros negros y grises corresponden a +1 y -1 respectivamente. Ya que las funciones de base se asemejan a "lineas locales", reciben el nombre de <i>randolets</i> [3].	38
3.20. Banco de filtros direccional piramidal. (a)Diagrama de bloques. Primero, una descompo- sición estándar multiescala dentro de ocho bandas es calculada, donde el canal pasobajas es submuestreado mientras que el pasoaltas no. Después, una descomposición direccio- nal con un DFB es aplicado a cada canal pasoaltas. (b) Resultado de la división de frecuencia, donde el número de direcciones es incrementado con la frecuencia [4].	39
3.21. Ilustración de imágenes base para una contourlet que satisfacen la relación de escala- miento de la curva. De la linea superior a la inferior, la escala es reducida por cuatro mientras el número de direcciones es duplicado [4].	40
3.22. Imágenes base de la PDFB que implementa la transformada curvelet. Estas tienen es- tructura lineal y satisfacen la relación de escalamiento anisotrópico [4].	40
3.23. Subespacios multiescala generados por la pirámide laplaciana [4].	42
3.24. Subespacios multidireccionales generados por el DFB [4].	44
3.25. Subespacios multidireccionales generados por la PDFB [4].	45
3.26. Rejillas de aproximación incrustadas en el dominio espacial para la PDFB que imple- menta la transformada curvelet. Estas son cuatro ilustraciones de los subespacios $W_{j,k}$ representando escalas burdas vs. escalas finas y direcciones horizontales vs. direcciones verticales. Cada subespacio es recorrido por la función prototipo curvelet dada en (3.12). Los intervalos de muestreo se emparejan con la ayuda de la función prototipo, por ejemplo de ancho w largo l , tal que los desplazamientos cubrirán el plano \mathbb{R}^2 [4].	47
3.27. Ejemplo de descomposición contourlet. (a) Imagen peppers con dos niveles de descom- posición de la pirámide laplaciana, con 4 direcciones en el primer nivel de la pirámide y ocho direcciones en el segundo nivel de descomposición de la pirámide. (b) Imagen barbara con el mismo nivel de descomposición y direcciones que la imagen peppers.	48
3.28. Ejemplo de funciones base.	49
3.29. Descomposición Contourlet.	51
4.1. Histograma de la imagen.	62
4.2. Histograma de la imagen.	63
4.3. Histograma de la imagen.	64
4.4. Histograma de la imagen.	65
4.5. Histograma de la imagen.	67
4.6. Gráficos de la variación de la PSNR respecto a α para los diferentes métodos.	68
4.7. Gráficos de la variación de la PSNR respecto a α para los diferentes métodos.	69

4.8.	Gráficos de la variación del coeficiente de correlación respecto a α para los diferentes métodos.	70
4.9.	Gráficos de la variación del coeficiente de correlación respecto a α para los diferentes métodos.	71
4.10.	Gráfico de los logos original y recuperado de la imagen marcada utilizando el primer método. (a) muestra el logo original. (b) muestra la imagen marcada. (c) muestra el logo recuperado. (d) muestra el logo recuperado después de un filtro de mediana y (e) muestra el logo recuperado sin una llave autorizada.	72
4.11.	Gráfico de los logos original y recuperado de la imagen marcada utilizando el segundo método. (a) muestra el logo original. (b) muestra la imagen marcada. (c) muestra el logo recuperado. (d) muestra el logo recuperado después de un filtro de mediana y (e) muestra un logo recuperado sin una llave autorizada.	72
4.12.	Gráfico de los logos original y recuperado de la imagen marcada utilizando el tercer método. (a) muestra el logo original. (b) muestra la imagen marcada. (c) muestra el logo recuperado. (d) muestra el logo recuperado después de un filtro de mediana y (e) muestra un logo recuperado sin una llave autorizada.	73
4.13.	Gráfico de los logos original y recuperado de la imagen marcada utilizando el cuarto método. (a) muestra el logo original. (b) muestra la imagen marcada. (c) muestra el logo recuperado. (d) muestra el logo recuperado después de un filtro de mediana y (e) muestra un logo recuperado sin una llave autorizada.	73
4.14.	Resultados de compresión JPEG utilizando el primer método.	75
4.15.	Resultados de compresión JPEG utilizando el segundo método.	76
4.16.	Resultados de compresión JPEG utilizando el tercer método.	77
4.17.	Resultados de compresión JPEG utilizando el cuarto método.	78
4.18.	(a) Lena marcada con porcentaje de compresión de 80%. (b) Logo recuperado de la imagen con porcentaje de compresión de 80% utilizando método 1.	79
4.19.	(a) Lena marcada con porcentaje de compresión de 20%. (b) Logo recuperado de la imagen con porcentaje de compresión de 20% utilizando método 1.	79
4.20.	(a) Lena marcada con porcentaje de compresión de 80%. (b) Logo recuperado de la imagen con porcentaje de compresión de 80% utilizando método 2.	79
4.21.	(a) Lena marcada con porcentaje de compresión de 20%. (b) Logo recuperado de la imagen con porcentaje de compresión de 20% utilizando método 2.	80
4.22.	(a) Lena marcada con porcentaje de compresión de 20%. (b) Logo recuperado de la imagen con porcentaje de compresión de 20% utilizando método 3.	80
4.23.	(a) Lena marcada con porcentaje de compresión de 80%. (b) Logo recuperado de la imagen con porcentaje de compresión de 80% utilizando método 3.	80
4.24.	(a) Lena marcada con porcentaje de compresión de 20%. (b) Logo recuperado de la imagen con porcentaje de compresión de 20% utilizando método 4.	81
4.25.	(a) Lena marcada con porcentaje de compresión de 80%. (b) Logo recuperado de la imagen con porcentaje de compresión de 80% utilizando método 4.	81
4.26.	Variación del coeficiente de correlación respecto al porcentaje de ruido utilizando método 1.	82
4.27.	Variación del coeficiente de correlación respecto al porcentaje de ruido utilizando método 2.	83
4.28.	Variación del coeficiente de correlación respecto al porcentaje de ruido utilizando método 3.	83
4.29.	Variación del coeficiente de correlación respecto al porcentaje de ruido utilizando método 4.	84
4.30.	(a) Lena marcada con porcentaje de ruido del 10%. (b) Logo recuperado de la imagen con porcentaje de ruido del 10% utilizando método 1.	85

4.31. (a) Lena marcada con porcentaje de ruido del 10%. (b) Logo recuperado de la imagen con porcentaje de ruido del 10% utilizando método 2.	85
4.32. (a) Lena marcada con porcentaje de ruido del 10%. (b) Logo recuperado de la imagen con porcentaje de ruido del 10% utilizando método 3.	85
4.33. (a) Lena marcada con porcentaje de ruido del 10%. (b) Logo recuperado de la imagen con porcentaje de ruido del 10% utilizando método 4.	86
4.34. Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro utilizando método 1.	86
4.35. Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro utilizando método 2.	87
4.36. Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro utilizando método 3.	88
4.37. Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro utilizando método 4.	88
4.38. (a) Imagen marcada después ser aplicada a un filtro de mediana de 3x3. (b) Logo recuperado de la imagen filtrada utilizando método 1.	89
4.39. (a) Imagen marcada después ser aplicada a un filtro de mediana de 3x3. (b) Logo recuperado de la imagen filtrada utilizando método 2.	89
4.40. (a) Imagen marcada después ser aplicada a un filtro de mediana de 3x3. (b) Logo recuperado de la imagen filtrada utilizando método 3.	89
4.41. (a) Imagen marcada después ser aplicada a un filtro de mediana de 3x3. (b) Logo recuperado de la imagen filtrada utilizando método 4.	90
4.42. Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando método 1.	90
4.43. Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando método 2.	91
4.44. Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando método 3.	91
4.45. Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando método 4.	92
4.46. (a) Imagen marcada con 50% de recorte. (b) Logo recuperado de la imagen recortada utilizando método 1.	92
4.47. (a) Imagen marcada con 50% de recorte. (b) Logo recuperado de la imagen recortada utilizando método 2.	93
4.48. (a) Imagen marcada con 50% de recorte. (b) Logo recuperado de la imagen recortada utilizando método 3.	93
4.49. (a) Imagen marcada con 50% de recorte. (b) Logo recuperado de la imagen recortada utilizando método 4.	93
4.50. Ejemplos de ataques geométricos a la imagen de lena.	94
4.51. Gráficos de la variación de la PSNR respecto a α para los diferentes métodos.	96
4.52. Gráficos de la variación del coeficiente de correlación respecto a α para los diferentes métodos.	97
4.53. Gráficos de la variación del coeficiente de correlación respecto al porcentaje de compresión JPEG para los métodos 1 y 2.	98
4.54. Gráficos de la variación del coeficiente de correlación respecto al porcentaje de compresión JPEG para los métodos 3 y 4.	99

4.55. Gráficos de la variación del coeficiente de correlación respecto al porcentaje de ruido agregado a las imágenes marcadas para los métodos 1 y 2.	100
4.56. Gráficos de la variación del coeficiente de correlación respecto al porcentaje de ruido agregado a las imágenes marcadas para los métodos 3 y 4.	101
4.57. Gráficos de la variación del coeficiente de correlación respecto al tamaño del filtro alicado a las imágenes marcadas para los métodos 1 y 2.	102
4.58. Gráficos de la variación del coeficiente de correlación respecto al tamaño del filtro aplicado a las imágenes marcadas para los métodos 3 y 4.	103
4.59. Gráficos de la variación del coeficiente de correlación respecto al porcentaje recortado a las imágenes marcadas para los métodos 1 y 2.	104
4.60. Gráficos de la variación del coeficiente de correlación respecto al porcentaje recortado de las imágenes marcadas para los métodos 3 y 4.	105
4.61. Ejemplos de ataques geométricos a la imagen de barbara.	106

Índice de tablas

4.1. Parámetros importantes de la imagen marcada de 'lena'	61
4.2. Varianza de las imágenes marcadas de "lena".	61
4.3. Parámetros importantes de la imagen marcada 'barbara'.	95

INTRODUCCIÓN

El avance de la tecnología ha dado lugar, que hoy en día las personas pueden intercambiar algún tipo de contenido digital y no exista un control en su distribución, lo que afecta directamente a los propietarios del contenido o a quienes poseen los derechos de autor. En las últimas décadas se han desarrollado técnicas para combatir este problema, una de ellas es el cifrado, sin embargo cuando se descifra la información, el contenido ya no tiene ninguna protección. Otra técnica, es la llamada marca de agua digital la cuál tiene mayores ventajas ya que permanece en el contenido aunque sufra diversas transformaciones, las aplicaciones de esta son muy diversas como firmas, huellas, para monitorear transmisiones, autenticación, etc. Las herramientas matemáticas que se utilizan para el marcado de agua son desde la transformada de Fourier hasta wavelets. Las wavelets han sido en los últimos una herramienta muy eficiente para el procesamiento de imágenes, sin embargo no es tan eficiente para representar orientaciones espaciales en imágenes en dos dimensiones, por lo que han surgido nuevas herramientas para tener una mejor manipulación de la información, tal es el caso de la transformada contourlet [5], la cual fue introducida por Martin Vetterli y Minh Do en el 2002. El algoritmo de marca de agua [1] es tomado como base para el desarrollo de nuestro trabajo, en el se plantea la inserción de la marca en el dominio de la transformada contourlet mediante la modulación de la marca con una técnica de espectro disperso. Se hicieron en [1] algunas modificaciones para mejorar la relación señal pico a ruido. En el desarrollo se presentan las simulaciones de ambos algoritmos de marcado de agua, en tales simulaciones se presentan las pruebas de robustez de los algoritmos a la compresión JPEG, a ruido, a recorte y técnicas de filtrado. Al final se hace un balance considerando las ventajas y desventajas de ambos métodos.

Hipótesis:

Debido a las características que posee la transformada contourlet y el algoritmo de marca de agua [1], la relación señal pico a ruido puede ser mejorada mediante la variación de la energía de las secuencias pseudoaleatorias, sin embargo esto podría traer una disminución de la robustez del algoritmo. Aún así el algoritmo propuesto debiese resistir a los tipos de ataques más comunes.

Objetivo:

El objetivo de esta tesis es explotar las propiedades de la transformada contourlet para desarrollar un algoritmo de inserción de marca de agua en imágenes digitales monocromáticas que permita mantener la mejor relación señal pico a ruido de la imagen marcada, así como una alta robustez contra los tipos de ataques conocidos en la literatura.

Objetivos específicos:

- Entender los diferentes métodos de inserción de una marca de agua en el dominio de la transformada.
- Entender las características que debe poseer una buena marca de agua.
- El trabajo publicado en [1], es tomado como referencia en el desarrollo de la tesis. Se buscará mantener, o mejorar, la mejor relación señal pico a ruido de la imagen marcada, así como una alta robustez a los tipos de ataques conocidos en la literatura.
- Hacer simulaciones y evaluaciones del algoritmo diseñado y comparar los resultados con otros algoritmos que utilizan la misma transformada.

Justificación

El crecimiento de las redes de comunicación digitales y el desarrollo de aplicaciones digitales permiten, el intercambio, manipulación, procesamiento y almacenamiento de información, por lo que personas de

todas partes del mundo puedan intercambiar o redistribuir cualquier tipo de contenido; aunque este posea o no derechos de autor. Por lo que, los autores y distribuidores de contenido necesitan ser capaces de demostrar que son los propietarios legítimos, vigilar el uso de sus productos, y de alguna forma prevenir la copia ilegal de información y la redistribución a terceras personas. En la actualidad, existen diferentes métodos tanto para la inserción de una marca de agua como para la detección de la misma, sin embargo aún no existe ningún algoritmo que permita mantener la mejor calidad visual y que no pierda robustez contra los ataques conocidos en la literatura.

Método

Para realizar la inserción de la marca de agua invisible en el dominio de la transformada contourlet en señales digitales es necesario comprender varios conceptos: tipos de marca de agua que existen, las técnicas básicas de inserción y detección de la marca en el dominio de la transformada, características de una marca de agua al igual que las propiedades de la transformada utilizada. La siguiente etapa consiste en el desarrollo de un algoritmo de inserción de la marca de agua de los datos de interés, y evaluar el desempeño del algoritmo en términos de su relación señal pico a ruido y su robustez a los diferentes tipos de ataques conocidos en la literatura.

Capítulo 1

Antecedentes

1.1. Introducción

El rápido crecimiento de las redes de comunicación digitales y el desarrollo de aplicaciones digitales que permiten la manipulación, procesamiento y almacenamiento de información, ha dado lugar a que personas de cualquier parte del mundo puedan redistribuir cualquier tipo de contenido; aunque este posea o no derechos de autor. Por lo que, los autores y los distribuidores del contenido necesitan ser capaces de demostrar que son los propietarios legítimos, vigilar el uso de sus productos, y de alguna forma prevenir la copia ilegal de información y la redistribución a terceras personas.

Para la solución, aunque parcial, de este problema surgió la *marca de agua digital*, la cual puede ser aplicada a cualquier señal electrónica. Las representaciones electrónicas de música, vídeo y fotografías son los tipos de señales más comunes con marca de agua digital. Para describir estas señales, nos referiremos en particular: a canción, vídeo o imagen como *trabajo*, y al conjunto de todos los posibles trabajos como *contenido*.

La *marca de agua digital* se define como la práctica de alterar imperceptiblemente un trabajo insertando un mensaje en este [6].¹

¹Algunos autores no consideran el término imperceptible como definitivo en la marca de agua digital, debido a que también existen marcas de agua perceptibles pero hemos adoptado esta definición debido a que en el desarrollo de este documento solo nos enfocaremos en marcas de agua digitales invisibles.

Por simplicidad nos referiremos a la marca de agua digital simplemente como *marca de agua*. En general, un sistema de marca de agua consiste en un sistema de incrustación de la marca de agua y un detector de la marca de agua como se muestra en la figura 1.1.

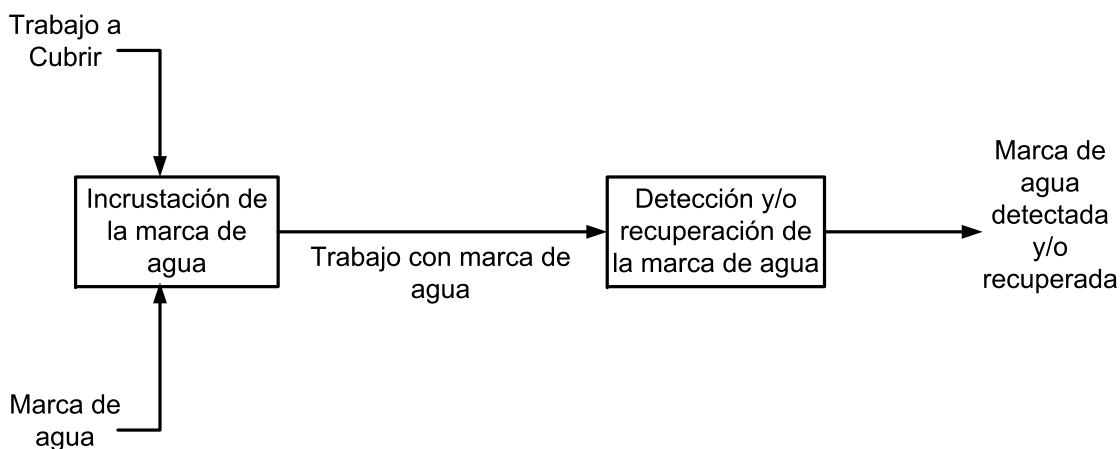


Figura 1.1: Sistema de marcado de agua genérico

La técnica de marca de agua está muy relacionada con la Criptografía y la Steganografía. Sin embargo, existen pequeñas diferencias que son importantes enfatizar de forma que no confundamos la marca de agua con la Criptografía o la Steganografía. Estas características serán explicadas con mayor detalle en la sección subsecuente.

1.2. Criptografía, Steganografía y Marca de Agua

1.2.1. Criptografía

La Criptografía (del griego *kryptos*, «ocultar», y *graphos*, «escribir», literalmente «escritura oculta») se define como el arte de cifrar y descifrar información utilizando técnicas que hagan posible el intercambio de mensajes de manera que sólo puedan ser leídos por las personas a quienes van dirigidos. Un proceso de encriptado y desencriptado se muestra en la figura 1.2, este consiste en un mensaje comúnmente conocido como texto plano, el método de encubrir el texto plano, de tal forma que su información quede oculta es el encriptado, el texto encriptado es también conocido como un texto cifrado. El proceso de volver

el texto encriptado de nuevo a su texto original es el desencriptado. La criptografía solo puede proteger el contenido de nuestro trabajo cuando este está en tránsito, pero una vez desencriptado, el contenido ya no tiene protección alguna.

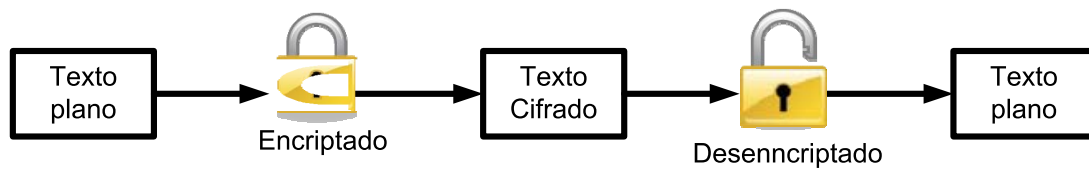


Figura 1.2: Criptografía para comunicaciones seguras

1.2.2. Steganografía

Mientras la Criptografía está relacionada en proteger el contenido de los mensajes, steganografía se refiere a ocultar su existencia. Steganografía se define como el estudio de las técnicas para ocultar la existencia de un mensaje secundario en la presencia de un mensaje principal. El mensaje principal es conocido como el portador de señal o mensaje portador; el mensaje secundario se refiere a la carga útil de la señal o mensaje de carga útil [7].

1.2.3. Marca de Agua

La Steganografía y la marca de agua describen métodos para incrustar información de manera transparente en un portador de la señal. Mientras la steganografía establece un canal de información encubierto para conexiones punto a punto, la marca de agua no necesariamente oculta el hecho de que la transmisión de información sea secreta a terceras personas. Además de la preservación de la calidad de la señal portadora, en general, una marca de agua tiene requerimientos adicionales de robustez contra manipulaciones destinadas a retirar o extraer de manera no autorizada la marca de agua del trabajo portador de

información.

1.3. Llaves

En ocasiones para agregar mayor seguridad se suelen utilizar llaves, estas se dividen en dos:

- *Llaves para cifrado* se utilizan para controlar la encriptación de un mensaje.
- *Llaves para marca de agua* se utilizan para controlar el proceso de inserción y detección de una marca de agua.

En general cuando hagamos mención al termino llave nos referiremos a la llave para marca de agua.

1.4. Tipos de marca de agua

- *Marca de agua robusta* este tipo de marcas son diseñadas para resistir ataques intencionales o no intencionales.
- *Marca de agua frágil* este tipo de marcas son incrustadas con muy baja robustez. Por lo que, pueden ser destruidas con mínimas manipulaciones.
- *Marca de agua publica y privada* este tipo de marcas son diferenciadas por el tipo de llave utilizada en el proceso de incrustación y recuperación de la marca de agua. Si la llave es conocida, el tipo de marca de agua es referido como publica, pero si la llave no es conocida entonces es referida como una marca de agua privada.

El desempeño de este tipo de marca de agua dependerá del tipo de aplicación que utilicemos para cada tipo de marca de agua.

1.5. Sistemas de marca de agua

Los sistemas de marca de agua se clasifican como:

- *Sistema de marca de agua ciego*, cuando el proceso de detección y/o recuperación de la marca se hace sin conocimiento del trabajo original y/o de la marca de agua.
- *Sistema de marca de agua informado*, cuando el proceso de detección y/o recuperación de la marca se realiza al menos con el trabajo original o la marca de agua.

1.6. Distorsiones y Ataques

En la práctica, un trabajo con marca de agua puede ser alterado intencionalmente o no intencionalmente, por lo que el sistema de marcado de agua aún después de ser alterado debe ser capaz de detectar y extraer la marca de agua. Este tipo de distorsiones deben ser tales que no exista una degradación excesiva, ya que de otro modo nuestro trabajo será inservible. Los tipos de ataques más comunes se presentan a continuación, algunos de ellos pueden ser intencionales o no intencionales dependiendo de la aplicación.

1.6.1. Ruido Aditivo

Este se ocasiona por el uso de ciertas aplicaciones o procesos, por ejemplo en el proceso de conversión de señales analógicas a digitales y digitales a analógicas o por errores ocasionados durante la transmisión. Sin embargo, un atacante puede introducir formas de ruido que son imperceptibles y afectar el umbral en el que el detector de la marca de agua trabaja.

1.6.2. Filtrado

El tipo de filtrado más común es el filtrado paso bajas, no introduce degradaciones considerables en trabajos que poseen una marca de agua, pero puede afectar considerablemente el desempeño, si la técnica utilizada para incrustar la marca de agua es por ejemplo de espectro disperso (Spread-Spectrum); ya que con esta técnica se esparce el contenido en altas frecuencias espectrales que no son despreciables para nuestra aplicación.

1.6.3. Recorte

Este es uno de los ataques más comunes, en el que en la mayoría de los casos el atacante está interesado en una pequeña parte del trabajo, tales partes pueden ser de una cierta imagen o secuencias de un vídeo. Por lo que, la marca de agua necesita ser esparcida a través de todo el trabajo, de forma que este tipo de ataque no afecte al trabajo.

1.6.4. Compresión

Este por lo general es un ataque no intencional el cual aparece muy seguido en aplicaciones multimedia. Prácticamente todo los trabajos que son actualmente distribuidos por Internet han sido comprimidos. Si la marca de agua es requerida para resistir diferentes niveles de compresión, en general es aconsejable realizar la incrustación de la marca de agua en el mismo dominio donde se lleva a cabo la compresión. Por ejemplo en el dominio de la transformada discreta coseno (DCT²) la marca de agua de la imagen es más robusta a la compresión JPEG³ que si se hubiese incrustado en el dominio espacial.

1.6.5. Rotación y Escalamiento

Este ha sido uno de los más grandes problemas en el marcado de agua, en particular a causa de su éxito en imágenes fijas. La detección y extracción basada en correlación falla cuando la rotación o el escalamiento son realizados en imágenes con marca de agua porque la marca de agua incrustada no comparte el mismo patrón espacial después de haber sido modificada. Obviamente, sería posible hacer búsquedas exhaustivas en diferentes ángulos de rotación y escalamientos con diferentes factores hasta que una pequeña correlación es encontrada. En aplicaciones de audio que poseen una marca de agua, también es relativamente fácil llevar a cabo una transformación no lineal del tiempo, que haga considerablemente difícil la detección de la marca de agua.

²De sus siglas en Inglés Discrete Cosine Transform

³Joint Photographic Experts Group

1.6.6. Promedio Estadístico

Un atacante puede estimar la marca de agua y entonces retirarla del trabajo restando la marca estimada. Esto es peligroso si la marca no depende substancialmente de la información. De modo que con diferentes trabajos que poseen una marca de agua sería posible mejorar la marca de agua estimada con un simple promedio.

1.6.7. Múltiples Marcas de Agua

Un atacante puede incrustar una marca de agua en un trabajo que ya posee una marca de agua, y posteriormente reclamar su propiedad.

1.7. Aplicaciones

1.7.1. Marcas de agua como firmas

El imprimir una leyenda de propiedad intelectual⁴ en un trabajo tiene muchas limitantes, debido a que una persona podría, por ejemplo en el caso de una imagen recortar una porción de esta en la que no se incluya la leyenda de propiedad intelectual. En este caso las marcas de agua pueden ser utilizadas como firmas de derecho de autor o propiedad. En esta aplicación se incrusta la identidad del titular de derechos de autor del trabajo como una marca de agua, la cual deberá permanecer aun cuando esta ya haya sido modificada de diferentes formas.

1.7.2. Marcas de agua para autenticación

Este tipo de marca de agua puede ser utilizada para aplicaciones en donde es muy importante verificar la legitimidad del contenido. En este caso se incrusta la información en el trabajo de forma que se pueda corroborar más tarde que este no ha sido alterado.

⁴Para indicar la propiedad intelectual de un trabajo se suele imprimir un símbolo; para trabajos visuales se utiliza el símbolo ©y para trabajos de audio el símbolo ®.

1.7.3. Marcas de agua para monitoreo de transmisiones

Las marcas de agua pueden ser utilizadas para la identificación de cuándo y dónde un trabajo está, o ha sido difundido, mediante el reconocimiento de las marcas de agua incrustadas en ellos. Sin embargo en algunas ocasiones la marca de agua puede degradar la calidad visual o audible del trabajo.

1.7.4. Marcas de agua para huellas digitales

Esta marca de agua es utilizada para rastrear la fuente de una copia ilegal. Cada copia disponible posee una marca de agua con una secuencia de bits única. Por lo que, si se hace una copia ilegal del trabajo puede ser fácilmente rastreable, ya que cada copia original tiene una única secuencia de bits incrustados en él.

1.7.5. Marcas de agua para control de copias

Para prevenir que un trabajo sea copiado, una marca de agua puede ser incrustada en él, de forma que cuando esté siendo copiada la marca de agua de un mensaje al dispositivo de grabado, este detendrá o permitirá su copia dependiendo de las reglas de uso determinadas por su propietario. Es indispensable que el dispositivo de grabado posea la marca de agua en su sistema con anterioridad, de forma que pueda identificar la marca y actuar acorde a esta.

1.8. Esbozo de la Tesis

La tesis esta organizada como sigue. En el **Capítulo 2** se explicarán las técnicas básicas que existen en la marca de agua, que tipos de marca de agua hay, además de la propiedades que debe poseer un marca de agua para ser considerada como una "buena" marca de agua. El **Capítulo 3** estudia en forma breve el avance de las técnicas para tratamiento de señales desde Fourier hasta Contourlet. La transformada contourlet como se verá más adelante es un herramienta que nos permite visualizar la información de una forma más selectiva, en diferentes niveles de resolución y en múltiples direcciones. De igual forma se explicara el algoritmo de inserción de marca de agua [1], el cual fue tomado como base para el desarrollo de este trabajo, se explotarán sus principales características y se identificarán sus defectos, para construir

un nuevo algoritmo que nos permita mantener o mejorar la relación señal pico a ruido. El **Capítulo 4** expone los resultados obtenidos de las simulaciones del desempeño de los algoritmos, así como parámetros importantes que nos permitan medir las diferencias entre la imagen marcada y la original. Las pruebas de robustez de los algoritmos serán hechas mediante el programa StirMark, que es uno de los programas más utilizados en la literatura del marcado de agua para medir la robustez. El **Capítulo 5** muestra las conclusiones, a partir de los resultados obtenidos de la simulaciones y hace mención de las oportunidades de mejora del trabajo realizado. El **Apéndice A** muestra el código de los algoritmos implementados bajo la plataforma de Matlab.

Capítulo 2

Marca de Agua Digital

2.1. Introducción

En esta capítulo se presentarán los requerimientos para realizar un buen marcado de agua y se discutirán algunas de técnicas básicas en el marcado de agua.

Las marcas de agua se clasifican de acuerdo a la percepción humana como:

- Marca de agua visible
- Marca de agua invisible

y de acuerdo a su esquema de incrustación como:

- Marca de agua en el dominio espacial
- Marca de agua en el dominio de la transformada

Sus principales características de cada una de ellas se explicaran en las secciones subsecuentes.

2.2. Técnicas básicas en la marca de agua

2.2.1. Marca de agua visible

La marca de agua visible por lo general es utilizada en imágenes o video y suele ser un patrón visual o un "logo" el cual es incrustado en alguna esquina de las imágenes como se muestra en la figura 2.1. Las características deseables en este tipo marcas de agua son:

- Es visible, pero no obstructiva, por lo tanto, se confirma directamente al propietario de la marca de agua.
- Difíciles de remover, y
- Se ajusta automáticamente, es decir, varia la intensidad de la marca de agua de forma que no obstruya las regiones importantes de la imagen.

Este tipo de marcas se han utilizado durante un largo tiempo, sin embargo aún queda mucho por hacer en aplicaciones como firmas de derecho de autor, control de copias y autenticación.

2.2.2. Marca de agua invisible

Como su nombre lo indica este tipo de marcas de agua son imperceptibles a simple vista y a comparación de las marca de agua visibles, las marcas de agua invisibles nos ofrecen dos potenciales ventajas:

- Nos son removidas cuando el trabajo es convertido a otros formatos o recortado y
- Sufren las mismas transformaciones que el trabajo en el cual fueron incrustadas.

En la figura 2.2 se muestra un ejemplo de una marca de agua invisible; para inserción de la marca se utilizo una técnica LSB¹ en el domino espacial que consiste en incrustar la marca de agua en los bits menos significativos del trabajo a cubrir.

¹De sus siglas en Inglés Least-significant-bit



(a) Imagen original



(b) Marca de agua



(c) Imagen con marca de agua visible

Figura 2.1: Ejemplo de marca de agua visible.



(a) Imagen original



(b) Marca de agua



(c) Imagen con marca de agua invisible

Figura 2.2: Ejemplo de marca de agua invisible.

2.2.3. Marca de agua en el dominio espacial

El proceso de incrustación de la marca de agua en el dominio espacial consiste en alterar ligeramente los valores de pixel que no son significativos en este dominio para sobreponer la marca de agua. Tiene ventajas como baja complejidad y fácil implementación, sin embargo son no robustas con respecto a ataques intencionales o no intencionales.

2.2.4. Marca de agua en el dominio de la transformada

La marca de agua en el dominio de la transformada es el esquema más utilizado y por lo general el esquema de codificación se refiere a una marca de agua multiplicativa que consiste en incrustar la marca de agua en los componentes más significativos del trabajo si los requerimientos de robustez son el principal objetivo para la protección del trabajo [8].

Transformadas

- *Transformada discreta coseno DCT* la marca de agua en este dominio es una de las más populares debido a que esta transformada es el estándar de codificación de imágenes y video² y por sus características de robustez, compresión y escalamiento. La transformada discreta coseno tipo II bidimensional se presenta en la ecuación 2.1

$$Y(u, v) = \frac{2c(u)c(v)}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \cos\left(\frac{(2i+1)u\pi}{2M}\right) \cos\left(\frac{(2j+1)v\pi}{2N}\right), \quad (2.1)$$

en donde $u = 0, 1, \dots, M-1, v = 0, 1, \dots, N-1$ y

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = 0; \\ 1, & \text{otro caso.} \end{cases}$$

- *Transformada discreta Wavelet DWT* esta técnica también es una de las más utilizadas ya que provee una excelente compactación de energía en tiempo y frecuencia, en la cual la energía tiende a compactarse espacialmente en cada subbanda. Para la transformada discreta wavelet, la relación

²Los estándares JPEG, MPEG-1, MPEG-2, H.261 y H.263.

entre el dominio de las señales espacial-temporal, $f(t)$, y la DWT de $f(t)$, $d(k, l)$, es

$$f(t) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} d(k, l) 2^{-k/2} \psi(2^{-k/2} t - l), \quad (2.2)$$

donde $\psi(\cdot)$ representa la función madre.

- *Transformada Contourlet* es una transformada multiresolución y multiescala; extensión de la transformada Wavelet, que será explicada en el capítulo tres de este trabajo.
- *Espectro disperso* es una técnica utilizada en comunicaciones que consiste en esparcir una señal angosta en un ancho de banda mucho mayor, al igual que la transformada Contourlet, esta técnica será explicada con mayor detalle en el capítulo tres de este trabajo.

2.3. Sistema de marca de agua

Como vimos en el capítulo uno, un sistema de marca de agua está formado por dos etapas: un proceso de incrustación de la marca de agua y un proceso de extracción o detección.

La figura 2.3 muestra un sistema de "marca de agua informado", el cual está formado por dos subsistemas: uno de incrustación de la marca y otro de detección de la marca. El subsistema de incrustación de la marca \hat{O} puede ser analizado como una función de tres variables como se muestra en la ecuación (2.3), donde O es el trabajo a cubrir, X es la marca de agua y $[K]$ es la llave; la llave puede ser opcional en el subsistema de incrustación, sin embargo, el uso de ella nos provee de más seguridad. El proceso de incrustación de la marca se lleva de acuerdo a la ecuación (2.4)

$$\hat{O} = f(O(i, j), X(i, j), [K]) \quad (2.3)$$

$$\hat{O} = O(i, j) + [K]X(i, j) \quad (2.4)$$

donde

$\hat{O}(i, j)$: es el trabajo marcado de dimensiones $i \times j$.

$O(i, j)$: es el trabajo original de dimensiones $i \times j$.

$X(i, j)$: es la marca de agua de dimensiones $i \times j$.

$[K]$: es la llave, $[\cdot]$ significa que puede ser opcional.

y la etapa de detección y/o recuperación de la marca de agua sería en función de las variables que se muestran en la ecuación (2.5).

$$\hat{X} = f(\hat{O}(i, j), O(i, j), [K]) \quad (2.5)$$

donde

$\hat{X}(i, j)$: es la marca de agua recuperada $i \times j$.

$\hat{O}(i, j)$: es el trabajo marcado de dimensiones $i \times j$.

$O(i, j)$: es el trabajo original de dimensiones $i \times j$.

$[K]$: es la llave, $[\cdot]$ significa que puede ser opcional.

La figura 2.4 muestra un sistema de marca de agua ciego, el subsistema de inserción de la marca de agua sería el mismo que en el sistema informado, pero su subsistema de recuperación y/o detección de la marca estaría dado en función de la ecuación (2.6), en el cual el trabajo original no es necesario.

$$\hat{X} = f(\hat{O}(i, j), [K]) \quad (2.6)$$

donde

$\hat{X}(i, j)$: es la marca de agua recuperada $i \times j$.

$\hat{O}(i, j)$: es el trabajo marcado de dimensiones $i \times j$.

$[K]$: es la llave, $[\cdot]$ significa que puede ser opcional.

(2.7)

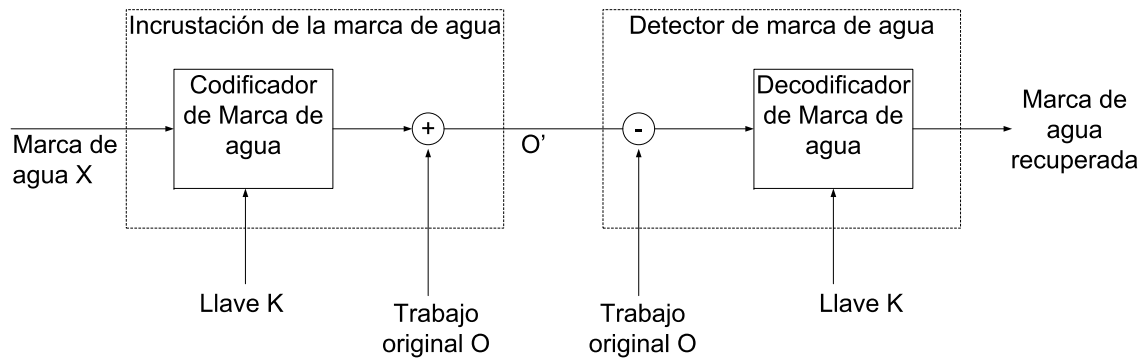


Figura 2.3: Sistema de marca de agua informado

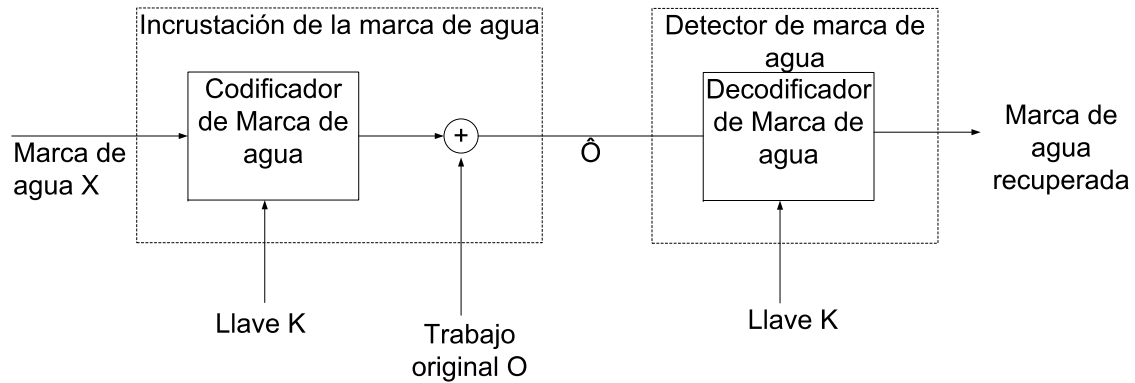


Figura 2.4: Sistema de marca de agua ciego

2.4. Propiedades de la marca de agua

Un sistema de marca de agua puede ser caracterizado por sus propiedades [9], sin embargo, la importancia de cada propiedad depende de los requerimientos de la aplicación y el lugar donde se desempeñará. Las propiedades que debe poseer un sistema de marca de agua son:

- *Efectividad de incrustación* es la probabilidad de detección de la marca de agua inmediatamente después de su incrustación, aunque es deseable tener el 100% de efectividad no siempre es posible debido a que su implementación requiere un alto costo con respecto a otras propiedades.
- *Robustez* es la de detectar la marca de agua después de operaciones comunes en el procesamiento de señales.
- *Fidelidad* se refiere a la similitud perceptual entre la imagen original y las versiones del trabajo marcado.
- *Seguridad* La habilidad de la marca de agua para resistir ataques hostiles.
- *Viabilidad del sistema* se refiere a que tan viable es implementar un sistema de marca de agua, tomando en cuenta costos, tiempo de procesamiento y características del sistema requerido.

Capítulo 3

Herramientas para la representación de imágenes

3.1. Introducción

Durante las últimas décadas se han desarrollado nuevas técnicas para representar las señales de una forma más eficiente; como técnicas se hace referencia a una transformación matemática que provea mayor información acerca de la imagen, la cual no es obvia en su representación original. Existe un sin número de estas transformadas como la Transformada de Fourier, la Short Time Fourier Transform, la Transformada Wavelet, la Transformada Ridgelet [10], la Transformada Curvelet [11], la Transformada Contourlet [5], etc. Unas de las transformadas más populares son la Transformada de Fourier y la transformada Wavelet, las cuales se explicarán en el desarrollo del capítulo. Una de las transformadas más recientes es la transformada *Contourlet* introducida por Minh N. Do y Martin Vetterli en el 2002. Esta transformada hace uso de una transformada tipo Wavelet para detección de bordes y posteriormente realiza una transformada local y direccional para detección de contornos en los segmentos de la imagen.

La transformada contourlet es adoptada para el desarrollo de este trabajo por que ofrece una flexible representación y descomposición direccional de imágenes, otra grande ventaja es que es implementada eficientemente con un banco de filtros iterados que dan lugar a rápidos algoritmos.

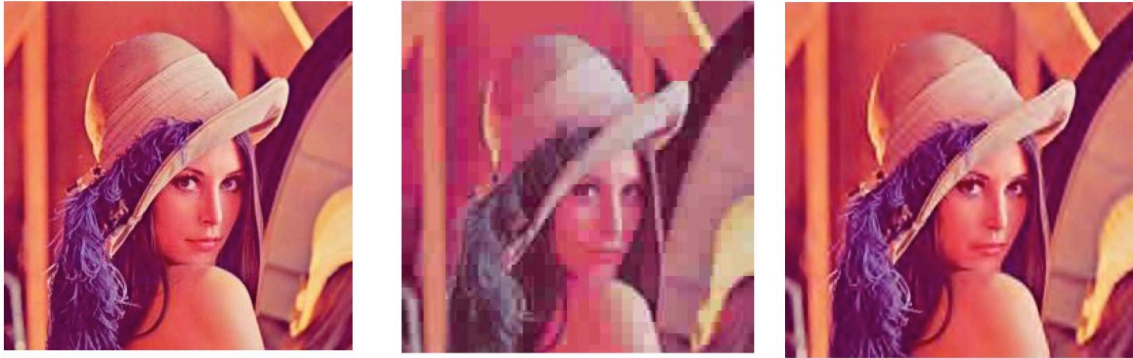


Figura 3.1: Ejemplo de compresión de imágenes, de izquierda a derecha; Lena original, Lena comprimida con un radio 43:1 con estándar JPEG, Lena comprimida con un radio 43:1 con estándar JPEG2000.

3.2. Transformada de Fourier

La transformada de Fourier descompone una señal en sus funciones exponenciales complejas de diferentes frecuencias y se define como

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad \text{Transformada directa}$$

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft} dt \quad \text{Transformada inversa}$$

Esta transformada es reversible, es decir, podemos pasar al dominio de la frecuencia y al dominio del tiempo sin problema alguno. Sin embargo, esta transformada no puede darnos información de ambos dominios a la vez, es decir, cuando estamos en el dominio de la frecuencia no es posible obtener información de la señal en tiempo y viceversa.

Esto no es problema cuando tratamos con señales estacionarias, ya que el contenido frecuencial de estas señales no cambia nunca en el tiempo, sin embargo, estas señales son poco comunes, por lo general siempre tenemos señales no estacionarias en las cuales su información frecuencial sí varía en el tiempo.

La transformada de Fourier solo da información de cuales componentes espectrales existen en una señal.

3.3. STFT Short Time Fourier Transform

Cuando necesitamos conocer la localización de una componente frecuencial en tiempo, es necesario acudir a otro tipo de técnicas que nos provean una representación de las señales tiempo-frecuencia. La STFT provee estas características y consiste en dividir la señal en pequeños segmentos, en donde estos segmentos de la señal pueden considerarse como estacionarios. Con este propósito, se elige una función ventana w , en donde el ancho de la ventana debe ser igual al segmento en el cual la señal es considerada estacionaria. Resumiendo lo anterior la STFT se define como

$$STFT(\tau, f) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-j2\pi ft} dt$$

Con la STFT no solo conocemos que componentes frecuenciales están en la señal, sino también donde están localizados en tiempo. Sin embargo, existe un problema que tiene sus raíces con el principio de incertidumbre de Heinsenberg. Este principio establece que no se puede conocer con precisión infinita la exacta representación de la señal, esto es, no se puede conocer más allá de una cierta precisión que componentes espectrales existen en un instante de tiempo. Pero lo que sí se puede conocer son los intervalos de tiempo en el cual ciertas bandas de frecuencias existen, lo cual es un problema de resolución.

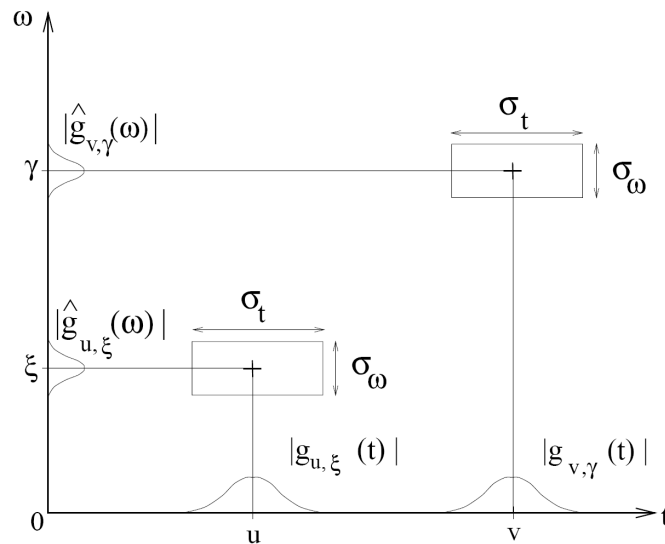


Figura 3.2: Principio de incertidumbre de Heinsenberg, representación del esparcimiento de la energía de dos átomos de Gabor [2]

Como nos podemos dar cuenta, el problema de la STFT es que la resolución esta relacionada con el ancho de la función ventana que sea usado. Este ancho de la ventana se le conoce como el *soporte* de la ventana. Si la función ventana es angosta, la STFT nos dará buena resolución en tiempo, pero mala resolución en frecuencia y si la función ventana es ancha la STFT nos dará buena resolución en frecuencia, pero mala resolución en tiempo.

3.4. Wavelets

3.4.1. Transformada Continua Wavelet CWT

La transformada Wavelet surgió como una solución mejorada al problema de resolución presente en la STFT. La transformada Wavelet es una representación tiempo-escala de la señal original, que se define en tiempo continuo como:

$$CWT(\tau, s) = \Psi(\tau, s) = \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t - \tau}{s} \right)$$

La transformada Wavelet es similar a la STFT, en el sentido de que ambas señales son multiplicadas por una función, como podemos observar de la definición de la transformada Wavelet, la señal original es multiplicada por la función ψ , la cual corresponde a la wavelet madre¹, es una función prototipo que permite generar a partir de ella otras funciones ventana. También podemos observar de la definición, que esta es una función de dos parámetros τ y s , parámetros de translación y escala respectivamente. El parámetro translación se relaciona con la ubicación de la ventana, ya que la ventana es desplazada a través de la señal. El parámetro escala provee en términos de frecuencia de manera general, para bajas frecuencias (altas escalas) una información global de la señal, mientras que en altas frecuencias (bajas escalas) provee información detallada de la señal.

3.4.2. Transformada Discreta Wavelet DWT

Hoy en día las computadoras son ampliamente usadas para hacer la mayoría de los cálculos en un procesamiento. Y la versión de la transformada continua Wavelet no es practica desde el punto de vista

¹El número de funciones wavelet es muy variado, en la actualidad existen wavelets continuas, discretas, ortogonales, bi-ortogonales etc.

que requiere un mayor uso de recursos y tiempo maquina para su cálculo, es por eso que surgió una versión discreta de la transformada Wavelet, la cuál provee una suficiente información de la señal original, con una significativa reducción del tiempo de cálculo.

El origen de la transformada discreta Wavelet (DWT) data de 1976 cuando Croiser, Esteban y Galand desarrollaron una técnica para descomponer señales discretas. Crochiere, Weber y Flanagan hicieron un trabajo similar sobre codificación de señales de voz en ese mismo año. Ellos nombraron su esquema de análisis como codificación por subbandas. En 1983, Burt definió una técnica muy similar a la codificación por subbandas que posteriormente la nombró como codificación piramidal, la cual es también conocida como análisis multiresolución. Después en 1989, Vetterli y Le Gall hicieron algunas mejoras a estos esquemas, removiendo la existencia de redundancia en el esquema de codificación piramidal.

La DWT es obtenida usando técnicas de filtrado. Al proceso de descomponer la señal se le conoce como *análisis* y al proceso de reconstrucción como *síntesis*. Es decir, filtros de diferentes frecuencias de corte son utilizados para analizar la señal en diferentes escalas. Las señales son pasadas a través de una serie de filtros paso-altas para analizar componentes de altas frecuencias, y a su vez filtros paso-bajas son utilizados para analizar componentes de bajas frecuencias.

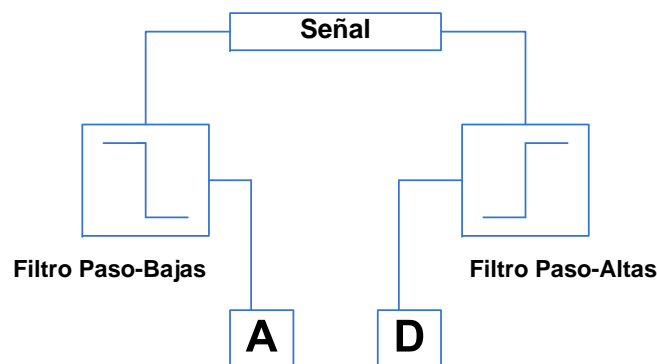


Figura 3.3: Proceso de descomposición

La DWT analiza la señal en diferentes bandas de frecuencia con diferentes resoluciones descomponiendo la señal en una aproximación reducida, e información detallada, como se muestra en la figura 3.3. Y su proceso inverso consiste en reconstruir la señal a partir de estas aproximaciones y detalles, como se

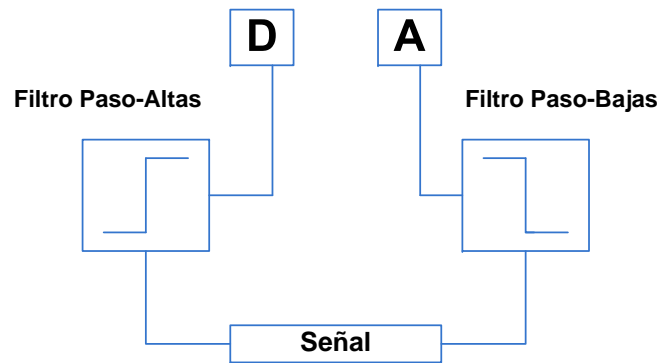


Figura 3.4: Proceso de reconstrucción

muestra en la figura 3.4. Es de notar que el desempeño de este proceso dependerá mucho de la selección de los filtros y por ende de las funciones wavelet..

3.4.3. Transformada Discreta Wavelet en dos Dimensiones DWT2

La DWT en dos dimensiones es ampliamente utilizada en procesamiento de imágenes e inclusive fue adoptada por el estándar JPEG2000 debido a que provee una buena representación de imágenes con pocos coeficientes. La DWT2 descompone una imagen en uno o más niveles, para cada nivel la DWT2 genera una matriz de coeficientes; estos coeficientes a su vez están clasificados en cuatro categorías: aproximaciones, detalles horizontales, detalles verticales y detalles diagonales.

Como podemos observar en la figura 3.6 la aproximación, es la matriz de coeficientes que posee la mayor energía, mientras que los detalles son coeficientes con muy poca energía, ya que estos solo dan detalles acerca de la orientación debido al tipo de filtrado que se hace a la imagen original.

Resumiendo, el éxito de la transformada Wavelet ha sido muy grande principalmente debido a que proveen una buena aproximación no lineal para funciones suaves por tramos en una dimensión (1-D), e incluso debido a su distribución de coeficientes a dado lugar al surgimiento de varios algoritmos para compresión de imágenes, por mencionar algunos: EZW, SPHIT, WDR, ASWDR [12].

Desafortunadamente, para el caso en dos dimensiones (2-D), estas solo son buenas en capturar discontinuidades de dimensión cero o punto, pero las funciones suaves por tramos en dos dimensiones se

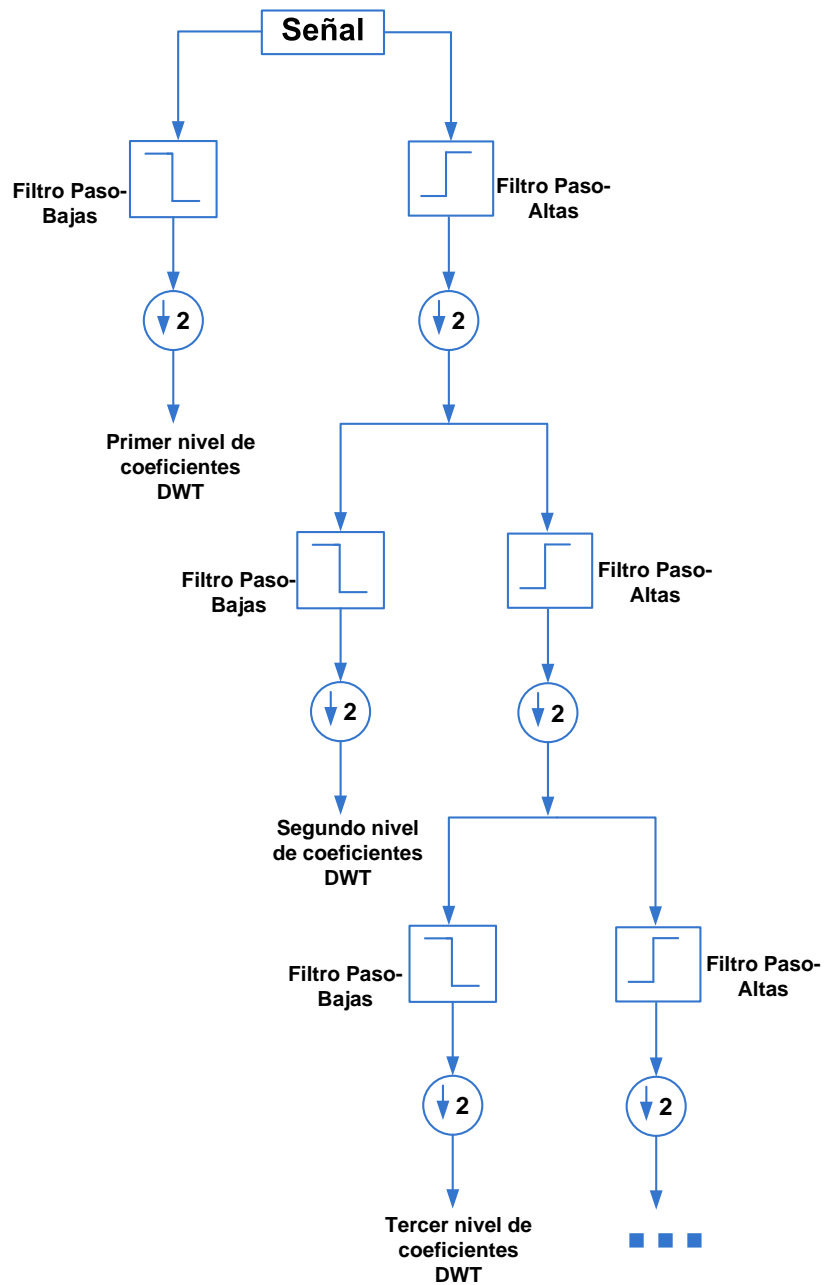
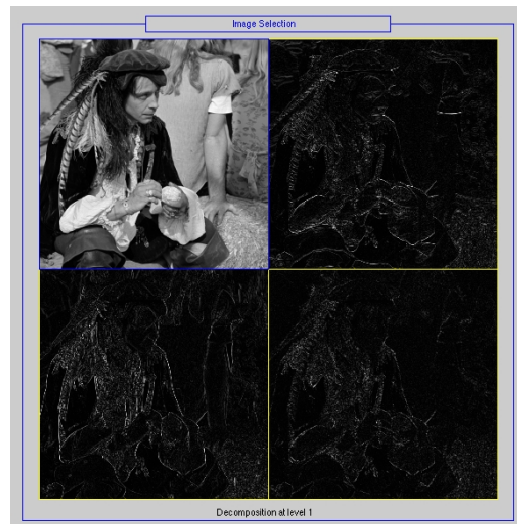


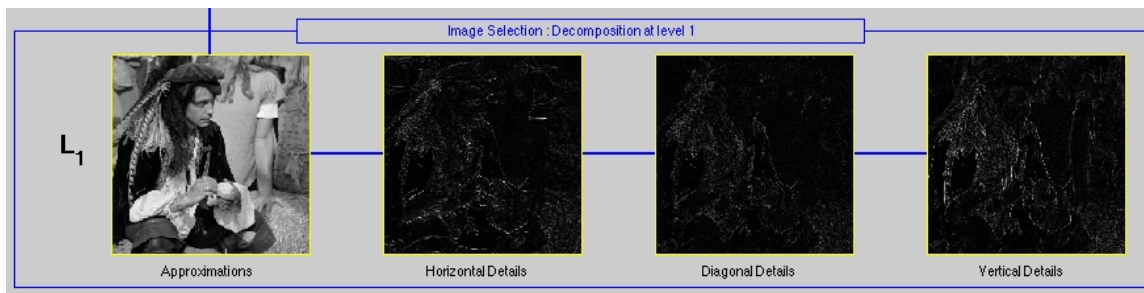
Figura 3.5: Ejemplo codificación de los coeficiente Wavelet por subbandas



(a) Imagen original



(b) Descomposición en el primer nivel



(c) Aproximaciones, detalles horizontales, verticales y diagonales

Figura 3.6: Ejemplo descomposición Wavelet en un nivel.

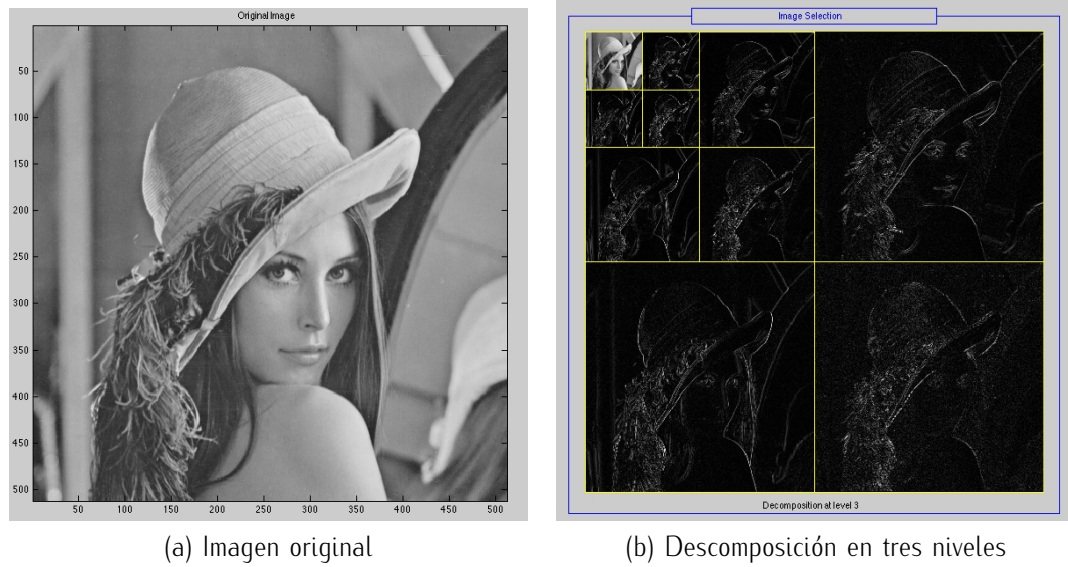


Figura 3.7: Ejemplo descomposición Wavelet en tres niveles.

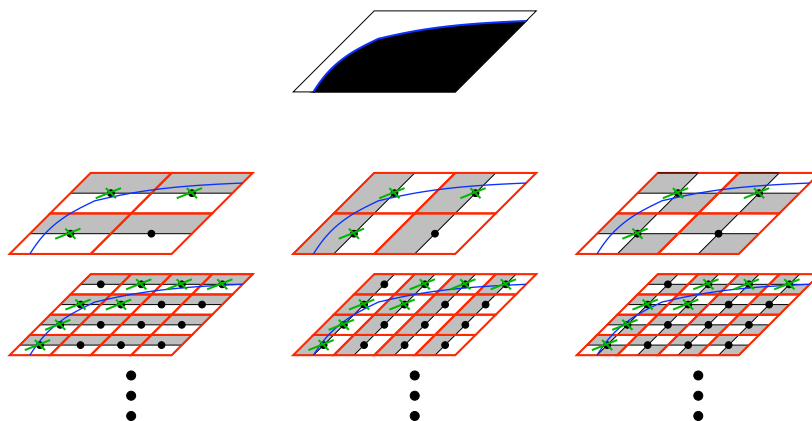
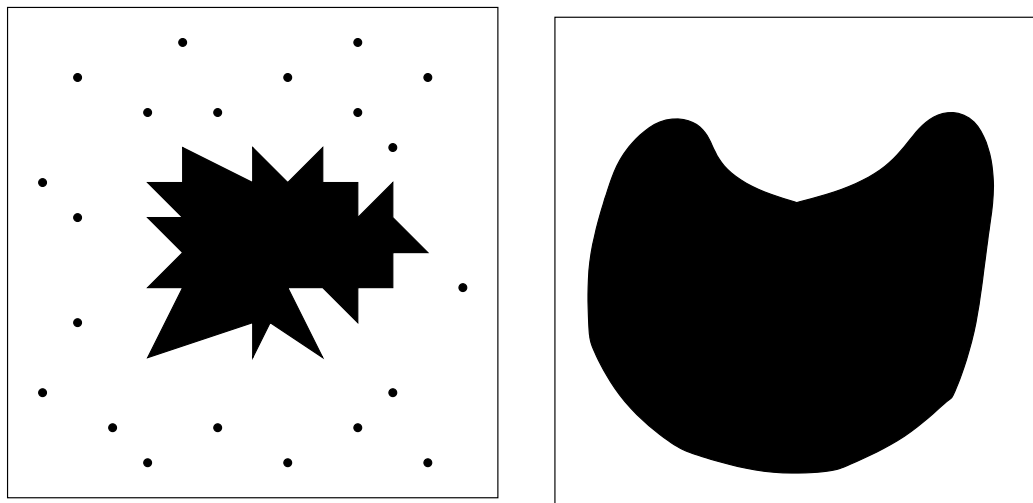


Figura 3.8: Las wavelets fallan en la captura de regularidades geométricas en imágenes [3].

asemejan a imágenes con discontinuidades de una dimensión. Intuitivamente, las wavelets en 2-D obtenidas a partir de un producto tensor de wavelets de una dimensión, son buenas en aislar la discontinuidad a un punto extremo, pero no detectan la suavidad a lo largo del contorno (ver figura 3.8). Esto indica que representaciones más eficaces son necesarias para mayores dimensiones.



(a) Bordes: imágenes punto con discontinuidades

(b) Contornos: Bordes localizados y direcciones regulares

Figura 3.9: Las wavelets no detectan la diferencia entre estas dos imágenes [3].

Minh N. Do [4] en su tesis doctoral propone una lista de cualidades deseables para nuevas representaciones de imágenes, basadas en lo anterior y en el sistema visual humano (HVS²), las cuales se nombran a continuación:

- ★ *Multiresolución*: La representación de imágenes deberá permitir aproximar exitosamente imágenes, empezando de una versión reducida hasta ir a una versión más fina en resolución.
- ★ *Localización*: Los elementos base en la representación deberán estar bien concentrados tanto en el dominio espacial como frecuencial.
- ★ *Muestreo Crítico*: La representación deberá ser con pequeña redundancia.

²De sus siglas en Inglés Human Visual System

- * *Direccionalidad*: La representación deberá contener funciones base orientadas en varias direcciones, mayores a tres direcciones. Las cuales son ofrecidas por las wavelets separables.
- * *Anisotropía*: Capturar contornos suaves en imágenes. La representación deberá contener funciones base con varias formas.

3.5. Transformada Curvelet

Como mencionamos al inicio del capítulo una de las transformadas más recientes es la transformada *curvelet*, introducida por Candès y Donoho [11], los cuales mostraron que su nuevo sistema de representación lograba una óptima representación para funciones suaves por tramos 2-D en \mathbb{R}^2 donde la curva de discontinuidad es una función C^2 .³

La transformada *curvelet* [13] es obtenida por un sistema de filtrado y posteriormente aplicando la transformada "Ridgelet" en intervalos a cada una de las imágenes pasabanda. En \mathbb{R}^2 , ridgelets son constantes a lo largo de líneas $x_1 \cos(\theta) + x_2 \sin(\theta) = cte$ y son wavelets (con escala s) a lo largo de la dirección ortogonal. En el dominio de frecuencia, tal función "ridgelet" es esencialmente localizada en la corona $|\omega| \in [2^s, 2^{s+1}]$ y alrededor del ángulo θ .

En resumen, la descomposición *curvelet* está formada por los siguientes pasos [11]:

1. Descomposición en subbandas del objeto dentro una secuencia de subbandas.
2. Poner en intervalos cada subbanda dentro de bloques de tamaño apropiado, dependiendo de su frecuencia central.
3. Aplicar la transformada "Ridgelet" sobre estos bloques.

El objetivo principal de la transformada *curvelet* es que por intervalos suaves, los segmentos de curvas suaves puedan ser capturadas eficientemente por la transformada "Ridgelet". La descomposición por subbandas es usada para mantener un número de "ridgelets" múltiples escalas, esto por el hecho de que las funciones "ridgelets" dan una escala para cierta subbanda. El tamaño de la ventana y las

³ C^p es el espacio de funciones continuas diferenciables p veces

subbandas de frecuencia estas relacionadas de tal forma que las curvelets tengan soporte obedeciendo a la relación de escalamiento anisotrópico para curvas [11,13]:

$$\text{ancho} \propto \text{longitud}^2, \quad (3.1)$$

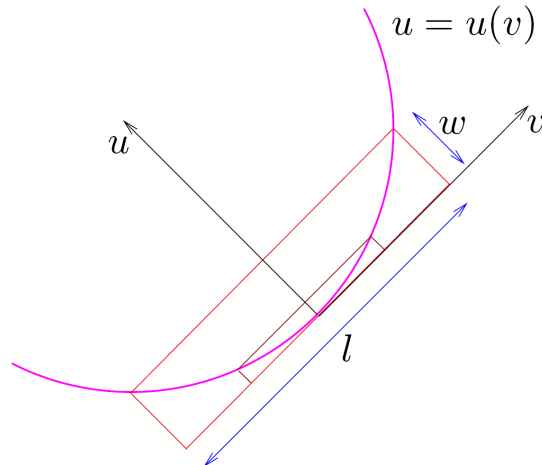


Figura 3.10: Relación de escalamiento anisotrópico para curvas. El soporte rectangular de las funciones base que se ajustan a la curva exhibe una relación cuadrática $\text{ancho} \propto \text{longitud}^2$. [3]

El enfoque original que tuvo la descomposición curvelet [11]trajo algunos problemas en su implementación para aplicaciones prácticas. Primero, su discretización no es tan trivial ya que su construcción original esta definida en el espacio continuo \mathbb{R}^2 y no es posible mantener todas sus características, especialmente en aplicaciones de compresión donde el muestreo crítico es deseable. Segundo, como la transformada curvelet fue definida en el dominio de frecuencia, no es tan trivial conocer como se lleva acabo el muestreo en el dominio espacial y finalmente como es una transformada basada en bloques las imágenes aproximadas poseen este "efecto de bloque".

3.6. Transformada Contourlet

La transformada contourlet como se verá en el desarrollo de la sección es una versión discreta de la transformada curvelet, la cual es una eficiente expansión direccional y multiresolución. La contourlet es construida via *banco de filtros* y puede ser vista como una extensión de *wavelets* con *direccionalidad*. Esta

construcción comienza en el dominio discreto lo cual se presta para desarrollo de algoritmos eficientes, y después converge a una construcción en el dominio continuo. Las varias etapas que conforman la transformada contourlet serán explicadas en las siguientes secciones.

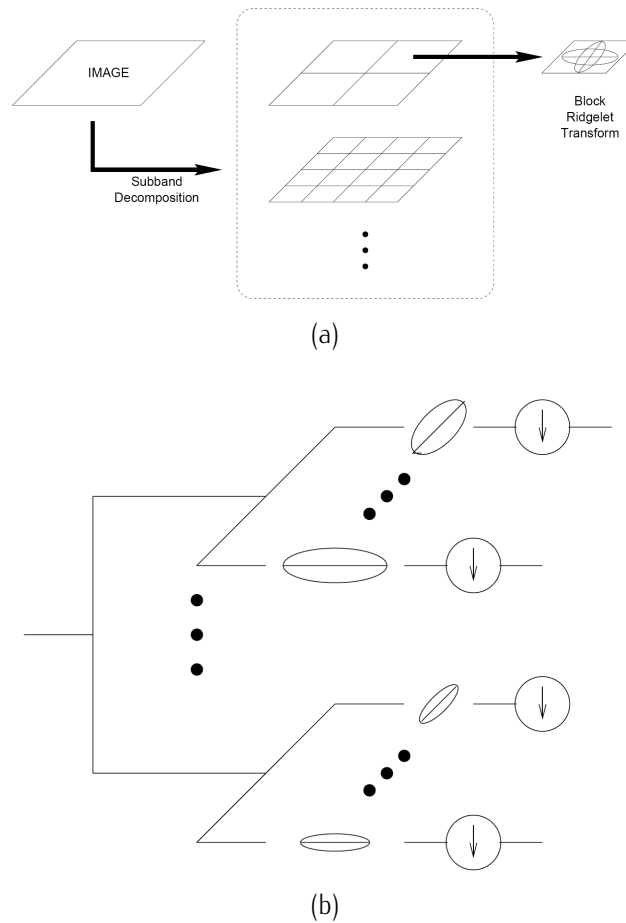


Figura 3.11: Dos esquemas de descomposición contourlet. (a) Esquema original basado en la transformada de bloque Ridgelet que es aplicada a cada una de las imágenes subbanda. (b) Esquema de banco de filtro propuesto, cada imagen es descompuesta por una estructura doble de banco de filtros [3].

3.6.1. Descomposición Multiescala

La forma en que la transformada contourlet obtiene una descomposición multiescala es con el uso de la Pirámide Laplaciana (LP^4) introducida por Burton y Adelson [14]. La LP en cada etapa genera una versión pasobajas reducida en resolución de la versión original y la diferencia entre la original y

⁴De sus siglas en Inglés Laplacian Pyramid

la predicción resulta en una imagen pasobanda, esta proceso puede ser iterado con la versión reducida interpolada.

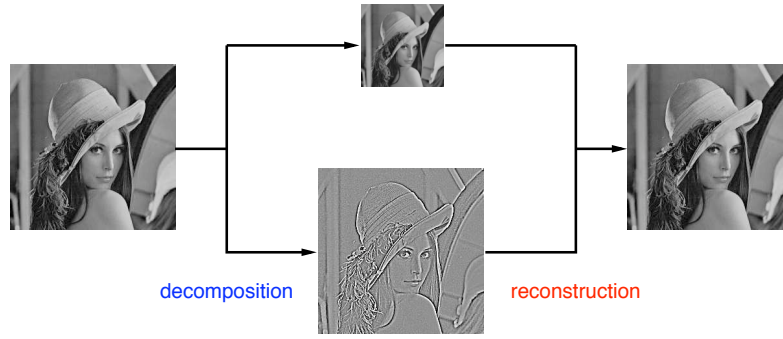


Figura 3.12: Ejemplo: Descomposición y reconstrucción de la pirámide laplaciana [3].

Una desventaja de la LP es que lleva implícito un sobremuestreo, sin embargo, la LP tiene una característica muy importante que es, que en cada nivel de la pirámide esta genera solo una imagen pasobanda(incluso en el caso multidimensional), la cual no tiene desorden de frecuencias. Este desorden de frecuencias ocurre en el banco de filtro wavelet, cuando un canal pasoaltas, después del submuestreo, es doblado dentro de la banda de frecuencias bajas, y su espectro es reflejado. Este efecto es evitado en la LP submuestreando solo el canal pasobajas.

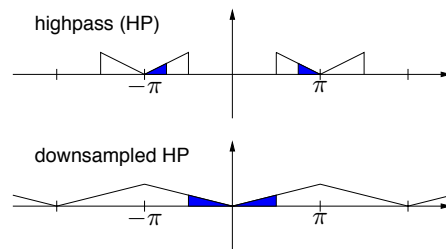


Figura 3.13: Ejemplo: Desorden de frecuencias debido al sobremuestreo del canal pasoaltas. *Arriba* Espectro después de filtrado pasoaltas. *Abajo* Espectro después de sobremuestreo. Las áreas rellenas indican que las altas frecuencias se doblan dentro de las bajas frecuencias [4].

La pirámide laplaciana que utiliza la transformada contourlet, tiene un esquema de reconstrucción diferente al usual (ver figura 3.14c). Minh Do y Martin Vetterli plantean un esquema de reconstrucción

basado en la teoría de marcos y sobremuestreo del banco de filtros. En [15] los autores demuestran que la LP con filtros ortogonales (es decir, $h[n] = g[-n]$ y $g[n]$ es ortogonal a sus translaciones con respecto a la cuadrícula del submuestreo) es un marco ajustado con los límites del marco igual a 1.

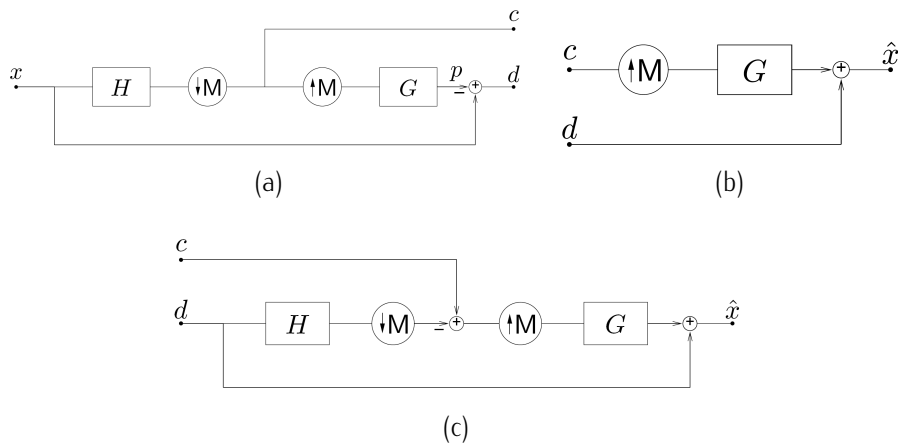


Figura 3.14: Pirámide Laplaciana. (a) Análisis: la salidas son, una versión reducida en resolución c y una diferencia d entre la imagen original de señal y la predicción. El proceso puede ser iterado descomponiendo la versión reducida repetidamente. (b) Síntesis Usual. (c) Esquema de reconstrucción propuesto para la Pirámide Laplaciana [4].

3.6.2. Descomposición Direccional

La descomposición direccional se lleva a cabo con banco de filtros direccionales (DFB⁵) en 2-D, este concepto fue introducido e 1992 por Banberger y Smith [16], este proceso maximiza el decimado mientras que al mismo tiempo provee una reconstrucción perfecta. Los DFB son implementados mediante una descomposición en estructura de árbol de l – niveles con 2^l subbandas y partición del espectro como se muestra en la figura 3.15.

La construcción original de los DFB involucra una modulación de la señal de entrada y el uso de filtro en forma de diamante. Además, para obtener una partición de espectro deseada se debe seguir una regla de árbol (veáse para más detalles [17, 18]). Como resultado, las regiones de frecuencia resultantes no siguen un simple orden.

Para simplificar el análisis de los DFB iterados, los autores propusieron un banco de filtros quin-

⁵De sus siglas en Inglés Directional Filter Bank

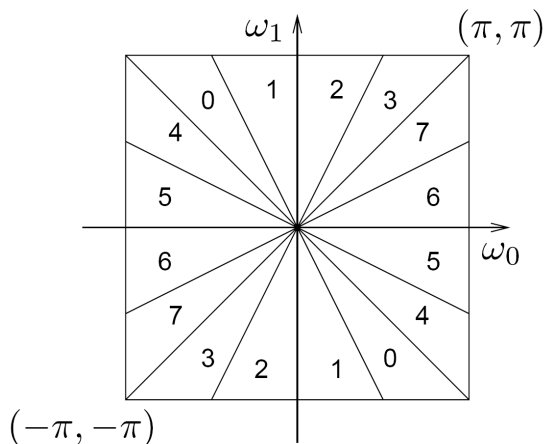


Figura 3.15: Partición frecuencial del banco de filtros direccional donde $l = 3$ y hay $2^3 = 8$ bandas frecuenciales [4].

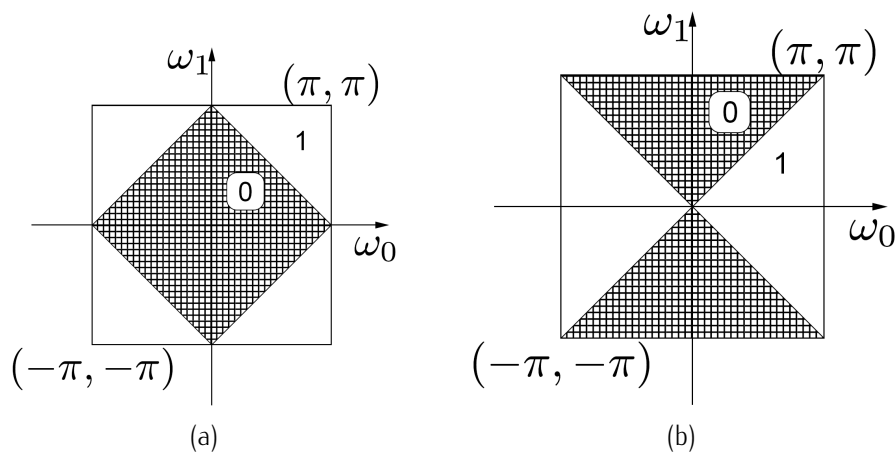


Figura 3.16: Dos posible configuraciones para los filtros en los QFB. (a) Filtros en forma de diamante. (b) Filtros en forma de ventilador [4].

con un banco de filtros quincunx (QFB⁶) con forma de ventilador (ver figura 3.16b). Esta nueva estructura evita la modulación de la imagen de entrada y tiene una regla más sencilla para descomposición de árbol. La partición frecuencial se realiza mediante una partición direccional de los QFB en forma de ventilador y por operaciones de rotación hechas por matrices de remuestreo.

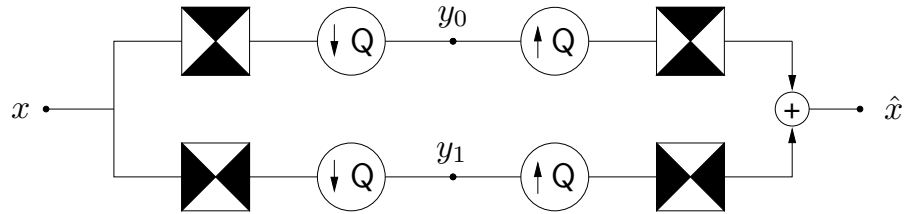


Figura 3.17: Partición del espectro en dos dimensiones usando el banco de filtros quincunx en forma de ventilador. Las regiones negras representan la respuesta ideal de cada filtro [3].

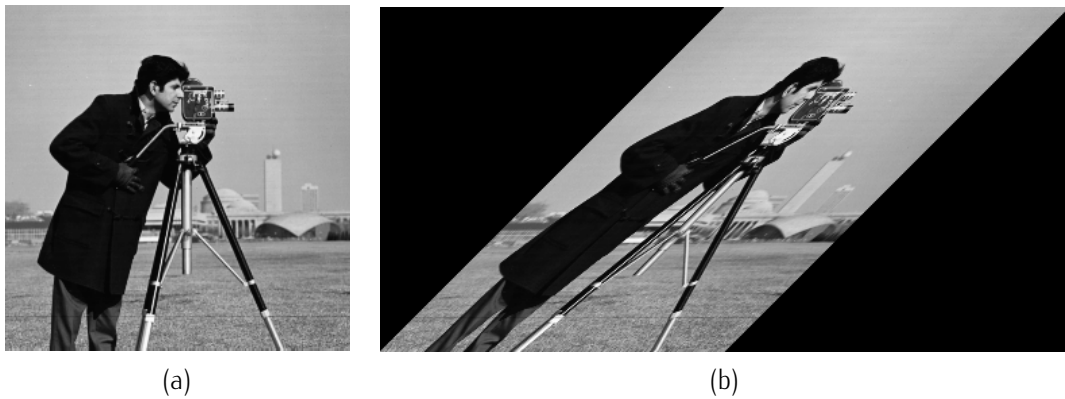


Figura 3.18: Ejemplo de una operación de remuestreo que es usado efectivamente como una operación de rotación para la descomposición del DFB. (a) Imagen "cameraman". (b) Imagen "cameraman" después del remuestreo [4].

Mediante el uso del concepto multitasa, el DFB de l -niveles en estructura de árbol se transforma en una estructura paralela de 2^l canales, con sus filtros equivalentes y todas aquellas matrices del muestreo. Cabe señalar que estos filtros sintetizados equivalen a $G_k^{(l)}$, $0 \leq k < 2^l$, y corresponden a las subbandas

⁶De sus siglas en Inglés Quincunx Filter Bank

de la figura 3.15. Las matrices de sobremuestreo tiene la siguiente forma diagonal:

$$S_k^{(l)} = \begin{cases} \text{diag}(2^{l-1}, 2) & \text{para } 0 \leq k < 2^{l-1} \\ \text{diag}(2, 2^{l-1}) & \text{para } 2^{l-1} \leq k < 2^l, \end{cases} \quad (3.2)$$

Las cuales corresponden a las subbandas horizontales y verticales respectivamente.

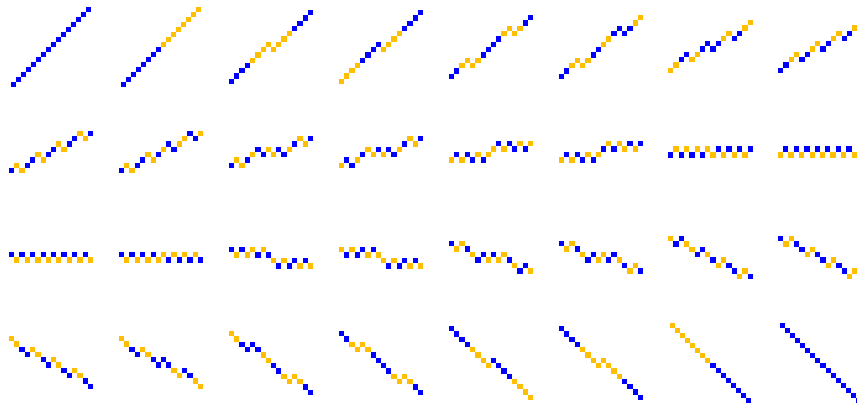


Figura 3.19: Respuesta al impulso de 32 filtros equivalentes para la primera mitad de los canales de un DFB de 6 niveles, los cuales usan filtro Haar. Los cuadros negros y grises corresponden a +1 y -1 respectivamente. Ya que las funciones de base se asemejan a “líneas locales”, reciben el nombre de *randolets* [3].

3.6.3. Descomposición Multiescala y Direccional

El DFB está diseñado para capturar componentes de alta frecuencia (representan direccionalidad) de imágenes. Por lo que las componentes de alta frecuencia no pueden ser tratadas de forma adecuada, por lo tanto el DFB no es eficiente en la representación de imágenes. Para mejorar esta situación, las componentes de baja frecuencia deberán ser removidas antes de pasar por el DFB.

Con esto en mente, la LP permite una descomposición en subbandas en cada una de sus imágenes pasabanda. Estas imágenes pueden ser alimentadas dentro de un banco de filtros tal que la información direccional pueda ser capturada eficientemente. Este proceso puede repetirse sobre la imagen reducida en resolución. La combinación de la pirámide laplaciana y el banco de filtros direccional resulta en un esquema llamado banco de filtros direccional piramidal (PDFB⁷), el cual provee una descomposición en

⁷De sus siglas en Inglés Pyramidal Directional Filter Bank

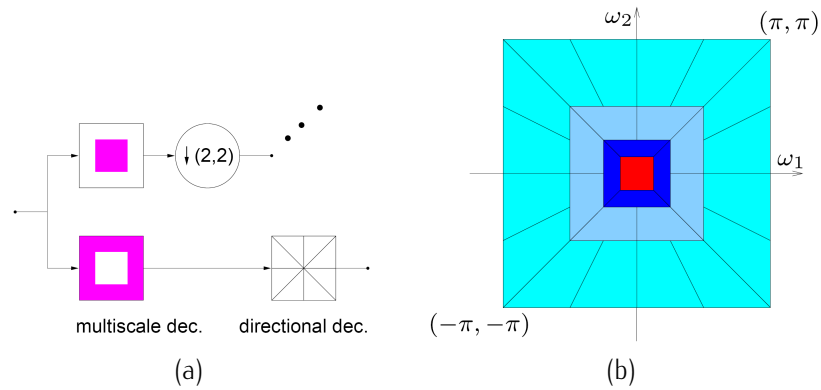


Figura 3.20: Banco de filtros direccional piramidal. (a) Diagrama de bloques. Primero, una descomposición estándar multiescala dentro de ocho bandas es calculada, donde el canal pasobajas es submuestreado mientras que el pasoaltas no. Después, una descomposición direccional con un DFB es aplicado a cada canal pasoaltas. (b) Resultado de la división de frecuencia, donde el número de direcciones es incrementado con la frecuencia [4].

subbandas direccionales en múltiples escalas.

Como la LP y DFB tienen una reconstrucción perfecta, la PDFB también la tendrá. La PDFB tiene el mismo factor de redundancia que la pirámide laplaciana. Otra característica que tiene la PDFB es que duplica el número de direcciones mientras que cada una de las escalas de la pirámide se hace más fina y también satisface la relación de escalamiento anisotrópico de curvelets. Esta relación en términos de la PDFB, esta dada por sus funciones base, es decir, un coeficiente de la pirámide en la subbanda b_j corresponde a una función base, que tiene como soporte local un cuadrado 2^j . Mientras que las funciones base de un DFB con $\lfloor n_0 - j/2 \rfloor$ niveles iterados tiene soporte en un rectángulo de longitud $2^{n_0-j/2}$ y ancho cercano a 1. Por lo tanto, la PDFB tiene una función base en el nivel j de la pirámide con soporte igual :

$$\text{ancho} \approx 2^j \quad \text{y} \quad \text{longitud} \approx 2^j \cdot 2^{n_0-j/2} = 2^{n_0} \cdot 2^{j/2}, \quad (3.3)$$

la cual claramente satisface la relación de escalamiento anisotrópico (3.1) de curvelets.

3.6.4. Análisis Multiresolución

En esta sección se mostrará el análisis de las propiedades de la PDFB como una estructura en el dominio continuo, hecha mediante una aproximación en la cual las matrices de muestreo de la PDFB son

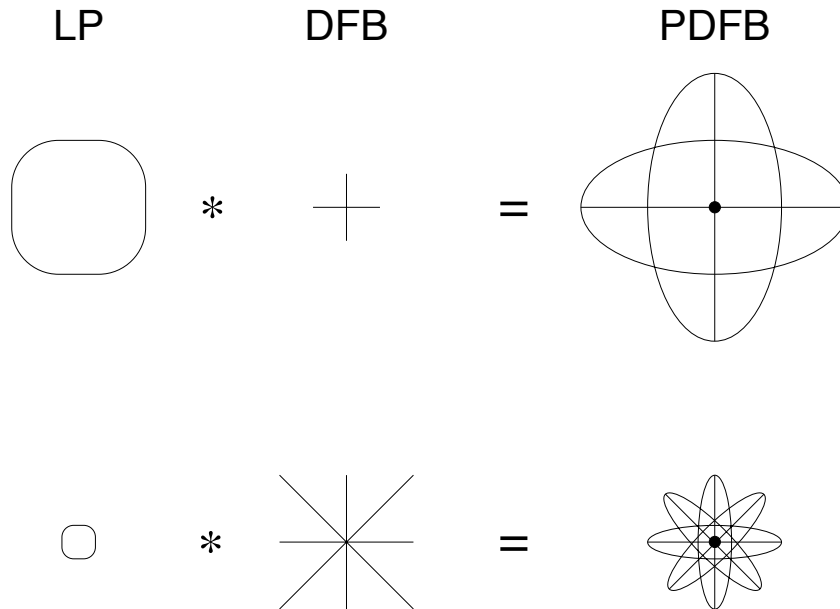


Figura 3.21: Ilustración de imágenes base para una contourlet que satisfacen la relación de escalamiento de la curva. De la línea superior a la inferior, la escala es reducida por cuatro mientras el número de direcciones es duplicado [4].

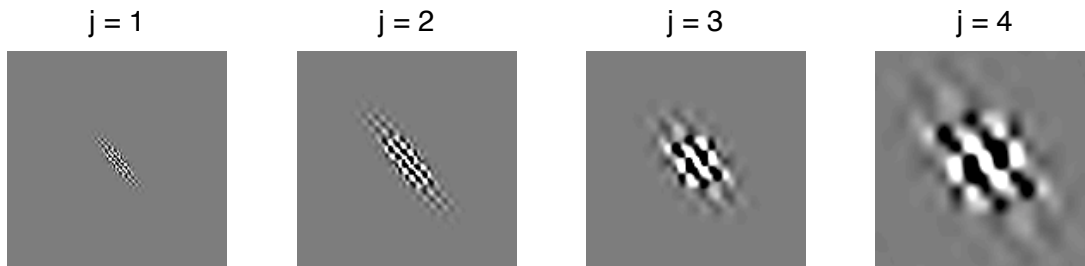


Figura 3.22: Imágenes base de la PDFB que implementa la transformada curvelet. Estas tienen estructura lineal y satisfacen la relación de escalamiento anisotrópico [4].

vistas como un sistema de rejillas rectangular.

Multiescala

Suponga, que la LP en la PDFB usa filtros ortogonales y submuestreo con orden dos en cada dimensión. Para ciertas aplicaciones, el filtro pasobajas G en la LP define únicamente una función de escalamiento ortogonal $\phi \in L^2(\mathbb{R}^2)$.

$$\phi = 2 \sum_{n \in \mathbb{Z}^2} g[n] \phi(2t - n)$$

en donde:

$$\phi_{j,n} = 2^j \phi \left(\frac{t - 2^j n}{2^j} \right), \quad j \in \mathbb{Z}, n \in \mathbb{Z}^2.$$

Donde la familia $\{\phi_{j,n}\}_{n \in \mathbb{Z}^2}$ es una base ortogonal de V_j para toda $j \in \mathbb{Z}$. La secuencia de subespacios jerarquizados satisface las propiedades siguientes:

$$\text{Invariancia al corrimiento: } f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_j, \quad \forall j \in \mathbb{Z}, k \in \mathbb{Z}^2$$

$$\text{Invariancia a la escala: } f(t) \in V_j \Leftrightarrow f(2^{-1}t) \in V_{j+1}, \quad \forall j \in \mathbb{Z}.$$

Es decir, V_j es un subespacio definido sobre una rejilla uniforme con intervalos $2^j \times 2^j$, la cual caracteriza la aproximación de la imagen a la resolución 2^j . Los detalles necesarios para incrementar la resolución de la imagen aproximada son preservados por la diferencia entre las imágenes de la LP. Sea W_j el complemento ortogonal de V_j en V_{j-1} (ver figura 3.23) se satisface:

$$V_{j-1} = V_j \oplus W_j$$

La pirámide laplaciana es también obtenida por un banco de filtros con submuestreo de sus salidas, donde cada componente polifásico de la diferencia de la señal corresponde a un canal separado de banco del filtros. $F_i(z)$, $0 \leq i \leq 3$ es el banco de filtros de síntesis para estos componentes polifásicos. $F_i(z)$, es un filtro pasoaltas. Al igual que en el banco de filtro wavelet, cada filtro es asociado a una función continua $\psi^{(i)}(t)$, donde:

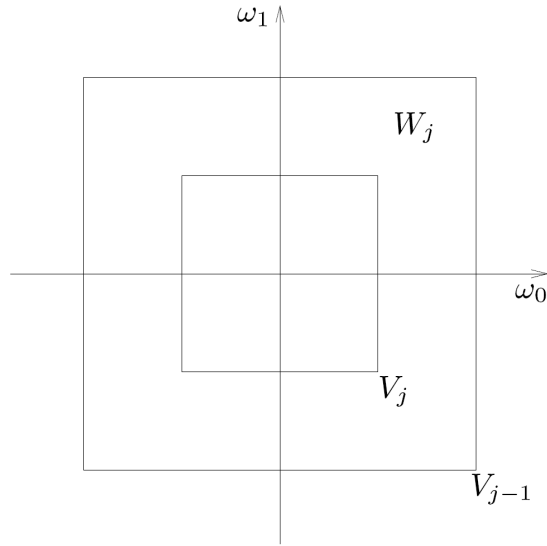


Figura 3.23: Subespacios multiescala generados por la pirámide laplaciana [4].

$$\psi^{(i)}(t) = 2 \sum_{n \in \mathbb{Z}^2} f_i[n] \phi(2t - n).$$

En [4] se demuestra que la LP con filtros ortogonales genera un análisis multiresolución. Entonces para una escala j , $\{\psi_{j,n}^{(i)}\}_{0 \leq i \leq |M|-1, n \in \mathbb{Z}^d}$ es un marco ajustado de W_j y para todas las escalas, $\{\psi_{j,n}^{(i)}\}_{0 \leq i \leq |M|-1, j \in \mathbb{Z}, n \in \mathbb{Z}^2}$ es un marco ajustado de $L^2(\mathbb{R}^d)$. En todos los casos con los límites del marco igual a 1.

Ahora, como W_{j+1} es generado por cuatro funciones prototipo, este no es un desplazamiento invariante de los subespacios, a menos que las $F_i(z)$ sean versiones desplazadas del filtro, o

$$F_i(z) = z^{-k_i} F(z)$$

donde k_i son los subconjuntos que representan el enrejado de submuestreo (2, 2)

$$k_0 = (0, 0)^T, k_1 = (1, 0)^T, k_2 = (0, 1)^T, k_3 = (1, 1)^T.$$

Basado en esto, podemos decir que W_{j+1} es un subespacio invariante al desplazamiento, denotado por:

$$\mu_{j, 2n+k_i} = \psi_{j+1, n}^{(i)} = \sum_{m \in \mathbb{Z}^2} f_i[m] \phi_{j, n+m}(t). \quad (3.4)$$

Con lo que se concluye que la familia $\{\mu_{j,n}\}_{n \in \mathbb{Z}^2}$ es un marco ajustado de W_{j+1} y se asemeja a una rejilla uniforme en \mathbb{R}^2 de intervalos $2^j \times 2^j$.

Direcciones Múltiples

La familia de banco de filtros direccional que se usan en la PDFB son filtros ortogonales y forman la familia

$$\{g_k^{(l)}[n - S_k^{(l)}m]\}_{0 \leq k \leq 2^l, m \in \mathbb{Z}^2} \quad (3.5)$$

que a su vez es una base direccional ortonormal de $l^2(\mathbb{Z}^2)$. $g_k^{(l)}$ son filtros direccionales, con $k = 0, \dots, 2^{l-1}-1$ para las direcciones $[-45^\circ, -45^\circ]$, y $k = 2^{l-1}, \dots, 2^l-1$ para las direcciones $[+45^\circ, +135^\circ]$. $S_k^{(l)}$ son las matrices diagonales de muestreo dadas en (3.2).

Aplicando el DFB a los subespacios de multiresolución V_j , se tiene

$$\theta_{j,k,n}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} g_k^{(l)}[m - S_k^{(l)}n] \phi_{j,m}(t) \quad (3.6)$$

Donde la familia $\{\theta_{j,k,n}^{(l)}\}_{m \in \mathbb{Z}^2}$ es una base ortonormal del subespacio direccional $V_{j,k}^{(l)}$ para cada $k = 0, \dots, 2^l - 1$. Estos subespacios son ortogonales con

$$V_{j,k}^{(l)} = V_{j,2k}^{(l+1)} \oplus V_{j,2k+1}^{(l+1)}, \quad y \quad (3.7)$$

$$V_j = \bigoplus_{k=0}^{2^l-1} V_{j,k}^{(l)}. \quad (3.8)$$

La figura 3.24 ilustra el subespacio dividido en “dos direcciones” por el DFB en el dominio de la frecuencia. En el dominio espacial, $V_{j,k}^{(l)}$ es el subespacio en una escala 2^j y una dirección k a lo largo de todas las 2^l direcciones.

Aplicando la descomposición direccional de la familia (3.5) sobre el subespacio de detalles W_{j+1} como lo hace la PDFB, se obtiene el siguiente resultado.

Definiendo

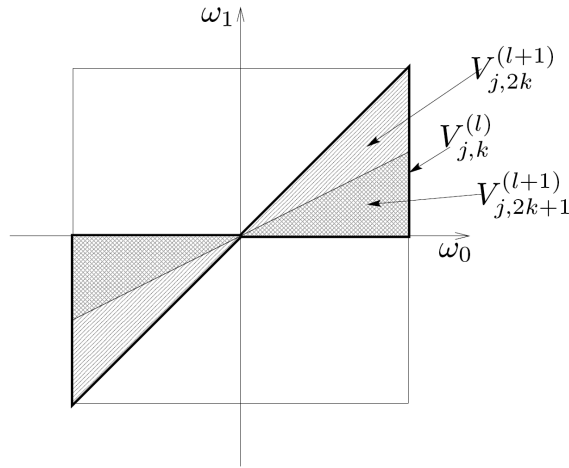


Figura 3.24: Subespacios multidireccionales generados por el DFB [4].

$$\rho_{j,k,n}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} g_k^{(l)}[m - S_k^{(l)}n] \mu_{j,m}(t) \quad (3.9)$$

La familia $\{\rho_{j,k,n}^{(l)}\}_{n \in \mathbb{Z}^2}$ es un marco ajustado de un subespacio $W_{j+1,k}^{(l)}$ con límites del marco igual a 1, para cada $k = 0, \dots, 2^l - 1$. Estos subespacios son ortogonales con

$$W_{j+1,k}^{(l)} = W_{j,2k}^{(l+1)} \oplus W_{j,2k+1}^{(l+1)}, \text{ y} \quad (3.10)$$

$$W_{j+1}^{(l)} = \bigoplus_{k=0}^{2^l-1} W_{j+1,k}^{(l)}. \quad (3.11)$$

La figura 3.25 muestra una representación gráfica de los subespacios, vistos en el dominio de la frecuencia. La razón por la cual $\{\rho_{j,k,n}^{(l)}\}_{n \in \mathbb{Z}^2}$ es un sistema sobrecompleto para $W_{j+1,k}^{(l)}$ es porque utiliza la misma rejilla de muestreo, que el subespacio mayor $V_{j,k}^l$.

Recordando que W_{j+1} no es invariante al corrimiento, sin embargo, el resultado siguiente establece que sus subespacios $W_{j+1,k}^{(l)}$ sí lo son, porque son generados por una sola función prototipo.

Dicho resultado se muestra a continuación, sea:

$$\rho_{j,k}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} g_k^{(l)}[m] \mu_{j,m}(t) \quad (3.12)$$

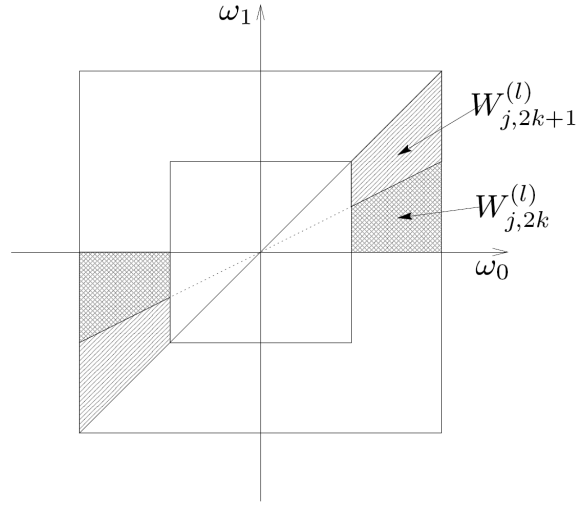


Figura 3.25: Subespacios multidireccionales generados por la PDFB [4].

Entonces para $l \geq 2$

$$\rho_{j,k,n}^{(l)}(t) = \rho_{j,k}^{(l)}(t - 2^j S_k^{(l)} n) \quad (3.13)$$

Por consiguiente, los subespacios $W_{j+1,k}^{(l)}$ satisfacen las siguientes propiedades de invariancia al corrimiento:

$$f(t) \in W_{j+1,k}^{(l)} \Leftrightarrow f(t - 2^j S_k^{(l)} n) \in W_{j+1,k}^{(l)}, \forall n \in \mathbb{Z}^2. \quad (3.14)$$

Esto nos dice que los subespacios multiescala multidireccionales $W_{j+1,k}^{(l)}$ están definidos sobre una rejilla rectangular con intervalos $2^{j+l-1} \times 2^{j+1}$ (o $2^{j+1} \times 2^{j+l-1}$), dependiendo de si estos son horizontales o verticales. Substituyendo (3.4) en (3.12), la función prototipo $\rho_{j,k}^{(l)}(t)$ se puede escribir directamente como la combinación lineal de la función de escalamiento $\phi_{j,m}(t)$ como

$$\begin{aligned} \rho_{j,k}^{(l)}(t) &= \sum_{i=0}^3 \sum_n g_k^{(l)}[2n + k_i] \left(\sum_{m \in \mathbb{Z}^2} f_i[m] \phi_{j,n+m} \right) \\ &= \sum_{m \in \mathbb{Z}^2} \underbrace{\left(\sum_{i=0}^3 \sum_{n \in \mathbb{Z}^2} g_k^{(l)}[2n + k_i] f_i[m - n] \right)}_{c_k^{(l)}[m]} \phi_{j,m}(t). \end{aligned} \quad (3.15)$$

Las secuencias $c_k^{(l)}[m]$ se asemejan a una suma de convoluciones entre $g_k^{(l)}[m]$ y $f_i[m]$, por lo tanto es un filtro pasoaltas y direccional. La ecuación (3.12) revela el aspecto contourlet de la función prototipo $\rho_{j,k}^{(l)}(t)$ para $W_{j+1,k}$, los cuales se ven como un agrupamiento de elementos de detección de bordes en una escala j y a lo largo de una dirección k .

Multiescala y Direcciones Múltiples

Finalmente, integrando sobre todas las escalas, se tiene el siguiente resultado para los marcos del espacio $L^2(\mathbb{R}^2)$.

Para una secuencia finita de enteros positivos $\{l_j\}_{j \leq j_0}$ la familia

$$\{\phi_{j_0,n}(t), \rho_{j,k,n}^{(l_j)}(t)\}_{j \leq j_0, 0 \leq k \leq 2^j - 1, n \in \mathbb{Z}^2} \quad (3.16)$$

es un marco ajustado de $L^2(\mathbb{R}^2)$. Para una secuencia finita de enteros positivos $\{l_j\}_{j \in \mathbb{Z}}$, la familia

$$\{\rho_{j,k,n}^{(l_j)}(t)\}_{j \in \mathbb{Z}, 0 \leq k \leq 2^j - 1, n \in \mathbb{Z}^2} \quad (3.17)$$

es un marco ajustado direccional wavelet de $L^2(\mathbb{R}^2)$. En cada caso, con límites del marco iguales a 1.

El marco ajustado en (3.16) provee una descomposición curvelet cuando el número de direcciones se duplica en cada una de las escalas más finas. Esto significa que si en la escala 2^{j_0} , se comienza con un DFB con l_{j_0} niveles entonces en las escalas más finas 2^j , $j < j_0$, el número de niveles de descomposición del DFB deberá ser:

$$l_j = \lfloor l_{j_0} - (j - j_0)/2 \rfloor, \quad \text{para } j \leq j_0. \quad (3.18)$$

Así la rejilla incrustada de aproximación para la curvelet del banco de filtros direccional piramidal en la escala 2^j , es $2^{\lfloor n_0 + j/2 \rfloor} \times 2^j$ para las direcciones horizontales y $2^j \times 2^{\lfloor n_0 + j/2 \rfloor}$ para las direcciones verticales, donde $n_0 = l_{j_0} - j_0/2 + 2$. La figura 3.26 ilustra este patrón de muestreo en diferentes escalas y direcciones. Lo principal a notar es que en el proceso de refinamiento, una dimensión espacial es refinada dos veces con respecto a la otra dimensión espacial.

La figura 3.27a muestra la descomposición contourlet de la imagen "Peppers", con solo dos niveles de descomposición para una mejor visualización, los componentes de menor energía se muestran en

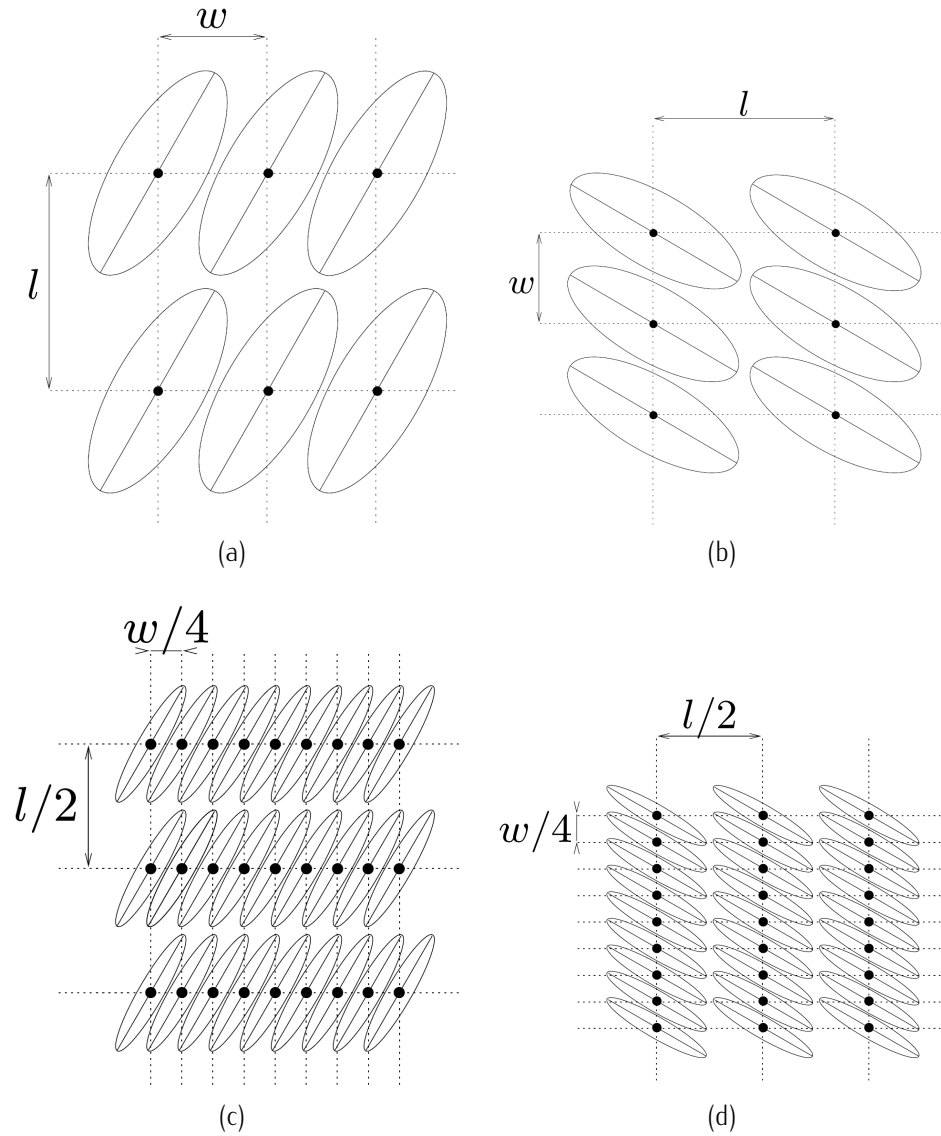
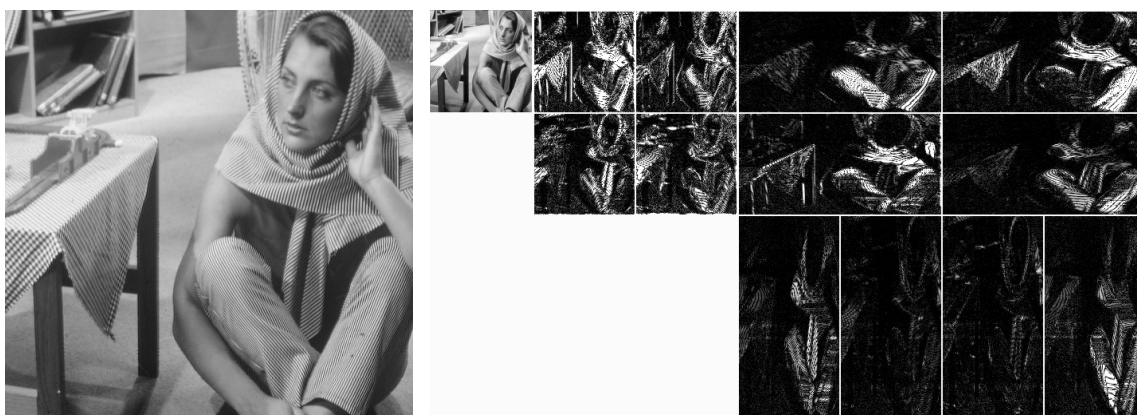


Figura 3.26: Rejillas de aproximación incrustadas en el dominio espacial para la PDFB que implementa la transformada curvelet. Estas son cuatro ilustraciones de los subespacios $W_{j,k}$ representando escalas burdas vs. escalas finas y direcciones horizontales vs. direcciones verticales. Cada subespacio es recorrido por la función prototipo curvelet dada en (3.12). Los intervalos de muestreo se emparejan con la ayuda de la función prototipo, por ejemplo de ancho w largo l , tal que los desplazamientos cubrirán el plano \mathbb{R}^2 [4].

color negro mientras que los de mayor energía en blanco. Esta ilustración muestra claramente como la transformada es flexible ya que obtiene una buena representación de la imagen en ambos dominios, y comienza con una representación burda de la imagen hasta una representación más fina. La figura 3.27b muestra la descomposición de la imagen "Barbara", con el mismo número de niveles y direcciones que la imagen anterior, sin embargo es importante resaltar que en esta imagen la transformada captura mejor la energía sobre las texturas de la ropa en las diferentes escalas y en las diferentes direcciones, lo que nos ayuda a concluir que la transformada contourlet como su nombre lo indica es una buena opción para capturar contornos.



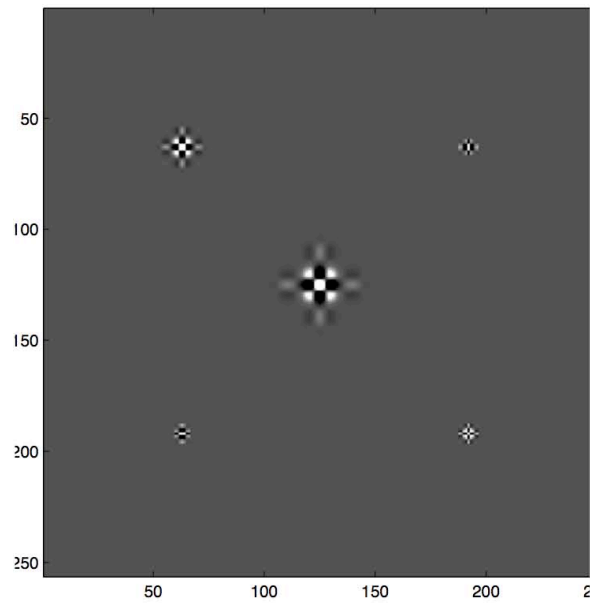
(a)



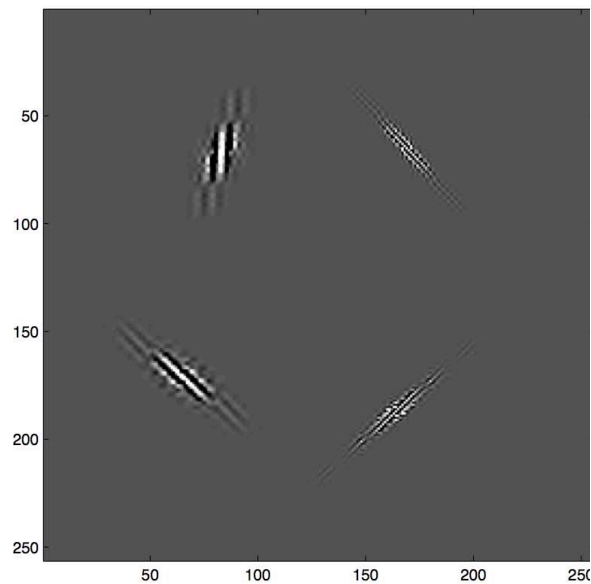
(b)

Figura 3.27: Ejemplo de descomposición contourlet. (a) Imagen peppers con dos niveles de descomposición de la pirámide laplaciana, con 4 direcciones en el primer nivel de la pirámide y ocho direcciones en el segundo nivel de descomposición de la pirámide. (b) Imagen barbara con el mismo nivel de descomposición y direcciones que la imagen peppers.

La figura 3.28 muestra la diferencia entre funciones base wavelet y funciones base contourlet, se puede observar como las bases contourlet capturan mayor energía en diferentes direcciones, además que cumplen con el principio de escalamiento anisotrópico.



(a) Funciones base wavelet



(b) Funciones base contourlet

Figura 3.28: Ejemplo de funciones base.

3.7. Descripción de algoritmos de inserción de la marca de agua

El primer algoritmo de inserción de la marca de agua fue tomado de [1], en este artículo los autores utilizan un método de inserción en el dominio de la transformada contourlet, con una técnica de espectro disperso para inserción de la marca y un detector correlacionado para la detección de la secuencia pseudoaleatoria incrustada. A continuación se dará una explicación con más detalle del algoritmo.

3.7.1. Algoritmo 1

Como se muestra en la figura 3.29, se usa una descomposición direccional que se duplica en cada uno de los niveles multiresolución de la pirámide. Es decir, si la imagen original es de tamaño $N \times N$, el segundo nivel de descomposición será de $(N/2) \times (N/2)$ y así sucesivamente. La marca será incrustada de dos formas, primero en el 75% de la banda Y (excepto Y_7 y Y_8) y segundo en todos los coeficientes de la banda Y , se seleccionaron estas bandas porque fueron las que presentaron mejor desempeño en las simulaciones; ambas con un conjunto de secuencias pseudoaleatorias.

Inserción de la marca de agua

La codificación de la marca de agua o mensaje se lleva a cabo de la siguiente forma:

Sea $\hat{\mathbf{b}} = (\hat{b}_1, \hat{b}_2, \dots, \hat{b}_M)^T$ la representación binaria de nuestra marca de agua o mensaje, donde, $\hat{b}_i \in \{0, 1\}$ y M es el número de bits en el mensaje a ser codificados. La marca de agua binaria $\hat{\mathbf{b}}$ es transformada en un vector $\mathbf{b} = (b_1, b_2, \dots, b_M)$, con $b_i \in \{1, -1\}$. Esta transformación de $1 \rightarrow -1$ y $0 \rightarrow 1$ es importante ya que nos permite reemplazar el operador $x-or$ usado en un campo finito del álgebra por la multiplicación [19]. Definiendo un conjunto de secuencias pseudoaleatorias \mathbf{P}_i generadas por una llave⁸, donde cada una de ellas corresponde a un bit b_i , el mensaje codificado puede ser obtenido por:

$$\mathbf{W} = \sum_{i=1}^M b_i \mathbf{P}_i \quad (3.19)$$

⁸El conjunto de secuencias pseudoaleatorias utilizadas en nuestro algoritmo, son binarias con media igual a 0 y variancia 1.

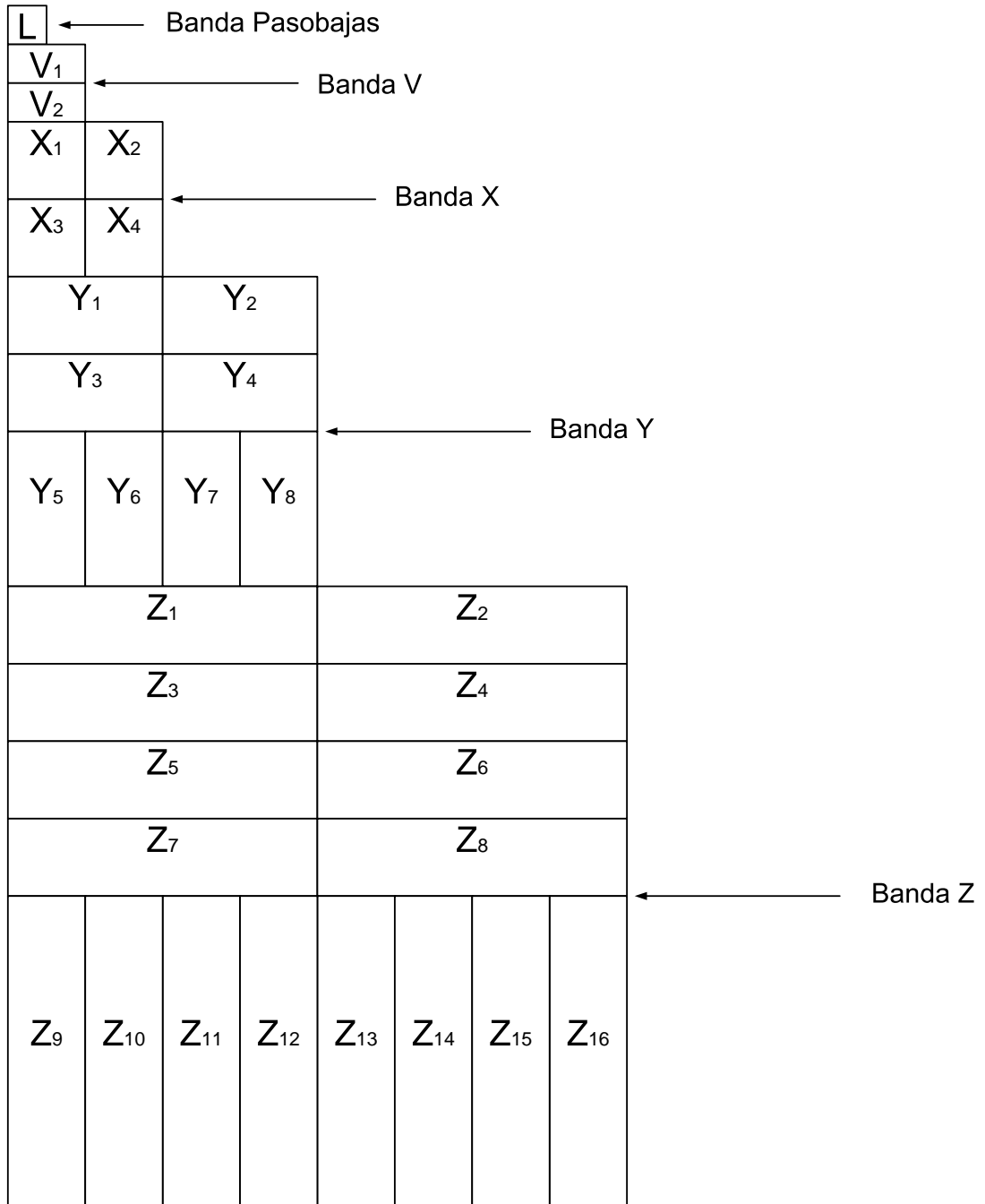


Figura 3.29: Descomposición Contourlet.

Claramente, la efectividad de este esquema depende de la elección específica de las secuencias pseudoaleatorias P_i . Una buena secuencia de espectro disperso es aquella que combina características estadísticas con criptografía.

Ahora bien, el proceso de incrustación de la marca de agua en forma de una secuencia pseudoaleatoria se lleva a cabo usando una técnica de espectro disperso con una regla aditiva [20], dada en la expresión siguiente:

$$Y'(i, j) = Y(i, j) + \alpha W(i, j), \quad 1 \leq i, j \leq (N/2). \quad (3.20)$$

donde $Y(i, j)$ representa los píxeles de la banda Y de la imagen transformada, $Y'(i, j)$ representa los correspondientes píxeles marcados y $W(i, j)$ representa los elemento de la matriz W . El factor de multiplicación α es un parámetro de control de la imperceptibilidad y robustez de la marca de agua. Después que los coeficientes deseados han sido marcados, estos son regresados a su posición original y la transformada inversa contourlet se lleva a cabo para obtener la imagen marcada.

Detección de la marca de agua

La marca de agua incrustada, es recuperada por un detector correlacionado. Después de recibir una marca que ha sufrido un posible ataque, se toma la transformada contourlet de la imagen. Las secuencias pseudoaleatorias son generadas usando la llave y se encuentra la correlación de los píxeles marcados con cada una de las secuencias pseudoaleatorias.

Sea \mathcal{Y} la representación de los píxeles después de la transformación, entonces la correlación para decidir el i -ésimo bit se encuentra como sigue.

$$\rho_i = \langle \mathcal{Y}, P_i \rangle, \quad (3.21)$$

$$\rho_i = \frac{\sum_{i=1}^{N/2} \mathcal{Y} P_i}{\sqrt{\sum_{i=1}^{N/2} \mathcal{Y}^2 \sum_{i=1}^{N/2} P_i^2}} \quad (3.22)$$

La media de los valores de correlación ρ_i es calculada y es seleccionado como umbral τ , para decidir el bit de la marca de agua. Si el valor de correlación es arriba de la media, por ejemplo $\rho_i > \tau$, entonces

se elige el valor de la marca de agua como 1, y en otro caso como 0. Así, la marca binaria entera es generada. Para verificar la semejanza entre la marca de agua original y la recuperada, se calcula el coeficiente de correlación de ambas imágenes.

3.7.2. Algoritmo 2(Propuesto)

En muchos esquemas de marcado de agua, la técnica de espectro disperso(SS) es utilizada como una técnica de modulación para incrustar la marca de agua, porque este esquema es robusto a interferencias de ruido, es decir, la cantidad de energía para poder eliminar la marca de agua debe ser muy alta. Sin embargo, en tal modulación la marca de agua por si misma es vista como una fuente de interferencia, por lo que entre más grande sea la marca de agua, con mayor interferencia resultará nuestra señal marcada.

Una deficiencia que tiene el algoritmo de marcado uno, es que el parámetro de control de imperceptibilidad α no toma en cuenta las características del lugar donde se inserta la marca. Para solucionar esto, lo que se puede hacer es hacer variar el parámetro α dependiendo de las características del lugar donde se insertará la marca de agua.

Variar α se puede hacer de varias formas. Una forma puede ser mediante una mascara de peso, la cual explota las características del sistema visual humano (HVS) para aumentar el desempeño del esquema de marcado de agua. En [21] los autores proponen un esquema en el dominio wavelet que toma en cuenta las texturas y luminancia de todas las subbandas para construir la mascara del lugar donde se insertará la marca. Otra forma, es explotar el conocimiento del codificador acerca de Y (el lugar donde se incrustará la marca), por lo que se puede realzar el desempeño mediante la modulación de la energía de la marca de agua insertada para compensar la señal de interferencia. En [22] los autores explotan esto mediante una técnica que ellos llaman "espectro disperso mejorado(ISS⁹)". Adoptamos, la segunda forma como propuesta de mejora del algoritmo, ya que no es tan trivial conocer como se distribuye la energía de las texturas y luminancia de una señal en el dominio contourlet ya que en este se tienen muchas direcciones en cada nivel de resolución, mientras que en wavelet solo se tiene los detalles horizontales, verticales y diagonales.

⁹Por sus siglas en inglés Improved Spread Spectrum

Inserción de la marca de agua

Para la inserción de la marca usaremos una versión lineal del método de ISS, ya que es más simple su análisis, pero aún así este provee una ganancia significativa en comparación del esquema tradicional de SS, por simplicidad consideraremos que solo insertaremos un bit de la marca de agua. Tomando como base el artículo [22] tenemos:

$$\mathbf{y}' = \mathbf{y} + (\alpha b - \lambda x)\mathbf{p}. \quad (3.23)$$

donde, como ya mencionamos antes \mathbf{y}' representa la correspondiente banda marcada, \mathbf{y} la banda donde se incrustará la señal de información, b es el bit de información, \mathbf{p} es la secuencia pseudoaleatoria, α y λ son parámetros de control de distorsión y $x \triangleq \langle \mathbf{y}, \mathbf{p} \rangle / \langle \mathbf{p}, \mathbf{p} \rangle$, el operador $\langle \cdot \rangle$ representa el producto interno. En el esquema tradicional de espectro disperso el valor de $\lambda = 0$.

Considerando que el canal de información es modelado como ruido aditivo:

$$\mathbf{s} = \mathbf{y}' + \mathbf{n}$$

La estadística del receptor es

$$\begin{aligned} r &= \frac{\langle \mathbf{s}, \mathbf{p} \rangle}{\langle \mathbf{p}, \mathbf{p} \rangle} \\ &= \frac{\langle \mathbf{y} + (\alpha b - \lambda x)\mathbf{p} + \mathbf{n}, \mathbf{p} \rangle}{\langle \mathbf{p}, \mathbf{p} \rangle} \\ &= \alpha b + (1 - \lambda)x + n. \end{aligned}$$

donde, $x \triangleq \langle \mathbf{y}, \mathbf{p} \rangle / \langle \mathbf{p}, \mathbf{p} \rangle$ y $n \triangleq \langle \mathbf{n}, \mathbf{p} \rangle / \langle \mathbf{p}, \mathbf{p} \rangle$. Por lo tanto, entre más cercano sea λ a 1, mayor será la influencia de x removida de r . La distorsión esperada del nuevo sistema al incrustar un bit de la marca de agua esta dada por:

$$\begin{aligned} E[D] &= E[\|\mathbf{y}' - \mathbf{y}\|] \\ &= E[\|(\alpha b - \lambda x)\mathbf{p}\|] = \left(\alpha^2 + \frac{\lambda^2 \sigma_y^2}{M \sigma_p^2} \right) \sigma_p^2. \end{aligned} \quad (3.24)$$

donde M es la longitud de \mathbf{y} , \mathbf{p} , \mathbf{s} y \mathbf{n} . Haciendo el promedio de la distorsión del nuevo sistema igual al tradicional SS, forzamos $E[D] = \sigma_p^2$, y por lo tanto

$$\alpha = \sqrt{\frac{M\sigma_p^2 - \lambda^2\sigma_y^2}{M\sigma_p^2}}. \quad (3.25)$$

Para calcular la probabilidad de error p_e , de [22] se tiene:

$$p_e = \frac{1}{2} \operatorname{erfc} \left(\frac{1}{\sqrt{2}} \sqrt{\frac{\frac{M\sigma_p^2}{\sigma_y^2} - \lambda^2}{\frac{\sigma_n^2}{\sigma_y^2} + (1 - \lambda)^2}} \right). \quad (3.26)$$

Esta es una función de λ , el ratio de la señal a ruido $(\sigma_y/\sigma_n)^2$ y la potencia relativa de la secuencia SS $M(\sigma_y/\sigma_n)^2$. Mediante la apropiada selección del parámetro λ , la tasa de errores de sistema usando ISS puede disminuir en varios ordenes en comparación del esquema de SS. La probabilidad de error puede ser minimizada para el valor óptimo de λ , calculando la probabilidad de error p_e cuando $\partial p/\partial \lambda = 0$:

$$\lambda_{opt} = \frac{1}{2} \left[\left(1 + \frac{\sigma_n^2}{\sigma_y^2} + \frac{M\sigma_p^2}{\sigma_y^2} \right) - \sqrt{\left(1 + \frac{\sigma_n^2}{\sigma_y^2} + \frac{M\sigma_p^2}{\sigma_y^2} \right)^2 - 4 \frac{M\sigma_p^2}{\sigma_y^2}} \right]. \quad (3.27)$$

De la ecuación anterior podemos concluir que para M suficientemente grande, $\lambda_{opt} \rightarrow 1$ y $\text{SNR} \rightarrow \infty$. El esquema de inserción de marca de agua usa la ecuación derivada de λ_{opt} para ajustar las propiedades deseadas y desempeño de todo el sistema de marca de agua. También, es posible diseñar un sistema con un valor predefinido para cierta tasa de bits en error y derivar para obtener λ_{opt} . Por otro lado, ISS no requiere cambios en el esquema de detección, debido a que el sistema de inserción de marca de agua es lineal.

Detección de la marca de agua

Al igual que en el algoritmo anterior la detección de la marca se lleva a cabo sin referencia de la imagen original. Consideraremos que la detección de la marca se lleva a cabo para un canal AWGN¹⁰. La prueba de hipótesis se formulará de la siguiente forma:

¹⁰Por sus siglas en inglés Additive White Gaussian Noise

$$H_0 : y'_i = y_i,$$

$$H_1 : y'_i = y_i + (ab - \lambda x) p_i.$$

es decir, H_0 es la hipótesis de que la imagen no este marcada y H_1 es la hipótesis de que la imagen esta marcada con un conjunto de secuencias pseudoaleatorias. Dado que hemos considerado el canal como AWGN. Tenemos:

$$p(\mathbf{y}'|H_0) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(y'_i - \mu_y)^2}{2\sigma_y^2}\right), \quad (3.28)$$

donde μ_y y σ_y^2 indican la media y varianza de las muestras de y . De la misma forma:

$$p(\mathbf{y}'|H_1) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(y'_i - \mu_y - (ab - \lambda x) p_i)^2}{2\sigma_y^2}\right). \quad (3.29)$$

Obteniendo la razón de máxima verosimilitud $\ell(\mathbf{y}')$, tenemos la estructura del detector:

$$\ell(\mathbf{y}') = \frac{p(\mathbf{y}'|H_1)}{p(\mathbf{y}'|H_0)} = \frac{\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(y'_i - \mu_y - (ab - \lambda x) p_i)^2}{2\sigma_y^2}\right)}{\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(y'_i - \mu_y)^2}{2\sigma_y^2}\right)}, \quad (3.30)$$

pasando a su forma logarítmica, tenemos:

$$\begin{aligned} \mathcal{L}(\mathbf{y}') &= \sum_{i=1}^n \frac{1}{\sqrt{2\sigma_y^2}} \left[(y'_i - \mu_y)^2 - (y'_i - \mu_y - (ab - \lambda x) p_i)^2 \right] \\ &= \frac{1}{\sqrt{2\sigma_y^2}} \left[\sum_{i=1}^n 2(ab - \lambda x) y'_i p_i - \sum_{i=1}^n 2(ab - \lambda x) \mu_y p_i - \sum_{i=1}^n (ab - \lambda x)^2 p_i^2 \right] \end{aligned} \quad (3.31)$$

Los últimos dos términos del paréntesis de la expresión anterior, no dependen de y'_i , con lo que se puede concluir que la correlación lineal de y' y p_i , es decir:

$$\rho_i = \frac{1}{n} \sum_{i=1}^n y'_i p_i,$$

es una estadística suficiente para la detección de la marca. Para conocer el umbral de detección de la marca T_ρ , adoptaremos el criterio de Neyman-Pearson, el cual, minimiza la probabilidad de perder la marca, sujeta a una tasa dada de falsa detección. Para determinar el criterio de decisión asumiremos que la variable de observación ρ esta formada por variables aleatorias. Sabemos que p_i son valores binarios, con media cero, variables aleatorias independientes, además de que no dependen de los coeficientes contourlet. También asumiremos que y' es una variable aleatoria independiente con media cero. Explotando el teorema del límite central, podemos asumir que ρ es normalmente distribuido. Bajo estas consideraciones tenemos:

$$\mu_{\rho|H_0} = E[\rho|H_0] = \frac{1}{n} E\left[\sum_{i=1}^n y_i p_i\right] = \frac{1}{n} \sum_{i=1}^n E[y_i] p_i = \mu_y \bar{p} = 0, \quad (3.32)$$

donde $\bar{p} = \sum w_i/n$ denota el promedio de las secuencias p_i . De forma similar, la varianza de ρ bajo la hipótesis H_0 , se puede escribir:

$$\sigma_{\rho|H_0}^2 = \frac{\sigma_y^2}{n^2} \sum_{i=1}^n E[y_i^2] + (\alpha b - \lambda x)^2 \sigma_y^2 E[p_i^2], \quad (3.33)$$

considerando el valor medio cuadrático de los coeficientes marcados, $E[(y'_i)^2]$, y recordando la regla de inserción de la marca de agua (3.23), podemos escribir:

$$E[(y'_i)^2] = E[(y_i)^2] + (\alpha b - \lambda x)^2 E[(p_i)^2] + 2(\alpha b - \lambda x) E[y_i \cdot w_i], \quad (3.34)$$

notando que $\sigma_y^2 = \sigma_y^4 = 1$, $E[y_i] = E[p_i] = 0$ y que p_i no depende de y , tenemos:

$$\sigma_{\rho|H_0}^2 = \frac{1}{n^2} \sum_{i=1}^n E[(y'_i)^2], \quad (3.35)$$

en la práctica $\sigma_{\rho|H_0}^2$ se estima simplemente como:

$$\sigma_{\rho|H_0}^2 \approx \frac{1}{n^2} \sum_{i=1}^n (y'_i)^2. \quad (3.36)$$

Considerando lo anterior podemos calcular la probabilidad de falsa alarma (P_f), obteniendo:

$$P_f = \int_{T_\rho}^{\infty} p(\rho|H_0) d\rho = \frac{1}{2} \operatorname{erfc} \left(\frac{T_\rho}{\sqrt{2\sigma_{\rho|H_0}^2}} \right), \quad (3.37)$$

Asumiendo que $P_f \leq 10^{-8}$, obtenemos:

$$T_\rho = 3,97 \sqrt{2\sigma_{\rho|H_0}^2}. \quad (3.38)$$

Finalmente, el criterio para decidir el i -ésimo bit queda expresada de la siguiente manera:

$$\Phi(\mathbf{y}') = \begin{cases} 1 & \text{si } \rho_i > T_\rho \\ 0 & \text{otro caso.} \end{cases}$$

de esta forma la marca de agua puede ser regenerada completamente. Un importante resultado de la utilización del criterio de Neyman-Pearson, es que el umbral puede ser calculado a *posteriori* sobre la imagen marcada. Además, la detección de la marca puede ser realizada sin el conocimiento de los parámetros de control de distorsión y robustez (α y λ), los cuales pueden ser ajustados con precisión en cada imagen.

Capítulo 4

Resultados

4.1. Introducción

En la simulación de nuestros algoritmos de inserción de marca de agua, hemos considerado dos imágenes las cuales son de las más comunes en procesamiento de señales, tales imágenes se conocen como 'lena' y 'barbara', ambas imágenes son de dimensiones 512 x 512 pixeles, y la marca de agua binaria es de tamaño 22 x 96 pixeles. Como mencionamos anteriormente, simulamos cuatro diferentes casos de inserción de marca de agua en el dominio de la transformada contourlet. El primer método consiste en marcar el 75% de la banda Y de tamaño 128 x 384 pixeles y el segundo método consiste en marcar la banda entera Y de tamaño 256 x 256 pixeles. Hemos elegido todas las bandas de Y excepto Y_7 y Y_8 para el primer caso. Los dos primeros métodos son utilizando el algoritmo de inserción de marca agua tomado de [1]. El método tres es el mismo que el primer método pero utilizando el algoritmo propuesto y el método cuatro es igual al segundo método pero utilizando el algoritmo propuesto.

Para la evaluación y comparación del desempeño de los métodos de marcado de agua, hemos utilizado el programa StirMark Benchmark 4 [23], el cuál realiza un gran numero de ataques y evaluaciones de la marca de agua. Los ataques en los que nos concentraremos serán la compresión JPEG, Ruido y Filtrado. Durante el desarrollo del capítulo se explicarán con detalle los resultados obtenidos.

4.2. Parámetros Importantes

Para evaluar las características entre la imagen marcada y la imagen original se utilizan los siguientes parámetros.

4.2.1. Histograma

Un histograma es una representación gráfica de la frecuencia de ocurrencia de cada valor de pixel en un imagen y solo proporciona información respecto a sus intensidades. Con la comparación del histograma de la imagen original y otra versión de la misma, se puede saber si esta ha sido expuesta a algún tipo de procesamiento.

4.2.2. Varianza

La varianza de una imagen, es una medida de la dispersión de los valores de pixel alrededor de la media de la imagen. Sea la imagen I de dimensiones M, N , la $\text{Var}[I]$ se calcula de la siguiente forma:

$$\begin{aligned} \text{Var}[I] &= E[I^2] - E[I]^2 \\ &= \left[\left(\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N I(i, j)^2 \right) - \left(\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N I(i, j) \right)^2 \right]. \end{aligned} \quad (4.1)$$

4.2.3. Relación señal pico a ruido

La relación o razón señal pico a ruido, es la proporción de la potencia máxima de una señal y la potencia del ruido que afecta la fidelidad de su representación. La PSNR es comúnmente usada como una medida de la calidad de una imagen reconstruida. Para obtener la PSNR, primero consideremos el error cuadrático medio (MSE), ó, distorsión, como:

$$D = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N E[|I(i, j) - \hat{I}(i, j)|^2], \quad (4.2)$$

donde I es la imagen original, \hat{I} es la imagen reconstruida y M, N son las dimensiones del tamaño de la imagen. La PSNR en decibelios se define como:

$$PSNR = 10\log_{10} \left(\frac{MAX_I^2}{D} \right) [dB]. \quad (4.3)$$

donde MAX_I es el máximo valor pico-a-pico en la imagen (típicamente 255 para imágenes de 8 bits).

4.3. Resultados de las simulaciones para la imagen "lena"

Después de obtener las gráficas de la variación de la PSNR y del coeficiente de correlación contra el factor de multiplicación (α), fijamos los valores del factor de multiplicación de modo que pudiésemos recuperar de las imágenes marcadas el logo con un coeficiente de correlación igual a 0.925 con respecto al logo original. En tabla 4.1 se muestran tales parámetros.

Parámetro	Método 1	Método 2	Método 3	Método 4
α	0.254	0.1947	0.0676	0.0496
PSNR[dB]	38.165	38.95	45.036	45.92
Correlación	0.925	0.925	0.925	0.925

Tabla 4.1: Parámetros importantes de la imagen marcada de 'lena'.

4.3.1. Histograma

La tabla 4.2 muestra la varianza de los valores de pixel de las imágenes marcadas con cada uno de los distintos métodos.

Parámetro	Imagen Original	Método 1	Método 2	Método 3	Método 4
Varianza	1.8289×10^3	1.8379×10^3	1.8364×10^3	1.8267×10^3	1.8269×10^3

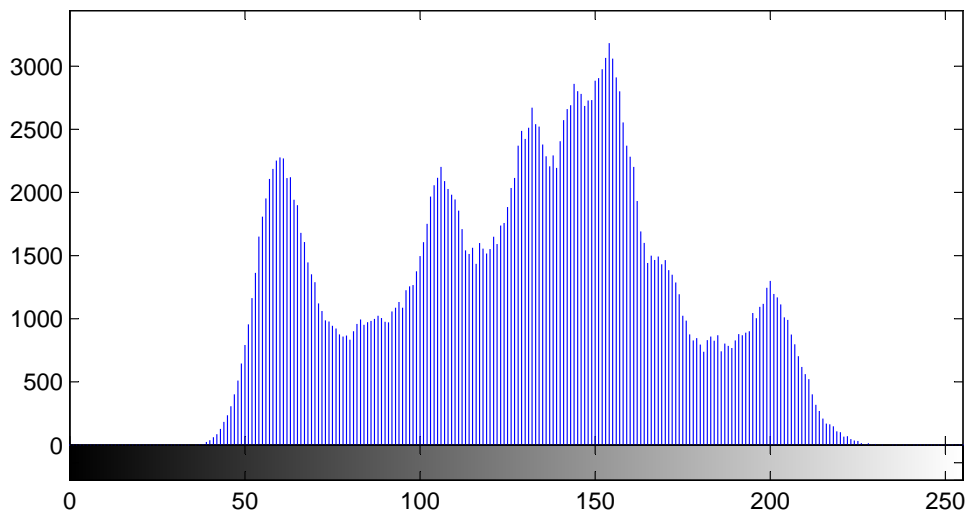
Tabla 4.2: Varianza de las imágenes marcadas de "lena".

Como mencionamos anteriormente el histograma nos permite visualizar la frecuencia de ocurrencia de cada valor de pixel. La figura 4.1 muestra el histograma de la imagen original, que nos servirá para comparar los histogramas de las imágenes marcadas.



(a) Imagen original Lena

Histograma imagen original lena



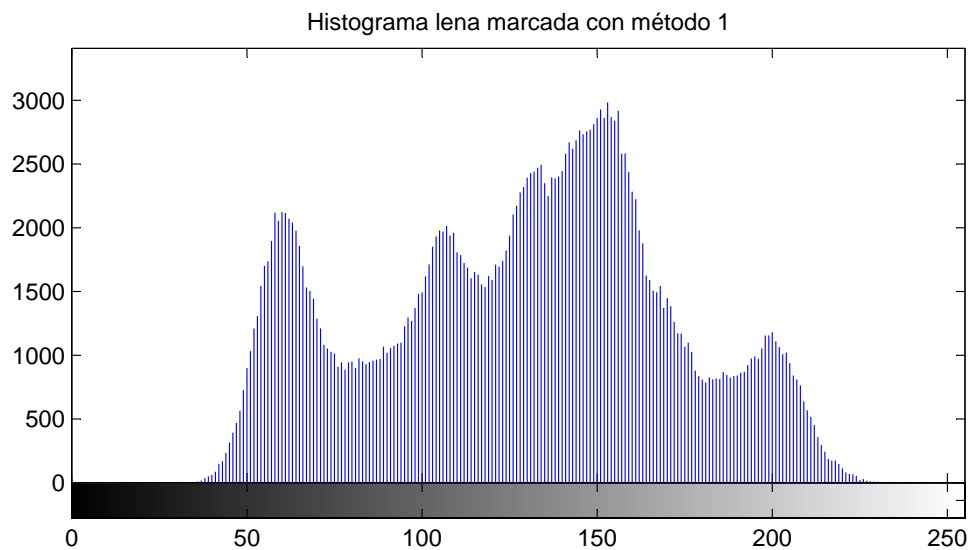
(b) Histograma

Figura 4.1: Histograma de la imagen.

La figura 4.2 muestra la imagen marcada con el primer método y su respectivo histograma. La imagen marcada tiene una PSNR de 38.165[dB] y si observamos los histogramas de las dos imágenes, veremos que no existe mucha diferencia entre ellos, sin embargo si comparamos los valores de la varianza entre ellos veremos que si existe una variación aunque no es muy grande.



(a) Imagen marcada con el primer método



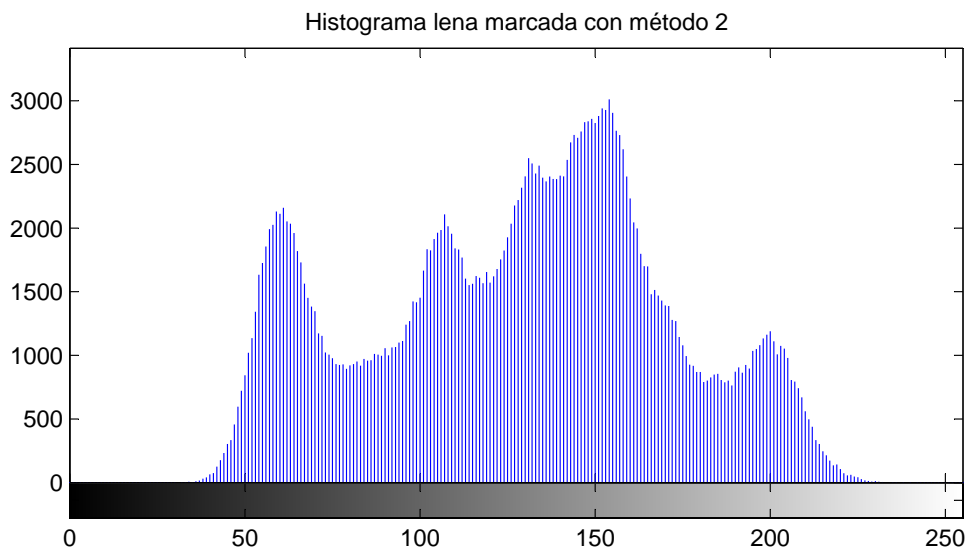
(b) Histograma

Figura 4.2: Histograma de la imagen.

La figura 4.3 muestra la imagen marcada con el segundo método y su respectivo histograma. La calidad visual de la imagen marcada mejora un poco respecto al primer método ya que se logra un valor de PSNR de 38.95[dB], al igual que la imagen marcada con el primer método la diferencia entre los histogramas no es muy significativa.



(a) Imagen marcada con el segundo método



(b) Histograma

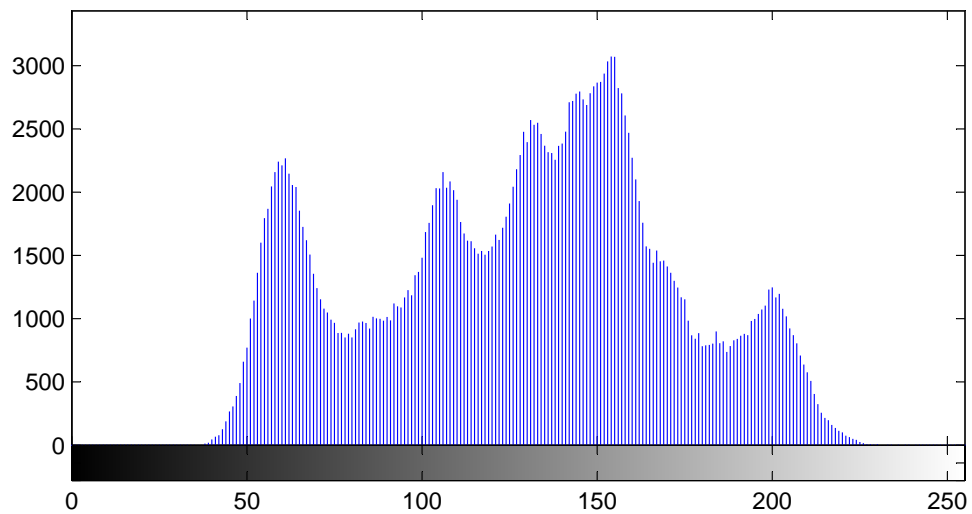
Figura 4.3: Histograma de la imagen.

La figura 4.4 muestra la imagen marcada con el tercer método y su respectivo histograma. La calidad visual de imagen marcada mejora considerablemente con respecto a los dos primeros métodos ya que se obtiene un imagen marcada con una PSNR igual a 45.036[dB].



(a) Imagen marcada con el tercer método

Histograma lena marcada con método 3



(b) Histograma

Figura 4.4: Histograma de la imagen.

La figura 4.5 muestra la imagen marcada con el cuarto método y su respectivo histograma. Esta imagen es la que posee mejor calidad visual con respecto a las demás imágenes marcadas ya que posee una PSNR igual a 45.92[dB], también es la que posee el histograma que es más semejante al de la imagen original. Podemos notar que entre más mejora la calidad de la imagen, el histograma se asemeja más al de la imagen original, hay una relación directa.

4.3.2. Variación de la PSNR

Las figuras 4.6 y 4.7 muestran las gráficas de la variación de la PSNR vs el factor de multiplicación. Podemos notar que al aumentar el factor de multiplicación la PSNR disminuye, aparentemente con los métodos 1 y 2 podríamos alcanzar imágenes con una PSNR de aproximadamente 67[dB] utilizando un factor de multiplicación suficientemente pequeño, mientras que con los métodos 3 y 4 podríamos alcanzar solo hasta 47[dB]. Pero basta con mirar las figuras 4.8 y 4.9 para deducir que para poder detectar y recuperar correctamente la marca de agua no es posible utilizar factores de multiplicación tan pequeños. Esto se debe a que entre más aumente el factor de multiplicación menor coeficiente de correlación obtendremos de la marca recuperada.

También podemos observar que entre menor sea el número de los coeficientes marcados, como en los métodos 1 y 3 donde la marca solo fue incrustada en el 75% de la banda Y la PSNR de las imágenes marcadas mejora, sin embargo, el coeficiente de correlación del logo recuperado disminuye. Y cuando se incrusta la marca en toda la banda Y la PSNR disminuye pero el coeficiente de correlación del logo recuperado mejora.

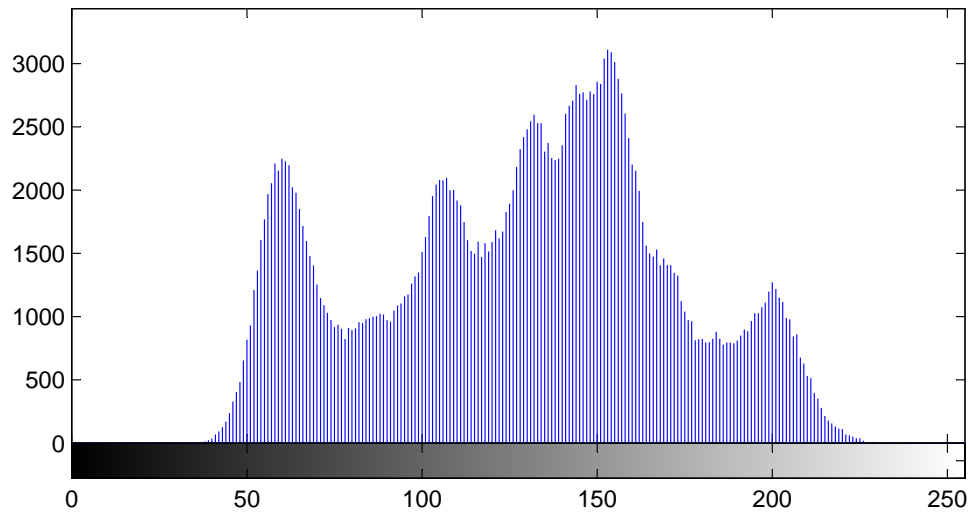
Con los métodos 3 y 4 propuestos se puede alcanzar una mejor calidad visual de la imagen marcada ya que la respuesta del detector es mucho más rápida y nos permite utilizar valores de factor de multiplicación muy pequeños de modo que podemos obtener imágenes marcadas con valores de PSNR arriba de 40[dB] y con valores de coeficiente de correlación arriba de 0.9.

La marca de agua binaria seleccionada es mostrada en las figuras 4.10a, 4.11a, 4.12a, 4.13a. Las imágenes marcadas de lena con cada uno de los métodos con coeficientes de correlación y PSNR que fueron presentados en la tabla 4.1 se muestran en las figuras 4.10b, 4.11b, 4.12b, 4.13b. Los logos recuperados de las imágenes son también mostrados en las figuras 4.10c, 4.11c, 4.12c, 4.13c. Estos logos



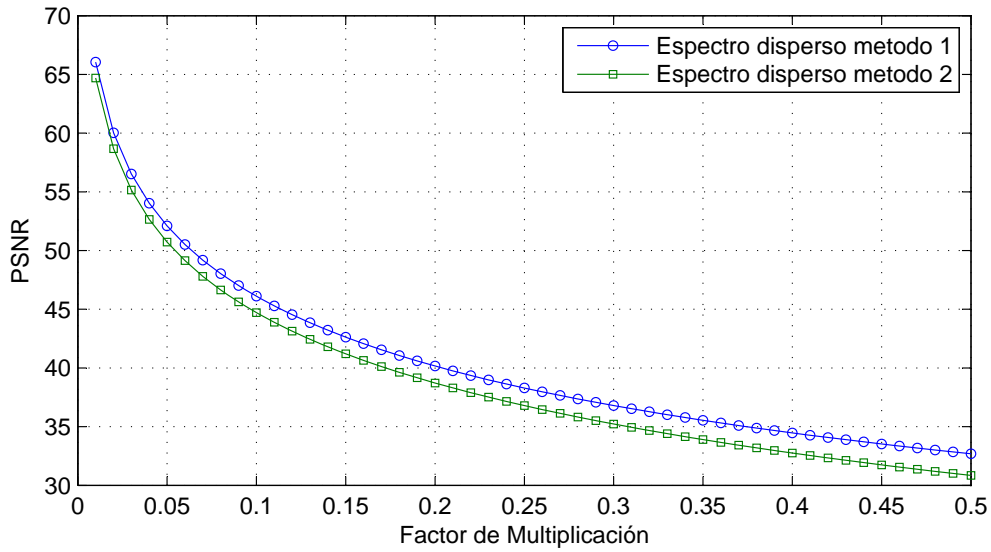
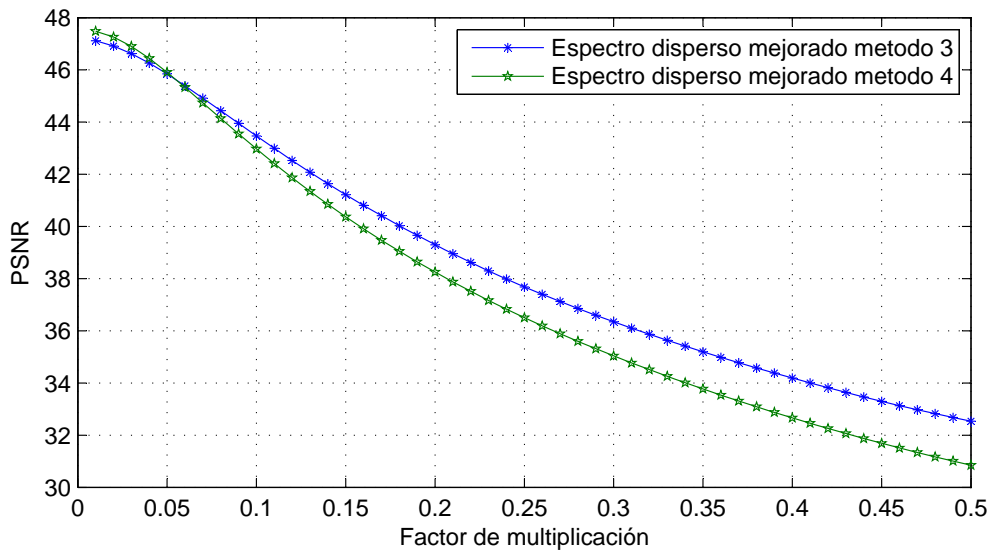
(a) Imagen marcada con el cuarto método

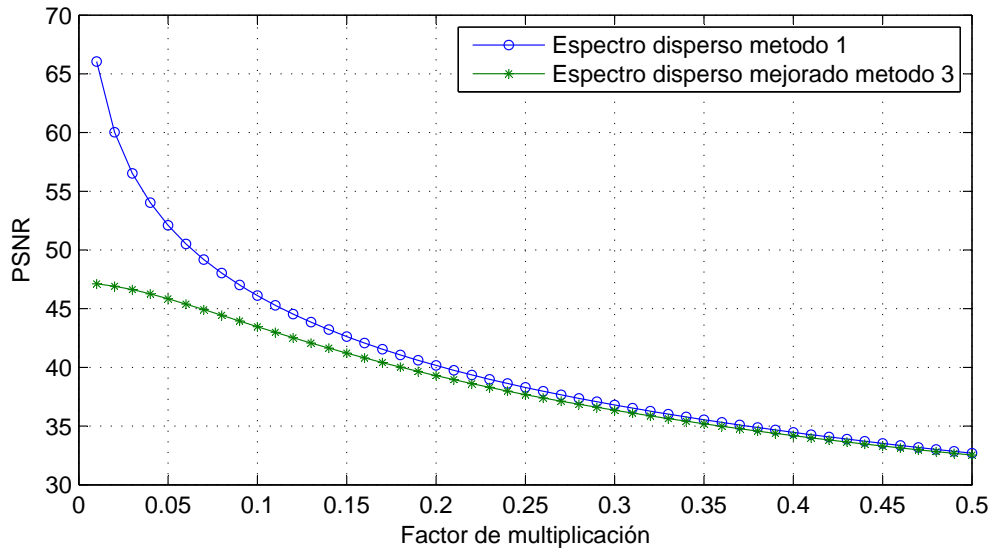
Histograma lena marcada con método 4



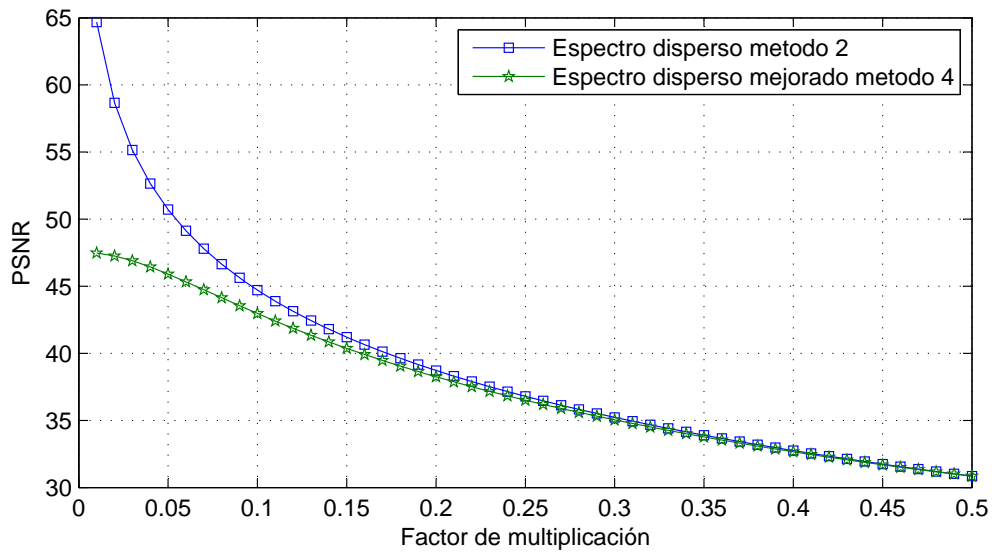
(b) Histograma

Figura 4.5: Histograma de la imagen.

(a) Variación de la PSNR respecto a α para los métodos 1 y 2(b) Variación de la PSNR respecto a α para los métodos 3 y 4Figura 4.6: Gráficos de la variación de la PSNR respecto a α para los diferentes métodos.

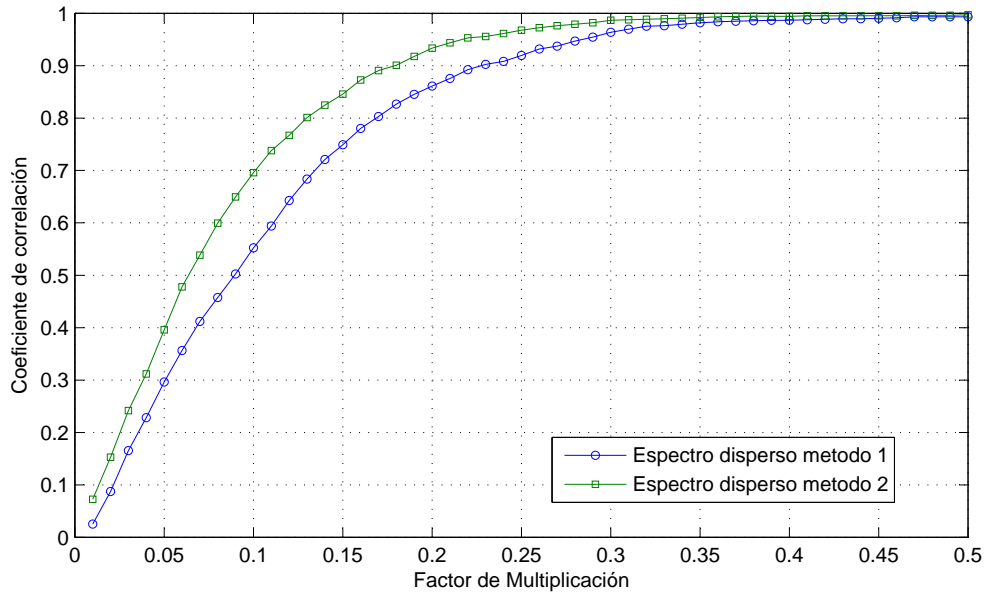
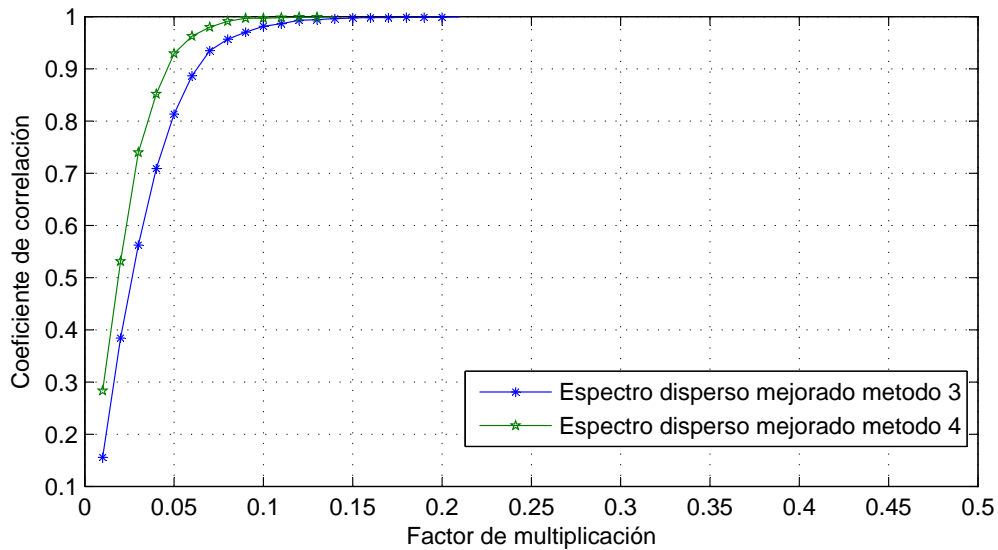


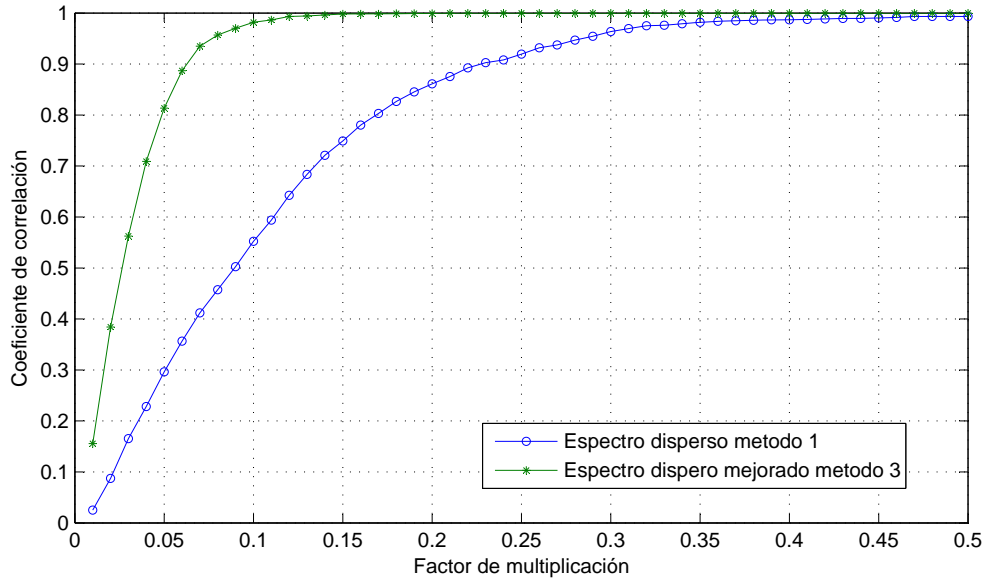
(a) Comparación del método 1 con el método propuesto 3



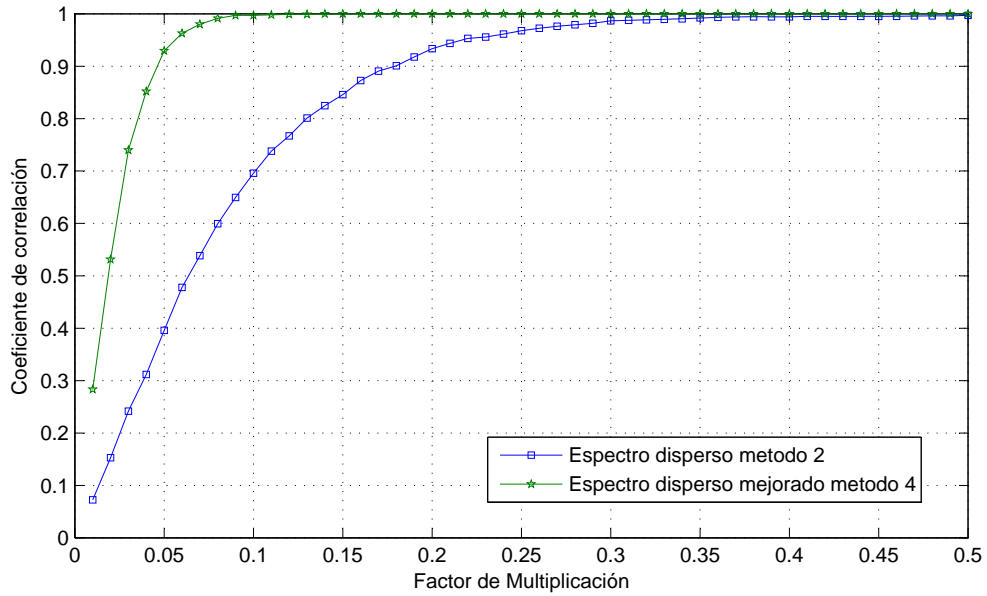
(b) Comparación del método 2 con el método propuesto 4

Figura 4.7: Gráficos de la variación de la PSNR respecto a α para los diferentes métodos.

(a) Variación del coeficiente de correlación respecto a α para los métodos 1 y 2(b) Variación del coeficiente de correlación respecto a α para los métodos 2 y 4Figura 4.8: Gráficos de la variación del coeficiente de correlación respecto a α para los diferentes métodos.



(a) Comparación del método 1 con el método propuesto 3



(b) Comparación del método 2 con el método propuesto 4

Figura 4.9: Gráficos de la variación del coeficiente de correlación respecto a α para los diferentes métodos.



Figura 4.10: Gráfico de los logos original y recuperado de la imagen marcada utilizando el primer método. (a) muestra el logo original. (b) muestra la imagen marcada. (c) muestra el logo recuperado. (d) muestra el logo recuperado después de un filtro de mediana y (e) muestra el logo recuperado sin una llave autorizada.



Figura 4.11: Gráfico de los logos original y recuperado de la imagen marcada utilizando el segundo método. (a) muestra el logo original. (b) muestra la imagen marcada. (c) muestra el logo recuperado. (d) muestra el logo recuperado después de un filtro de mediana y (e) muestra un logo recuperado sin una llave autorizada.



Figura 4.12: Gráfico de los logos original y recuperado de la imagen marcada utilizando el tercer método. (a) muestra el logo original. (b) muestra la imagen marcada. (c) muestra el logo recuperado. (d) muestra el logo recuperado después de un filtro de mediana y (e) muestra un logo recuperado sin una llave autorizada.



Figura 4.13: Gráfico de los logos original y recuperado de la imagen marcada utilizando el cuarto método. (a) muestra el logo original. (b) muestra la imagen marcada. (c) muestra el logo recuperado. (d) muestra el logo recuperado después de un filtro de mediana y (e) muestra un logo recuperado sin una llave autorizada.

recuperados son filtrados con un filtro de mediana para obtener una mejor calidad visual y son mostrados en las figuras 4.10d, 4.11d, 4.12d, 4.13d. Las figuras 4.10e, 4.11e, 4.12e, 4.13e muestran los logos recuperados con una llave no autorizada.

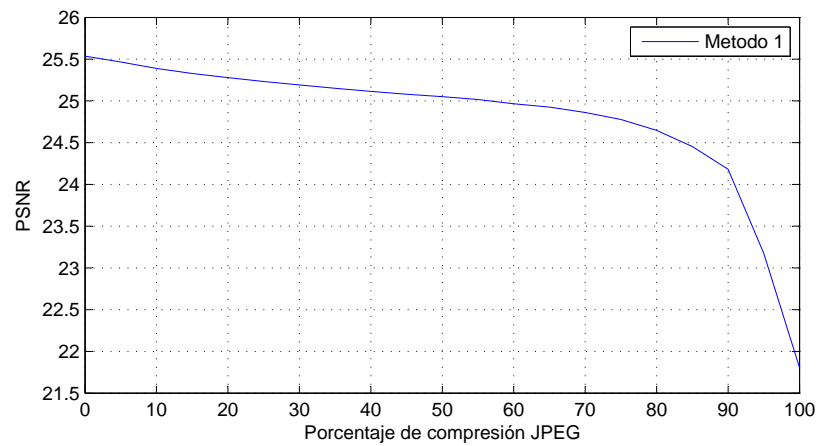
4.3.3. Prueba de compresión JPEG

La figura 4.14a muestra la variación de la PSNR respecto al porcentaje de compresión JPEG utilizando el primer método, la PSNR disminuye considerablemente aunque el porcentaje de compresión sea del 0%, ya que disminuye cerca de 25.5[dB]. La figura 4.14b muestra la variación del coeficiente de correlación respecto al porcentaje de compresión utilizando el primer método, podemos observar que el logo puede ser recuperado de forma adecuada con porcentajes de compresión de hasta 80%.

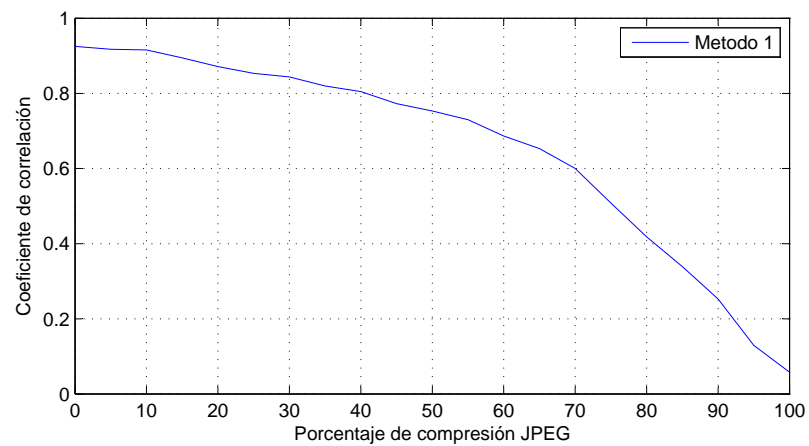
La figura 4.15a muestra la variación de la PSNR respecto al porcentaje de compresión JPEG utilizado el segundo método, esta gráfica es muy similar a la del primer método, ya que no difieren mucho los valores de un método a otro. La figura 4.15b muestra la variación del coeficiente de correlación respecto al porcentaje de compresión utilizando el segundo método. A igual que el primer método el logo puede ser recuperado de forma adecuada con porcentajes de compresión de hasta el 80%.

La figura 4.16a muestra la variación de la PSNR respecto al porcentaje de compresión JPEG utilizando el tercer método, el comportamiento de esta gráfica es muy parecido a la de los dos primeros métodos de hecho no hay grandes variaciones de una a otra gráfica. La figura 4.16b muestra la variación del coeficiente de correlación respecto al porcentaje de compresión JPEG, podemos observar en la gráfica que el método 3 es muy vulnerable a la compresión JPEG ya que solo podríamos recuperar el logo de forma adecuada para porcentajes de compresión menores al 35%.

La figura 4.17a muestra la variación de la PSNR respecto al porcentaje de compresión JPEG utilizando el cuarto método, al igual que en los métodos anteriores la variación de la PSNR es muy similar a los otros métodos, sin embargo, como podemos observar de la figura 4.17b al igual que el tercer método la detección empeora y solo podemos recuperar el logo de forma adecuada para porcentajes de compresión menores de 35%, mientras que, en los dos primeros métodos podemos utilizar porcentajes de compresión menores del 80%. La diferencia que existe entre los primeros dos métodos y los propuestos, es que en los métodos propuestos hemos modulado la amplitud de las secuencias pseudoaleatorias lo que no permite alcanzar

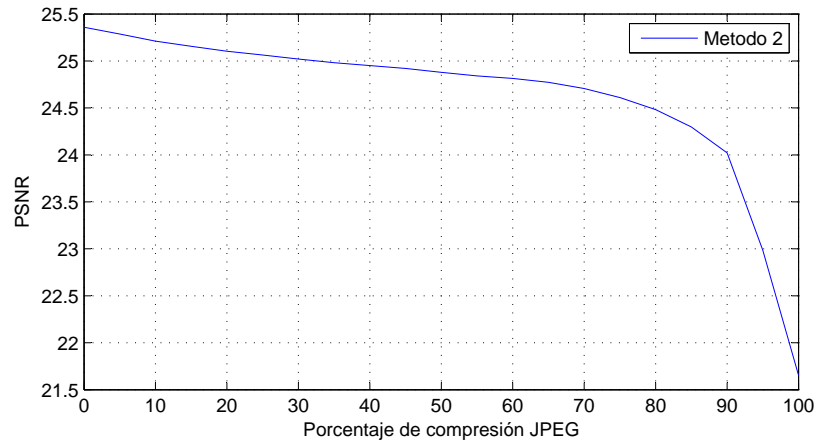


(a) Variación de la PSNR respecto al porcentaje de compresión JPEG

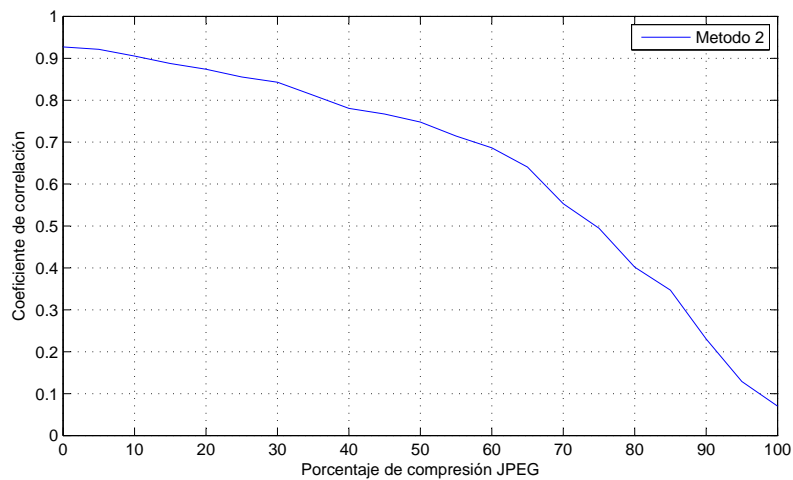


(b) Variación del coeficiente de correlación respecto al porcentaje de compresión JPEG

Figura 4.14: Resultados de compresión JPEG utilizando el primer método.

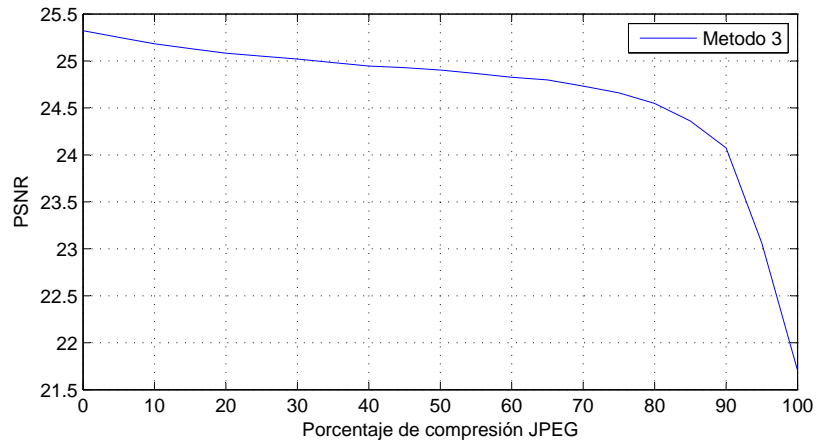


(a) Variación de la PSNR respecto al porcentaje de compresión JPEG

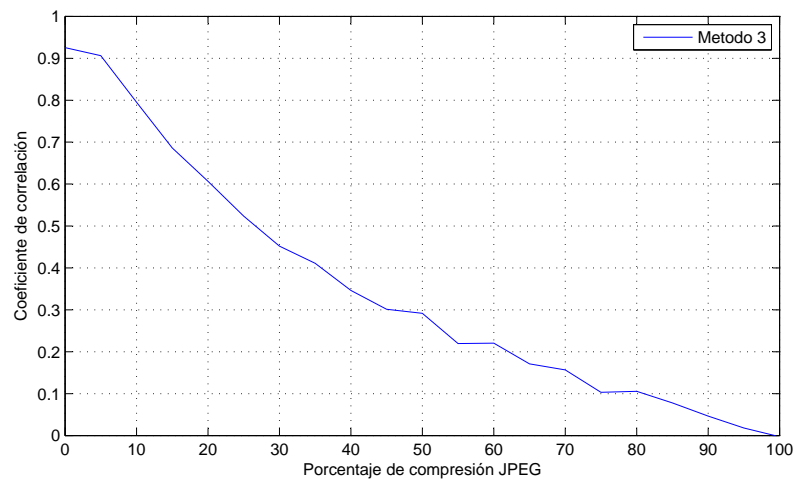


(b) Variación del coeficiente de correlación respecto al porcentaje de compresión JPEG

Figura 4.15: Resultados de compresión JPEG utilizando el segundo método.



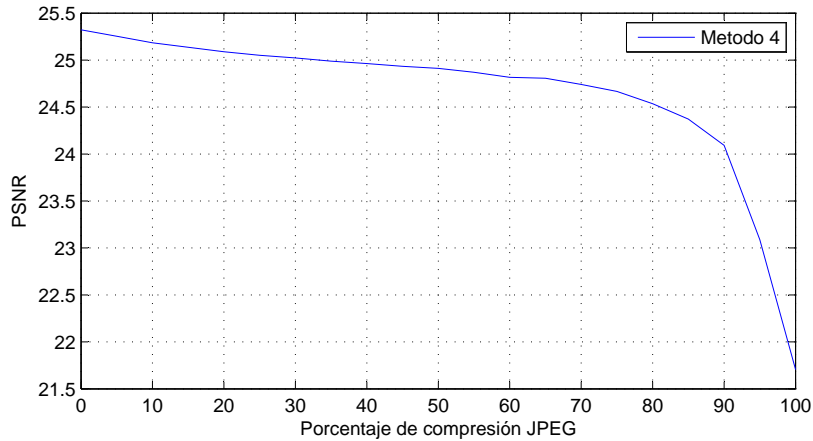
(a) Variación de la PSNR respecto al porcentaje de compresión JPEG



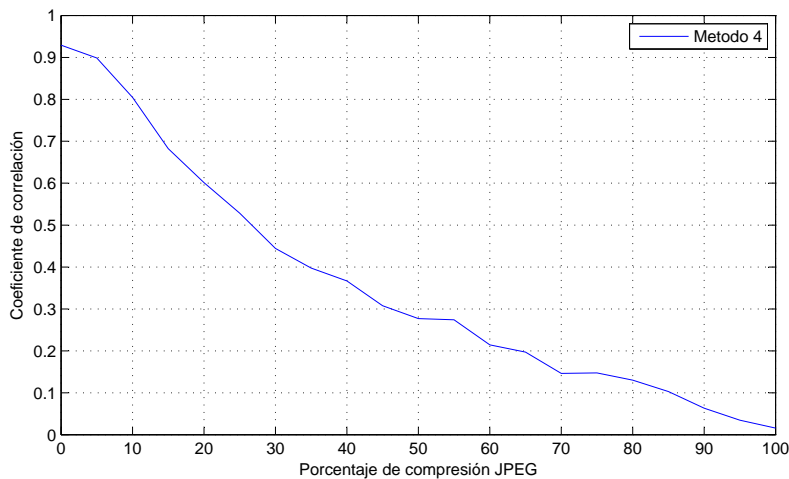
(b) Variación del coeficiente de correlación respecto al porcentaje de compresión JPEG

Figura 4.16: Resultados de compresión JPEG utilizando el tercer método.

imágenes marcadas con una alta PSNR, sin embargo, como vemos de los resultados de la compresión jpeg, la detección empeora y se vuelve más vulnerable a los ataques.



(a) Variación de la PSNR respecto al porcentaje de compresión JPEG



(b) Variación del coeficiente de correlación respecto al porcentaje de compresión JPEG

Figura 4.17: Resultados de compresión JPEG utilizando el cuarto método.

Las figuras 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.24 y 4.25 muestran ejemplos de los logos recuperados de las imágenes marcadas con factores de compresión JPEG del 80% y 20% para los cuatro métodos. Nótese que para los métodos 3 y 4 los logos ya no tienen ningún patrón visual con el logo original para el factor de compresión del 80%.



Figura 4.18: (a) Lena marcada con porcentaje de compresión de 80%. (b) Logo recuperado de la imagen con porcentaje de compresión de 80% utilizando método 1.

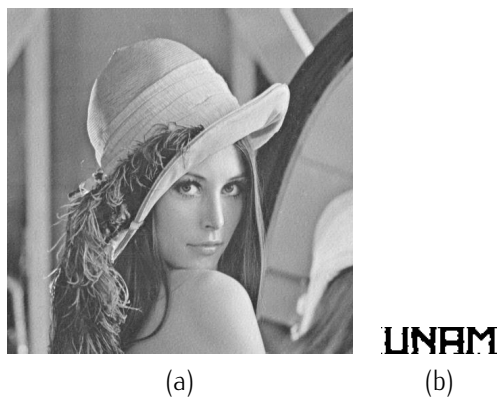


Figura 4.19: (a) Lena marcada con porcentaje de compresión de 20%. (b) Logo recuperado de la imagen con porcentaje de compresión de 20% utilizando método 1.



Figura 4.20: (a) Lena marcada con porcentaje de compresión de 80%. (b) Logo recuperado de la imagen con porcentaje de compresión de 80% utilizando método 2.

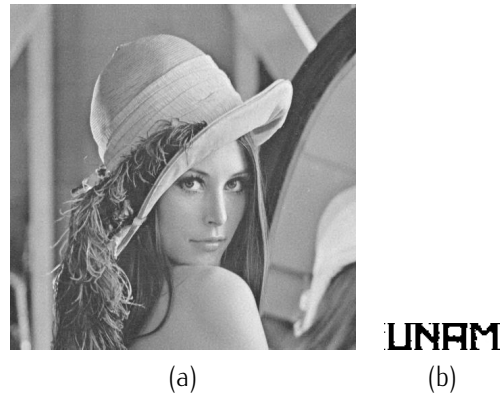


Figura 4.21: (a) Lena marcada con porcentaje de compresión de 20%. (b) Logo recuperado de la imagen con porcentaje de compresión de 20% utilizando método 2.



Figura 4.22: (a) Lena marcada con porcentaje de compresión de 20%. (b) Logo recuperado de la imagen con porcentaje de compresión de 20% utilizando método 3.



Figura 4.23: (a) Lena marcada con porcentaje de compresión de 80%. (b) Logo recuperado de la imagen con porcentaje de compresión de 80% utilizando método 3.



Figura 4.24: (a) Lena marcada con porcentaje de compresión de 20%. (b) Logo recuperado de la imagen con porcentaje de compresión de 20% utilizando método 4.



Figura 4.25: (a) Lena marcada con porcentaje de compresión de 80%. (b) Logo recuperado de la imagen con porcentaje de compresión de 80% utilizando método 4.

4.3.4. Prueba de Ruido

La figura 4.26 muestra la gráfica de la variación del coeficiente de correlación respecto al porcentaje de ruido agregado a la imagen utilizando el primer método, vemos que este tipo de ataque es más severo que la compresión JPEG ya que para un porcentaje de ruido de apenas el 10% la correlación que guarda el logo recuperado con el logo original es de apenas 0.4, después de un porcentaje del 15% difícilmente se recuperara un logo que presente un patrón visual similar al logo original.

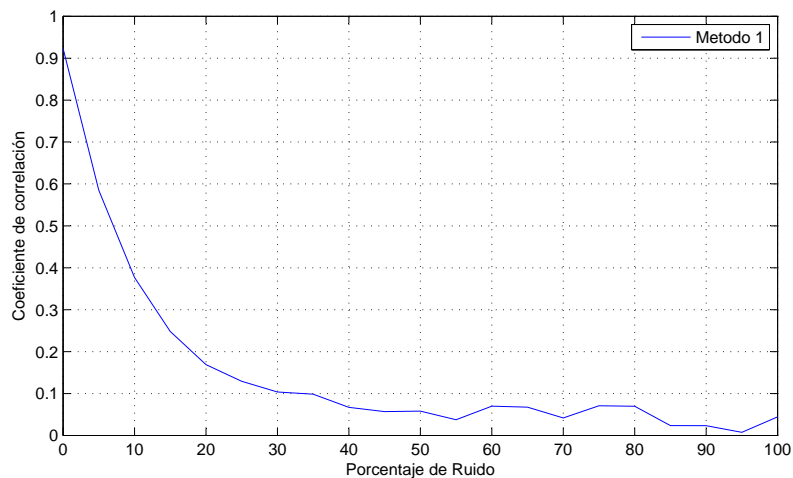


Figura 4.26: Variación del coeficiente de correlación respecto al porcentaje de ruido utilizando método 1.

La figura 4.27 muestra la gráfica de la variación del coeficiente de correlación respecto al porcentaje de ruido agregado a la imagen utilizando el segundo método, como podemos ver el comportamiento de la gráfica es muy similar al del primer método y al igual que el primer método difícilmente se recuperara un logo con un patrón visual similar al logo original para factores de ruido mayores del 15%.

La figura 4.28 muestra la gráfica de la variación del coeficiente de correlación respecto al porcentaje de ruido agregado a la imagen utilizando el tercer método, analizando la gráfica podemos observar que la calidad visual del logo recuperado disminuye más rápidamente que los dos primeros métodos, y para poder recuperar logos con un patrón visual similar al logo original necesitamos un porcentaje de ruido menor al 5%.

La figura 4.29 muestra la gráfica de la variación del coeficiente de correlación respecto al porcentaje

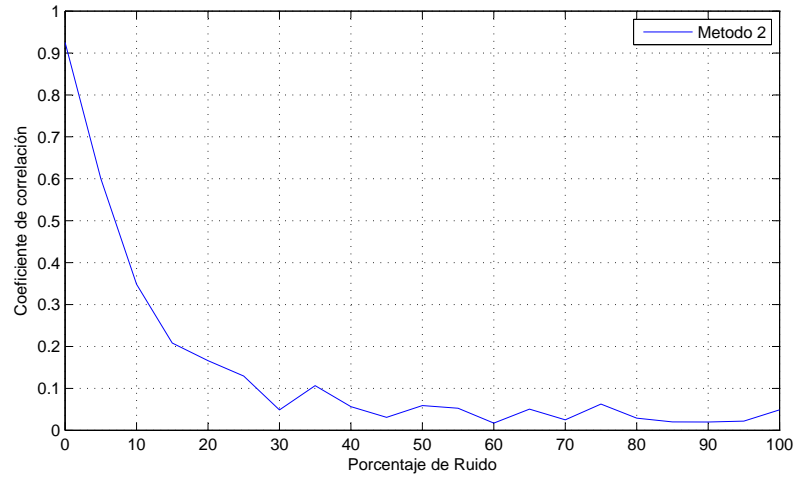


Figura 4.27: Variación del coeficiente de correlación respecto al porcentaje de ruido utilizando método 2.

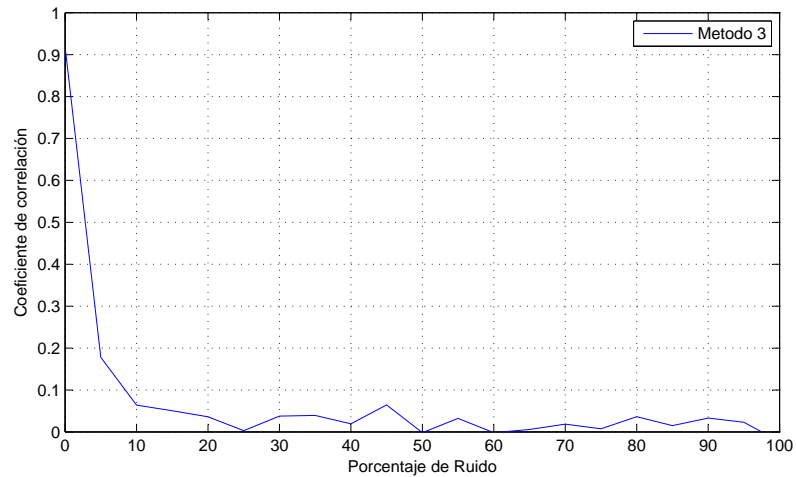


Figura 4.28: Variación del coeficiente de correlación respecto al porcentaje de ruido utilizando método 3.

de ruido agregado a la imagen marcada utilizando el cuarto método. Al igual que en el tercer método la calidad visual de los logos recuperados disminuye más rápidamente que los dos primeros métodos y para poder obtener un logo con un patrón visual similar al logo original necesitamos un porcentaje de ruido menor al 5%, mientras que para los dos primeros métodos se puede utilizar porcentajes de ruido menores al 15%.

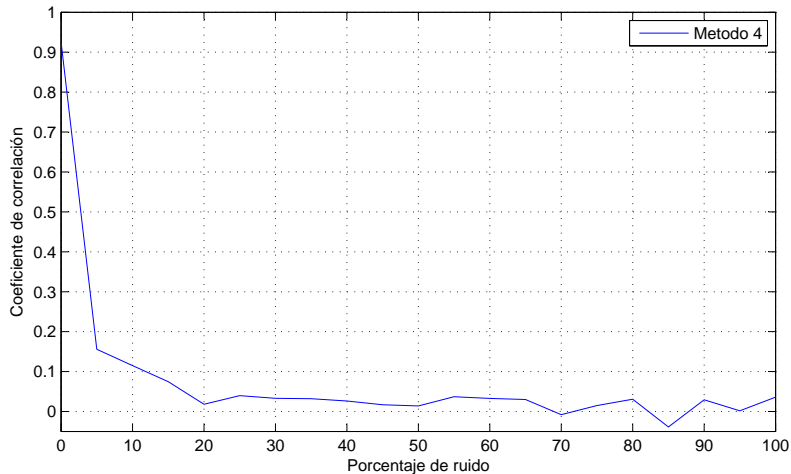


Figura 4.29: Variación del coeficiente de correlación respecto al porcentaje de ruido utilizando método 4.

Las figuras 4.30, 4.31, 4.32, 4.33 muestran las imágenes marcadas con un porcentaje de ruido del 10%, los dos primeros métodos permiten recuperar el logo con un patrón visual reconocible, pero para los métodos 3 y 4, los logos recuperados ya no poseen un patrón visual con el logo original.

4.3.5. Prueba de Filtro de Mediana

La figura 4.34 muestra la gráfica de la variación del coeficiente de correlación respecto al tamaño de filtro de mediana utilizando el primer método, ese filtro obtiene la mediana de los valores de los píxeles dentro una ventana y sustituye el valor del píxel central por el resultado. Este tipo de ataque como podemos observar de la gráfica 4.34 afecta demasiado la recuperación de la marca. Por ejemplo solo podemos recuperar el logo correctamente con un filtro de mediana con un tamaño de ventana 3 para mayores ventanas la recuperación ya no es exitosa.

La figura 4.35 muestra la gráfica de la variación del coeficiente de correlación respecto al tamaño de

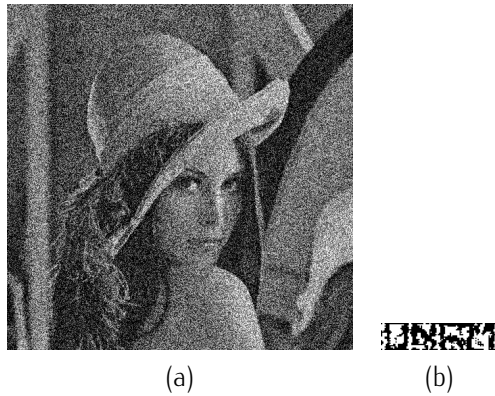


Figura 4.30: (a) Lena marcada con porcentaje de ruido del 10%. (b) Logo recuperado de la imagen con porcentaje de ruido del 10% utilizando método 1.

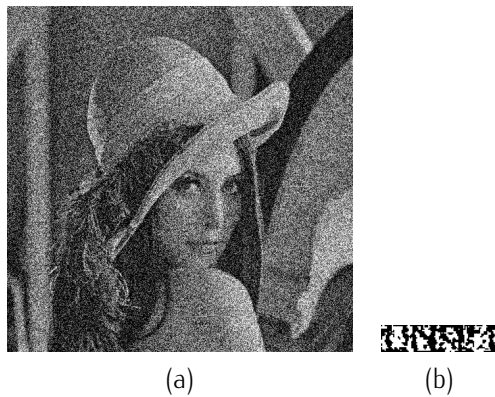


Figura 4.31: (a) Lena marcada con porcentaje de ruido del 10%. (b) Logo recuperado de la imagen con porcentaje de ruido del 10% utilizando método 2.

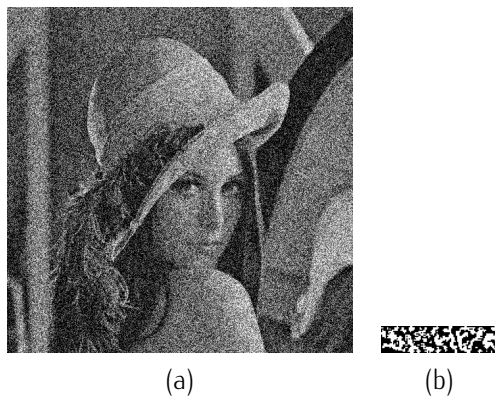


Figura 4.32: (a) Lena marcada con porcentaje de ruido del 10%. (b) Logo recuperado de la imagen con porcentaje de ruido del 10% utilizando método 3.



Figura 4.33: (a) Lena marcada con porcentaje de ruido del 10%. (b) Logo recuperado de la imagen con porcentaje de ruido del 10% utilizando método 4.

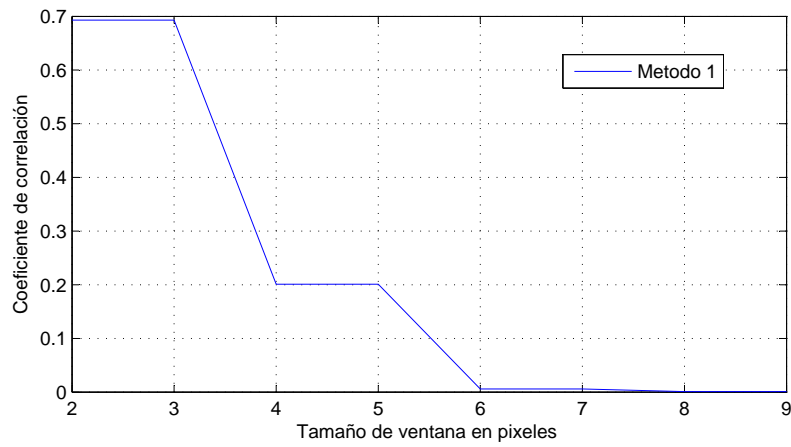


Figura 4.34: Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro utilizando método 1.

filtro de mediana utilizando el segundo método, esta gráfica es muy parecida a la del primer método y solo se puede recuperar el logo satisfactoriamente para filtros de tamaño de ventana menores o iguales a 3.

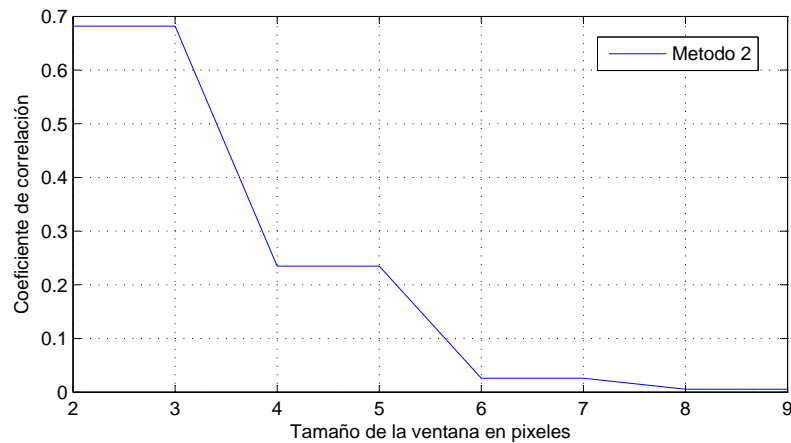


Figura 4.35: Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro utilizando método 2.

La figura 4.36 muestra la gráfica de la variación del coeficiente de correlación respecto al tamaño de filtro de mediana utilizando el tercer método. Al igual que los dos primeros métodos el logo puede ser recuperado de la imagen solamente si se le es aplicado un filtro de mediana de tamaño de ventana inferior a 3, aunque no con la misma calidad visual que los con dos primeros métodos.

La figura 4.37 muestra la gráfica de la variación del coeficiente de correlación respecto al tamaño del filtro de mediana utilizando el cuarto método, el comportamiento de la gráfica es muy similar al método 3, y solo se puede recuperar el logo para un filtro de tamaño de ventana inferior a 3.

Las figuras 4.38, 4.39, 4.40 y 4.41 muestran las imágenes marcadas después de un filtro de mediana de tamaño 3×3 y sus respectivos logos recuperados utilizando los cuatro métodos. Nótese que los dos primeros métodos recuperan los logos con una calidad visual mejor.

4.3.6. Prueba de Recorte

La figura 4.42 muestra la variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando el método 1. La gráfica muestra grandes saltos con respecto a los diferentes

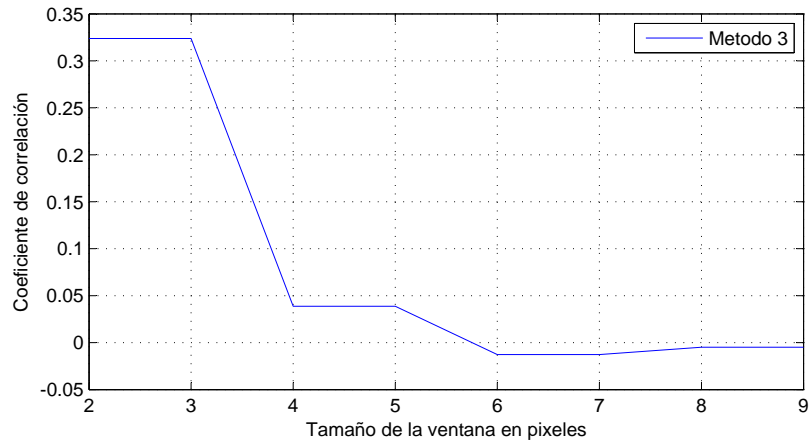


Figura 4.36: Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro utilizando método 3.

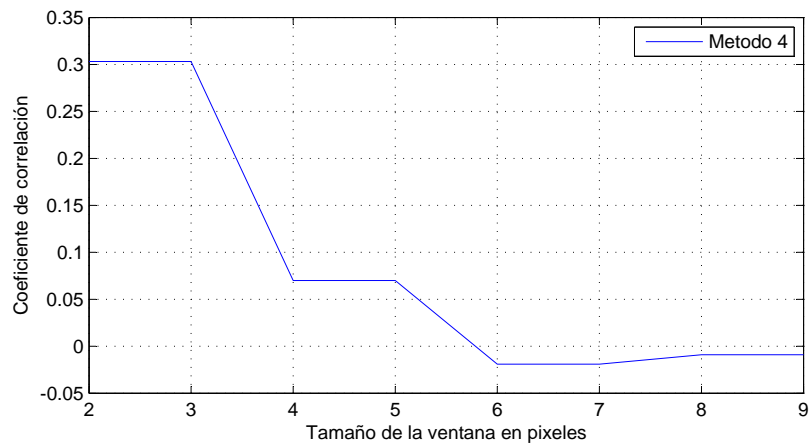


Figura 4.37: Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro utilizando método 4.

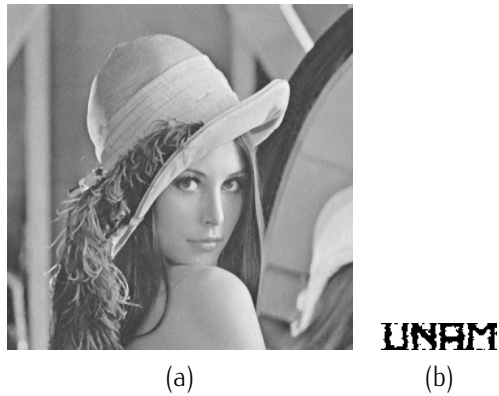


Figura 4.38: (a) Imagen marcada después ser aplicada a un filtro de mediana de 3x3. (b) Logo recuperado de la imagen filtrada utilizando método 1.



Figura 4.39: (a) Imagen marcada después ser aplicada a un filtro de mediana de 3x3. (b) Logo recuperado de la imagen filtrada utilizando método 2.



Figura 4.40: (a) Imagen marcada después ser aplicada a un filtro de mediana de 3x3. (b) Logo recuperado de la imagen filtrada utilizando método 3.



Figura 4.41: (a) Imagen marcada después ser aplicada a un filtro de mediana de 3x3. (b) Logo recuperado de la imagen filtrada utilizando método 4.

porcentajes de recorte de la imagen, pero aun así es posible recuperar el logo adecuadamente con un porcentaje de recorte de hasta el 50%.

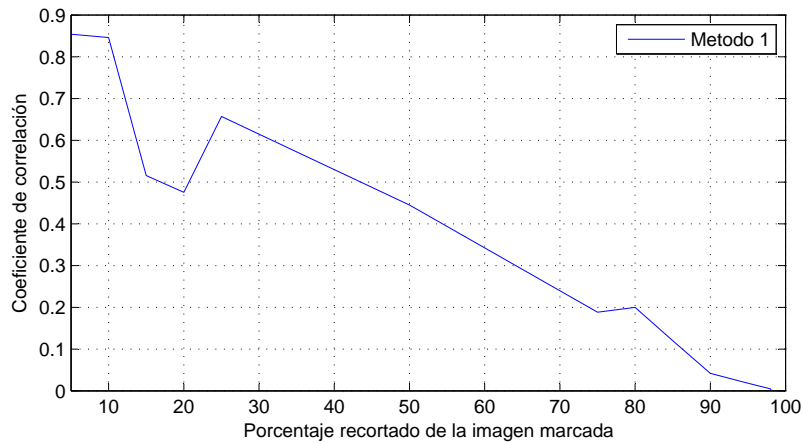


Figura 4.42: Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando método 1.

La figura 4.43 muestra la variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando el método 2. La gráfica es muy semejante al primer método y de igual forma se puede recuperar el logo con hasta un 50% de recorte.

La figura 4.44 muestra la variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando el método 3. La gráfica muestra que el método 3 es muy susceptible al ataque de recorte ya que para poder recuperar de forma adecuada el logo, el porcentaje recortado de la imagen no

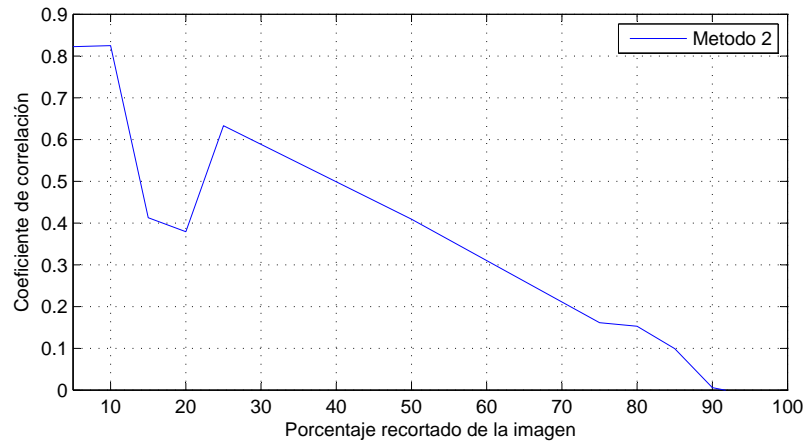


Figura 4.43: Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando método 2.

debe de exceder del 10%.

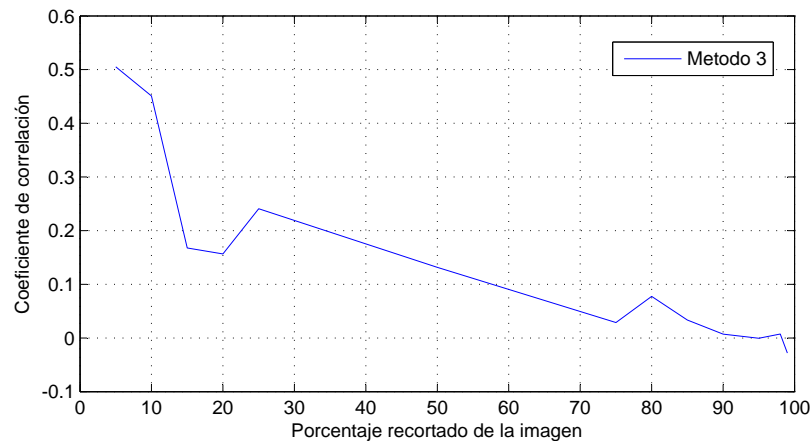


Figura 4.44: Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando método 3.

La figura 4.45 muestra la variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando el método 4. El comportamiento de la gráfica es muy similar a la del método 3, y no existe mucha diferencia entre los resultados obtenidos. Al igual que en el método tres para poder recuperar el logo de forma adecuada el porcentaje recortado de la imagen no debe exceder del 10%.

Las figuras 4.46, 4.47, 4.48 y 4.49 muestran ejemplos de las imágenes marcadas recortadas y con sus

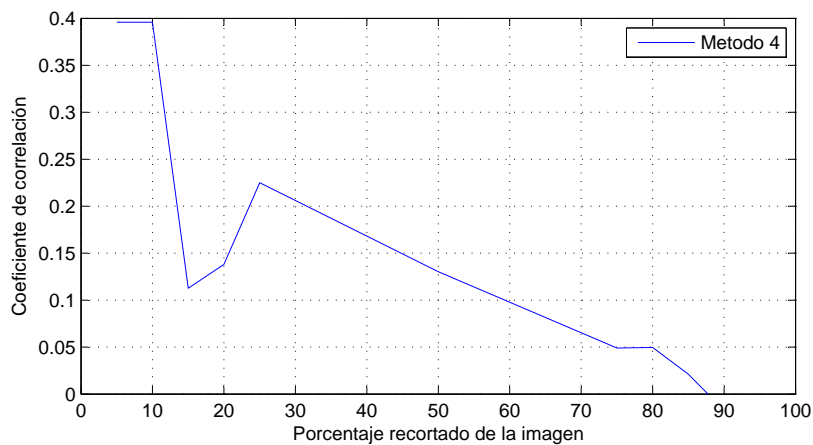


Figura 4.45: Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen utilizando método 4.

logos recuperados utilizando los cuatro métodos. Podemos ver que para los primeros dos métodos los logos son recuperados todavía con un patrón visual parecido al logo original mientras que con los dos últimos la correlación de los logos es muy baja.

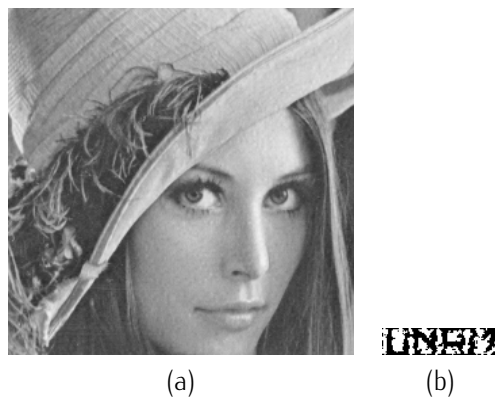


Figura 4.46: (a) Imagen marcada con 50% de recorte. (b) Logo recuperado de la imagen recortada utilizando método 1.

4.3.7. Prueba de Rotación y escalamiento

Los resultados acerca de rotación y escalamiento, no son presentados aquí porque la robustez a grandes rotaciones depende del mecanismo de sincronización y no es característico del algoritmo de inserción y recuperación de la marca de agua [24]. La figura 4.50 muestran ejemplos de ataques de

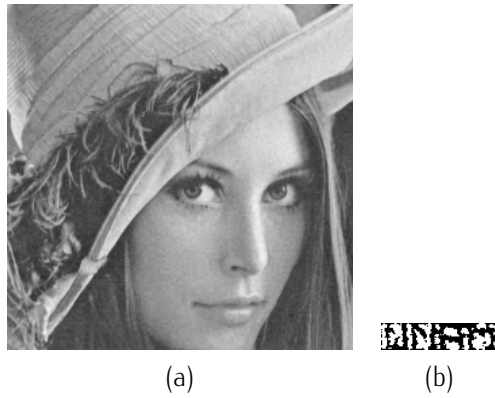


Figura 4.47: (a)Imagen marcada con 50% de recorte. (b) Logo recuperado de la imagen recortada utilizando método 2.

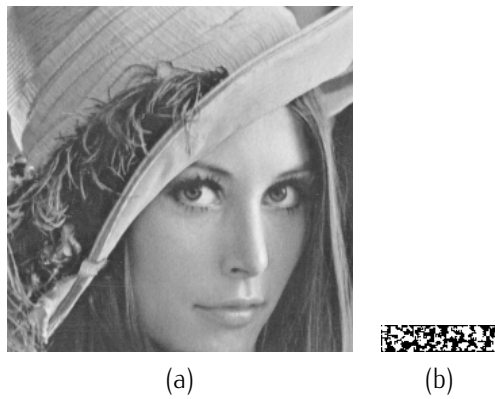


Figura 4.48: (a)Imagen marcada con 50% de recorte. (b) Logo recuperado de la imagen recortada utilizando método 3.

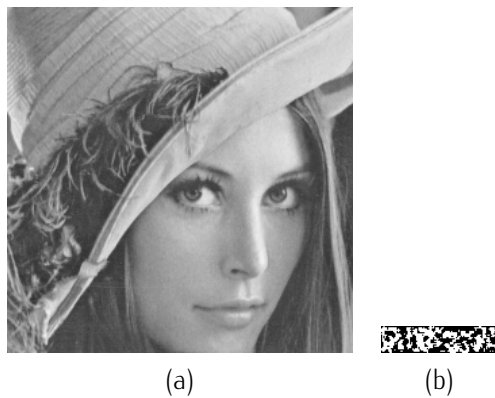


Figura 4.49: (a)Imagen marcada con 50% de recorte. (b) Logo recuperado de la imagen recortada utilizando método 4.

rotación y escalamiento.

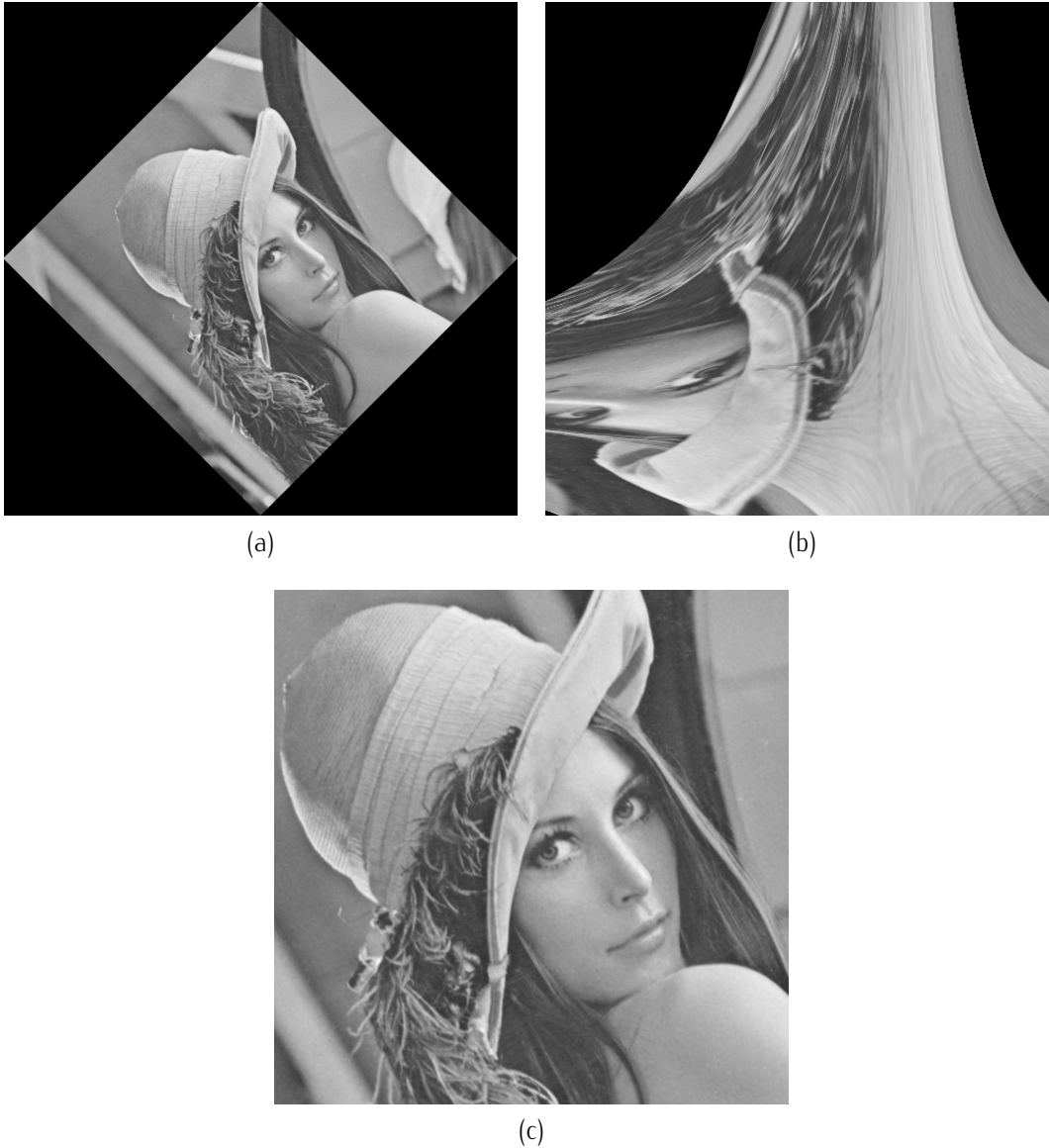


Figura 4.50: Ejemplos de ataques geométricos a la imagen de lena.

4.4. Resultados de las simulaciones para la imagen “barbara”

Después de terminar las simulaciones con la imagen de “lena”, se utilizó la imagen de “barbara” para saber que variantes podría tener el algoritmo de marca de agua. Al igual que el primer algoritmo, después

de obtener las gráficas de la variación de la PSNR respecto al factor de multiplicación, fijamos el valor del factor de multiplicación de modo que pudiésemos recuperar el logo con un coeficiente de correlación de 0.9 respecto al original. Los parámetros obtenidos son mostrados en la tabla 4.3.

Parámetro	Imagen Original	Método 1	Método 2	Método 3	Método 4
α	-	0.39	0.31	0.11	0.078
PSNR[dB]	-	34.72	35.28	40.98	41.81
Correlación	-	0.9	0.9	0.9	0.9
Variancia	2.982×10^3	3.0037×10^3	3.0011×10^3	2.9786×10^3	2.9784×10^3

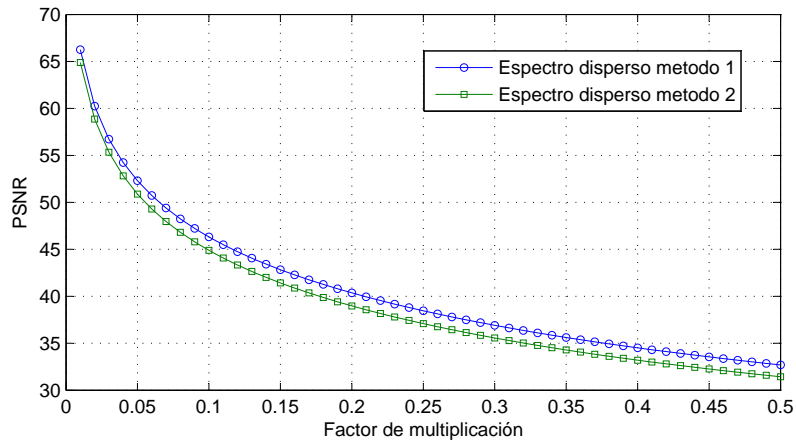
Tabla 4.3: Parámetros importantes de la imagen marcada 'barbara'.

Al igual que con la imagen de lena los métodos que nos ofrecen mejor calidad visual de la imagen marcada son los métodos 3 y 4. Las figuras 4.51 y 4.52 muestran la variación de la PSNR respecto al factor de multiplicación y la variación del coeficiente de correlación respecto al factor de multiplicación. El comportamiento de estas gráficas es muy similar al de las simulaciones con la imagen de "lena", y también la respuesta del detector es mucha más rápida con los métodos 3 y 4 que con los métodos 1 y 2.

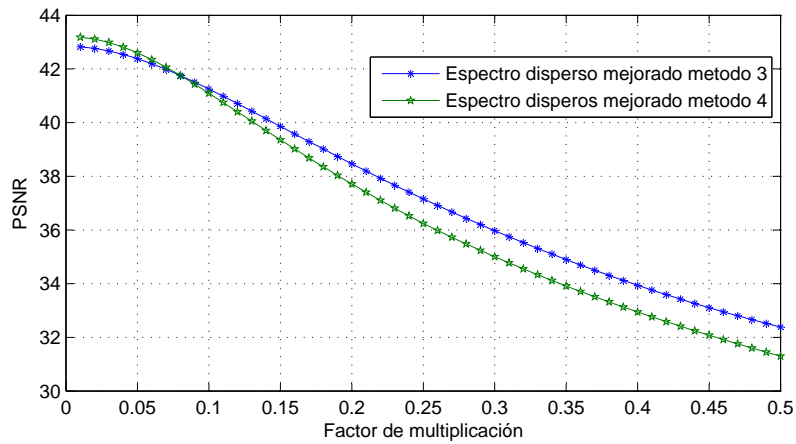
Las figuras 4.53 y 4.54 muestran las gráficas de la variación del coeficiente de correlación respecto al factor de compresión JPEG para los cuatro métodos. Al igual que en las simulaciones de la imagen "lena", los métodos 1 y 2 presentan mayor robustez a este tipo de procesamiento ya que nos permiten recuperar el logo con un patrón visual parecido al original, de las imágenes marcadas con porcentaje de compresión de 80% mientras que para los métodos 3 y 4 solo podemos manejar porcentajes de compresión menores al 60%.

Las figuras 4.55 y 4.56 muestran las gráficas de la variación del coeficiente de correlación respecto al porcentaje de ruido agregado a las imágenes marcadas para los cuatro métodos. Este tipo de ataque es el que más afecta la recuperación de la marca de agua ya que solo es posible recuperarla con buena calidad para porcentajes de ruido menores del 10% para los métodos 1 y 2, mientras que, para los métodos 3 y 4 para porcentajes de ruido menores de 5%.

Las figuras 4.57 y 4.58 muestran las gráficas de la variación del coeficiente de correlación respecto

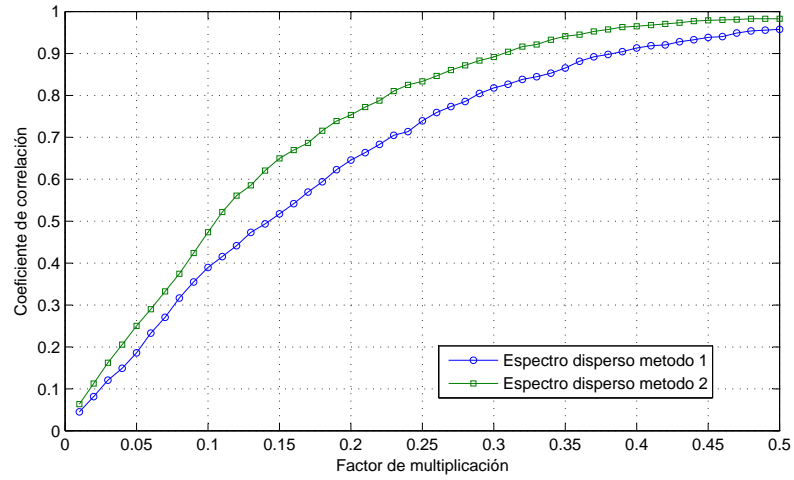
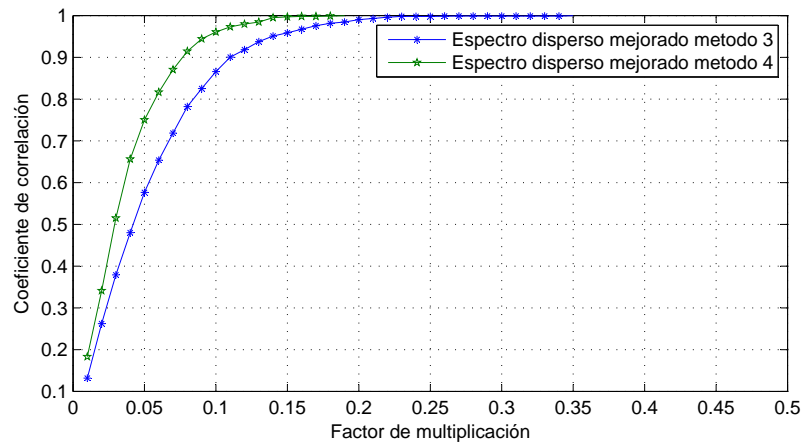


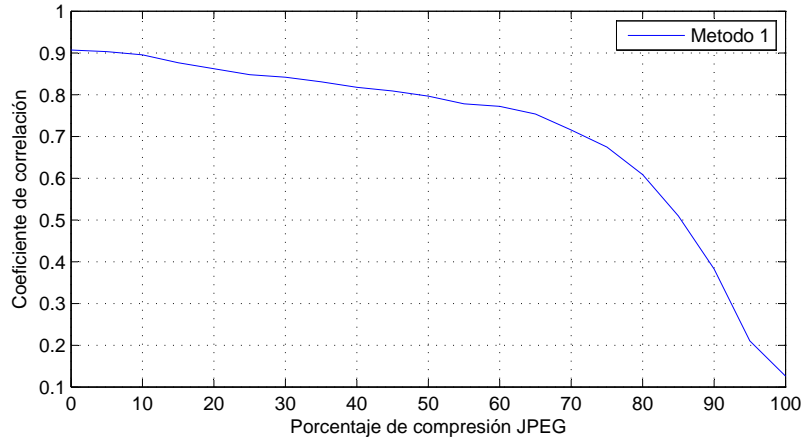
(a) Variación de la PSNR respecto a α para los métodos 1 y 2



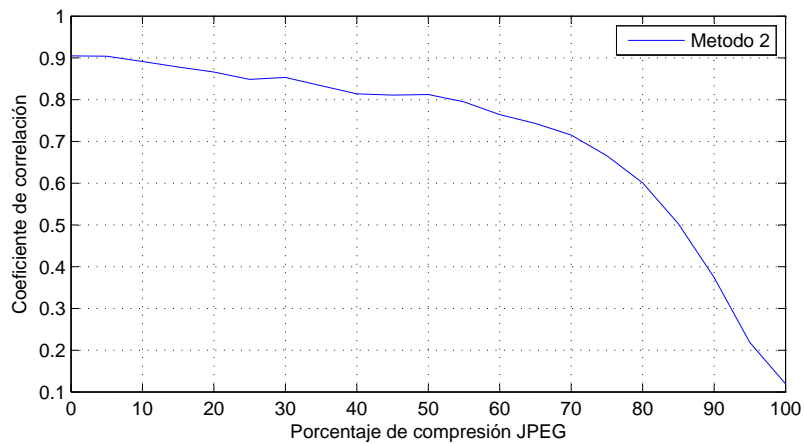
(b) Variación de la PSNR respecto a α para los métodos 3 y 4

Figura 4.51: Gráficos de la variación de la PSNR respecto a α para los diferentes métodos.

(a) Variación del coeficiente de correlación respecto a α para los métodos 1 y 2(b) Variación del coeficiente de correlación respecto a α para los métodos 3 y 4Figura 4.52: Gráficos de la variación del coeficiente de correlación respecto a α para los diferentes métodos.

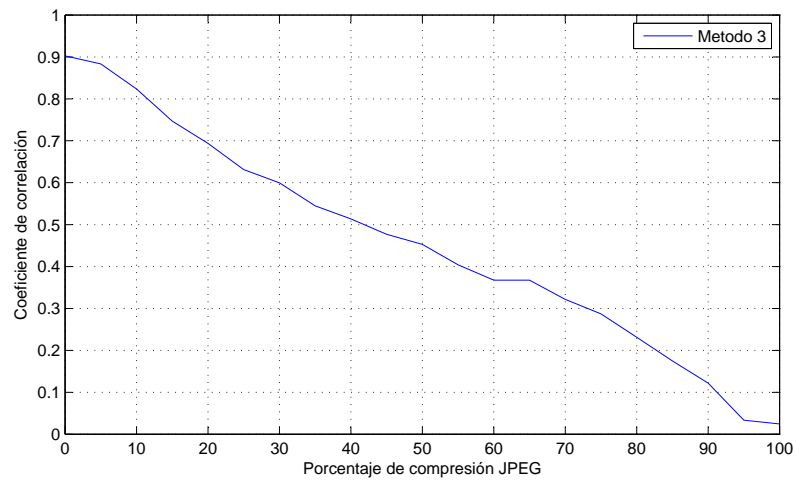


(a) Variación del coeficiente de correlación respecto al porcentaje de compresión JPEG utilizando el método 1

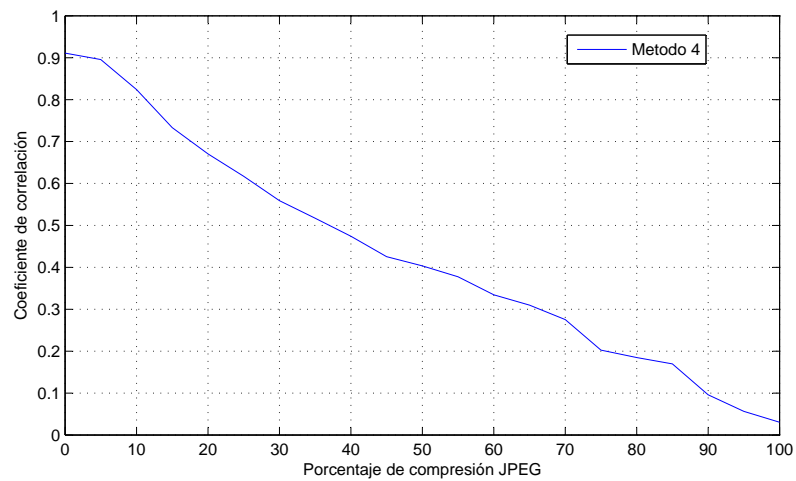


(b) Variación del coeficiente de correlación respecto al porcentaje de compresión JPEG utilizando el método 2

Figura 4.53: Gráficos de la variación del coeficiente de correlación respecto al porcentaje de compresión JPEG para los métodos 1 y 2.

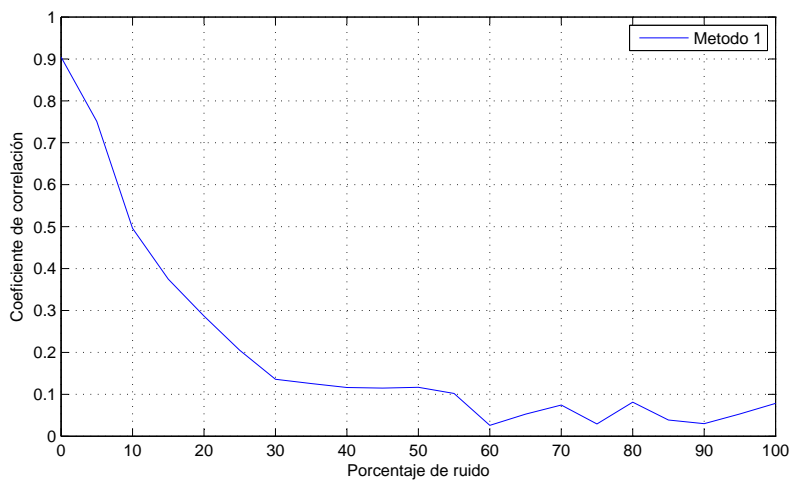


(a) Variación del coeficiente de correlación respecto al porcentaje de compresión JPEG utilizando el método 3

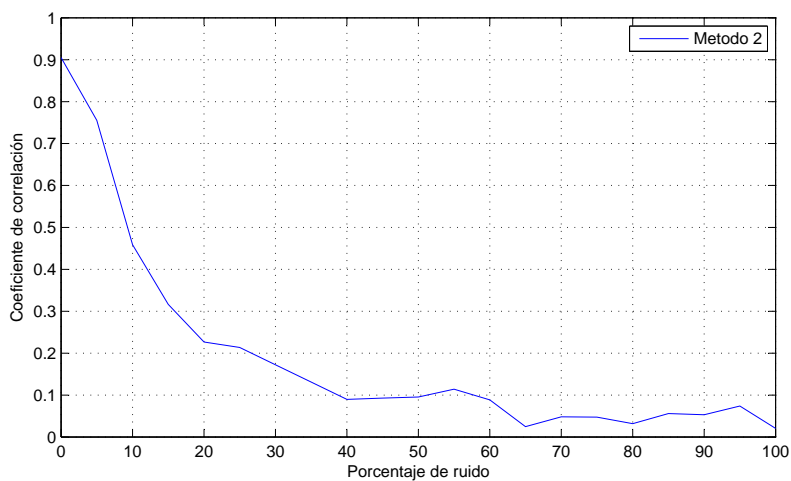


(b) Variación del coeficiente de correlación respecto al porcentaje de compresión JPEG utilizando el método 4

Figura 4.54: Gráficos de la variación del coeficiente de correlación respecto al porcentaje de compresión JPEG para los métodos 3 y 4.

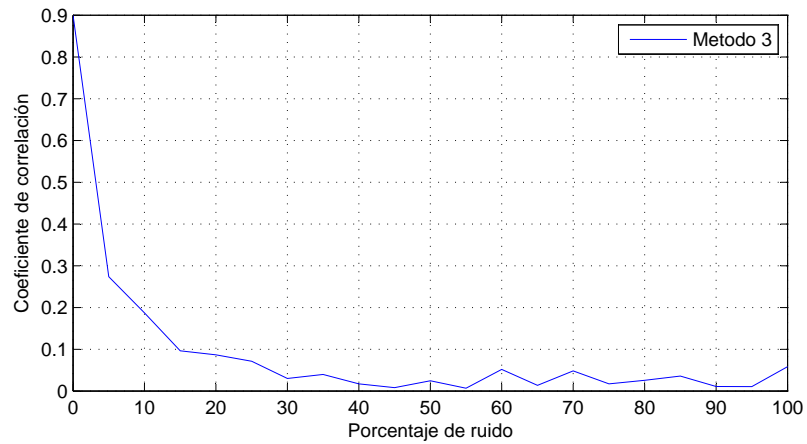


(a) Variación del coeficiente de correlación respecto al porcentaje de ruido agregado a la imagen marcada utilizando el método 1

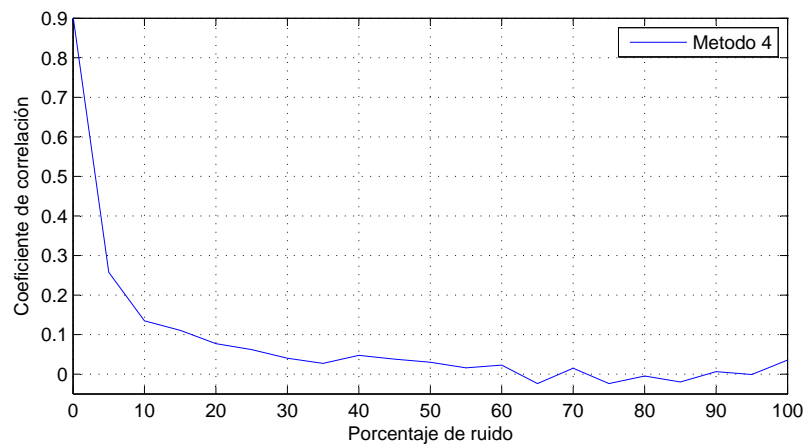


(b) Variación del coeficiente de correlación respecto al porcentaje de ruido agregado a la imagen marcada utilizando el método 2

Figura 4.55: Gráficos de la variación del coeficiente de correlación respecto al porcentaje de ruido agregado a las imágenes marcadas para los métodos 1 y 2.



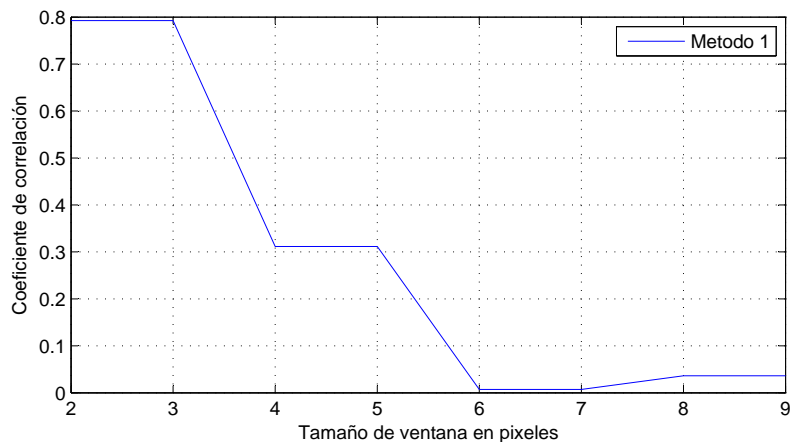
(a) Variación del coeficiente de correlación respecto al porcentaje de ruido agregado a la imagen marcada utilizando el método 3



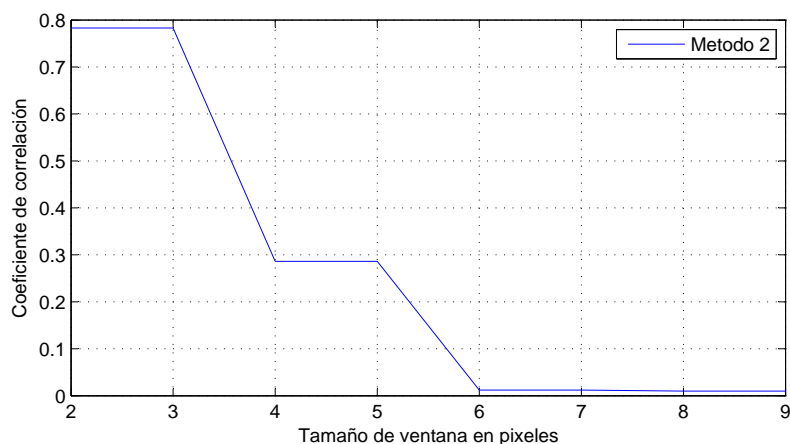
(b) Variación del coeficiente de correlación respecto al porcentaje de ruido agregado a la imagen marcada utilizando el método 4

Figura 4.56: Gráficos de la variación del coeficiente de correlación respecto al porcentaje de ruido agregado a las imágenes marcadas para los métodos 3 y 4.

al tamaño del filtro de mediana aplicado a las imágenes marcadas para los cuatro métodos. Al igual que con la simulaciones de la imagen "lena", el algoritmo solo puede recuperar el logo después de un ataque de filtro de mediana de tamaño no mayor a 3, los métodos que presentan mayor correlación del logo recuperado son los métodos 1 y 2.



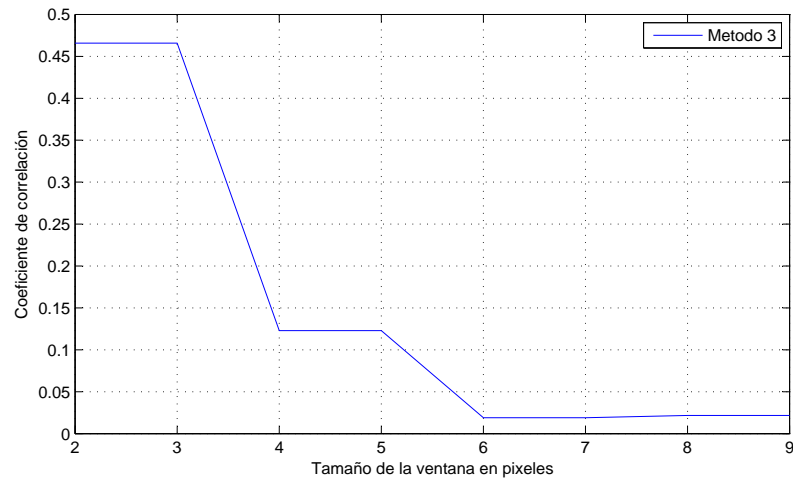
(a) Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro de mediana aplicado a la imagen marcada utilizando el método 1



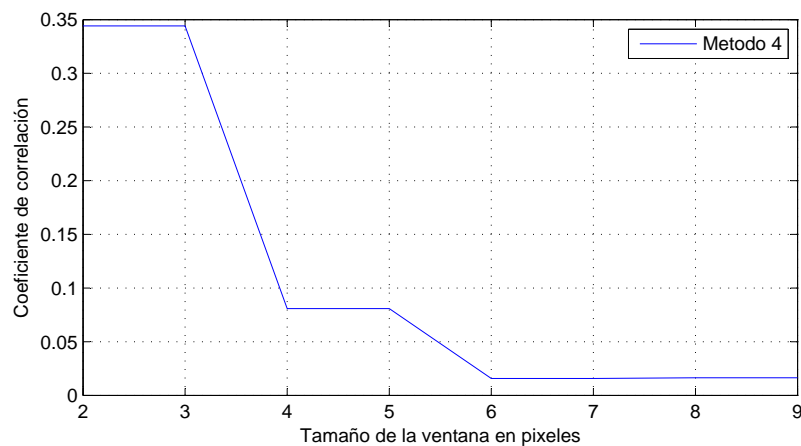
(b) Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro de mediana aplicado a la imagen marcada utilizando el método 2

Figura 4.57: Gráficos de la variación del coeficiente de correlación respecto al tamaño del filtro aplicado a las imágenes marcadas para los métodos 1 y 2.

Las figuras 4.59 y 4.60 muestran la variación del coeficiente de correlación de las imágenes marcadas para los cuatro métodos.

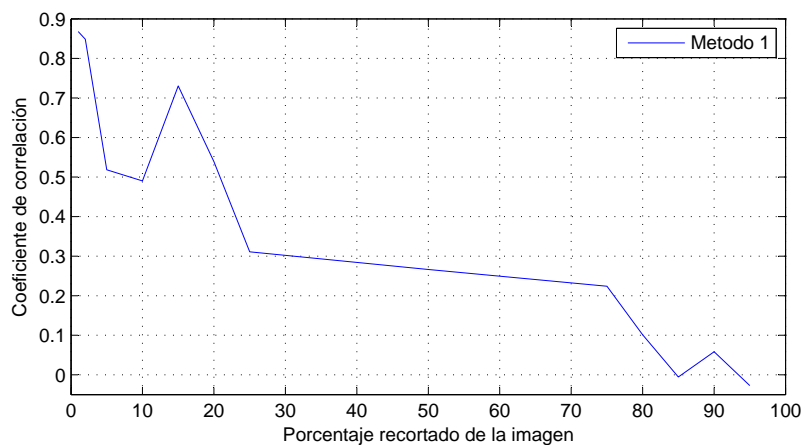


(a) Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro de mediana aplicado a la imagen marcada utilizando el método 3

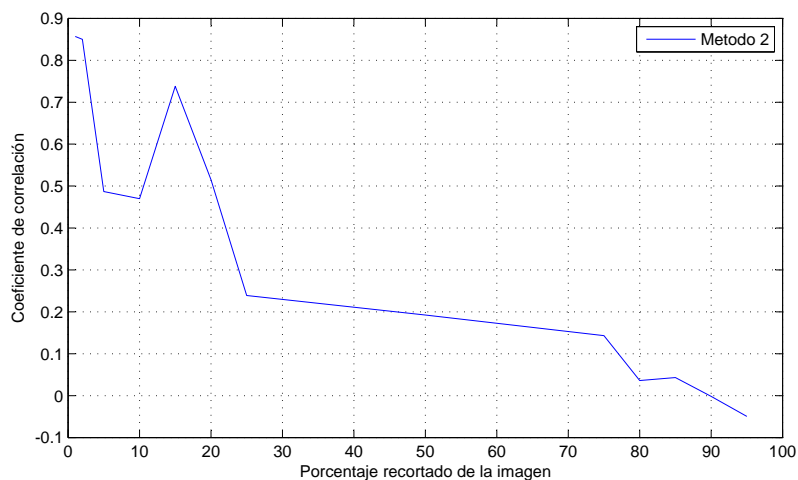


(b) Variación del coeficiente de correlación respecto al tamaño de la ventana del filtro de mediana aplicado a la imagen marcada utilizando el método 4

Figura 4.58: Gráficos de la variación del coeficiente de correlación respecto al tamaño del filtro aplicado a las imágenes marcadas para los métodos 3 y 4.

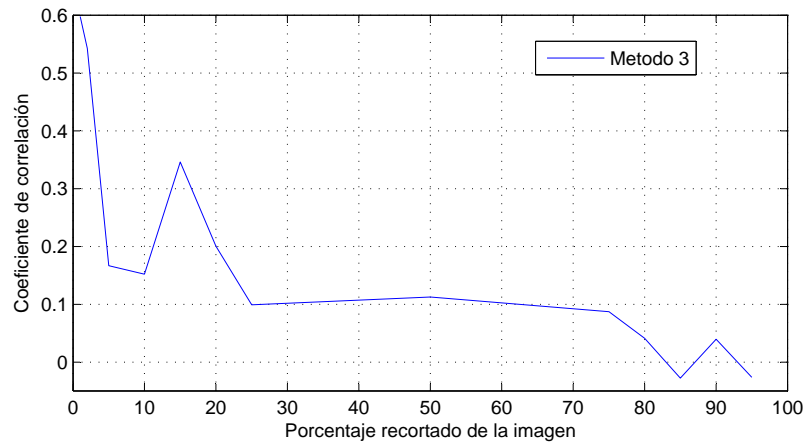


(a) Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen marcada utilizando el método 1

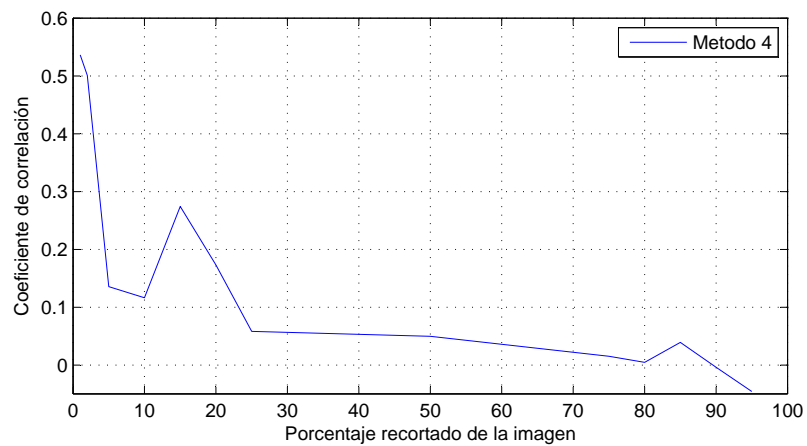


(b) Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen marcada utilizando el método 2

Figura 4.59: Gráficos de la variación del coeficiente de correlación respecto al porcentaje recortado a las imágenes marcadas para los métodos 1 y 2.



(a) Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen marcada utilizando el método 3



(b) Variación del coeficiente de correlación respecto al porcentaje recortado de la imagen marcada utilizando el método 4

Figura 4.60: Gráficos de la variación del coeficiente de correlación respecto al porcentaje recortado de las imágenes marcadas para los métodos 3 y 4.

La figura 4.61 muestra ejemplos de ataques geométricos aplicados a la imagen de “barbara” con marca de agua. Al igual que las pruebas hechas con “lena”, las gráficas presenta grandes saltos de un porcentaje de recorte a otro, y también los métodos que nos permiten recuperar los logos con un mayor coeficiente de correlación con el logo original son los métodos 1 y 2. Los métodos 3 y 4 solo nos permiten recuperar el logo con porcentaje de recorte de entre el 15% y 20%.

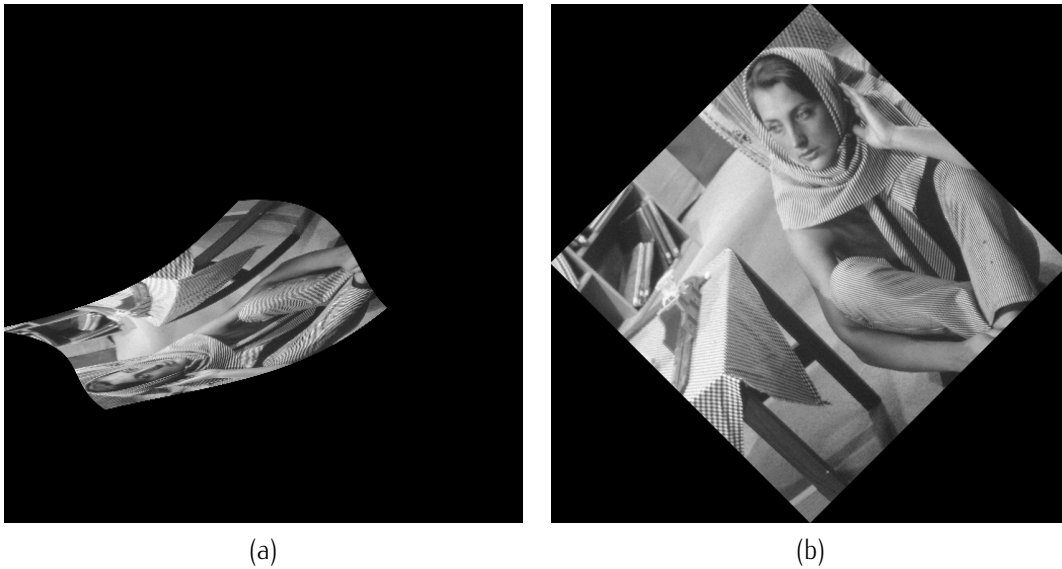


Figura 4.61: Ejemplos de ataques geométricos a la imagen de barbara.

Tomando todo lo anterior en consideración, podemos concluir que las imágenes marcadas poseen el mismo comportamiento para los diferentes tipos de ataques, por lo que los algoritmos se pueden generalizar para cualquier imagen digital y a partir del desempeño de la gráfica del factor de multiplicación vs. PSNR podemos establecer un umbral para obtener imágenes con marca de agua con un alto valor de la PSNR.

Capítulo 5

Conclusiones

La transformada contourlet, resultó ser una buena herramienta para descomponer una imagen e incrustar en ella una marca de agua, ya que provee un buen agrupamiento de la información. Además, que su implementación se lleva a cabo con un banco de filtros que dan lugar a algoritmos rápidos.

La implementación del algoritmo (1) [1], nos ayudó a identificar tanto los niveles de descomposición de la imagen, que nos proveen una mayor robustez e imperceptibilidad de la marca, como las características de la técnica de espectro disperso para modular la marca de agua o mensaje que se desee incrustar, la principal característica de esta técnica es que es robusta al ruido, ya que la cantidad de energía (o distorsión) para eliminar la marca de agua incrustada debe ser muy grande. Por otra parte, el algoritmo (1) nos dio un punto de partida para proponer un algoritmo algoritmo (2), con variantes no solo en la incrustación de la marca sino también en la detección.

Cuando realizamos las gráficas de la variación del factor de multiplicación vs PSNR para ambos algoritmos, nos dimos cuenta que la PSNR disminuye conforme el valor del factor de multiplicación aumenta, para el algoritmo (1) la disminución de la calidad visual disminuye de una forma exponencial mientras que para el algoritmo (2) la gráfica es más acotada y su disminución es más lenta. Para el caso de las gráficas de la variación del coeficiente de correlación vs factor de multiplicación, constatamos que al aumentar el factor de multiplicación la detección mejoraba, la respuesta del detector del algoritmo (2) es aproximadamente 5 veces más rápida que la del detector del algoritmo (1), lo que nos permite utilizar valores del factor de multiplicación más pequeños y alcanzar valores de la PSNR más altos. La PSNR de

la imagen marcada mejora aproximadamente en 7[dB] de la utilización del algoritmo (2) a la del algoritmo (1) para el caso de 'lena' y 6.5[dB] para el caso de 'barbara'.

Después de que las imágenes marcadas sufrieron varios ataques con el programa StirMark, realizamos las gráficas del coeficiente de correlación vs cada uno de los ataques. Para el caso de la compresión JPEG, observamos que el algoritmo (1) es más robusto ya que nos permite recuperar la marca de agua con un buen patrón visual hasta porcentajes de compresión del 80% mientras que para el caso del algoritmo (2) solo lo podemos recuperar para compresiones menores al 35%. En el caso de la prueba de ruido las imágenes marcadas permitieron una recuperación favorable para el caso de un porcentaje de ruido del 15% para el algoritmo (1), mientras que el algoritmo (2) solo permite la recuperación de la marca para porcentajes de ruido del 5%. El ataque del filtro de mediana es demasiado severo para poder detectar la marca para un tamaño de ventana de más de 3x3 para ambos métodos, sin embargo, con el algoritmo (1) la marca se recupera con mayor correlación a la marca original. Por último, el ataque de recorte el algoritmo (1) tuvo mejor desempeño que el algoritmo (2) ya que la marca se recuperó exitosamente para porcentajes de recorte del 50% mientras que para el algoritmo (2) solo se recuperó para porcentajes menores al 10%. En cuanto al tipo de ataques podemos concluir que el algoritmo (1) es más robusto que el algoritmo (2) en todos los ataques a los que fueron expuestos las imágenes marcadas. La robustez del algoritmo (2) disminuye debido a que la energía de las secuencias pseudoaleatorias es modulada para compensar la interferencia de la señal. Obviamente el hecho de modular la energía de las secuencias pseudoaleatorias nos permite alcanzar valores de la PSNR más altos, pero esto a cambio en una disminución de robustez. ¿Qué algoritmo es mejor que otro?, la aplicación sería la que nos daría la respuesta.

5.1. Trabajo a futuro

El algoritmo propuesto en la tesis está lejos de ser óptimo ya que este depende de muchos factores. Primero, la elección de método para generar las secuencias pseudoaleatorias tiene un impacto directo en la fiabilidad y seguridad de la marca insertada. En nuestro algoritmo utilizamos las secuencias que genera Matlab, por lo que podría estudiarse a futuro otro tipo de secuencias pseudoaleatorias conocidas en la literatura del marcado de agua como las secuencias de máxima longitud o simplemente secuencias-m,

códigos de oro, códigos Kasami, secuencias Legendre, entre otros, y averiguar que ventajas o desventajas podría traer el uso de tales secuencias. Segundo, en la elección del nivel de descomposición de la imagen marcada, tanto en la pirámide laplaciana como en el número de direcciones utilizadas, ya que en nuestro algoritmo sólo probamos incrustar la marca en el segundo nivel de descomposición, por lo que podría explorarse que efectos tiene en cuanto a robustez e imperceptibilidad la marca, cuando esta es incrustada en otros niveles de descomposición de la transformada o en una específica dirección. Tercero, en las pruebas hechas nos dimos cuenta que el esquema de inserción de la marca de agua empeoraba, o mejoraba dependiendo del tipo de filtros utilizados, tanto en la descomposición piramidal como en la direccional, por lo que también se podría enfocar a un trabajo futuro el diseño de filtros digitales que provean una respuesta más plana para la estructura (quincunx) que utiliza la transformada contourlet. Finalmente, la detección de la marca se lleva a cabo, bajo la consideración de que la distribución de los coeficientes donde se insertara la marca es Gaussiana, lo cual está lejos de serlo, por lo que se podría plantear a futuro una prueba de hipótesis, en la cual su estadística utilice la distribución generalizada Gaussiana para estimar la forma de distribución de los coeficientes y así detectar la marca.

Bibliografia

- [1] M. Jayalakshmi, S. Merchant, and U. B. Desai, "Blind Watermarking in Contourlet Domain with Improved Detection.," In *IHH-MSP*, pp. 449–452 (IEEE Computer Society, 2006).
- [2] S. Mallat, *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)* (Academic Press, 1999).
- [3] M. Do, in *Discrete Geometrical Image Processing Using the Contourlet Transform* (2003).
- [4] M. Do, Ph.D. thesis, Lausanne, 2001, prix EPFL de doctorats – EPFL doctorate award (2001).
- [5] M. Do and M. Vetterli, "Contourlets: a new directional multiresolution image representation," *Signals, Systems and Computers*, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on **1**, 497–501 vol.1 (2002).
- [6] I. Cox, M. L. Miller, and J. A. Bloom, *Digital watermarking* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002).
- [7] M. Arnold, M. Schmucker, and S. D. Wolthusen, *Techniques and Applications of Digital Watermarking and Content Protection* (Artech House, Inc., Norwood, MA, USA, 2003).
- [8] J.-S. Pan, H.-C. Huang, and L. C. Jain, *Intelligent Watermarking Techniques* (Series on Innovative Intelligence, 2004), Vol. 7.
- [9] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE (USA)* **87**, 1079–1107 (1999).

-
- [10] E. J. Candès and D. L. Donoho, "Ridgelets: A Key to Higher-Dimensional Intermittency?," .
- [11] E. J. Candès and D. L. Donoho, "Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges," (1999).
- [12] *The Transform and Data Compression Handbook*, K. R. Rao and P. Yip, eds., (CRC Press, Inc., Boca Raton, FL, USA, 2000).
- [13] E. J. Candès and D. L. Donoho, "Curvelets, Multiresolution Representation, and Scaling Laws," .
- [14] P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a compact image code," *IEEE Transactions on Communications* **COM-31,4**, 532–540 (1983).
- [15] M. Do and M. Vetterli, "Framing pyramids," 2001.
- [16] R. Bamberg and M. Smith, "A filter bank for the directional decomposition of images: theory and design," (1992).
- [17] S.-I. Park, M. J. T. Smith, and R. M. Mersereau, "A new directional filter bank for image analysis and classification," In *ICASSP '99: Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference*, pp. 1417–1420 (IEEE Computer Society, Washington, DC, USA, 1999).
- [18] S.-I. Park, Ph.D. thesis, Georgia, 2001.
- [19] J. J. K. Ó. Ruanaidh and G. Csurka, "A Bayesian Approach to Spread Spectrum Watermark Detection and Secure Copyright Protection for Digital Image Libraries," In ,
- [20] I. Cox, J. Kilian, T. Leighton, and T. Shanon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing* **6**, 1673–1687 (1997).
- [21] H. Malvar and D. Florêncio, "Improved spread spectrum: a new modulation technique for robust watermarking," *IEEE Transactions on Image Processing* **51**, 898–905 (2003).
- [22] M. Barni and F. Bartolini, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE Transactions on Image Processing* **10**, 783–791 (2001).

-
- [23] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on Copyright Marking Systems," In *Information Hiding*, pp. 218–238 (1998).
- [24] M. Barni, F. Bartolini, A. D. Rosa, and A. Piva, "A new decoder for the optimum recovery of nonadditive watermarks," *IEEE Transactions on Image Processing* **10**, 755–766 (2001).

Apéndice A

Código de los algoritmos implementados

Las implementaciones de los dos algoritmos se llevaron a cabo en Matlab, los códigos utilizados se muestran a continuación, para poder ejecutar los programas es necesario contar con el toolbox de la transformada contourlet¹.

A.1. Código de Algoritmo 1

Los programas están organizados como sigue, la función principal para el algoritmo de inserción de marca de agua 1 se llama 'sswatermark.m', la cual a su vez utiliza las funciones 'Gpsecuencias.m', esta función genera las secuencias pseudoaleatorias, 'EmbedPnwatermark.m', la cual hace la inserción de la marca, 'Recwatermark.m' es la función que recupera la marca de agua y la función 'PSNR.m' calcula la relación señal pico a ruido entre la imagen original y la imagen marcada.

¹El toolbox de la transformada contourlet puede ser descargado de <http://www.ifp.uiuc.edu/minhdo/software/>

```

%*****
%****      Proyecto : Marca de agua en el dominio contourlet      ****
%****      utilizando técnica de espectro disperso                ****
%****      implemetadado por MAZAY OSWALDO JIMENEZ SALINAS        ****
%*****
%*****
%*****      Archivo : sswatermark.m      *****
%*****      Date : 2 Julio de 2008      *****
%*****      Plataforma : Matlab        *****
%*****
%*****
%*****
%*****
%*****
%*****
%*****
%*****
%*****
%*****
%*****
%*****      SSWATERMARK      *****
%*****

%%Inicio de variables en cero
clear all
close all
clc

%%Seleccion del metodo utilizar
%%Metodo 1 insertar marca en el 75% de la banda Y
%%Metodo 2 insertar marca en banda Y

metodo = '1';
%%Seleccion de factor de multiplicacion ej:0.1, 0.25, etc
alfa = 0.25;

%%Leer imagen original ej: barbara, lena, etc
co = imread('barbara.pgm');
co = double(co); %Convierte la imagen en formato double

%Mostrar imagen original
imshow ( co, [] ) ; %muestra imagen en patalla
title( sprintf('Imagen Original') ); %Agrega titulo

%%Leer marca de agua
wml=imread('logo.bmp');

%%Mostrar marca de agua
figure
imshow(wml,[]); %Muestra imagen en pantalla

%%Convertir la marca de agua en -1 y 1

wm = double(wml.*-2 + 1);

%%Genera secuencias pseudoaleatorias del tamaño de la marca de agua wm

pn = Gpsecuencias(wm, metodo);

```



```
%%Transformada Contourlet de la imagen original

%%Parametros:
nlevels = [1, 2, 3, 4]; % Nivel de descomposicion
pfilter = '5/3' ;      % Filtro Piramidal
dfilter = 'dmaxflat7'; % Filtro Direccional

Co = pdfbdec( co, pfilter, dfilter, nlevels ); %descomposición contourlet

%%Incrustacion de la marca de agua

Ca = EmbedPnWatermark(Co, pn, alfa, metodo, wm);

%%Reconstruccion de la imagen con marca de agua

cw = pdfbrec( Ca, pfilter, dfilter ); %reconstrucción contourlet

%% Calculo de PSNR

psnr = PSNR(cw , co);

%% Mostrar imagen reconstruida y original

figure

subplot(1,2,1), imshow ( co, [] ) ;
title( sprintf('Imagen Original') );
subplot(1,2,2), imshow( cw, [] );
title(sprintf('Imagen con marca de agua con PSNR : %f dB',psnr) );

%%Recuperacion de la marca

Mrec = Recwatermark(cw, pn, pfilter, dfilter, nlevels, metodo);

%%Muestra marca de agua reconstruida

figure
imshow(Mrec);
```

```

% Funcion que inserta la marca de agua dependiendo del metodo utilizado

function ytr = EmbedPnWatermark(Co, pn, alfa, metodo, wm)

if metodo == '1'
    %%Esparcir cada bit de la marca con una secuencia pseudoaleatorias
    PN = zeros(128 , 384);
    for i = 1 : numel (wm)
        PN = PN + (wm(i) .* pn{1,1}{1,i});
    end

    %%Seleccion de subbandas

    Y1 = Co{1,4}{1,1};
    Y2 = Co{1,4}{1,2};
    Y3 = Co{1,4}{1,3};
    Y4 = Co{1,4}{1,4};
    Y5 = Co{1,4}{1,5};
    Y6 = Co{1,4}{1,6};

    Y = [ ];

    Y(1:64,1:128) = Y1;
    Y(1:64,129:256) = Y2;
    Y(65:128,1:128) = Y3;
    Y(65:128,129:256) = Y4;
    Y(1:128,257:320) = Y5;
    Y(1:128,321:384) = Y6;

    %% Insercion marca

    Y = Y + alfa*PN;

    Co{1,4}{1,1} = Y(1:64,1:128);
    Co{1,4}{1,2} = Y(1:64,129:256);
    Co{1,4}{1,3} = Y(65:128,1:128);
    Co{1,4}{1,4} = Y(65:128,129:256);
    Co{1,4}{1,5} = Y(1:128,257:320);
    Co{1,4}{1,6} = Y(1:128,321:384);

elseif metodo == '2'

    %%Esparcir cada bit de la marca con una secuencia pseudoaleatoria
    PN = zeros(256);
    for i = 1 : numel (wm)
        PN = PN + (wm(i) .* pn{1,1}{1,i});
    end

    %%Seleccion de subbandas

    Y1 = Co{1,4}{1,1};
    Y2 = Co{1,4}{1,2};
    Y3 = Co{1,4}{1,3};
    Y4 = Co{1,4}{1,4};
    Y5 = Co{1,4}{1,5};
    Y6 = Co{1,4}{1,6};

```

```
Y7 = Co{1,4}{1,7};
Y8 = Co{1,4}{1,8};

Y = [ ];

Y(1:64,1:128) = Y1;
Y(1:64,129:256) = Y2;
Y(65:128,1:128) = Y3;
Y(65:128,129:256) = Y4;
Y(129:256,1:64) = Y5;
Y(129:256,65:128) = Y6;
Y(129:256,129:192) = Y7;
Y(129:256,193:256) = Y8;

%% Insercion marca

Y = Y + alfa*PN;

Co{1,4}{1,1} = Y(1:64,1:128);
Co{1,4}{1,2} = Y(1:64,129:256);
Co{1,4}{1,3} = Y(65:128,1:128);
Co{1,4}{1,4} = Y(65:128,129:256);
Co{1,4}{1,5} = Y(129:256,1:64);
Co{1,4}{1,6} = Y(129:256,65:128);
Co{1,4}{1,7} = Y(129:256,129:192);
Co{1,4}{1,8} = Y(129:256,193:256);
end

ytr = Co;
```

```

%% Funcion que recupera la marca de agua dependiendo del metodo utilizado

function Mrec = Recwatermark(cw, pn, pfilter, dfilter, nlevels, metodo)

if metodo == '1'

    CW = pdfbdec( cw, pfilter, dfilter, nlevels );

    %%Seleccion de subbandas
    Y1 = CW{1,4}{1,1};
    Y2 = CW{1,4}{1,2};
    Y3 = CW{1,4}{1,3};
    Y4 = CW{1,4}{1,4};
    Y5 = CW{1,4}{1,5};
    Y6 = CW{1,4}{1,6};

    Y = [];

    Y(1:64,1:128) = Y1;
    Y(1:64,129:256) = Y2;
    Y(65:128,1:128) = Y3;
    Y(65:128,129:256) = Y4;
    Y(1:128,257:320) = Y5;
    Y(1:128,321:384) = Y6;

    [M N] = size(pn{1,1}{1,1});
    l_pn = length(pn{1,1});

    ro = [];

    for i = 1 : l_pn
        ro ( i ) = sum(sum(Y.*pn{1,1}{1,i}))/sqrt( sum(sum(Y.^2)) * sum(sum(pn{1,1}{1,i}.^2)) );
    end

    Ro = mean(ro);

    Mrec = [];

    for i = 1 : l_pn
        if ro ( i ) >= Ro
            Mrec ( i ) = 0;
        else
            Mrec ( i ) = 1;
        end
    end

elseif metodo == '2'
    cd contourlet_toolbox;

    CW = pdfbdec( cw, pfilter, dfilter, nlevels );

    cd ..

```

```

%%Seleccion de subbandas
Y1 = CW{1,4}{1,1};
Y2 = CW{1,4}{1,2};
Y3 = CW{1,4}{1,3};
Y4 = CW{1,4}{1,4};
Y5 = CW{1,4}{1,5};
Y6 = CW{1,4}{1,6};
Y7 = CW{1,4}{1,7};
Y8 = CW{1,4}{1,8};

Y = [ ];

Y(1:64,1:128) = Y1;
Y(1:64,129:256) = Y2;
Y(65:128,1:128) = Y3;
Y(65:128,129:256) = Y4;
Y(129:256,1:64) = Y5;
Y(129:256,65:128) = Y6;
Y(129:256,129:192) = Y7;
Y(129:256,193:256) = Y8;

[M N] = size(pn{1,1}{1,1});
l_pn = length(pn{1,1});

ro = [];

for i = 1 : l_pn
    ro ( i ) = sum(sum(Y.*pn{1,1}{1,i}))/sqrt( sum(sum(Y.^2)) * sum(sum(pn{1,1}{1,i}).
^2) ) );
end

Ro = mean(ro);

Mrec = [];

for i = 1 : l_pn
    if ro ( i ) >= Ro
        Mrec ( i ) = 0;
    else
        Mrec ( i ) = 1;
    end
end

end

Mrec = reshape (Mrec,22,96);

```

```
%funcion que genera secuencias pseudoaleatorias dependiendo de tamaño de
%marca a incrustar y el metodo

function pn = Gpsecuencias(wm , metodo)

rand('twister', 5489); %inicializacion del estado del generador de numeros
                        %aleatorios
pn = {}; %arreglo que alojara las secuencias

%%%Seleccion del metodo

if metodo == '1' %genera secuencias para metodo 1
    for i = 1 : numel (wm)
        pn{1,1}{1,i} = round(rand(128,384)).*2 - 1;
    end
elseif metodo == '2' %genera secuencia spara metodo 2
    hm = hadamard(256);
    for i = 1 : numel (wm)
        pn{1,1}{1,i} = round(rand(256)).*2 - 1;
    end
end

end

% Encuentra PSNR entre la imagen de entrada (in) y la estimada (est)
% en decibeles (dB).
%
% Referencia: Vetterli & Kovacevic, "Wavelets and Subband Coding", p. 386

function r = PSNR(in, est)

error = in - est;

r = 10 * log10(max(in(:)).^2/ mean(abs(error(:)).^2));
```

A.2. Código de Algoritmo 2

La función principal para la implementación del algoritmo de marca de agua 2 se llama 'isswatermark.m', esta a su vez utiliza las funciones 'issEmbedpwatermark.m', la cual hace la inserción de la marca de agua mediante la técnica de espectro disperso mejorado, la función 'issRecwatermark.m' la cual recupera la marca de la imagen, la función 'lambdopt.m' a cual calcula el parámetro λ_{opt} , la función 'productinter.m' la cual calcula el producto interno y las funciones 'Gpsecuencias.m' y 'PSNR.m' mencionas anteriormente.

```

%*****
%****      Proyecto : Marca de agua en el dominio contourlet      ****
%****      utilizando técnica de espectro disperso mejorado      ****
%****      implemetadado por MAZAY OSWALDO JIMENEZ SALINAS      ****
%*****
%*****
%*****      Archivo : sswatermark.m      *****
%*****      Date : 2 Julio de 2008      *****
%*****      Plataforma : Matlab      *****
%*****
%*****
%*****
%*****
%*****
%*****
%*****
%*****
%*****      SSWATERMARK      *****
%*****

%%Inicio de variables en cero
clear all
close all
clc

%%Seleccion del metodo utilizar
%%Metodo 1 insertar marca en el 75% de la banda Y
%%Metodo 2 insertar marca en banda Y

metodo = '1';

%%Seleccion de factor de multiplicacion ej:0.1, 0.25, etc

alfa = 0.11;

%%Leer imagen original
co = imread('barbara.pgm');
co = double(co);

%Mostrar imagen original
imshow ( co, [] );
title( sprintf('Imagen Original') );

%%Leer marca de agua

wm1=imread('logo.bmp');

%%Mostrar marca de agua
figure
imshow(wm1,[]);

%%Convertir la marca de agua en -1 y 1

wm = double(wm1.*-2 + 1);

%%Genera conjunto de secuencias pseudoaleatorias del tamaño de la marca de agua wm

```



```

% Funcion que inserta la marca de agua dependiendo del metodo utilizado

function ytr = EmbedPnWatermark(Co, pn, alfa, metodo, wm)

if metodo == '1'

    %%Seleccion de subbandas

    Y1 = Co{1,4}{1,1};
    Y2 = Co{1,4}{1,2};
    Y3 = Co{1,4}{1,3};
    Y4 = Co{1,4}{1,4};
    Y5 = Co{1,4}{1,5};
    Y6 = Co{1,4}{1,6};

    Y = [ ];

    Y(1:64,1:128) = Y1;
    Y(1:64,129:256) = Y2;
    Y(65:128,1:128) = Y3;
    Y(65:128,129:256) = Y4;
    Y(1:128,257:320) = Y5;
    Y(1:128,321:384) = Y6;

    %% Insercion marca

    for i = 1 : numel (wm)
        lambda = lambdaopt(Y);
        Pxu = productinter(Y,pn{1,1}{1,i});
        variancia_pn = var(pn{1,1}{1,i}(:,1));
        Y = Y + ( alfa*wm(i) - lambda*(Pxu/variancia_pn) ) * pn{1,1}{1,i};
    end

    Co{1,4}{1,1} = Y(1:64,1:128);
    Co{1,4}{1,2} = Y(1:64,129:256);
    Co{1,4}{1,3} = Y(65:128,1:128);
    Co{1,4}{1,4} = Y(65:128,129:256);
    Co{1,4}{1,5} = Y(1:128,257:320);
    Co{1,4}{1,6} = Y(1:128,321:384);

elseif metodo == '2'

    %%Seleccion de subbandas

    Y1 = Co{1,4}{1,1};
    Y2 = Co{1,4}{1,2};
    Y3 = Co{1,4}{1,3};
    Y4 = Co{1,4}{1,4};
    Y5 = Co{1,4}{1,5};
    Y6 = Co{1,4}{1,6};
    Y7 = Co{1,4}{1,7};
    Y8 = Co{1,4}{1,8};

    Y = [ ];

```

```
Y(1:64,1:128) = Y1;
Y(1:64,129:256) = Y2;
Y(65:128,1:128) = Y3;
Y(65:128,129:256) = Y4;
Y(129:256,1:64) = Y5;
Y(129:256,65:128) = Y6;
Y(129:256,129:192) = Y7;
Y(129:256,193:256) = Y8;

%%% Insercion marca

for i = 1 : numel (wm)
    lambda = lambdaopt(Y);
    Pxu = productinter(Y,pn{1,1}{1,i});
    variancia_pn = var(pn{1,1}{1,i}(:,1));
    Y = Y + ( alfa*wm(i) - lambda*(Pxu/variancia_pn) ) * pn{1,1}{1,i};
end

Co{1,4}{1,1} = Y(1:64,1:128);
Co{1,4}{1,2} = Y(1:64,129:256);
Co{1,4}{1,3} = Y(65:128,1:128);
Co{1,4}{1,4} = Y(65:128,129:256);
Co{1,4}{1,5} = Y(129:256,1:64);
Co{1,4}{1,6} = Y(129:256,65:128);
Co{1,4}{1,7} = Y(129:256,129:192);
Co{1,4}{1,8} = Y(129:256,193:256);

end

ytr = Co;
```

```

%%% Funcion que recupera la marca de agua dependiendo del metodo utilizado
function Mrec = Recwatermark(cw, pn, pfilter, dfilter, nlevels, metodo)
%%%Deteccion de la marca

if metodo == '1'
    cd contourlet_toolbox;

    CW = pdfbdec( cw, pfilter, dfilter, nlevels );

    cd ..
    %%%Seleccion de subbandas
    Y1 = CW{1,4}{1,1};
    Y2 = CW{1,4}{1,2};
    Y3 = CW{1,4}{1,3};
    Y4 = CW{1,4}{1,4};
    Y5 = CW{1,4}{1,5};
    Y6 = CW{1,4}{1,6};

    Y = [];

    Y(1:64,1:128) = Y1;
    Y(1:64,129:256) = Y2;
    Y(65:128,1:128) = Y3;
    Y(65:128,129:256) = Y4;
    Y(1:128,257:320) = Y5;
    Y(1:128,321:384) = Y6;

    [M N] = size(pn{1,1}{1,1});
    l_pn = length(pn{1,1});

    ro = [];

    for i = 1 : l_pn
        ro (i) = sum(sum(Y.*pn{1,1}{1,i}))/ (M*N);
    end

    sigmab2 = sum(sum(Y.^2));
    sigmab2= sigmab2/((l_pn*M*N)^2);
    Ro= 3.97*sqrt(2*sigmab2);

    Mrec = [];

    for i = 1 : l_pn
        if ro (i) >= Ro
            Mrec ( i ) = 0;
        else
            Mrec ( i ) = 1;
        end
    end

elseif metodo == '2'
    cd contourlet_toolbox;

```

```

CW = pdfbdec( cw, pfilter, dfilter, nlevels );

cd ..

%%Seleccion de subbandas
Y1 = CW{1,4}{1,1};
Y2 = CW{1,4}{1,2};
Y3 = CW{1,4}{1,3};
Y4 = CW{1,4}{1,4};
Y5 = CW{1,4}{1,5};
Y6 = CW{1,4}{1,6};
Y7 = CW{1,4}{1,7};
Y8 = CW{1,4}{1,8};

Y = [ ];

Y(1:64,1:128) = Y1;
Y(1:64,129:256) = Y2;
Y(65:128,1:128) = Y3;
Y(65:128,129:256) = Y4;
Y(129:256,1:64) = Y5;
Y(129:256,65:128) = Y6;
Y(129:256,129:192) = Y7;
Y(129:256,193:256) = Y8;

[M N] = size(pn{1,1}{1,1});
l_pn = length(pn{1,1});

ro = [];

for i = 1 : l_pn
    ro (i) = sum(sum(Y.*pn{1,1}{1,i}))/ (M*N);
end

sigmab2 = sum(sum(Y.^2));
sigmab2= sigmab2/((l_pn*M*N)^2);
Ro= 2.63*sqrt(2*sigmab2);

Mrec = [];

for i = 1 : l_pn
    if ro (i) >= Ro
        Mrec ( i ) = 0;
    else
        Mrec ( i ) = 1;
    end
end
end

Mrec = reshape (Mrec,22,96);

```

```
%%% Calculo de lambda optima

function lamb = lambdaopt ( Y )

Y = double(Y);
N = numel(Y);
variancia_y = var(Y(:,1));
variancia_n = 1; %%%asumido que es gaussiano
variancia_pn = 1; %%%la secuencias pseudoaleatorias se diseñaron con variancia unitaria
p_termino = 1 + (variancia_n/variancia_y) + N*(variancia_pn/variancia_y);
s_termino = p_termino^2 - 4*N*(variancia_pn/variancia_y);
lamb = 0.5*( p_termino - sqrt(s_termino) );

%%%Funcion que calcula el producto interno entre dos vectores
function Pxu = productinter ( Px , Pu)

Pxu = sum(sum(Px.*Pu))/numel(Px);
```

Índice alfabético

- Algoritmo 1, 50
- Algoritmo 2(Propuesto), 53
- Análisis Multiresolución, 39
- Aplicaciones, 7

- Código de Algoritmo 1, 115
- Código de Algoritmo 2, 123
- Código de los algoritmos implementados, 115
- Compresión, 6
- Conclusiones, 107
- Criptografía, 2
- Criptografía, Steganografía y Marca de Agua, 2

- Descomposición Direccional, 35
- Descomposición Multiescala, 33
- Descomposición Multiescala y Direccional, 38
- Descripción del algoritmo de inserción de la marca de agua, 50
- Detección de la marca de agua, 52, 55
- Direcciones Múltiples, 43
- Distorsiones y Ataques, 5

- Esbozo de la Tesis, 8

- Filtrado, 5

- Histograma, 60, 61

- Inserción de la marca de agua, 50, 54
- Introducción, 1, 11, 21, 59

- Llaves, 4

- Múltiples Marcas de Agua, 7
- Marca de Agua, 3
- Marca de agua en el dominio espacial, 15
- Marca de agua invisible, 12
- Marca de agua visible, 12
- Marcas de agua como firmas, 7
- Marcas de agua para autenticación, 7

- Marcas de agua para control de copias, 8
- Marcas de agua para huellas digitales, 8
- Marcas de agua para monitoreo de transmisiones, 8
- Modelo de la marca de agua, 16
- Multiescala, 41
- Multiescala y Direcciones Múltiples, 46

- Parámetros Importantes, 60
- Promedio Estadístico, 7
- Propiedades de la marca de agua, 19
- Prueba de compresión JPEG, 74
- Prueba de Filtro de Mediana, 84
- Prueba de Recorte, 87
- Prueba de Rotación y escalamiento, 92
- Prueba de Ruido, 82

- Recorte, 6
- Relación señal pico a ruido, 60
- Resultados de las simulaciones para la imagen "lena", 61
- Resultados de las simulaciones para la imagen barbara, 94
- Rotación y Escalamiento, 6
- Ruido Aditivo, 5

- Sistemas de marca de agua, 4
- Steganografía, 3
- STFT Short Time Fourier Transform, 23

- Técnicas básicas en la marca de agua, 12
- Trabajo a futuro, 108
- Transformada Contourlet, 32
- Transformada Curvelet, 31
- Transformada de Fourier, 22
- Transformadas, 15
- Transformada Continua Wavelet CWT, 24
- Transformada Discreta Wavelet DWT, 24
- Transformada Discreta Wavelet en dos Dimensiones DWT, 26

Variación de la PSNR, 66

Varianza, 60

Wavelets, 24