



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

CENTRO DE INVESTIGACIONES EN INGENIERÍA
Y CIENCIAS APLICADAS

**ALGORITMO DE RECOCIDO SIMULADO PARA LA COMPRESIÓN DE
IMÁGENES 2D APLICADO A LA PREDICCIÓN DE LOS SIGNOS DE LOS
COEFICIENTES DE LA TRANSFORMADA WAVELET.**

TESIS PROFESIONAL
PARA OBTENER EL GRADO DE:

MAESTRÍA EN INGENIERÍA Y CIENCIAS APLICADAS
OPCIÓN TERMINAL TECNOLOGÍA ELÉCTRICA

P R E S E N T A

L.I. PEDRO MORENO BERNAL

ASESOR: DR. MARCO ANTONIO CRUZ CHÁVEZ
COASESOR: DR. ABELARDO RODRÍGUEZ LEÓN

CUERNAVACA, MOR.

NOVIEMBRE 2011



UNIVERSIDAD AUTONOMA
DEL ESTADO DE MORELOS

Centro de Investigación en
Ingeniería y Ciencias Aplicadas

OFICIO No. 138/2011
Asunto: Aprobación de Tesis

C. PEDRO MORENO BERNAL
PRESENTE.

Por este conducto le notifico que su tesis de Maestría titulada:

**"ALGORITMO DE RECOCIDO SIMULADO PARA LA COMPRESIÓN DE IMÁGENES 2D
APLICADO A LA PREDICCIÓN DE LOS SIGNOS DE LOS COEFICIENTES DE LA TRANSFORMADA
WAVELET"**

Fue aprobada en su totalidad por el jurado revisor y examinador integrado por los ciudadanos:

NOMBRE	FIRMA
DRA. OCOTLAN DIAZ PARRA	
DR. MARTIN HERIBERTO CRUZ ROSALES	
M.C. JESUS DEL CARMEN PERALTA ABARCA	
DR. MARCO ANTONIO CRUZ CHAVEZ	
DR. ABELARDO RODRÍGUEZ LEÓN	

Por consiguiente, se autoriza a editar la presentación definitiva de su trabajo de investigación para culminar en la defensa oral de mismo.

Sin otro particular aprovecho la ocasión para enviarle un cordial saludo.

ATENTAMENTE
"POR UNA HUMANIDAD CULTA"

DRA. MARISOL GUIZADO RODRIGUEZ
SECRETARIA DE INVESTIGACIÓN Y POSGRADO
DE LA FCQeI-CIICAp

Resumen

Para este trabajo de investigación se aplicó un algoritmo de Recocido Simulado para maximizar el costo de la solución aplicado al problema de la predicción de signo de los coeficientes wavelet para la compresión de imágenes 2D, para el cual se realizó un estudio con diferentes estructuras de vecindad, con el objetivo de mejorar el desempeño del algoritmo. Parte del problema requiere generar una solución inicial de forma aleatoria ya que no existen benchmarks para este tipo de problema.

Se evaluó la eficiencia y eficacia de un conjunto de estructuras de vecindad para tres y cinco coeficientes vecinos. En base al análisis de desempeño realizado se demuestra que la estructura de vecindad que mejores resultados provee es la estructura sencilla, la cual se utilizó para evaluar el algoritmo.

Se aplicó una metodología de sintonización, la cual permitió realizar el análisis de sensibilidad de los parámetros de control del algoritmo de Recocido Simulado, lo que permitió trabajar al algoritmo con el mejor desempeño en eficiencia y eficacia.

Además se realizó una comparación entre el algoritmo de Recocido Simulado y un algoritmo Genético reportado en la literatura para resolver el mismo problema. El análisis de desempeño muestra que el algoritmo de Recocido Simulado es más eficiente que el algoritmo Genético para tres coeficientes vecinos e igualmente de eficaces en encontrar una buena solución.

La aportación de este trabajo de investigación para resolver el problema tratado, es aplicar la predicción de signo con 5 coeficientes vecinos, lo cual mejora significativamente el costo de la solución.

Las pruebas de sintonización y experimentales fueron realizadas en un equipo de cómputo del laboratorio de Optimización y Software del CIICAp, posteriormente, se comparó la eficiencia con el algoritmo genético y de acuerdo a los resultados obtenidos, el algoritmo de Recocido Simulado propuesto demostró ser eficaz y eficiente.

Abstract

For this research we applied a simulated annealing algorithm to maximize the cost of the solution applied to the problem of predicting the sign of the wavelet coefficients for 2D image compression, for which a study was conducted with different neighborhood structures with the aim of improving the performance of the algorithm. Part of the problem requires generating a random initial solution as there are no benchmarks for this type of problem. We evaluated the efficiency and effectiveness of a set of neighborhood structures for three and five neighboring coefficients. Based on performance analysis conducted shows that the neighborhood structure that provides better results is simple structure, which was used to evaluate the algorithm. We applied a tuning methodology, which allowed for the sensitivity analysis of the control parameters of simulated annealing algorithm, allowing the algorithm to work with the best performance in efficiency and effectiveness. Furthermore, a comparison between the simulated annealing algorithm and a genetic algorithm reported in the literature to solve the same problem. The performance analysis shows that the simulated annealing algorithm is more efficient than the genetic algorithm for three neighboring coefficients and equally effective in finding a good solution. The contribution of this research to solve the problem treaty is to apply the prediction coefficients sign with 5 neighboring coefficients, which significantly improves the cost of the solution. The experimental tuning and testing were performed in the computer laboratory of Software Optimization in CIICAp, then compared the efficiency with genetic algorithm according to the results obtained, the proposed simulated annealing algorithm proved to be effective and efficient.

Agradecimientos

A CONACYT por brindarme el apoyo económico para la realización de mis estudios de posgrado.

Al Dr. Marco Antonio Cruz Chávez, director de este trabajo de investigación, por la orientación y consejos para la realización y culminación de este trabajo.

Al Dr. Otoniel Mario López Granado de la universidad Miguel Hernández de Elche, España.

A los integrantes del comité tutorial y revisores de tesis: Dr. Marco Antonio Cruz Chavez, Dr. Martin Heriberto Cruz Rosales, Dr. Abelardo Rodríguez León, Dra. Ocotlán Díaz Parra y M.C. Jesús del Carmen Peralta Abarca, por sus comentarios, orientación y consejos para la realización de este trabajo de investigación.

Nomenclatura

SA Siglas en inglés para el algoritmo de Recocido Simulado (Simulated Annealing).

DWT Siglas en inglés para Transformada Wavelet Discreta (Discrete Wavelet Transform).

DCT Siglas en inglés para Transformada del Coseno Discreto (Discrete Cosine Transform).

N Sigla en inglés para identificar el vecino del coeficiente al Norte (North)

W Sigla en inglés para identificar el vecino del coeficiente al Oeste (West)

E Sigla en inglés para identificar el vecino del coeficiente al Este (East)

S Sigla en inglés para identificar el vecino del coeficiente al Sur (South)

HL Sigla en inglés para denotar la subbanda que combina frecuencias altas-bajas (High-Low)

LH Sigla en inglés para denotar la subbanda que combina frecuencias bajas-altas (Low-High)

HH Siglas en inglés para denotar la subbanda que combina frecuencias altas-altas (High-High)

NSP Sigla en inglés de Neighbor Sign Patterns.

LTW Sigla en inglés de Lower Tree Wavelet

Lena y Bárbara Imágenes estándar de prueba.

Parámetros de control de Recocido Simulado

T_0	Valor inicial del parámetro de control (Temperatura inicial)
β	Coefficiente de parámetro de control (Decremento temperatura)
T_f	Valor final del parámetro de control (Enfriamiento)
L_k	Longitud de la Cadena de Markov
S	Solución Inicial
S'	Solución Inicial perturbada.

Contenido

Índice de Figuras	i
Índice de Tablas	iii
Índice de Algoritmos	iv
Capítulo 1 Introducción	1
1.1. Estado del arte.....	3
1.2. Objetivo de la investigación	9
1.3. Alcance de la investigación	10
1.4. Contribución de la tesis	10
1.5. Organización de la tesis	11
Capítulo 2 Predicción de signo de los coeficientes de la Transformada Wavelet	13
2.1. Análisis de señales mediante wavelet	14
2.2. Transformada Wavelet	15
2.3. Codificación del signo.....	18
2.4. Contexto de la vecindad	23
Capítulo 3 Algoritmo de Recocido Simulado	28
3.1. Descripción del Algoritmo de Recocido Simulado	29
3.2. Generación de una primera solución.....	35
3.3. Metodología de sintonización	39
3.4. Análisis de complejidad del Algoritmo de Recocido Simulado	42
Capítulo 4 Resultados Experimentales	44
4.1. Descripción del equipo utilizado	44
4.2. Análisis de Sensibilidad	45

4.3. Resultados Experimentales	47
4.4. Análisis de eficacia y eficiencia del Algoritmo	50
4.5. Análisis de los resultados	50
4.5.1. Primer análisis: imagen Lena	51
4.5.2. Segundo análisis: imagen Bárbara	60
Capítulo 5 Conclusiones y trabajos futuros.....	72
5.1. Conclusiones	72
5.2. Trabajos futuros.....	73
Referencias.....	74
 Apéndices	
A. Estructuras de datos utilizadas en el Algoritmo	80
B. Complejidad Temporal por pasos	81
C. Código fuente del Algoritmo de Recocido Simulado	83
Glosario de términos	85

Índice de Figuras

Figura 2-1. Nivel de descomposición en subbandas wavelet.....	16
Figura 2-2. Descomposición wavelet de a) uno b) dos c) tres niveles	18
Figura 2-3. Procesado de una imagen	19
Figura 2-4. Ordenación de los coeficientes en forma de árbol	19
Figura 2-5. DWT de segundo nivel de una imagen de 8 x 8 pixeles	20
Figura 2-6. Mapa de símbolos resultantes del LTW	21
Figura 2-7. Ejemplo de codificación utilizando LTW	22
Figura 2-8. Ejemplo de transformada wavelet	22
Figura 2-9. Mapa de signos de los vecinos	23
Figura 2-10. Distribución de coeficientes vecinos a analizar por tipo de subbanda para 3 vecinos	24
Figura 2-11. Distribución de coeficientes vecinos a analizar por tipo de subbanda para 5 vecinos.	25
Figura 2-12. Integración del signo en el codificador	27
Figura 3-1. Definición formal de entropía	31
Figura 3-2. Representación del espacio de soluciones de una estructura de vecindad	32
Figura 3-3. Estructura de vecindad sencilla.....	34
Figura 3-4. Estructura de vecindad pares aleatorios	34
Figura 3-5. Estructura de vecindad triples aleatorios	34
Figura 3-6. Estructura de vecindad cuádruples aleatorios	35
Figura 3-7. Vector predictor solución inicial para tres coeficientes vecinos	35
Figura 3-8. Combinación del patrón de signos de los vecinos	36
Figura 3-9. Ejemplo para evaluar el coeficiente actual en la subbanda HL_4	37
Figura 4-1. Gráfica comparativa de estructuras de vecindad para las imágenes de prueba	49
Figura 4-2. Imagen de Lena	51
Figura 4-3. Imagen comprimida con predicción de signo a 0.0625 bpp.....	55
Figura 4-4. Imagen comprimida con predicción de signo a 0.125 bpp.....	56
Figura 4-5. Imagen comprimida con predicción de signo a 0.25 bpp	57
Figura 4-6. Imagen comprimida con predicción de signo a 0.5 bpp	58
Figura 4-7. Imagen comprimida con predicción de signo a 1 bpp	59

Figura 4-8. Imagen comprimida con predicción de signo a 2 bpp	60
Figura 4-9. Imagen de Bárbara.....	61
Figura 4-10. Imagen comprimida con predicción signo a 0.0625 bpp	64
Figura 4-11. Imagen comprimida con predicción signo a 0.125 bpp	65
Figura 4-12. Imagen comprimida con predicción signo a 0.25 bpp	66
Figura 4-13. Imagen comprimida con predicción signo a 0.5 bpp.....	67
Figura 4-14. Imagen comprimida con predicción signo a 1 bpp.....	68
Figura 4-15. Imagen comprimida con predicción signo a 2 bpp.....	69
Figura 4-16. Comparativa Lena para tres y cinco coeficientes vecinos	70
Figura 4-17. Comparativa Bárbara para tres y cinco coeficientes vecinos.....	71

Índice de Tablas

Tabla 2-1. Probabilidad de distribución del mapa de signos para la subbanda HL ₆ de la imagen Lena	25
Tabla 2-2. Elementos del vector NSP para tres vecinos.....	26
Tabla 3-1. Función de distribución de Boltzmann	29
Tabla 3-2. Tabla de símbolos de los signos.....	36
Tabla 3-3. Criterio de aceptación modificado para la función de Boltzmann	38
Tabla 4-1. Equipos de cómputo utilizados	44
Tabla 4-2. Rangos utilizados en los parámetros de control para realizar el análisis de sensibilidad.....	46
Tabla 4-3. Valores de los parámetros de control, fijados de acuerdo al análisis de sensibilidad.....	47
Tabla 4-4. Resultados experimentales de las estructuras de vecindad para las imágenes de prueba.....	48
Tabla 4-5. Predicción de signo para la imagen de Lena.....	52
Tabla 4-6. Estudio de la entropía y ahorro en bps para la imagen Lena tres coeficientes vecinos	53
Tabla 4-7. Estudio de la entropía y ahorro en bps para la imagen Lena cinco coeficientes vecinos	54
Tabla 4-8. Diferencia de aciertos entre tres y cinco coeficientes vecinos para Lena	54
Tabla 4-9. Predicción de signo para la imagen de Bárbara	62
Tabla 4-10. Estudio de la entropía y ahorro en bps para la imagen Bárbara con tres coeficientes vecinos	62
Tabla 4-11. Estudio de la entropía y ahorro en bps para la imagen Bárbara con cinco coeficientes vecinos	63
Tabla 4-12. Diferencia de aciertos entre tres y cinco coeficientes vecinos para Bárbara.....	63

Índice de Algoritmos

Algoritmo 3-1. Algoritmo de Metrópolis.....	30
Algoritmo 3-2. Algoritmo general de búsqueda por vecindad.....	33
Algoritmo 3-3. Algoritmo Recocido Simulado para la predicción de los signos de los coeficientes wavelet	38

Capítulo 1

Introducción

En la actualidad existen numerosas herramientas y algoritmos matemáticos empleados para el tratamiento digital de imágenes. La gran mayoría de estas herramientas están basadas en el análisis de las texturas y en el empleo de métodos estadísticos, filtrados, transformaciones, etcétera, para realizar cálculos que permitan obtener índices cuantitativos que describan la distribución de niveles de gris o niveles digitales en una imagen [Fernández A., 2007], proporcionando así, información útil para tareas posteriores de clasificación, segmentación, recuperación de imágenes en bases de datos, compresión, etcétera.

Una de estas herramientas matemáticas es la conocida con el nombre de transformada wavelet discreta (TWD), la cual presenta un buen comportamiento a la hora de realizar un estudio de la relación espacio-frecuencia en el análisis de imágenes, donde funciones matemáticas segmentan los datos o señales en diferentes componentes de frecuencia y proceden a estudiar cada una de ellas con una resolución ajustada a su escala [Colom, 2001].

La transformada wavelet es una poderosa herramienta para compresión de imágenes. Algunos de los codificadores de imágenes, incluyendo el estándar JPEG2000 [ISO/IEC 15444-1], emplea transformada wavelet en sus algoritmos. La aplicación de la transformada wavelet sobre una imagen provee una matriz de coeficientes donde cada coeficiente está formado por un valor de cierta magnitud y un signo (positivo o negativo), por lo que el éxito del algoritmo de compresión depende de la compresión de dichos coeficientes.

La mayoría de los codificadores basados en transformadas wavelet, para imágenes como video, codifican el signo en crudo (raw coding), es decir, emplean un bit por signo, ya que se sabe que el signo de los coeficientes

wavelet es equiprobable [Shapiro, 1996], por lo que no es necesario emplear ningún método de compresión para el signo de los coeficientes.

Sin embargo, se han realizado distintos estudios al respecto aplicando técnicas basadas en la agrupación de muestras obtenidas en contextos, que al parecer pueden compactar la información del signo, con lo que mejoraría el rendimiento global del codificador. Un ejemplo de ello es el estándar de compresión JPEG2000, el cual codifica el signo de los coeficientes wavelet empleando información basada en contextos, observando el carácter del signo de los vecinos tanto horizontales como verticales del coeficiente a codificar.

Autores como Chrysafis y Ortega [Chrysafis y Ortega, 1997], LoPresto, Ramchandram y Orchard [LoPresto *et al.*, 1997], Xiong [Xiong *et al.*, 1997] y X. Wu [Wu, 1997] han realizado análisis basados en diferentes modelos de contexto para codificar los coeficientes wavelet y el signo, en los cuales el modelado para el signo y la textura son compartidos. Este tipo de modelos basan su análisis en los diferentes coeficientes vecinos en función del tipo de subbanda wavelet a analizar.

Dada la gran variedad de posibles combinaciones a analizar para predecir el signo de cada coeficiente wavelet, la combinación de los coeficientes vecinos y los signos, los distintos tipos de subbandas, etcétera, se hace necesario emplear herramientas de optimización.

Una metaheurística capaz de encontrar buenas correlaciones entre el coeficiente a codificar y sus vecinos horizontales y verticales, podrá hacer una buena predicción del signo, por lo tanto, encontraremos una solución óptima o subóptima a la predicción de signo dentro de un conjunto de coeficientes wavelet que ayudaría a codificar (enviar información al codificador aritmético) con un símbolo, indicando si la predicción es correcta o no. El obtener altos índices de acierto, envía al codificador mucho más símbolos de acierto que de fallo, haciendo que la cantidad de información necesaria en el codificador disminuya y por tanto aumente la tasa de compresión.

1.1. Estado del Arte

Muchos autores han estudiado diferentes comportamientos basados en diferentes modelos de contexto para encontrar un patrón que pueda ser aplicado a la eficiente codificación de los signos de los coeficientes wavelet, debido a ello, existen diferentes propuestas aplicadas en varias técnicas de codificación de imágenes, en la literatura se han propuesto diferentes métodos aplicados a dicho problema. Posiblemente los primeros autores que consideraron codificar el signo fueron Schwartz, Zandi y Boilek [Zandi et. al 1995], quienes utilizaron un signo de un pixel vecino para compararlo en sus algoritmos de modelos de contexto. Posteriormente [Wu 1997] expande el esfuerzo al considerar seis vecinos en el modelo de contexto de los signos de los coeficientes.

X. Wu [Wu, 1997] propone una nueva técnica llamada ECECOW “Embedded Conditional Entropy Coding of Wavelet Coefficients”, este sistema ofrece escalabilidad en la codificación de la calidad y el tamaño de la imagen de una forma más natural y convencional que otro tipo de sistemas de compresión de imágenes. Este método codifica los coeficientes wavelet uniformemente cuantificados en un plano de bits, explorando de los más a los menos significantes en orden de ocho de las subbandas de más bajas frecuencias a las más altas frecuencias. El plano de bits a codificar se ocupa sólo de dos símbolos fuentes: 0 o 1. Cuando el bit actual es 1 y este bit es el más significativo, ECECOW codifica el signo del coeficiente. Este tipo de conversión de coeficientes wavelet a flujos de símbolos binarios es similar al de CREW [Zandi *et al.*, 1995]. La codificación de imágenes consiste en los símbolos adyacentes en los planos de tiempo y frecuencia, estimando la probabilidad condicional que sirve como modelo estadístico de la fuente. Al conjunto de observaciones en que la probabilidad del símbolo actual es condicionado se le llama modelo de contexto. Es por ello que en la compresión de imagen/video basado en transformadas wavelet, el conocimiento de las propiedades intrínsecas de los coeficientes wavelet y el ingenio algorítmico son requeridos para formar modelos de contextos. Guiados por las observaciones, ECECOW

modela un coeficiente para sus vecinos en la misma subbanda y por el espacio correspondiente de los coeficientes en la subbandas padres, según el nivel de descomposición de la transformada wavelet. Para una máxima compresión, ECECOW trata todos los bits conocidos hasta el momento de la codificación de los coeficientes vecinos en la subbanda actual y subbandas padres como eventos modelados potenciales, a diferencia de los algoritmos EZW [Shapiro, 1993] y SPIHT [Said y Pearlman, 1996], el modelo de contexto de ECECOW contiene información futura buscando generalmente series de tiempo para reducir la incertidumbre. Para utilizar dicho contexto, propone un modelo estadístico de muy alto orden.

La orientación de las diferentes subbandas presta un adaptivo contexto para seleccionar una regla que reduzca el número de parámetros de modelos. Para la subbanda LH predominan las estructuras verticales y para la subbanda HL predominan las estructuras horizontales. Por lo tanto, dichas estructuras en las imágenes de prueba, Lena y Bárbara exhiben patrones de los signos de los coeficientes wavelet claramente visibles que sugieren que el signo de bits de los coeficientes wavelet son compresibles, a pesar que los coeficientes en las subbandas de alta frecuencia parecen tener una media de ceros en el proceso. Durante el procedimiento de codificación del plano de bit incrustados, ECECOW considera para el signo del coeficiente wavelet tres estados: + (positivo), - (negativo) y 0 (nulo). Respecto al plano de bits, el signo del coeficiente es aún desconocido por el codificador si el más significativo bit del coeficiente está por debajo en el plano de bits, es decir, si está en una subbanda hija. En este caso, ECECOW asigna un estado 0 al coeficiente, de otra manera asigna + o - al coeficiente por el significado convencional del signo. El estado 0 es un concepto dinámico, ya que puede cambiar a positivo o negativo según el proceso de codificación avanza más profundamente en el plano de bits. Se distingue 0 de + y - porque con tal distinción se obtiene un modelo de contexto más revelador de los signos de los coeficientes wavelet.

Esta técnica es efectiva para un modelo de contexto de alto orden de disposición de octavas de dos dimensiones de los coeficientes wavelet para compresión de imágenes. Esta técnica conduce a una incrustación para codificar la entropía condicional de los coeficientes wavelet, además que

mejora el rendimiento de previos codificadores de imágenes wavelet en compresión de imagen con y sin pérdida de información.

Deever y Hemami [Deever y Henami, 2000] en “What’s your sign? Efficient Sign Coding for Embedded Wavelet Image Coding” proveen un análisis detallado del comportamiento de los signos de los coeficientes wavelet derivados de una transformación multi-escala, utilizando filtros biortogonales Daubechies’ 9/7. Además consideran la predicción del coeficiente del vector base en la escala más fina. El soporte del vector base es un bloque de 7x9 píxeles. El sub-espacio está formado por todos los vecinos del vector base en todas las orientaciones (HL,LH, HH) en la misma escala. La proyección del sub-espacio es una combinación lineal de los vectores base de los vecinos. Los coeficientes de esta combinación lineal son utilizados con el valor codificado de los coeficientes wavelet vecinos para formar la predicción. La predicción de los signos de los coeficientes wavelet sin importantes vecinos en la imagen de prueba Bárbara, codifican 1 bit por píxel, lo que permite incrementar la eficiencia del codificador aritmético. Esto genera una reducción de más del 10% en la tasa de bits necesarios para codificar los signos correspondientes.

Taubman [Taubman, 2000] muestra un novedoso algoritmo de compresión de imágenes llamado EBCOT “Embedded Block Coding with Optimized Truncation”, en donde la codificación del signo se utiliza en más de una vez por cada muestra, inmediatamente después de una muestra insignificante, encuentra muestras significantes durante la codificación de ceros y en la operación de codificación de la longitud. Resulta que el signo de los bits de las muestras adyacentes, exhiben dependencias estadísticas substanciales que son efectivamente explotadas para mejorar la eficiencia de la codificación. Para explotar esta redundancia en la información del signo, se utilizan cinco modelos de contexto para codificar de acuerdo a la información disponible sobre los signos de los vecinos inmediatos horizontales y verticales.

El rendimiento de compresión que ofrece este algoritmo sin un conjunto de precedentes de características de flujos de bits, que incluyen escalabilidad en la resolución, escalabilidad en SNR y capacidad de acceso aleatorio, pueden co-existir o existir dentro de un único flujo de bits sin sacrificar la eficiencia de compresión.

Nuevamente Deever y Hemami [Deever y Hemami, 2003] en “Efficient Sign coding and Estimation of Zero-Quantized Coefficients in Embedded Wavelet Image Codecs”, muestran un modelo de contexto intra-banda para codificar el signo, específicamente, analizan las correlaciones del signo a lo largo y a través de los bordes. Esta información es particularmente útil para comprimir los signos de las subbandas HL y LH, que por lo general procesan la información de los bordes direccionalmente fuertes. Esto se debe a que las correlaciones verticales a menudo permanecen muy altas a lo largo de los bordes verticales en las imágenes. Los coeficientes de los vecinos a lo largo del borde son considerados valiosa información de contexto y se espera que tengan el mismo signo que el del coeficiente que se desea codificar. Estas observaciones son independientes del tipo de filtro wavelet empleado. Para el tipo de filtro Daubechies' 9/7, los signos de los coeficientes wavelet son negativos entre los bordes, esta fuerte correlación negativa se da porque para colores claros los signos de los coeficientes wavelet son positivos y para colores oscuros los signos de los coeficientes wavelet son negativos. Los cuatro vecinos más cercanos son considerados para el signo de la intra-banda en diferentes modelos de contexto. Los contextos 3^4 (cada vecino puede ser positivo, negativo o nulo) son cuantificados a un total de siete contextos, los cuales fueron evaluados y se tomaron los tres mejores según los resultados del algoritmo en las tres tasas diferentes de bits. La compresión del signo y la estimación de coeficientes cuantificados-ceros mejoran la total eficiencia de codificación de las imágenes probadas, teniendo una ganancia de 0.5 dB en algunas de ellas. Aproximadamente el 20% de la ganancia se atribuye a la proyección de la predicción del signo y la estimación de la cuantificación de ceros.

Para el año 2006 Oliver y Malumbres [Oliver y Malumbres, 2006] introducen un nuevo algoritmo de compresión de imagen basado en la eficiente construcción de árboles bajos de los coeficientes wavelets. La principal contribución del codificador Lower Tree Wavelet (LTW) es la utilización de árboles de coeficientes, no sólo como un eficiente método de agrupamiento, sino también

como un camino rápido para codificar. La complejidad es menor que otros codificadores de coeficientes wavelet como SPIHT y JPEG2000.

Su rendimiento de compresión mejora en 0.2 – 0.4 dB ante el algoritmo SPIHT y en 0.35 dB con la imagen Lena ante JPEG 2000.

Mbainabeye [Mbainabeye y Nouredine, 2007] en “Optimal Image Compression Based on Sign and Magnitude Coding of Wavelet Coefficients” propone un nuevo enfoque que separa la codificación del signo y la información de la magnitud de los coeficientes wavelet de la imagen. Además la imagen es descompuesta en el dominio wavelet y consideran a los coeficientes como los datos que proporcionan dos tipos de información: el signo y la magnitud. En este enfoque, un único símbolo llamado Significante se utiliza para codificar la magnitud que es mayor o igual al umbral.

La probabilidad en la cuantización de los coeficientes wavelet de ser positivos o negativos, es calculada plano por plano de bits. Es conocido que la probabilidad de encontrar un coeficiente negativo en la subbanda LL es igual a cero, por lo tanto, es posible encontrar en las subbandas HL, LH y HH coeficientes negativos. Por lo que proponen un algoritmo para codificar el mapa de signos que indica la presencia de coeficientes positivos y negativos en las subbandas HL, LH y HH. Un mapa de signos es generado por el símbolo ‘0’ para coeficientes positivos y por el símbolo ‘1’ para coeficientes negativos, estos símbolos son los que codifican la entropía. Se obtienen resultados comparados con el estándar JPEG2000 en términos de calidad (PSNR) para las imágenes de prueba Lena y Bárbara, dicha comparación se realiza en términos de calidad visual en imágenes descifradas en 0.15 bpp.

Kim Sungjei [Sungjei *et al.*, 2009] propone un modelo de contexto para el signo en modelos basados en contextos adaptivos, en función de la dirección para la codificación del signo, en codificadores de imagen empotrados basados en wavelet, se realiza en cascada de 2D a transformadas de 1D. La primera dirección filtrada tiene nueve direcciones, incluyendo direcciones verticales o 90°. En la segunda dirección del filtro, también son nueve direcciones con dirección horizontal o 0°. Esta dirección adaptiva de la transformada wavelet puede soportar la transformada wavelet convencional, aunque la imagen tenga

características dominantes horizontales o verticales. Los vecinos se deciden a lo largo de las características intrínsecas de la imagen, así los vecinos elegidos para el contexto del signo del coeficiente actual son estructurados con una alta correlación. Por lo tanto, el método construye el mejor contexto para los signos del coeficiente actual, utilizando los vecinos a lo largo de la dirección del filtro. DA-DWT (Direction-Adaptive Discrete Wavelet Transform) con 4 niveles de descomposición es aplicado a las imágenes de prueba conocidas como Bárbara y Lena de tamaño 512 x 512 píxeles. Utilizando tres vecinos adyacentes por la izquierda del coeficiente a codificar para la subbanda HL y para la subbanda LH los tres vecinos adyacentes superiores del coeficiente a codificar, en total $3^3 = 27$ contextos de signos son utilizados para codificar el signo de los coeficientes wavelet en las subbandas HL, HI y HH. Este modelo de contexto reduce los bits hasta 5% para las imágenes de prueba, es decir, con este modelo de contexto adaptivo, la codificación del signo alcanza un ahorro de bits de 5.5% en las pruebas experimentales.

En “E-LTW: An enhanced LTW Encoder with sign Coding and Precise Rate Control” [López *et al.*, 2009], un simple contexto basado en un enfoque de codificación del signo encuentran correlaciones a lo largo y a través de los bordes, con el fin de identificar a los vecinos con las más altas correlaciones en el codificador LTW, además definen un conjunto de vecinos con una distancia máxima de dos posiciones, estos son los coeficientes West, North, West-West, North-North, North-West y NorthNorth-West. El codificador realiza un recorrido Morton [Morton, 1966] en la etapa de codificación, lo cual no hace posible tener información de los vecinos East y South. Todos los vecinos tienen tres posibles valores para el signo, positivo(+), negativo (-) o nulo(*), pero el coeficiente a ser codificado solo tiene dos posibles valores para el signo, positivo (+) o negativo (-). Esto forma un total de $2*6^3$ posibles combinaciones de los signos vecinos. El nivel de descomposición aplicado a las imágenes de prueba Lena y Bárbara es de seis. Para la subbanda HL los signos de los vecinos que mejor están correlacionados con el signo del coeficiente actual son los que se encuentra al N, NN y W. Tomando en cuenta la simetría para la subbanda LH, los vecinos que mejor correlación tienen con el coeficiente actual son W, WW y N y para la subbanda HH son N, W y NW en busca de los bordes diagonales, por lo que

cada tipo de subbanda tiene un máximo de 3^3 combinaciones de los signos vecinos. La máxima ganancia de compresión es de 17.35% para la imagen Bárbara de 1 bpp y el mínimo es 2.94% para la imagen Lena en 0.0625 bpp.

García y López [García *et al.*, 2011] introducen una mejora al codificador LTW, el cual llaman S-LTW, donde utilizan un algoritmo genético para mejorar el rendimiento de la codificación de signos de los coeficientes wavelet. En esta nueva aproximación predicen eficientemente el signo de los coeficientes wavelet utilizando un vector predictor, el cual incluyen en un codificador de imagen no empotrado (S-LTW). Los resultados muestran que la capacidad de codificar el signo en la compresión tiene una ganancia de 17.35%, siendo la tasa de distorsión (R/D) mejorada arriba de 0.25 dB.

1.2. Objetivo de la investigación

1. Optimizar la tasa de bits a través de la predicción de los signos de los coeficientes wavelet para comprimir una imagen 2D, aplicando un algoritmo de Recocido Simulado.
2. Realizar las pruebas necesarias del algoritmo de Recocido Simulado y comparar la calidad de la solución obtenida con los resultados obtenidos por el algoritmo Genético [García *et al.*, 2011], los cuales han sido reportados en la literatura. Las pruebas realizadas se llevaron a cabo utilizando el mismo número de instancias de prueba para poder realizar una comparación directa y confiable.
3. Evaluar una estructura de vecindad que mejore el desempeño del algoritmo de Recocido Simulado.
4. Aumentar el número de vecinos de tres a cinco con el objetivo de ampliar la vecindad que permita una mejor explotación del espacio de soluciones para encontrar una mejor calidad en la solución ya obtenida con tres vecinos.

1.3. Alcance de la investigación

1. El algoritmo de Recocido Simulado será evaluado con diferentes estructuras de vecindad como son: perturbación simple, pares aleatorios, triples aleatorios, cuádruples aleatorios y una estructura de vecindad híbrida, que permita una mejor explotación del espacio de soluciones.

2. Se realizará un estudio de sensibilidad a los parámetros de control del algoritmo de Recocido simulado, lo que mejorará el tiempo de convergencia y mejorará la calidad de las soluciones del algoritmo propuesto.

3. Los resultados obtenidos durante las pruebas experimentales realizadas por el algoritmo de Recocido Simulado se compararon con los obtenidos por el algoritmo Genético reportado en la literatura, tomando en cuenta la evaluación con el mismo número de instancias y con una nueva vecindad.

4. Las instancias de prueba son generadas de forma aleatoria y se utilizan tanto para el algoritmo propuesto como el algoritmo genético, lo cual servirá para realizar una comparación en cuanto a la eficiencia y eficacia del algoritmo.

5. No existe un conjunto de restricciones, por lo tanto, no se generará un modelo matemático.

1.4. Contribución de la tesis

En este trabajo de investigación se aplicó un algoritmo de Recocido Simulado con dos modificaciones para maximizar el costo de la solución, para la predicción de los signos de los coeficientes wavelet aplicado a la compresión

de imágenes 2D. Una estructura de vecindad sencilla prueba ser eficiente y eficaz. En base a la investigación sobre estructuras de vecindad, la estructura híbrida demuestra encontrar la calidad de la solución pero con un costo computacional mayor que la estructura sencilla. Por lo que en este trabajo de investigación se evalúa el desempeño de la estructura de vecindad simple para el problema de la predicción de los signos de los coeficientes wavelet y se concluye que para este problema el desempeño de dicha estructura es mejor que las otras estructuras de vecindad.

La aportación de este trabajo de investigación para resolver el problema tratado, es aplicar la predicción de signo con 5 coeficientes vecinos, para mejorar significativamente el costo de la solución.

Como parte fundamental del algoritmo de Recocido Simulado, el análisis de sensibilidad de los parámetros de control (parámetro de control inicial, longitud de cadena de Markov, factor decremento beta y parámetro de paro), son términos independientes de las restricciones. De acuerdo a lo anterior, se propone una metodología para la realización del análisis de sensibilidad, el cual lleva a cabo la sintonización de las variables de entrada del algoritmo, permitiendo conocer los límites inferior y superior al cambio de valor de ciertos parámetros y de esta forma determinar el rango de valores dentro del cuál, la solución es buena.

1.5. Organización de la Tesis

La organización del presente trabajo de investigación se basa en capítulos en la siguiente forma: en el capítulo uno se da una introducción al problema a tratar; se realiza una breve explicación del estado del arte y se enumeran los objetivos y alcances de la investigación. En el capítulo dos se dará una breve visión de la necesidad de utilizar distintas técnicas de compresión de imágenes basadas en transformadas wavelet. En el capítulo tres se analizará cómo funciona el algoritmo de Recocido Simulado, se describe como encontrar la solución inicial y cómo se ha realizado la implementación para realizar la predicción de signo, la estructura de vecindad utilizada, el comportamiento, la metodología de sintonización y el análisis de complejidad del algoritmo. En el

capítulo cuatro se muestran los resultados experimentales de la evaluación del algoritmo de Recocido Simulado, que prueban el correcto funcionamiento del algoritmo mostrando gráficamente la calidad de los resultados obtenidos, además se detallará el análisis de sensibilidad, eficiencia y eficacia, así como una serie de pruebas y evaluaciones para comparar la calidad de la solución obtenida por el algoritmo genético reportado en la literatura y el algoritmo de Recocido Simulado para las imágenes estándar de prueba Lena y Bárbara, utilizadas para todo tipo de pruebas en el tratamiento digital de imágenes, para la combinación de 3 signos con 3 coeficientes vecinos. Se realizaron pruebas para la vecindad propuesta de la combinación de 3 signos con 5 coeficientes vecinos y se detalla un análisis del rendimiento de forma teórica. En el capítulo cinco expondremos las conclusiones obtenidas tras la realización del presente trabajo de investigación, así como los trabajos a futuro.

Por último se detalla la lista bibliográfica consultada para la realización de este trabajo de investigación.

Capítulo 2

Predicción de signo de los coeficientes de la Transformada Wavelet

En la actualidad, el tratamiento digital de imágenes ha progresado vigorosamente, las técnicas de procesamiento son empleadas para resolver problemas muy diversos que requieren normalmente métodos capaces de realzar la información de las imágenes para la interpretación y el análisis humano [González y Woods, 1996].

Para este caso, el interés se centra en los procedimientos para extraer la información de la imagen de forma conveniente para el procesamiento por computadora. En el campo de la informática existen multitud de algoritmos y aplicaciones dedicados a la compresión tanto de imágenes digitales como audio y video. Dentro de la teoría de la información se buscan técnicas que ayuden a reducir el tamaño de la información a transmitir o almacenar. Esto se complica a medida que se presentan diferentes tipos de problemas, también si no se diseña un buen método de compresión eficiente que ayude a reducir el tamaño de la información. Perder información es la incapacidad de un método o algoritmo para reconstruir el producto original, a partir de otro al que se ha aplicado un proceso de compresión, sin embargo puede sacrificarse información original sin que el ser humano perciba los cambios, logrando así mejores tasas de compresión.

Las técnicas de compresión se clasifican generalmente en dos grupos: técnicas de compresión sin pérdida de información y técnicas de compresión con pérdida de información. Las técnicas de compresión sin pérdida de información se caracterizan porque la tasa de compresión que proporcionan está limitada por la entropía (redundancia de datos) de la señal original. Las técnicas de compresión con pérdida de información pueden lograr tasas de compresión muy elevadas, pero sacrificando cierta pérdida de información que no será posible recuperar durante el proceso de descompresión. Nosotros nos

centraremos en el análisis de la información de imágenes digitales para tratar ésta con pérdida de información [Effelsberg y Steinmetz, 1998].

El sistema visual del hombre responde con diferente grado de sensibilidad a la información que recibe, si podemos reducir la información a la que es menos sensible, podremos descartarla sin afectar a la percepción de la imagen. La eliminación de esta redundancia es conocida como redundancia visual, la cual requiere de un proceso de cuantificación de información. Este proceso generará una pérdida de información irreversible.

Una imagen digital consiste inicialmente en una matriz de píxeles que representan niveles de intensidad, pero este formato no es el más adecuado para las tareas de análisis. Es por ello que mediante el análisis de señales se puede extraer información relevante mediante transformaciones.

Existen dos razones para transformar una imagen de una representación a otra: que la transformación permita aislar elementos importantes de la imagen facilitando así su análisis o bien, que la transformación disponga los datos de manera más compacta para que la imagen pueda almacenarse y transmitirse de una manera más eficiente.

Utilizando técnicas y algoritmos de compresión de imágenes se puede lograr una buena tasa de compresión de bits, aunado a ello, codificando los signos de los coeficientes wavelet, se puede obtener una mayor tasa de compresión reduciendo la entropía.

2.1. Análisis de señales mediante Wavelet

Algunos tipos de información utilizados en la percepción automatizada son los momentos estadísticos, los coeficientes de la transformada de Fourier, transformada Wavelet y las medidas de distancia multidimensionales. Por lo que es necesario el análisis basado en texturas. Una textura es un atributo que presenta la distribución espacial de los niveles de gris de una determinada región [IEEE Standard 610.4,1990].

En los últimos años muchos de los métodos y técnicas orientados a la codificación de imágenes han sido enfocados a la transformada wavelet, las cuales se han convertido en una herramienta estándar debido a su gran capacidad de compresión de datos [Lewis y Knowles, 1992].

La transformada Wavelet Discreta (DWT) es similar a la transformada Coseno discreta (DCT) en cuanto a la descomposición de la señal original en coeficientes asignados a funciones base [Yves, 1994].

La DCT ésta limitada a funciones coseno que requieren bastantes operaciones aritméticas. Por otra parte, la teoría de wavelet utiliza un rango amplio de funciones base, dando como resultado una menor complejidad computacional sin sacrificar la calidad de la imagen.

2.2. Transformada Wavelet

La transformada Wavelet es una herramienta matemática de reciente desarrollo que permite descomponer una señal, no sólo en componentes frecuenciales, sino también espaciales. Los coeficientes de la DWT nos sitúan distintos componentes frecuenciales en posiciones espaciales mediante un análisis por subbandas. No entraremos en términos matemáticos, sino en términos generales para una mejor comprensión es necesario definir lo que entendemos por imagen digital.

Una imagen digital es una imagen $f(x,y)$ que se ha discretizado, tanto en las coordenadas espaciales como en el brillo, puede considerarse como una matriz donde sus índices de fila y columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Los elementos de este tipo se denominan elementos de la imagen, o más comúnmente pixeles o pels, abreviaturas de su denominación inglesa "picture elements".

La aplicación de la DWT sobre una imagen proporciona una matriz de coeficientes conocidos como coeficientes wavelet. Cuando a una imagen le es aplicada la DWT se obtienen cuatro tipos de coeficientes: aproximaciones, detalles horizontales, detalles verticales y detalles diagonales. Todo coeficiente wavelet estará formado por un valor de cierta magnitud y su correspondiente signo (positivo o negativo).

La principal característica de la transformada wavelet es su capacidad para representar la señal en el dominio espectral y temporal de forma simultánea, es decir, representa la señal espacial en los ejes tiempo-frecuencia-amplitud. Se precisa de técnicas de filtrado digital (pasando la señal por filtros paso bajo y paso alto), en lo que se conoce como descomposición en subbandas o piramidal.

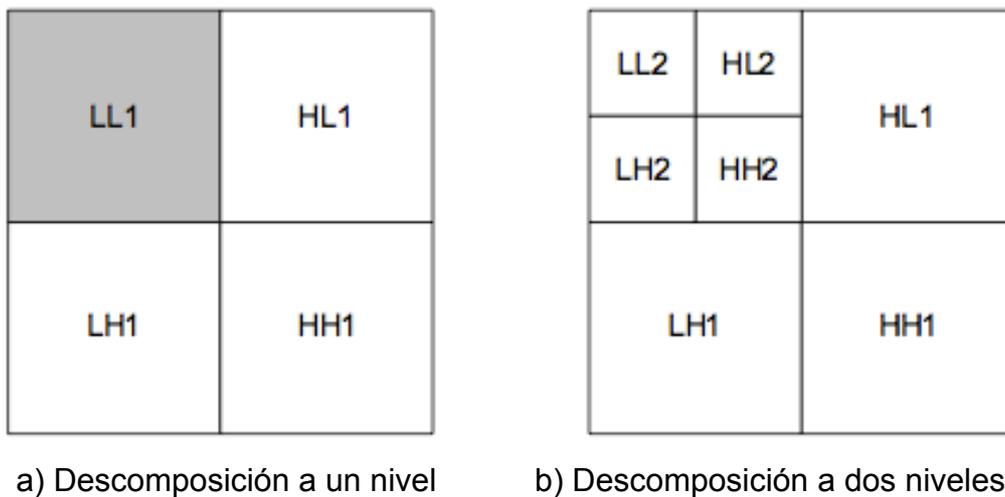


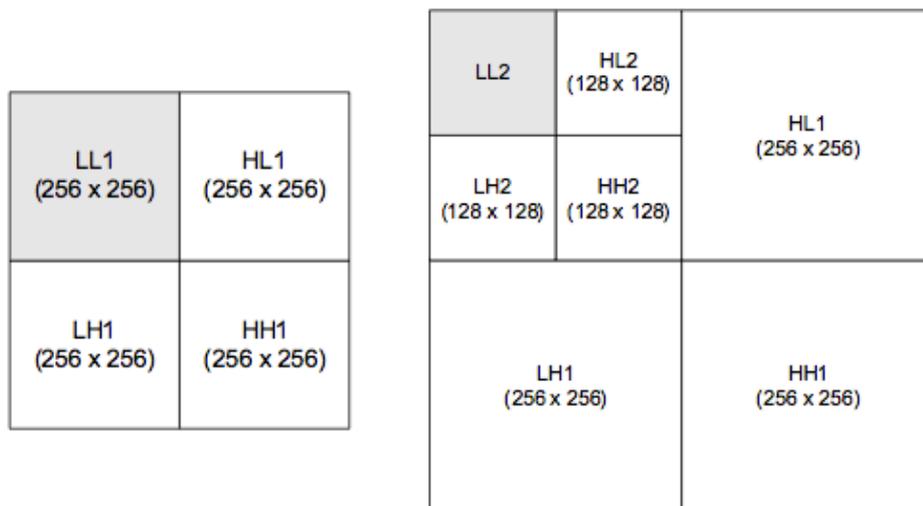
Figura 2-1. Nivel de descomposición en subbandas wavelet.

En la figura 2-1 se muestra la típica descomposición en subbanda para una imagen en un nivel (a) y dos niveles (b). Se puede observar que en el primer nivel de descomposición se obtienen cuatro subbandas. La subbanda LL_1 corresponde a un filtrado paso bajo de la imagen de entrada y las tres restantes con los niveles de detalle vertical, horizontal y diagonal (HL_1, LH_1 y HH_1 , respectivamente), se obtienen filtrando paso bajo y paso alto las filas y columnas de la imagen.

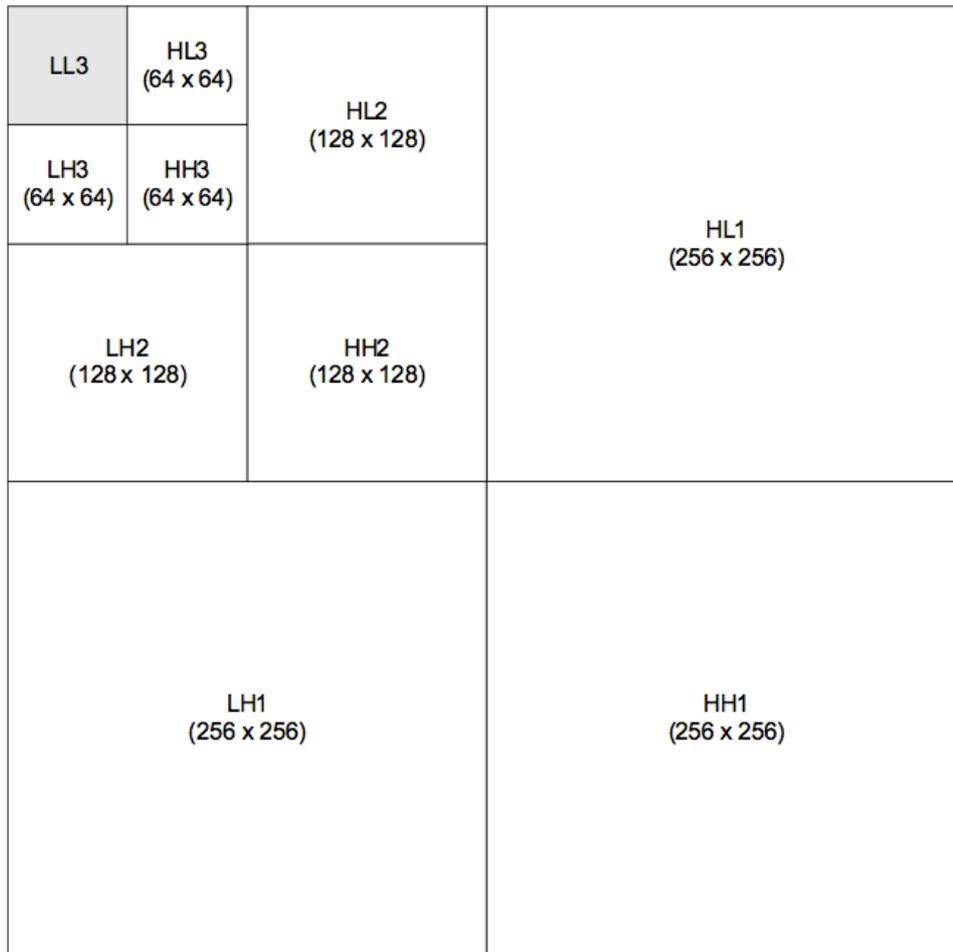
Es decir, la subbanda de detalle vertical HL_1 implica un filtrado paso alto de las filas y paso bajo de las columnas. La subbanda LH_1 implica un filtro paso bajo a las filas y paso alto a las columnas, finalmente para la subbanda HH_1 se aplica un filtro paso alto en filas y columnas.

El segundo nivel de descomposición repite el proceso descrito pero tomando como imagen de entrada la subbanda LL_1 . De esta forma, la transformada wavelet permite el estudio de una imagen en varias resoluciones, ofreciendo desde una visión más general de la imagen hasta otras más particulares que permiten apreciar los más mínimos detalles.

En este trabajo de investigación se ha aplicado la transformada wavelet de seis dimensiones (DWT-6D), de esta forma conseguiremos un nivel de detalle adecuado. En la figura 2-2 se muestra la descomposición a tres niveles de una imagen de 512 x 512 pixeles de resolución de profundidad de 8 bits/pixel. Al ser una imagen cuadrada, todas las subbandas al pasar de un nivel de descomposición a otro tendrán la misma dimensión, por ejemplo, en el análisis de la subbanda HL para una descomposición de seis niveles dentro de la matriz de 512 x 512 pixeles, habrá un total de seis matrices de tamaños: 256 x 256 bits, 128 x 128 bits, 64 x 64 bits, 32 x 32 bits, 16 x 16 bits y 8 x 8 bits respectivamente.



a) Descomposición de un nivel b) Descomposición de dos niveles



c) Descomposición de tres niveles

Figura 2-2. Descomposición wavelet de a) uno b) dos c) tres niveles.

Únicamente analizaremos los signos de las subbandas HL, LH y HH, ya que la subbanda LL tendrá positivos todos los signos de sus coeficientes.

2.3. Codificación del signo

Para comprender la codificación del signo, explicaremos como se comporta un codificador de imágenes. Nos centraremos en el codificador LTW [Oliver y Malumbres, 2006], empleado para la obtención de los resultados de este trabajo de investigación. En la figura 2-3 se muestra el esquema general para el procesado de una imagen.

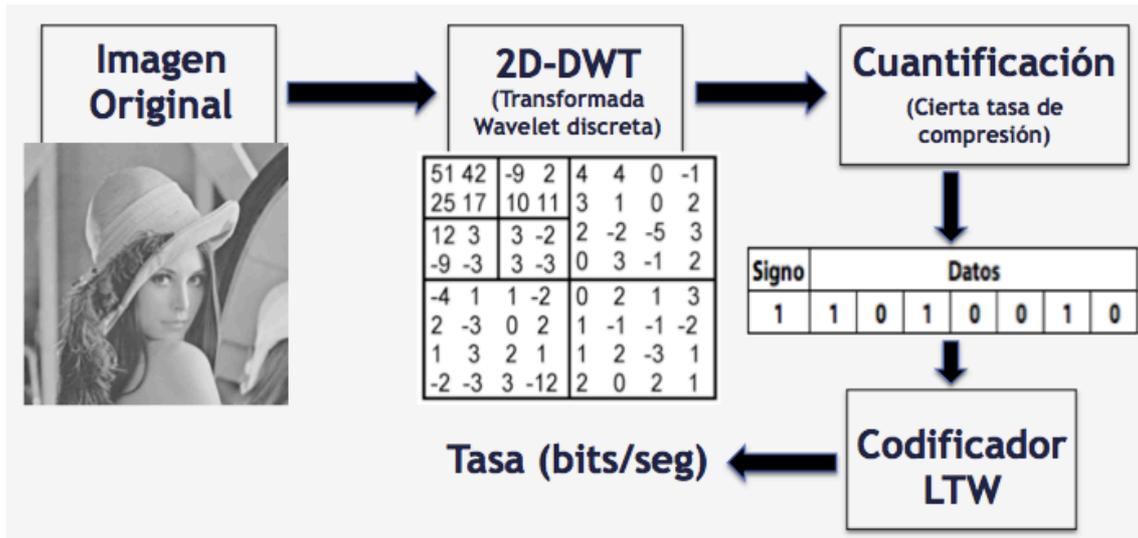


Figura 2-3. Procesado de una imagen.

El LTW es un codificador basado en la transformada wavelet, el cual intenta mantener la eficiencia en la compresión producida por el estándar de compresión JPEG 2000, bajo un costo computacional sensiblemente menor. Para ello, el codificador LTW es aplicado a un conjunto de coeficientes C , resultado de la aplicación previa de un proceso transformado (DWT). Cada elemento $C_{i,j}$ del conjunto de coeficientes C , será denominado coeficiente transformado. El resto de la transformación de la imagen será aplicado de forma recursiva hasta completar el nivel de descomposición N deseado para obtener los detalles horizontales, verticales y diagonales de las subbandas HL_1 , LH_1 y HH_1 respectivamente.

El codificador LTW ordena los coeficientes en forma de árbol, este tipo de ordenación será empleada para reducir la redundancia de coeficientes entre subbandas. De esta forma se logra reducir el número de símbolos necesarios para codificar la imagen. La figura 2-4 muestra la ordenación en forma de árbol.

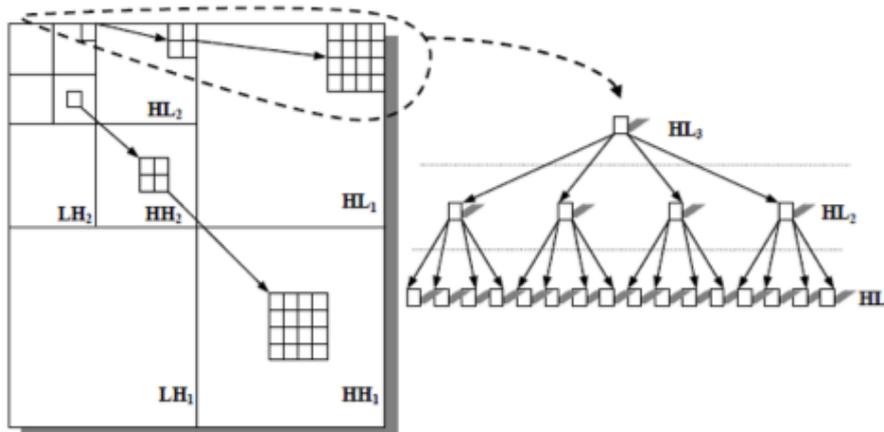


Figura 2-4. Ordenación de los coeficientes en forma de árbol.

El mapa de coeficientes wavelet se construye tras el proceso de cuantización de los coeficientes, como se muestra en la figura 2-5 una pequeña imagen de 8 x 8 que ha sido transformada utilizando nivel de descomposición DWT-2D.

51 42	-9 2	4 4 0 -1
25 17	10 11	3 1 0 2
12 3	3 -2	2 -2 -5 3
-9 -3	3 -3	0 3 -1 2
-4 1 1 -2	0 2 1 3	
2 -3 0 2	1 -1 -1 -2	
1 3 2 1	1 2 -3 1	
-2 -3 3 -12	2 0 2 1	

Figura 2-5. DWT de segundo nivel de una imagen de 8x8 píxeles.

Utilizando esta imagen muestra, explicaremos cómo funciona el codificador basado en árboles. Se define $rplanos$ como el número de bits menos significantes a remover. Los coeficientes han sido cuantificados y el parámetro seleccionado es $rplanos=2$. Con respecto al parámetro máximo será computado como $maxplano=\lceil \log_2(51) \rceil=6$. Cuando la cuantización es aplicada utilizando $rplanos=2$, los valores dentro del intervalo $-(2)^2 \dots 2^2$ son absolutamente cuantificados. Así, todos los coeficientes significantes pueden ser representados utilizando de 3 a 6 bits y por lo tanto, el conjunto de símbolos necesarios para representar el mapa de coeficientes significantes es $\{3,4,5,6,3^L,4^L,5^L,6^L,L,I\}$. De forma genérica, se emplea la siguiente notación para el conjunto de símbolos: los símbolos llamados LOWER symbol (L),

representan los coeficientes cuyas raíces son las inferiores en el árbol, mientras que el resto de los coeficientes no son codificados y son denominados LOWER_COMPONENT (representados con un *). La figura 2-6 muestra los símbolos resultantes de aplicar el algoritmo LTW a la imagen de muestra.

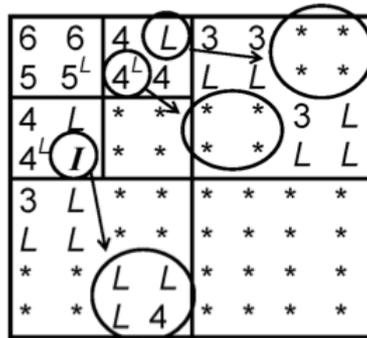


Figura 2-6. Mapa de símbolos resultantes del LTW

Para los casos en los que un coeficiente tenga un valor insignificante, estos serán etiquetados como ISOLATED_LOWER (representados por I). Para los coeficientes más significativos se emplea un símbolo indicando el número de bits necesarios para poder ser representados. También en el caso de que tengamos un coeficiente que sea la raíz del árbol, se suele emplear el número de bits necesarios para representarlo con el superíndice L.

El proceso de codificación del algoritmo se realiza en dos etapas. En la primera, llamada computación de símbolos, todas las subbandas son analizadas en bloques de coeficientes de tamaño 2 x 2, desde el primer nivel de descomposición hasta el último. En el primer nivel de descomposición, si los cuatro coeficientes de cada bloque 2 x 2 son insignificantes, entonces serán considerados parte del mismo bloque, parte inferior del árbol, etiquetados como LOWER_COMPONENT.

Entonces, cuando se analizan niveles de subbandas superiores, si un bloque de tamaño 2 x 2 tiene sus cuatro coeficientes insignificativos y todos son descendientes directos se etiquetan como LOWER_COMPONENT.

Sin embargo, cuando se encuentra un coeficiente significativo dentro de un bloque, el análisis por bloque no continúa, ya que para este caso, cada coeficiente tendrá que ser analizado individualmente. Cada coeficiente insignificativo dentro de un bloque pasará a ser etiquetado como LOWER, pero

si todos sus descendientes son LOWER_COMPONENT, serán etiquetados con el símbolo ISOLATE_LOWER.

Por otra parte, para cada coeficiente significativo se empleará el número de bits necesarios para representar dicho símbolo. Pero si todos los descendientes de un coeficiente significativo son insignificativos, es decir LOWER_COMPONENT, utilizaremos un símbolo especial indicando el número de bits necesarios para representarlo junto con el superíndice L.

En la segunda etapa, las subbandas son codificadas partiendo de la subbanda LL_N , tal y como se muestra en la figura 2-7.

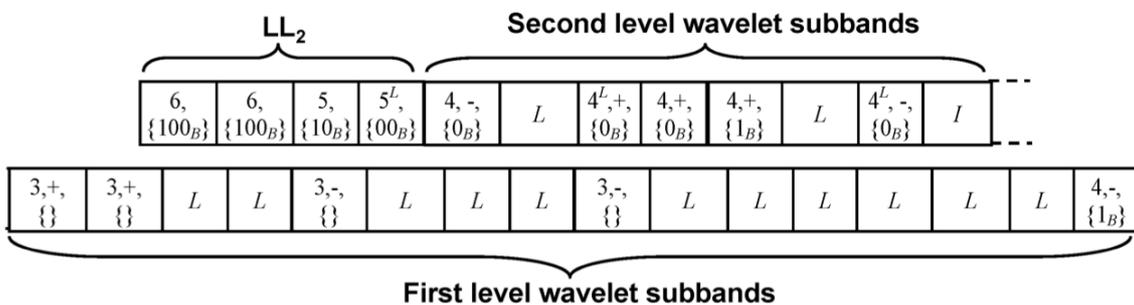


Figura 2-7. Ejemplo de codificación utilizando LTW

Además, este orden de escalonado proporciona una versión reducida en tamaño de la imagen original, figura 2-8. Para cada subbanda en los bloques de tamaño 2×2 , los símbolos computados en la primera etapa son codificados mediante una función de entropía.



a) Imagen original de Bárbara b) Organización de los coeficientes wavelet

Figura 2-8. Ejemplo de transformada wavelet.

2.4. Contexto de la vecindad

En esta sección del capítulo justificaremos el criterio que va a seguir el algoritmo de Recocido simulado para realizar la predicción de signo de los coeficientes wavelet.

El contexto es determinado por un conjunto de vecinos, donde para codificar el signo del coeficiente wavelet se utiliza información de los signos de los coeficientes de los vecinos horizontales y verticales en las direcciones N (North), E (East), W (West) y S (South). El codificador LTW emplea el llamado recorrido Morton [Morton, 1966] para recorrer las subbandas de menor a mayor frecuencia (zig-zag), esto hace que al recorrer el mapa de signos de una imagen no tengamos información disponible para los vecinos E (East) y S (South). Esto es una restricción para muchos codificadores no progresivos (no empotrados) cuando buscamos correlación del signo entre los vecinos compartidos. Para identificar buenos vecinos como predictores de signo y con alta correlación, se define un conjunto de vecinos con una distancia máxima de dos posiciones a partir del coeficiente X a evaluar [López 2010], en este caso, el conjunto de posibles vecinos serán: **W**(West), **N**(North), **WW**(West-West), **NN**(North-North) y **NW**(North-West) y también **NNW** (North North-West), véase figura 2-9.

	NNW	NN	
	NW	N	
WW	W	X	E
		S	

Figura 2-9. Mapa de signos de los vecinos

Todos los vecinos tendrán tres posibles valores para el signo, positivo (+), negativo (-) o no significativo (*), mientras que el coeficiente a evaluar tendrá dos posibles valores para el signo, positivo (+) o negativo (-), ya que los no significativos no son evaluados, esto hace que tengamos un total de 2×6^3 combinaciones de signos vecinos (Neighbor Sign Patterns, NSP), donde el

número dos representa los posibles valores para el signo (positivo o negativo) del coeficiente a evaluar, la base representa seis vecinos y la potencia representa los posibles valores para el signo (positivo, negativo o no significativo) para cada vecino.

Hemos obtenido la probabilidad de distribución de las combinaciones del signo y disminuido severamente los vecinos intentando agrupar el mayor número de posibilidades para encontrar el mayor número de aciertos entre comparar el signo resultante del NSP con el signo del coeficiente a evaluar (**función objetivo del problema**).

Sobre la imagen a analizar aplicamos nivel 6 de descomposición wavelet y un bajo nivel de cuantización para obtener los coeficientes transformados. Una vez aplicado lo anterior, obtenemos la probabilidad de distribución del patrón de signos (NSP) para cada coeficiente wavelet. De esta forma, para la subbanda HL, los vecinos cuyos signos están más correlados con el signo del coeficiente a evaluar son N, NN y W. Teniendo en cuenta la simetría para la subbanda LH, los vecinos serán W, WW y N, mientras que para la subbanda HH serán N, NW y W, ver figura 2-10. Así para cada tipo de subbanda somos capaces de reducir el patrón de signos vecinos (NSP) a un máximo de 3³ posibles combinaciones.

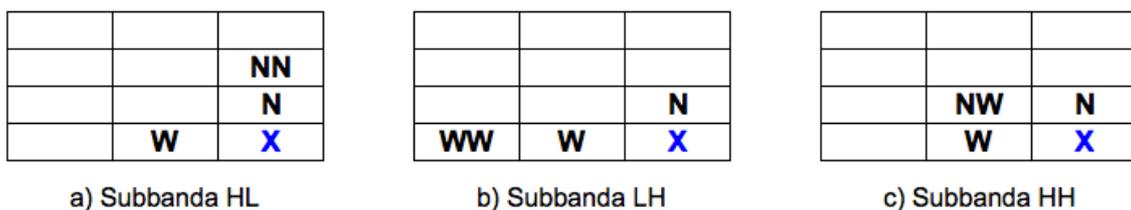


Figura 2-10. Distribución de vecinos a analizar por tipo de subbanda para tres vecinos.

En la tabla 2-1, se muestra la probabilidad de predecir el signo de los vecinos para la subbanda HL de 6 niveles de descomposición wavelet para la imagen de Lena de 8 x 8 pixeles. Podemos observar que, con una probabilidad superior al 20%, el signo del coeficiente será positivo cuando lo sean sus vecinos N, NN, y W. También podemos observar que cuando los vecinos N y

NN tienen el mismo signo y el vecino W tiene signo opuesto, el coeficiente también tendrá el mismo signo que el vecino W con una probabilidad de 25%.

Tabla 2-1. Probabilidad de distribución del mapa de signos para la subbanda HL₆ de la imagen Lena.

Coefficiente	N	NN	W	Ocurrencias	Probabilidad (%)
+	+	+	+	13	20.31
+	+	+	-	8	12.50
-	-	-	+	8	12.50
-	+	+	+	6	9.38
-	-	+	+	6	9.38
Resto de combinaciones				23	35.93

La distribución del patrón de signos (NSP) para cinco vecinos consta de 3⁵ elementos de la combinación de 3 signos con cinco vecinos e investigando con el Dr. Otoniel López de la Universidad Miguel Hernández en Alicante, España, los vecinos cuyos signos están más correlados con el signo del coeficiente a evaluar para la subbanda HL son los que tratan de encontrar correlación en los bordes verticales tanto a lo largo, como cuando los cruzamos, por lo que se propone para la subbanda HL: N, NN, NNN, W, WW, para la subbanda LH: W, WW, WWW, N, NN, opuestamente a la HL busca bordes horizontales y para la subbanda HH: N, W, NW, NNWW, NNNWWW busca bordes diagonales, ver figura 2-11. Así para cada tipo de subbanda somos capaces de reducir el patrón de signos vecinos (NSP) a un máximo de 3⁵ posibles combinaciones.

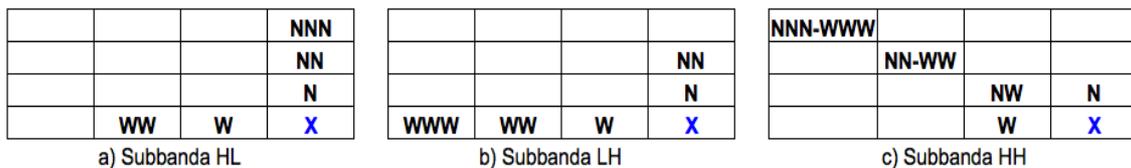


Figura 2-11. Distribución de coeficientes vecinos a analizar por tipo de subbanda para cinco vecinos.

Después de que se analizan todas las posibles combinaciones de los vecinos para cada tipo de subbanda y para cada nivel de descomposición wavelet, podemos hacer una predicción de signo del coeficiente actual.

Se parte de un vector formado por la combinación de los posibles signos de los coeficientes vecinos por el número de vecinos, consideremos tres posibles

valores para el signo negativo(-), positivo(+) y nulo (*), para cada tipo de subbanda (HL, LH y HH).

Se consideran tres y cinco vecinos a analizar para realizar la predicción de signo de los coeficientes wavelet, con tres vecinos se considera adecuada [López, 2010] y la aportación de este trabajo de investigación es analizar la predicción de signo con cinco vecinos. Por lo tanto, el vector para tres vecinos tendrá 27 elementos resultado la combinación de tres coeficientes vecinos por tres signos como lo podemos ver en la tabla 2-2.

Tabla 2-2. Elementos del vector NSP para tres vecinos.

Combinación	Vecino 1	Vecino 2	Vecino 3	Signo
0	*	*	*	Inicializamos de forma aleatoria almacenando valores que serán cero para predicción de signo positiva y 1 para predicción de signo negativa. (Solución inicial)
1	*	*	+	
2	*	*	-	
3	*	+	*	
4	*	+	+	
5	*	+	-	
6	*	-	*	
7	*	-	+	
8	*	-	-	
9	+	*	*	
10	+	*	+	
11	+	*	-	
12	+	+	*	
13	+	+	+	
14	+	+	-	
15	+	-	*	
16	+	-	+	
17	+	-	-	
18	-	*	*	
19	-	*	+	
20	-	*	-	
21	-	+	*	
22	-	+	+	
23	-	+	-	
24	-	-	*	
25	-	-	+	
26	-	-	-	

Para el caso de cinco coeficientes vecinos el vector tendrá 243 elementos, debido al tamaño no es posible mostrarlo como imagen o tabla, pero la combinación se realiza de la siguiente manera: para el vecino 1 cada 81 elementos cambia de signo, el vecino 2 cada 27 elementos cambia de signo, el vecino 3 cada 9 elementos cambia de signo, el vecino 4 cada 3 elementos cambia de signo y el vecino 5 cada elemento es un signo diferente.

Una vez definidos los tamaños de los vectores (**solución inicial**) la combinación para tres y cinco coeficientes vecinos y el contexto para cada tipo de subbanda, se inicializan los vectores de forma aleatoria almacenando valores que serán cero para predicción de signo positiva y uno para predicción de signo negativa.

El vector resultante con el mayor número de aciertos será el que mejor predicción de signo ha encontrado, éste vector predictor es incrustado dentro del codificador LTW para la compresión de imágenes 2D, como se puede ver en la figura 2-12.

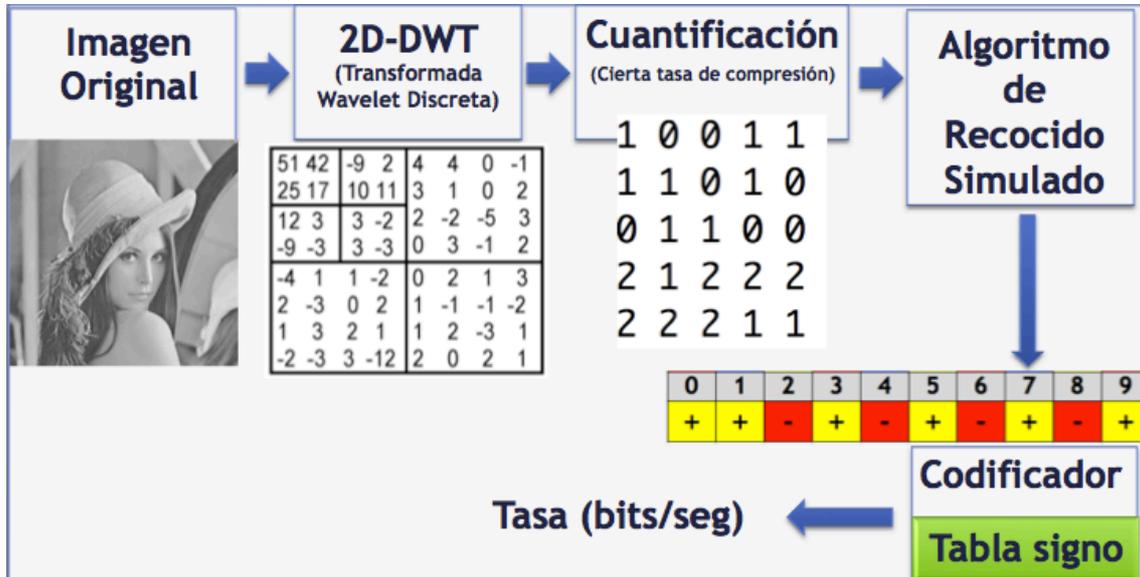


Figura 2.12. Integración del signo en el codificador.

Capítulo 3

Algoritmo de Recocido Simulado

Recocido Simulado (RS) conocido en Inglés como *Simulated Annealing*, es una técnica basada en el recocido de sólidos y fue propuesta por primera vez por [Kirkpatrick *et al.*, 1983]. Esta técnica basa su analogía en el proceso de recocido de sólidos, cuando un material es sometido a altas temperaturas al punto de fusión, posteriormente es enfriado gradualmente, esto hace que las moléculas sean acomodadas de tal forma que la energía potencial de la configuración de las moléculas sea mínima, alcanzando una mejor resistencia.

El Recocido Simulado es una técnica como solución a problemas de optimización de gran tamaño. Ha sido aplicado al problema del agente viajero TSP [Kirkpatrick *et al.*, 1983]. También se ha empleado con éxito para muy diversos propósitos: diseño automático de circuitos integrados [Kirkpatrick *et al.*, 1983], supresión automática de ruido en imágenes digitales al objeto de restaurarlas [Stuart y Geman, 1984] y análisis sintáctico del lenguaje humano [Hirts y Selman, 1987] entre otros.

El Recocido Simulado de un objeto para mejorar alguna de sus propiedades consiste en producir al azar cambios en su descripción. Si es posible definir una temperatura en relación a tal propiedad, el proceso comienza a altas temperaturas en la que todos los cambios aleatorios son aceptados, en consecuencia el aspecto del objeto inicial cambia continuamente, y continua con un paulatino y lento descenso de la temperatura, de manera que cada vez el número de cambios aceptados disminuye más y más y una forma particular del objeto acaba por emerger. Finalmente cuando la temperatura es tan baja que ningún cambio se acepta, el algoritmo se detiene con el objeto inicial profundamente alterado, objeto que probablemente pueda encontrarse para la propiedad de interés, dicha propiedad es la función objetivo.

La probabilidad de aceptación puede adoptar diferentes métodos estocásticos como son el método Monte Carlo cuando la distribución de probabilidad es la

de Boltzmann. El algoritmo básico es el de Metrópolis, que permite simular el comportamiento de un sistema a temperatura constante. El fundamento matemático es posible a través de la teoría de probabilidades, en particular, a través de cadenas de Markov.

El algoritmo de Recocido Simulado como una secuencia de algoritmos de metrópolis a temperatura decreciente en forma monótona, legitima el uso de la temperatura en un sentido muy amplio del cual derivan las ecuaciones fundamentales del algoritmo a partir de la teoría de la información.

3.1. Descripción del Algoritmo de Recocido Simulado

El algoritmo de Recocido Simulado es una búsqueda local iterativa guiada a través de un proceso estocástico, donde un estado j es aceptado con una probabilidad dada por la función de distribución de Boltzmann.

El algoritmo de Metrópolis simula el cambio de energía en el proceso de enfriamiento de un sistema físico. Las leyes de la termodinámica establecen que a una temperatura T , la probabilidad de un aumento de energía de magnitud ΔE viene dada por la expresión de la tabla 3-1.

Tabla 3-1. Función de distribución de Boltzmann.

$$P(\Delta E) = e^{-\Delta E / (k \cdot T)}, \text{ donde } k \text{ es la constante de Boltzmann.}$$

La simulación de enfriamiento físico en la transición de líquido a sólido se apoya en el algoritmo de metrópolis [Metrópolis *et al.*, 1953] de la siguiente manera: dado un estado actual i con su temperatura T , generar un estado j mediante una perturbación, si la diferencia de energías (ΔE) entre el estado actual y el perturbado es menor o igual que cero, el estado j es aceptado como actual, si $\Delta E > 0$ el estado es aceptado con probabilidad dada por la función de distribución de Boltzmann; el algoritmo 3-1 muestra el algoritmo de Metrópolis con un número α aleatorio generado uniformemente distribuido entre $(0, 1)$, si α

$< \text{Probabilidad de aceptación el estado } j \text{ se acepta como el actual, si } \alpha \geq \text{Probabilidad de aceptación el estado } j \text{ se rechaza hasta alcanzar el equilibrio térmico para lo cual es necesario que la distribución de probabilidad de las configuraciones alcance el equilibrio estocástico para la distribución de Boltzmann [Laanhover et al., 1992].$

Algoritmo 3-1. Algoritmo de Metrópolis.

Dado un estado actual i con su temperatura T
 Generar un estado j mediante una perturbación.
 Si $\Delta E \leq 0$ el estado j es aceptado como actual.
 Si $\Delta E > 0$ el estado es aceptado con probabilidad
 $\text{Probabilidad de aceptación} = e^{(\Delta E / k^* T)}$
 Con un número α aleatorio generado uniformemente
 distribuido entre $(0, 1)$
 Si $\alpha < \text{Probabilidad de aceptación}$ el estado j se
 acepta como el actual
 Si $\alpha \geq \text{Probabilidad de aceptación}$ el estado j se
 rechaza
 Se itera hasta que se alcanza el equilibrio térmico.

El algoritmo de Recocido Simulado originalmente está diseñado para minimizar el costo de una función objetivo. Para este trabajo de investigación se requiere maximizar el costo de la función objetivo. Por lo que se realizaron dos modificaciones sobre el criterio de aceptación del Recocido Simulado (ver tabla 3-5). La primera modificación acepta como soluciones buenas valores que maximicen la función de costos y la segunda modificación valida soluciones con alto número de aciertos omitiendo el signo negativo de la función de Boltzmann. La función objetivo como ya se explicó anteriormente en el capítulo 2, plantea encontrar el mayor número de aciertos entre la comparación de cada NSP y el signo del coeficiente a evaluar.

El objetivo del algoritmo de Recocido Simulado aplicado al problema de la predicción de los signos de los coeficientes wavelet, es encontrar una tabla donde para cada NSP (V_k) hay una predicción de signo ($S_{i,j}$) para cada coeficiente $C_{i,j}$, de forma que no existe una relación unívoca entre combinar los signos de los vecinos, por ejemplo, para el mismo patrón V_k , no siempre la predicción $S_{i,j}$ es positiva o negativa. Es posible que para el mismo patrón V_k , la predicción de signo $S_{i,j}$ tenga mayor tendencia a ser positiva que negativa. Pero

el problema es aún más complejo porque una predicción de signo para un NSP puede corresponder bien para una imagen específica y no para otras.

El algoritmo de Recocido Simulado para un conjunto de vecinos ya identificados como buenos predictores de signo, obtiene para cada tipo de subbanda (HL, LH y HH), la distribución del mapa de signos con el objetivo de que se minimice la entropía (ver figura 3-1).

Entropía o incertidumbre de la fuente, define la cantidad medida de información (en unidades r-arias por símbolo) obtenida observando una sola salida de la fuente. Conforme incrementa su magnitud, más incertidumbre y por tanto más información esta asociada a la fuente. Si los símbolos de la fuente son igualmente probables, la entropía queda maximizada y la fuente proporciona el máximo valor posible de información promedio por cada símbolo de la fuente [González, 1996]. En términos generales, podemos definir la entropía de un mensaje (datos del mensaje, suceso, predicción, etcétera), como el límite máximo al que se puede comprimir un mensaje, usando un enfoque símbolo a símbolo sin ninguna pérdida de información [Shannon, 1948], el límite de compresión (en bits) es igual a la entropía multiplicada por la longitud del mensaje.

$$\begin{aligned}
 H(X) = E(I(X)) &= \sum_{i=1}^n p(x_i) \log_a \left(\frac{1}{p(x_i)} \right) \\
 &= - \sum_{i=1}^n p(x_i) \log_a p(x_i)
 \end{aligned}$$

Figura 3-1. Definición formal de entropía.

En busca de mejorar el costo de la solución hemos utilizado diferentes estructuras de vecindad aplicadas al algoritmo de Recocido Simulado, debido a que muchos problemas tratados en la optimización combinatoria requieren de una búsqueda de soluciones aproximadas y ajustadas a la complejidad y naturaleza del problema [Cruz y Martínez, 2010]. Para este trabajo de investigación cinco estructuras diferentes de búsqueda de vecindad son aplicadas en busca de una mejor solución en base a la función objetivo. La

parte esencial de las estructuras de vecindad son el tamaño y su estructura, mientras el tamaño sea más grande, la calidad de soluciones óptimas puede ser mejor y ello lleva a encontrar una solución final. Para elegir la mejor estructura de vecindad se realizó un análisis de eficiencia y eficacia sobre un conjunto de estructuras para determinar cuál se adecuada mejor al problema de la predicción de signos de los coeficientes wavelet.

Una vecindad es el conjunto de soluciones las cuales se pueden alcanzar a partir de una solución s por medio de un movimiento σ/s [Papadimitriou y Steiglitz, 1998], que puede ser un intercambio, inserción o eliminación entre los elementos de una solución s . Tomando en cuenta la definición anterior, una vecindad es definida como el conjunto de soluciones cercanas de una solución inicial, $s \in S$ en una instancia del problema, tal que, el conjunto $N(s)$ es factible en un punto cercano a s . El conjunto $N(s)$ (ver figura 3-2) llamado vecindad de s , indica que cada solución $s' \in N(s)$ puede ser alcanzado directamente desde el valor actual de la función objetivo en un sólo paso. De acuerdo con esto la vecindad es defina por la función $N: S \rightarrow 2^S$ [Martínez, 2010]. Para mejorar la solución s , es necesario movernos paso por paso desde la solución inicial factible hacia una solución que proporciona el mínimo valor de la función objetivo, la cual usualmente implica el costo. Para nuestro problema de la predicción de los signos de los coeficientes wavelet, la función objetivo es maximizar el número de aciertos que representan la función objetivo del problema.

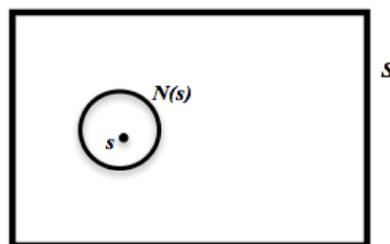


Figura 3-2. Representación del espacio de soluciones de una estructura de vecindad.

El procedimiento comienza de un punto s y el conjunto de soluciones $N(s)$, es elegido a través de una perturbación s' dada por un procedimiento estocástico,

esto es, si $f(s') \geq f(s)$, s' es remplazada por la solución s y pasa a ser la solución actual del problema, el algoritmo 3-2 se muestra el algoritmo general de búsqueda de vecindad.

Algoritmo 3-2. Algoritmo general de búsqueda por vecindad.

```

Generar solución inicial S
Hacer
    S' = solución inicial perturbada
    Si  $f(s') \geq f(s)$  entonces
        S' = mejor solución
        S ← S'
    Fin-si
Mientras la solución continúe mejorando
  
```

Las técnicas de búsqueda por vecindad se implementan para mejorar una solución maximizando el costo de la función objetivo, recordemos que para este trabajo de investigación se requiere de maximizar el número de aciertos de la predicción de signo de los coeficientes wavelet.

Para elegir la estructura que encuentra mejores soluciones al problema definimos un movimiento que permita alcanzar una solución s' por medio de una perturbación sobre la solución s , siempre y cuando se cumpla el criterio de selección. Este proceso será de forma iterativa hasta alcanzar el número de iteraciones determinadas en el algoritmo, si la solución encontrada no mejora, significa que hemos encontrado el óptimo local.

El tamaño de la vecindad, esta determinado por la cadena de Markov, para el caso de tres vecinos el tamaño de la vecindad será de 3^3 y 3^5 para predicción de signo con cinco coeficientes vecinos. Los elementos son seleccionados de forma aleatoria.

El proceso de perturbación aplicado a una estructura de vecindad simple es realizado sobre una solución s inicializada de forma aleatoria con valores de cero para signo positivo y uno para valores de signo negativo, una vez obtenida la solución s' , seleccionamos un elemento de forma aleatoria e intercambiamos el valor de dicho elemento, si el valor es cero se cambia por el valor uno y viceversa (ver figura 3-3).

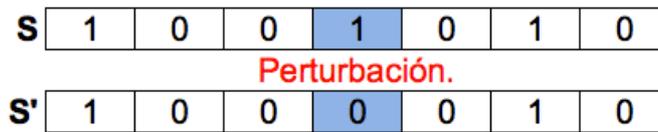


Figura 3-3. Estructura de vecindad sencilla.

Una estructura de vecindad con pares aleatorios, genera una solución inicial s factible, donde a partir de s se seleccionan de forma aleatoria dos elementos que intercambian su valor para formar la solución s' (ver figura 3-4).



Figura 3-4. Estructura de vecindad pares aleatorios.

Una estructura de vecindad con triples aleatorios, genera una solución inicial s factible, donde a partir de s se seleccionan de forma aleatoria tres elementos que intercambian su valor de la siguiente forma, el elemento uno toma el valor del elemento tres y el elemento dos toma el valor del elemento uno, utilizamos este criterio de forma arbitraria para formar la solución s' (ver figura 3-5).

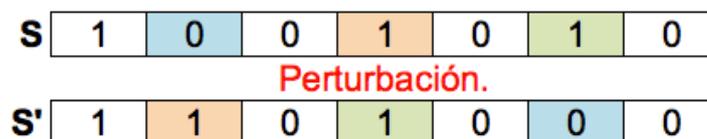


Figura 3-5. Estructura de vecindad triples aleatorios.

Una estructura de vecindad con cuádruples aleatorios, genera una solución inicial s factible, donde a partir de s se seleccionan de forma aleatoria cuatro elementos que intercambian su valor de la siguiente forma, el elemento uno toma el valor del elemento tres y viceversa y el elemento dos toma el valor del elemento cuatro y viceversa, utilizamos este criterio de forma arbitraria para formar la solución s' (ver figura 3-6).

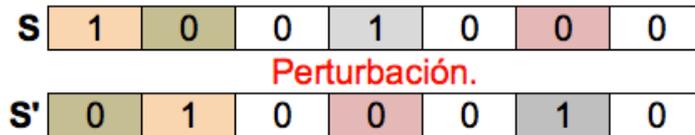


Figura 3-6. Estructura de vecindad cuádruples aleatorios.

Una estructura híbrida, esta compuesta por las cuatro estructuras mencionadas anteriormente y selecciona de forma aleatoria cuál de ellas será utilizada para realizar la perturbación sobre la solución inicial s y así obtener la solución s' que nos permitirá evaluar el costo de la función objetivo.

Tanto la estructura de vecindad simple como la híbrida son iguales en eficacia, pero la estructura sencilla es mejor en eficiencia. En la figura 4-1 y en la tabla 4-3 se puede observar el comportamiento de las diferentes estructuras de vecindad que nos demuestran que tanto la estructura sencilla como la híbrida alcanzan el óptimo local con una desviación estándar de cero, por este comportamiento y la complejidad de ambas estructuras, en el algoritmo de Recocido Simulado la estructura sencilla se utilizó para la predicción del signo de los coeficientes wavelet.

3.2. Generación de una primera solución

El signo de los coeficientes puede ser positivo, negativo o no significativo, pero los no significativos no vale la pena evaluarlos por lo que solamente se utilizan los signos positivo y negativo.

Una solución inicial es generada de forma aleatoria ya que no existen benchmarks para este problema, donde solamente tomamos valores que representan con el número cero los signos que son positivos y con el número uno los signos que son negativos como se puede ver en la figura 3-7.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	Índices
1	1	1	0	0	1	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	1	1	1	1	0	1	Signo
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	Ocurrencias
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	Aciertos

Figura 3-7. Vector predictor solución inicial para tres coeficientes vecinos.

Para determinar un patrón de signos vecinos (NSP) hacemos las posibles combinaciones entre los diferentes signos de los diferentes vecinos como se muestra en la figura 3-8.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	Índice
*	+	-	*	+	-	*	+	-	*	+	-	*	+	-	*	+	-	*	+	-	*	+	-	*	+	-	V ₃
*	*	*	+	+	+	-	-	-	*	*	*	+	+	+	-	-	-	*	*	*	+	+	+	-	-	-	V ₂
*	*	*	*	*	*	*	*	*	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	V ₁

Figura 3-8. Combinación del patrón de signos de los vecinos.

Una vez definido el NSP nos apoyamos de la tabla 3-2 para representar los signos por números para comenzar a evaluar el coeficiente actual según el contexto de la subbanda a analizar y el nivel de descomposición wavelet.

Tabla 3-2. Tabla de símbolos de los signos.

0	Positivo
1	Negativo
2	Nulo

Describiremos como es que se lleva a cabo el proceso de evaluación de la solución, para el ejemplo utilizaremos el contexto de la subbanda HL, para el cual, los coeficientes vecinos que mejor predicción tienen son N, W y NN. Tomamos el coeficiente actual a evaluar, representado por el número en color rojo en la figura 3-9 e identificamos a sus vecinos, V1 tendrá el valor del signo del vecino W, V2 tendrá el signo del vecino N y V3 el del signo NN. Una vez identificados los signos de los coeficientes vecinos, buscamos la combinación asociada en el vector de NSP de la figura 3-8, por lo tanto, V1=0, V2=1 y V3=2, recordemos que en la tabla 3-4 esta la equivalencia entre números y su respectivo símbolo e identificamos el índice en el vector de la figura 3-8 que contiene dicha combinación, el cual es el número 15 en el vector predictor de la solución identificamos el signo que corresponde al índice 15 en el vector predictor de la figura 3-7, para este caso equivale a uno, y comparamos si el coeficiente actual a evaluar es igual al valor de vector predictor, aumentamos el número de aciertos en uno puesto que la equivalencia se cumple e incrementamos el número de ocurrencias en uno ya que funciona como un

contador del número de veces que ocurre un evento, en caso de no ser equivalente el número de aciertos no incrementa pero el número de ocurrencias sí lo hace.

Cabe recordar que el número de aciertos es el costo de la función objetivo del problema.

LL	HL														
LH	HH	HL	2	HL	HL	HL	HL								
LH	LH	HH	HH	HL	1	HL	HL	HL	HL						
LH	LH	HH	HH	HL	HL	HL	HL	HL	HL	0	1	HL	HL	HL	HL
LH	LH	LH	LH	HH	HH	HH	HH	HL							
LH	LH	LH	LH	HH	HH	HH	HH	HL							
LH	LH	LH	LH	HH	HH	HH	HH	HL							
LH	LH	LH	LH	HH	HH	HH	HH	HL							
LH	HH														
LH	HH														
LH	HH														
LH	HH														
LH	HH														
LH	HH														
LH	HH														

Figura 3-9. Ejemplo para evaluar el coeficiente actual en la subbanda HL₄.

El procedimiento será el mismo para el vector predictor de cinco coeficientes vecinos y se toma como su vecindad 3⁵ elementos.

El algoritmo de Recocido Simulado para la predicción de signo de los coeficientes wavelet utiliza una estructura de vecindad sencilla. Anteriormente mencionamos en que consisten las dos modificaciones que se realizaron a los criterios de evaluación del Algoritmo de Recocido Simulado. Inicializamos los parámetros, T₀ representa la temperatura inicial, T_f representa la temperatura final y S representa la configuración de la solución inicial.

Mientras la temperatura inicial sea menor o igual que la temperatura final, generamos un estado S' por medio de una perturbación de S, evaluamos la función objetivo de S' y si dicha función objetivo es mayor o igual que la función objetivo de S, el estado se acepta como el estado actual, en caso contrario, se acepta o rechaza de acuerdo al criterio de aceptación de la función de probabilidad de Boltzmann (ver tabla 3-3).

Algoritmo 3-3. Algoritmo Recocido Simulado para la predicción de los signos de los coeficientes wavelet.

```

Inicializar ( $T_0, S, T_f, \beta, L$ )
Repetir
  Repetir
     $S' = S$ 
    Perturbar( $S'$ )
    EvaluaSignoVecinosMatriz(subbanda, imagen,  $S'$ , nivelW)
    Si  $f(S') \geq f(S)$  entonces
       $S = S'$ 
    Fin-si
    Sino
      Probabilidad de aceptación =  $e^{(f(S') - f(S) / T_0)}$ 
       $\alpha =$  Aleatorio de  $[0,1)$ 
      Si  $\alpha <$  Probabilidad de aceptación entonces
         $S = S'$ 
      Fin si.
  Hasta alcanzar el equilibrio ( $L$ )
   $T_0$  decrementa en base a un coeficiente  $\beta$ 
Hasta alcanzar el equilibrio térmico

```

Para el criterio de aceptación generamos una variable α con la siguiente restricción $0 \geq \alpha < 1$, comparamos en la función de probabilidad de Boltzmann, si $\alpha <$ *probabilidad de aceptación*, aceptamos el estado S' como una solución posible del problema hasta alcanzar el equilibrio con un decremento en base al parámetro β , esto es, si la temperatura T_0 es muy grande, la probabilidad de aceptación será mayor, en caso contrario, la probabilidad de aceptación será muy cercana a cero y estaremos alcanzando el equilibrio.

Tabla 3-3. Criterio de aceptación modificado para la función de Boltzmann.

$$\text{Probabilidad de aceptación} = e^{(f(S') - f(S) / T_0)}$$

El vector resultante de la ejecución del algoritmo del algoritmo de Recocido Simulado es almacenado en un fichero de texto plano en el que indicamos la subbanda analizada y cual es la mejor predicción encontrada así como información relativa al número de ocurrencias totales, números de ocurrencias por combinación, número de aciertos y fallos por combinación, etcétera.

3.3. Metodología de Sintonización

Para llevar a cabo una adecuada sintonización de los parámetros de control de nuestro algoritmo, es necesario llevar a cabo un análisis de sensibilidad.

El análisis de sensibilidad es una evaluación del comportamiento de las variables críticas de un problema (parámetros de control), con la finalidad de establecer un rango numérico, dentro del cuál la solución obtenida por el algoritmo sigue siendo buena, además que permite conocer que tan sensible es el algoritmo a ciertos cambios en los valores de ciertas variables propias del problema.

El análisis de sensibilidad tiene su origen en la programación lineal (PL) debido a que facilita la toma de decisiones. Los cambios llevados a cabo durante dicha evaluación pueden ser en el entorno general del problema, en la empresa o bien en los datos característicos del problema.

El objetivo primordial del análisis de sensibilidad, es encontrar la adecuada sintonización de los parámetros de control del algoritmo, de modo que este tenga una mejora en cuanto a la eficiencia y eficacia. Los valores más estudiados en este análisis de forma general, son los coeficientes de la función objetivo y los términos independientes de las restricciones.

Por lo tanto, un objetivo fundamental en el análisis de sensibilidad es identificar los parámetros sensibles (es decir, los parámetros cuyos valores no pueden cambiar sin que cambie la solución óptima) [Hillier y Lieberman, 2010].

El parámetro de control inicial se toma de forma arbitraria, comenzaremos con 1000 y será dividido entre la cantidad mínima de pruebas que es 30, para obtener el valor de decremento o incremento [Cruz, 2005]. Para el algoritmo de Recocido Simulado éste parámetro no debe ser muy grande de tal forma que el algoritmo no tarde mucho o que éste tarde muy poco. Para la longitud de la cadena de Markov con un rango numérico de uno a tres veces el tamaño de la

estructura de vecindad, no existen un solo valor medio. Para el resto de los parámetros que cuentan con un sólo valor medio del cual partimos con 15 incrementos siendo el límite superior y 15 decrementos siendo el límite inferior, tendremos un valor arriba y por debajo de la solución inicial de la cual partiremos, por ejemplo, el coeficiente de control β cuenta con un valor promedio obtenido de la literatura, dicho valor es 0.98 y partiendo de éste valor se realizan pruebas con 15 incrementos y 15 decremento. El valor del parámetro de control final o enfriamiento cuenta con un valor promedio de 0.001, del cual realizamos el mismo número de incremento y decrementos que para el parámetro anterior.

Hemos tomado la explicación de la metodología de sintonización de las tesis de [Cruz, 2005] y de [Martínez, 2010] para encontrar una adecuada proporción en los valores correspondiente a los parámetros de control, tomando en cuenta tanto el problema como el método de solución implementado.

a) Selección de los parámetros de control.

Para determinar los parámetros de control del algoritmo, es necesario llevar a cabo una revisión de publicaciones que se relacionen con el algoritmo implementado, de esta forma es posible analizar los parámetros tomados en cuenta en investigaciones anteriores [Martínez, 2010].

Otra forma de identificar los parámetros de control, es identificar los parámetros críticos que influyen de cierta manera en la calidad de la solución.

b) Establecer Rangos de Evaluación.

Una vez que hemos identificado los parámetros de control, es necesario establecer los rangos que van a ser utilizados para el análisis de sensibilidad a cada uno de los parámetros. En caso de existir parámetros probados en la literatura para el mismo problema, especificar un rango de acción para cada parámetro de control será más sencillo. El tener una adecuada sintonización nos permite identificar la proporción adecuada entre los parámetros de control dentro del cual el algoritmo obtiene buenas soluciones.

c) Pruebas a Rangos de Evaluación.

Una vez establecidos los rangos se calcula una serie de muestras, de acuerdo al tamaño del rango, que permita evaluar el comportamiento del algoritmo cuando los parámetros de control toman valores distintos.

Una vez obtenido el conjunto de muestras, se procede a realizar las pruebas experimentales, para lo cuál se recomienda realizar conjuntos mínimo de 30 pruebas para cada una de las muestras. Para llevar a cabo una adecuada sintonización de los parámetros de control, es necesario realizar un barrido de los valores correspondientes a una de las variables, manteniendo fijos los demás, hasta identificar el valor que mejore la calidad de las soluciones. Una vez obtenido el mejor valor para esta variable, se fija dicho valor y se comienza con la variación de otro parámetro, llevando a cabo el mismo proceso hasta obtener el conjunto de valores que permitan al algoritmo mejorar la eficiencia y eficacia. Una vez obtenido los mejores valores para cada parámetro, realizamos muestras para un rango más pequeño, tomando como punto medio, el valor fijado, de modo que se vuelva a realizar el proceso de sintonización hasta fijar nuevamente los valores.

d) Sintonización de Parámetros.

Los valores obtenidos al final de cada una de las muestras al término de la evaluación, serán considerados como los valores de sintonización de los parámetros de control del algoritmo. Estos serán aquellos valores que influyan de manera positiva en la calidad de la solución, lo que permite tener un mejor desempeño del algoritmo.

3.4. Análisis de Complejidad del Algoritmo Recocido Simulado

Una vez definidos los parámetros de control que hacen que el algoritmo funcione correctamente, es necesario realizar un estudio que permita conocer su comportamiento y así poder medir su rendimiento, centrándose principalmente en su simplicidad y el uso eficiente de los recursos.

La complejidad computacional de un algoritmo se puede clasificar de acuerdo a la dificultad de resolverlo en función de los siguientes parámetros:

Espacio. Cantidad de memoria requerida para almacenamiento de los datos durante la ejecución del algoritmo.

Tiempo. Duración de la ejecución del algoritmo.

Ambos parámetros representan el costo requerido por el algoritmo según el tipo de problema que se está tratando, para encontrar una solución. El tiempo de ejecución de un algoritmo o complejidad temporal $T(n)$, donde n es el tamaño de la entrada, está en función de los parámetros: Datos de entrada, velocidad del procesador y complejidad del algoritmo. La complejidad temporal representa el número de instrucciones simples (asignaciones, comparaciones, operaciones aritméticas, etcétera) que serán ejecutadas por el algoritmo. Por lo regular se considera la complejidad del algoritmo en el peor de los casos, aunque también es importante conocer la complejidad en el mejor y el caso promedio.

Para clasificar si un algoritmo es considerado como bueno o malo nos basamos en las siguientes convenciones:

1. Todos los algoritmos, desde constantes hasta polinomiales, son polinomiales y son clasificados como buenos algoritmos.

2. Todos los algoritmos exponenciales y factoriales, son exponenciales y son clasificados cómo malos algoritmos.

La complejidad del algoritmo Recocido Simulado se calcula en una primera etapa, un análisis de las instrucciones requeridas por el algoritmo para encontrar una solución al problema tratado, de esta forma obtenemos la función de complejidad temporal del algoritmo.

Para el caso del Recocido Simulado, m representa el número de signos vecinos y n es la cantidad de signos generados en el vector con respecto al patrón de signos vecinos (NSP). Por lo tanto la complejidad para el algoritmo de Recocido simulado con una estructura sencilla para el peor de los casos tiene una complejidad (ver apéndice B) de:

$$O((n^2+n)(3^m) \ln 2^n)$$

Donde n es el número de elementos que componen el vector NSP, m equivale al número de coeficientes vecinos que utilizamos para la predicción de signos y 2^n es el espacio de soluciones.

Capítulo 4

Resultados Experimentales

En este capítulo se presentan los resultados obtenidos en las pruebas experimentales realizadas al algoritmo de Recocido Simulado para la predicción de signo de los coeficientes wavelet para las imágenes de prueba Lena y Bárbara. Estos resultados son comparados con un algoritmo genético reportado en [García y López, 2011] aplicado a la predicción de signo para tres vecinos para las imágenes de prueba Lena y Bárbara [<http://www.kodak.com/digitalImaging/samples/classic.shtml>].

4.1. Descripción del equipo utilizado

Para realizar las pruebas experimentales correspondientes al algoritmo de Recocido Simulado y el algoritmo genético, se utilizaron los equipos de cómputo con las características mostradas en la tabla 4-1.

Tabla 4-1. Equipos de cómputo utilizados.

Metaheurística	Descripción de equipo utilizado		
	Procesador	Memoria	S. O.
Algoritmo Genético	Intel Core 2 CPU - 1.8 GHz	5 Gb	Windows 7 64 bits
Algoritmo de Recocido Simulado	Intel Celeron - 1.6 GHz	1 Gb	Windows XP Home Ed.

Metaheurística	Tiempo de ejecución (seg)	
	Lena	Bárbara
	HL	LH
Algoritmo Genético	3.743	9.39
Algoritmo de Recocido Simulado	2.1	2.2

El codificador LTW se puede descargar de la URL <http://www.grc.upv.es/Software/lw.html>

El equipo que se utilizó para ejecutar el algoritmo de Recocido Simulado fue debido a que el algoritmo genético utilizó un procesador a 1.8 GHz y para

poder tener una aproximación sobre el tiempo de ejecución utilizamos un procesador inferior.

4.2. Análisis de Sensibilidad

El método para realizar el análisis de sensibilidad para los parámetros del algoritmo de Recocido Simulado para la predicción de signo de los coeficientes wavelet a través de la metodología explicada en el capítulo tres.

a) Selección de los parámetros de control.

De acuerdo con el análisis realizado al problema tratado, las variables utilizadas para llevar a cabo el análisis de sensibilidad del algoritmo de Recocido Simulado son las siguientes:

Parámetro de control valor inicial (T_0)

Parámetro de control valor final o criterio de paro (T_f)

Coefficiente de control (β)

Longitud de la Cadena de Markov (LCM)

b) Establecer Rangos de Evaluación.

Una vez identificados los parámetros de control, establecemos los rangos para realizar el análisis de sensibilidad para cada una de las variables identificadas en el paso uno de la metodología.

En el análisis realizado a cada una de las variables se pudo observar que si el coeficiente de control tiene un valor cercano a uno, es decir 0.999, el parámetro de control T_0 disminuye lentamente, en caso contrario, si es más próximo a cero, es decir 0.965, el parámetro de control T_0 disminuye rápidamente alcanzando un enfriamiento más pronto.

Tabla 4-2. Rangos utilizados en los parámetros de control para realizar el análisis de sensibilidad.

a) Tres vecinos

Parámetro de Control	Límite Superior	Límite Inferior
T_0	1000	5
β	0.994	0.965
T_f	2	0.0001
LCM	27	

b) Cinco vecinos.

Parámetro de Control	Límite Superior	Límite Inferior
T_0	1000	5
β	0.994	0.965
T_f	2	0.0001
LCM	243	

c) Pruebas a Rangos de Evaluación.

La longitud de la cadena de Markov, está definida por el tamaño de la vecindad la cuál será utilizada por la estructura de vecindad simple en el algoritmo de Recocido Simulado para la predicción de signo para tres y cinco vecinos respectivamente.

Para cada uno de los rangos mostrados en la tabla 4-2, se calcularon un mínimo de 30 pruebas. Una vez encontrado el valor para el parámetro de control inicial T_0 , se fija para comenzar la variación de los siguientes parámetros de control.

El segundo parámetro a sintonizar es el coeficiente de control β partiendo de un valor de 0.98 [Cruz, 2005]. Una vez realizadas 30 pruebas, seleccionamos el mejor valor del parámetro que mejores resultados obtuvo, fijamos el parámetro para encontrar el siguiente.

Para encontrar el parámetro de control final o enfriamiento, partimos de un valor medio 0.001 realizando los debidos incrementos y decrementos mencionados en el capítulo anterior. Una vez obtenido el valor que mejor desempeño mostró en el algoritmo, procedemos a realizar las últimas 30 pruebas con los cuatro parámetros de control que mejores resultados obtuvieron (ver tabla 4-3).

Tabla 4-3. Valores de los parámetros de control, fijados de acuerdo al análisis de sensibilidad.

Parámetro de Control	Tres vecinos	Cinco vecinos
T_0	5	5
β	0.965	0.965
T_f	2	2
LCM	27	243

d) Sintonización de Parámetros.

Los valores obtenidos para cada uno de los parámetros de control evaluados, al término de las pruebas experimentales con las variables mostradas en la tabla 4-3, dan como resultado la sintonización de los parámetros de control.

La sintonización de los parámetros de control se lleva a cabo con la finalidad de identificar aquellos valores correspondientes a las variables de control, que influyen de manera positiva en la calidad de la solución, lo que permite tener una mejora en el desempeño del algoritmo.

4.3. Resultados Experimentales

Para conocer el comportamiento de un algoritmo, es necesario recurrir al cálculo de parámetros estadísticos como son la media y desviación estándar. La media permite obtener el promedio de las soluciones encontradas para cada instancia. La desviación estándar permite conocer que tan dispersas están las soluciones con respecto a la media aritmética. De acuerdo a los datos obtenidos se puede determinar si un algoritmo es bueno o no para cierto tipo de problema.

En base a las pruebas experimentales que se realizaron sobre las diferentes estructuras de vecindad con los valores sintonizados, como se muestra en la tabla 4-4, para las imágenes de Lena y Bárbara podemos visualizar que las

estructuras de vecindad que mejores resultados obtienen en base a la desviación estándar son la estructura simple y la estructura híbrida. Debido a la complejidad de ambas estructuras, la sencilla tiene una menor complejidad computacional que la híbrida, por lo que se eligió la estructura de vecindad sencilla para realizar la perturbación sobre la solución en el algoritmo, con ello tenemos una mejor eficacia.

Se pudo observar que las estructuras de pares, triples y cuádruples aleatorios para el peor de los casos no alcanzaban el costo más alto, siendo la sencilla e híbrida que alcanzaban el mayor costo de la solución. Por lo que el promedio para el mejor de los casos era el costo alcanzado como Mejor y por lo tanto la desviación estándar es de cero, indicando que parece ser ese costo de la solución el óptimo local. Sucede lo mismo para Lena y Bárbara.

Tabla 4-4. Resultados experimentales de las estructuras de vecindad para las imágenes de prueba a) Lena y b) Bárbara.

a) Lena

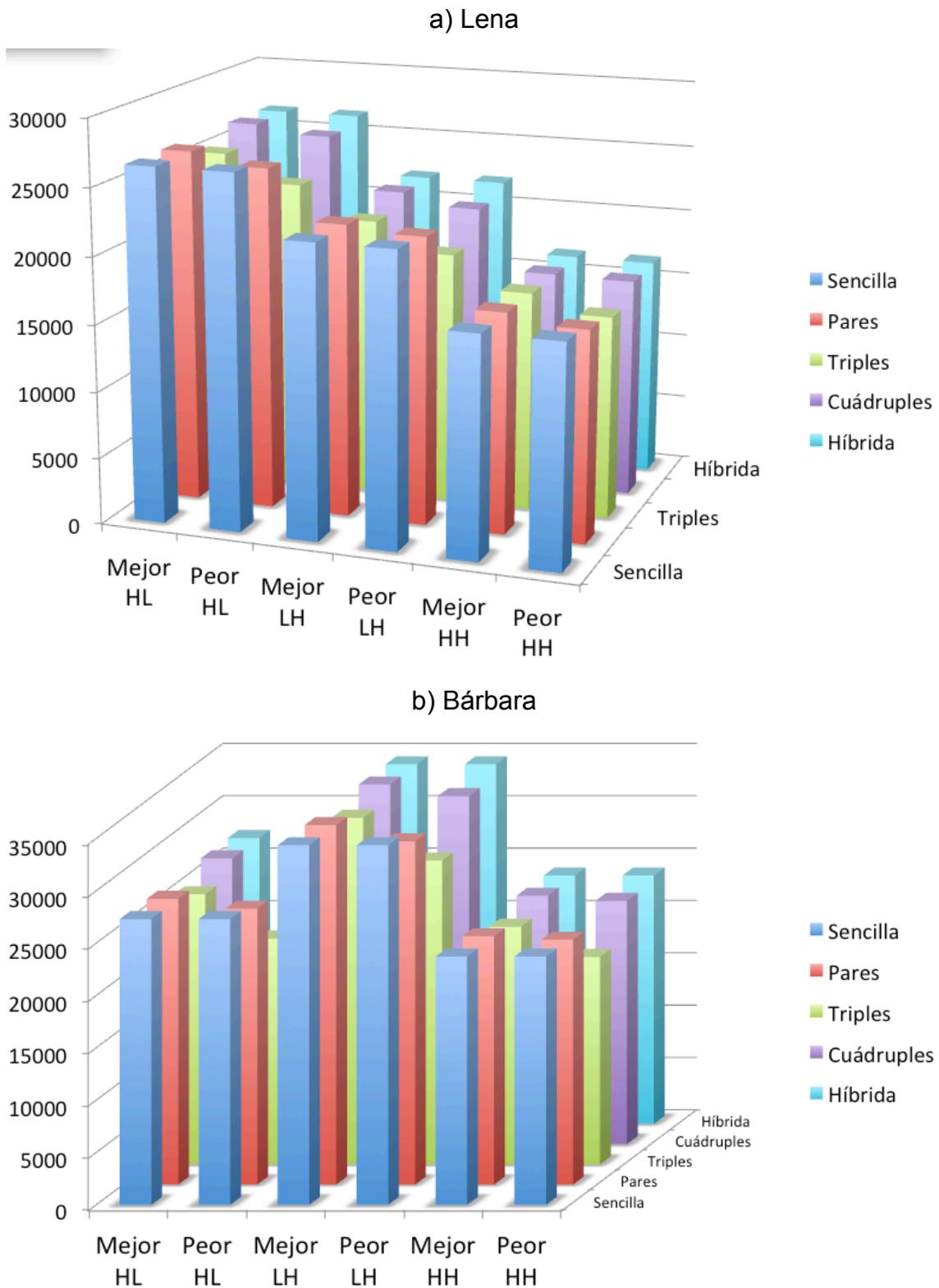
Estructura	Lena											
	HL				LH				HH			
	Mejor	Peor	Promedio Mejor	Dev. Estándar	Mejor	Peor	Promedio Mejor	Dev. Estándar	Mejor	Peor	Promedio Mejor	Dev. Estándar
Sencilla	26474	26474	26474	0	21953	21953	21953	0	16548	16548	16548	0
Pares A.	26474	25634	25314.56	203.07	21953	21539	21835.83	127.2	16548	15846	16502.6	128.14
Triples A.	25229	23244	24343.7	614.29	20928	18868	20082.83	586.36	16497	15241	15793.33	313.4
Cuádruples A.	26474	25869	26342.56	157.48	21953	21085	21731.46	271.41	16548	16482	16527.16	19.53
Híbrida A.	26474	26474	26474	0	21953	21953	21953	0	16548	16548	16548	0

b) Bárbara

Estructura	Bárbara											
	HL				LH				HH			
	Mejor	Peor	Promedio Mejor	Dev. Estándar	Mejor	Peor	Promedio Mejor	Dev. Estándar	Mejor	Peor	Promedio Mejor	Dev. Estándar
Sencilla	27310	27310	27310	0	34351	34351	34351	0	23728	23728	23728	0
Pares A.	27303	26368	27106.43	250.48	34351	32793	34102.9	332.18	23728	23394	23631.5	112.23
Triples A.	25802	21577	23811.57	905.59	33113	29025	31107.63	1015.29	22712	19815	21906.2	644.87
Cuádruples A.	27310	22799	26985.26	829.1	34351	33243	34059.33	413.45	23728	23241	23606.63	151.25
Híbrida A.	27310	27310	27310	0	34351	34351	34351	0	23728	23728	23728	0

En la gráfica de la figura 4-1 se puede apreciar los resultados mostrados en la tabla 4-4 inciso a y b.

Figura 4-1. Gráfica comparativa de estructuras de vecindad para las imágenes de prueba a) Lena y b) Bárbara.



4.4. Análisis de Eficacia y Eficiencia del Algoritmo

El análisis de eficacia y eficiencia se lleva a cabo para evaluar tanto la calidad de las soluciones obtenidas por un algoritmo, como los recursos y tiempos necesarios para su ejecución. Para realizar dicho análisis se comparó la calidad de la solución obtenida por el algoritmo genético reportado en la literatura y el algoritmo de Recocido Simulado propuesto para el mismo problema.

La calidad de la solución obtenida por el algoritmo de Recocido Simulado para la predicción de signo para tres vecinos es la misma que encontró el algoritmo genético. Ambos algoritmos son igual de eficaces, pero el algoritmo de Recocido Simulado demuestra ser más eficiente.

La predicción de signo para cinco vecinos que realiza el algoritmo de Recocido Simulado en la imagen Bárbara para las subbandas HL y LH muestra una mejora significativa, alrededor de 1000 aciertos más que el costo de la solución encontrada con tres vecinos, siendo este más eficaz. Para Lena no son tan significativos ya que se trata de una imagen con poca textura, pocas altas frecuencias, lo que hace que haya menos coeficientes significativos y por tanto vecinos con signo para poder predecir.

4.5. Análisis de los Resultados

Para comprobar que el algoritmo de Recocido Simulado de acuerdo a la aportación de la tesis mejora la calidad de la solución para cinco vecinos, detallamos de forma teórica la entropía y el ahorro en bits en el apéndice E.

A continuación se detallan en dos puntos los análisis de evaluación del algoritmo de Recocido Simulado para las imágenes de Lena y Bárbara para tres y cinco vecinos.

4.5.1. Primer Análisis: Imagen Lena

El primer estudio del comportamiento del RS se realizó sobre una imagen estándar de prueba ampliamente utilizada para todo tipo de algoritmos de procesamiento de imágenes (compresión y eliminación de ruido) y relacionado con muchas publicaciones científicas, figura 4-2, se trata de una imagen con un alto nivel de detalle en tonos grises a 8 bits por pixel, cuyo tamaño es de 512 x 512 píxeles y con muy poca textura.



Figura 4-2. Imagen de Lena.

Se analizó cada una de las tres subbandas (HL, LH y HH), en un total de 87,360 coeficientes wavelet a analizar, del total de 262,144 coeficientes wavelet que tiene la imagen.

Uno de los principales objetivos a conseguir del algoritmo es la correcta predicción del signo por cada coeficiente analizado.

En la tabla 4-5 se muestra para cada tipo de subbanda HL, LH y HH, los resultados obtenidos en la predicción de signo por NSP utilizando tres coeficientes vecinos. Los datos se han obtenido aplicando el RS con los siguientes parámetros: $T_0=5$, $T_f=2$, cadena de Markov= 3^3 , la configuración de la solución inicial es de forma aleatoria y mostramos el vector predictor para cada subbanda.

Tabla 4-5. Predicción de signo para la imagen de Lena.

Combinación	Vecino 1	Vecino 2	Vecino 3	Subbanda HL	Subbanda LH	Subbanda HH
				Predicción: N,NN,W	Predicción: W,WW,N	Predicción N,W,NW
0	*	*	*	+	+	+
1	*	*	+	-	-	+
2	*	*	-	+	+	-
3	*	+	*	+	+	-
4	*	+	+	-	-	-
5	*	+	-	+	+	-
6	*	-	*	-	-	+
7	*	-	+	-	-	+
8	*	-	-	+	+	+
9	+	*	*	+	-	-
10	+	*	+	-	-	-
11	+	*	-	+	+	-
12	+	+	*	+	+	-
13	+	+	+	-	-	-
14	+	+	-	+	+	-
15	+	-	*	-	-	-
16	+	-	+	-	-	-
17	+	-	-	+	+	+
18	-	*	*	-	+	+
19	-	*	+	-	-	+
20	-	*	-	+	+	+
21	-	+	*	-	+	+
22	-	+	+	-	-	+
23	-	+	-	+	+	+
24	-	-	*	-	-	+
25	-	-	+	-	-	+
26	-	-	-	+	+	+

Para poder tener una medida comparativa de lo buena o mala que puede llegar a ser la predicción, desde un punto de vista probabilístico, hacemos uso de la medida de la entropía.

En términos generales, podemos definir la entropía de un mensaje (datos del mensaje, suceso, predicción, etcétera), como el límite máximo al que se puede comprimir un mensaje, usando un enfoque símbolo a símbolo sin ninguna pérdida de información [García, 2010], el límite de compresión (en bits) es igual a la entropía multiplicada por la longitud del mensaje.

Algunas técnicas como el popular algoritmo para compresión de datos Ziv-Lempel [Ziv y Lempel, 1977] o una de sus variantes el LZW [Welch, 1984], son algoritmos de compresión sin pérdida, no usan probabilidades de los símbolos aislados, sino que usan las probabilidades conjuntas de pequeñas secuencias de símbolos para codificar el mensaje, por lo que pueden lograr un nivel de

compresión mayor. Hemos utilizado la técnica LZW sobre un codificador LTW agrupando los aciertos frente a las ocurrencias totales en una subbanda dada para el análisis.

En la tabla 4-6 se muestran los resultados obtenidos para la imagen de Lena con la configuración antes mencionada para tres vecinos. Como se puede observar, el mayor ahorro de compresión se obtiene en la subbanda HL con un ahorro de 1626.01 bits, mientras que los peores resultados se obtiene en las subbanda HH, con apenas un ahorro de 283.35 bits.

Los datos necesario para el estudio de la entropía y ahorra se basan en las siguientes formulas:

Probabilidad: Número de aciertos * 100 / Ocurrencias

Entropía: Suma $(-(\text{Probabilidad}/100) * \text{Log}_2(\text{Probabilidad}/100))$

Ratio (bps): Entropía * Ocurrencias

Ahorro (bits): Ocurrencias – Ratio

Ganancia (%): Ahorro * 100 / Ocurrencias

Tabla 4-6. Estudio de la entropía y ahorro en bps para la imagen Lena tres coeficientes vecinos.

Lena	Subbanda		
	HL	LH	HH
Ocurrencias	43131	36198	29685
Aciertos	26474	21953	16548
Fallos	16657	14245	13137
Probabilidad (%)	61.3804456	60.6469971	55.7453259
Entropía (bits)	0.96230046	0.96703986	0.99045461
Ratio (bps)	41504.9812	35004.9088	29401.6451
Ahorro (bits)	1626.01884	1193.09125	283.354851
Ganancia (%)	3.76995396	3.29601428	0.95453883

En la tabla 4-7 se muestran los resultados obtenidos para la imagen de Lena con la configuración antes mencionada para cinco vecinos. Como se puede observar, el mayor ahorro de compresión se obtiene en la subbanda HL con un ahorro de 1955.09 bits, mientras que los peores resultados se obtiene en las subbanda HH, con apenas un ahorro de 386.27 bits.

Tabla 4-7. Estudio de la entropía y ahorro en bps para la imagen Lena cinco coeficientes vecinos.

Lena	Subbanda		
	HL	LH	HH
Ocurrencias	43131	36198	29685
Aciertos	26943	22137	16833
Fallos	16188	14061	12852
Probabilidad (%)	62.4678306	61.1553124	56.7054068
Entropía (bits)	0.95467076	0.96378995	0.9869874
Ratio (bps)	41175.9046	34887.2684	29298.721
Ahorro (bits)	1955.09536	1310.73157	386.279022
Ganancia (%)	4.53292379	3.62100549	1.30125997

Podemos comprobar obteniendo la diferencia del total de aciertos encontrados para tres y cinco vecinos, que con la nueva vecindad el costo de la solución aumenta 469 aciertos más para la subbanda HL y para la subbanda LH apenas 185 aciertos más, ver tabla 4-8.

Tabla 4-8. Diferencia de aciertos entre tres y cinco coeficientes vecinos para Lena.

	HL	LH	HH
Tres vecinos	26474	21953	16548
Cinco vecinos	26943	22137	16833
Diferencia	469	184	285

La imagen de Lena comprimida/descomprimida con diferentes niveles de bits por pixel (bpp) utilizando el vector predictor de signo obtenido por el algoritmo de Recocido Simulado para tres vecinos dentro del codificador S-LTW[García et al., 2011] se muestran en las siguientes figuras.



Figura 4-3. Imagen comprimida con predicción de signo a 0.0625 bpp.



Figura 4-4. Imagen comprimida con predicción de signo a 0.125 bpp.



Figura 4-5. Imagen comprimida con predicción de signo a 0.25 bpp.



Figura 4-6. Imagen comprimida con predicción de signo a 0.5 bpp.



Figura 4-7. Imagen comprimida con predicción de signo a 1 bpp.



Figura 4-8. Imagen comprimida con predicción de signo a 2 bpp.

4.6.2. Segundo Análisis: Imagen Bárbara

El segundo análisis es aplicado sobre otra imagen de prueba ampliamente utilizada y mostrada en la figura 4-9, se trata de una imagen en tonos grises a 8 bits por pixel, cuyo tamaño es de 512 x 512 pixeles. En este caso, a diferencia de la imagen de Lena mostrada en la figura 4-2, ésta contiene mucha textura.



Figura 4-9. Imagen de Bárbara.

Del mismo modo que ocurre para la imagen de Lena, en este caso también tendremos que analizar un total de 87,360 coeficientes distribuidos en las tres subbandas HL, LH y HH. En la tabla 4-9 se muestran, para cada tipo de subbanda, los resultados obtenidos en la predicción de signo por NSP utilizando tres vecinos. Los datos se obtuvieron aplicando el RS con los mismos parámetros que para la imagen de Lena. A diferencia de la tabla 4-4, se añadió los valores de la probabilidad de acierto y de fallo por NSP, como se puede observar hay NSP en dos de las tres subbandas con una probabilidad de acierto superior al 80% como se puede ver en los elementos 16, 20 y 23 de la subbanda HL y el elemento 25 de la subbanda HH

Tabla 4-9. Predicción de signo para la imagen de Bárbara.

Combinación	Vecino 1	Vecino 2	Vecino 3	Subbanda HL		Subbanda LH		Subbanda HH				
				Predicción: N,NN,W		Predicción: W,WW,N		Predicción N,W,NW				
				Acierto(%)	Fallo(%)	Acierto(%)	Fallo(%)	Acierto(%)	Fallo(%)			
0	*	*	*	+	51,27	48,73	+	50,16	49,84	-	50,19	49,81
1	*	*	+	-	68,37	31,63	-	66,06	33,94	+	51,67	48,33
2	*	*	-	+	72,48	27,52	+	66,58	33,42	-	51,08	48,92
3	*	+	*	+	50,34	49,66	+	56,68	43,32	-	63,69	36,31
4	*	+	+	-	75,87	24,13	-	71,31	28,69	-	61,49	38,51
5	*	+	-	+	64,16	35,84	+	60,05	39,95	-	61,32	38,68
6	*	-	*	+	52,59	47,41	-	52,89	47,11	+	63,43	36,57
7	*	-	+	-	64,20	35,80	-	56,12	43,88	+	66,20	33,80
8	*	-	-	+	78,77	21,23	+	71,84	28,16	+	61,65	38,35
9	+	*	*	-	52,16	47,84	+	56,34	43,66	-	60,29	39,71
10	+	*	+	-	77,90	22,10	-	62,21	37,79	-	63,12	36,88
11	+	*	-	+	68,93	31,07	+	68,64	31,36	-	62,32	37,68
12	+	+	*	+	59,21	40,79	+	60,96	39,04	-	77,11	22,89
13	+	+	+	-	71,37	28,63	-	63,78	36,22	-	59,97	40,03
14	+	+	-	+	74,28	25,72	+	69,33	30,67	-	78,05	21,95
15	+	-	*	-	63,53	36,47	-	56,31	43,69	+	53,81	49,19
16	+	-	+	-	81,55	18,45	-	70,02	29,98	-	63,37	36,63
17	+	-	-	+	55,80	44,20	+	65,21	34,79	+	57,13	42,87
18	-	*	*	+	56,38	43,62	-	56,05	43,95	+	58,41	41,59
19	-	*	+	-	64,00	36,00	-	70,19	29,81	+	65,06	34,94
20	-	*	-	+	80,17	19,83	+	62,29	37,71	+	61,47	38,53
21	-	+	*	+	67,46	32,54	+	55,58	44,42	-	50,33	49,67
22	-	+	+	-	52,61	47,39	-	64,05	35,95	-	54,59	45,41
23	-	+	-	+	82,09	17,91	+	71,83	28,17	+	65,77	34,23
24	-	-	*	-	53,68	46,32	-	65,91	34,09	+	74,61	25,39
25	-	-	+	-	73,92	26,08	-	68,16	31,84	+	80,00	20,00
26	-	-	-	+	73,80	26,20	+	65,20	34,80	+	52,30	47,70

En la tabla 4-10 se presentan los resultados obtenidos en cuanto al nivel de compresión para la imagen Bárbara. Del mismo modo como ocurre para la imagen Lena, se puede observar que el mayor ahorro de compresión se obtiene en la subbanda HL con un ahorro de 3439.72 bits, mientras que los peores resultados se obtienen en la subbanda HH con un ahorro de 1767.01 bits

Tabla 4-10. Estudio de la entropía y ahorro en bps para la imagen Bárbara con tres coeficientes vecinos.

Bárbara	Subbanda		
	HL	LH	HH
Ocurrencias	40808	53391	37876
Acertos	27310	34351	23728
Fallos	13498	19040	14148
Probabilidad (%)	66.9231523	64.3385589	62.6465308
Entropía (bits)	0.9157096	0.93983696	0.95334751
Ratio (bps)	37368.2772	50178.8351	36108.9903
Ahorro (bits)	3439.72282	3212.16486	1767.00972
Ganancia (%)	8.42904042	6.01630399	4.66524903

En la tabla 4-11 se presentan los resultados obtenidos en cuanto al nivel de compresión para la imagen Bárbara con cinco vecinos, se puede observar que el mayor ahorro de compresión se obtiene en la subbanda LH con un ahorro de 4728,08 bits, mientras que los peores resultados se obtienen en la subbanda HH con un ahorro de 2067,94 bits.

Tabla 4-11. Estudio de la entropía y ahorro en bps para la imagen Bárbara con cinco coeficientes vecinos.

Bárbara	Subbanda		
	HL	LH	HH
Ocurrencias	40808	53391	37876
Aciertos	27880	35952	24115
Fallos	17439	12928	13761
Probabilidad (%)	68.3199373	67.3371917	63.668286
Entropía (bits)	0.90086827	0.91144414	0.94540226
Ratio (bps)	36762.6322	48662.9142	35808.056
Ahorro (bits)	4045.3678	4728.08576	2067.94399
Ganancia (%)	9.91317339	8.85558569	5.45977398

Podemos comprobar obteniendo la diferencia del total de aciertos encontrados para tres y cinco vecinos, que con la nueva vecindad el costo de la solución aumenta 1601 aciertos más para la subbanda LH y para la subbanda HH apenas 387 aciertos más, ver tabla 4-11.

Tabla 4-12. Diferencia de aciertos entre tres y cinco coeficientes vecinos para Bárbara.

	HL	LH	HH
Tres vecinos	27310	34351	23728
Cinco vecinos	27880	35952	24115
Diferencia	570	1601	387

La imagen de Bárbara comprimida/descomprimida con diferentes niveles de bits por pixel (bpp) utilizando el vector predictor de signo obtenido por el algoritmo de Recocido Simulado para tres vecinos dentro del codificador S-LTW [García *et al.*, 2011] se muestran en las siguientes figuras.



Figura 4-10. Imagen comprimida con predicción signo a 0.0625 bpp.



Figura 4-11. Imagen comprimida con predicción signo a 0.125 bpp.



Figura 4-12. Imagen comprimida con predicción signo a 0.25 bpp.



Figura 4-13. Imagen comprimida con predicción signo a 0.5 bpp.



Figura 4-14. Imagen comprimida con predicción signo a 1 bpp.



Figura 4-15. Imagen comprimida con predicción signo a 2 bpp.

En el figura 4-16 se muestra una comparativa de los datos obtenidos de la imagen Lena para tres coeficientes vecinos frente a los obtenidos con cinco coeficientes vecinos. En ella podemos observar como para las tres subbandas la ganancia obtenida para el caso de cinco coeficientes vecinos es superior a tres coeficientes vecinos. En cuanto a los valores para la entropía y la probabilidad de acierto, en ambos casos los valores obtenidos son similares.

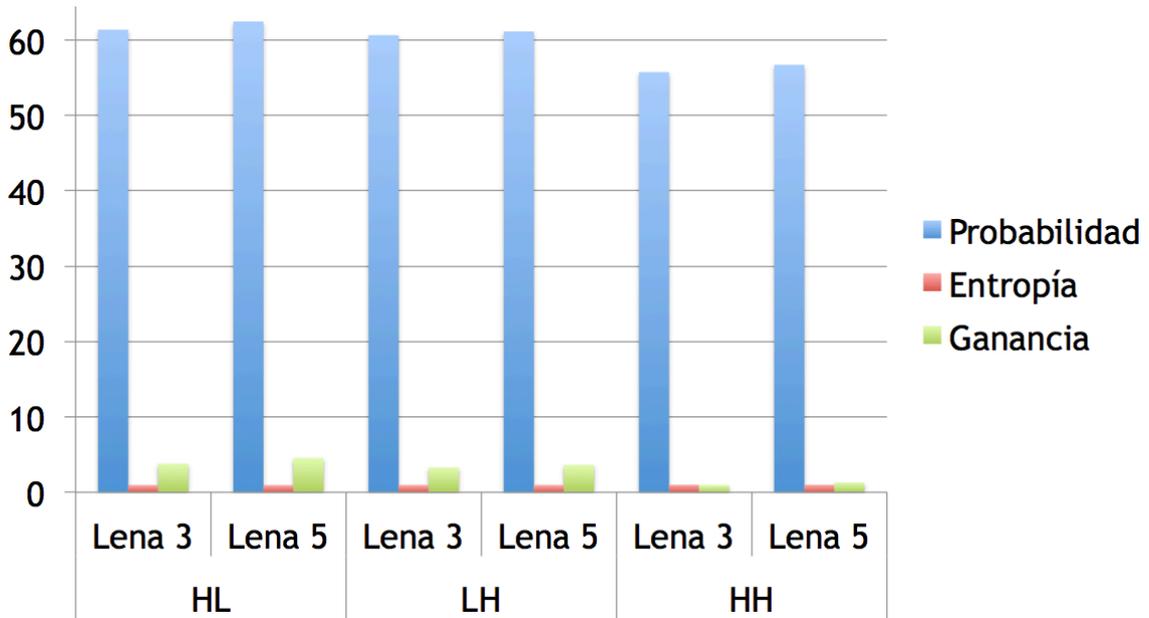


Figura 4-16. Comparativa Lena para tres y cinco coeficientes vecinos.

En el figura 4-17 se muestra una comparativa de los datos obtenidos de la imagen Bárbara para tres y cinco coeficientes vecinos. En ella podemos observar como para las tres subbandas la ganancia obtenida para el caso de cinco coeficientes vecinos es superior a tres coeficientes vecinos. En cuanto a los valores para la entropía y la probabilidad de acierto, en ambos casos los valores obtenidos son similares.

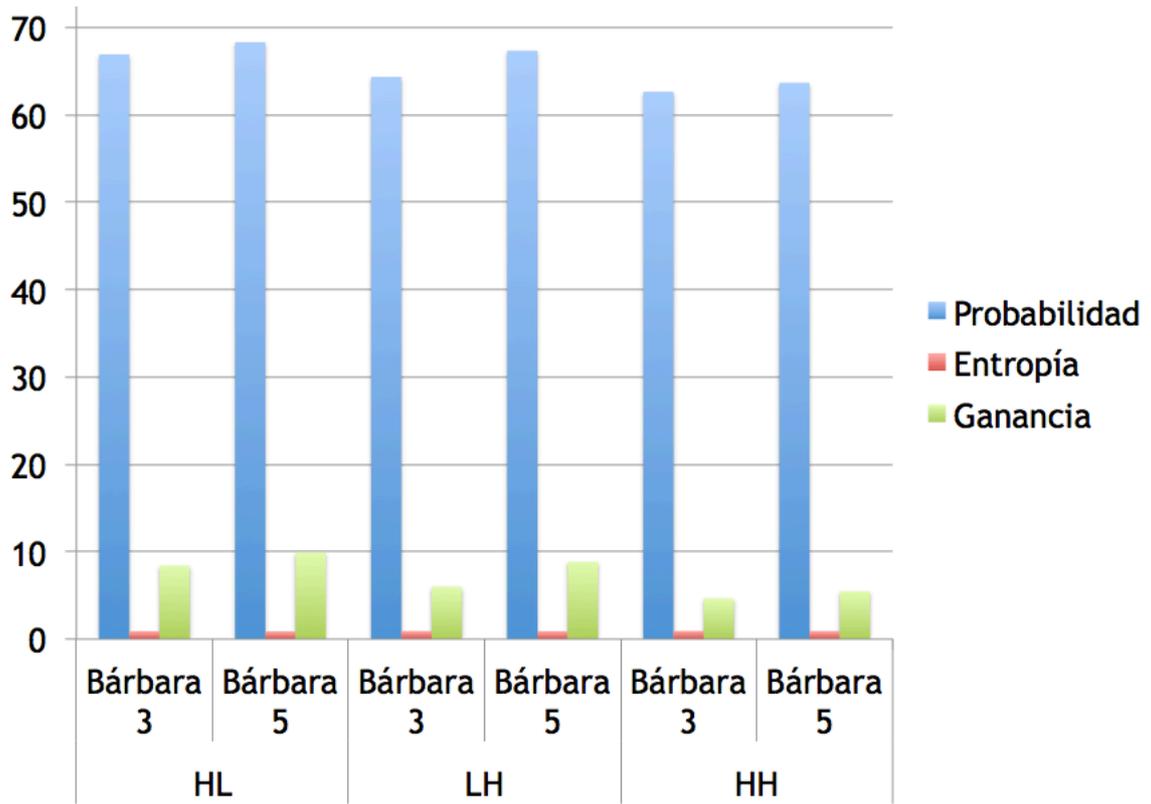


Figura 4-17. Comparativa Bárbara para tres y cinco coeficientes vecinos.

Capítulo 5

Conclusiones y Trabajos Futuros

5.1. Conclusiones

El algoritmo de Recocido Simulado para evaluar 3 coeficientes vecinos es eficaz y demostró ser más eficiente que el algoritmo genético en la predicción de signos de los coeficientes wavelet.

Obtiene una ganancia de compresión arriba del 17% mejorando el rendimiento global del codificador.

El ampliar los coeficientes vecinos a 5 para realizar la predicción de signos, demostró ser más eficaz, encontrando alrededor de 1000 aciertos más sobre el costo de la solución en algunas subbandas, con un tiempo aproximado de 22 segundos.

Esto es debido al aumento del espacio de soluciones.

Por ahora el vector resultante 3^5 no ha podido ser probado dentro del codificador LTW puesto que sólo soporta predicción con un vector 3^3 .

Se comprobó que el algoritmo de Recocido Simulado no es adecuado para este problema, por lo que se recomienda utilizar un algoritmo de búsqueda local iterada.

5.2. Trabajos futuros

Se realizará un estudio exhaustivo tomando como base el número de coeficientes vecinos mayor que cinco para analizar la predicción de signos de los coeficientes wavelet.

Dependiendo el número de vecinos, definir una estructura con diferentes vecinos para formar contextos y encontrar un mejor costo de la solución.

Referencias

[Chrysafis and Ortega, 1997] Chrysafis C., Ortega A. “Efficient context-based entropy coding for lossy wavelet image compression”, Proc. Of 1997 Data Compression Conference, pp. 241-250, Mar. 1997.

[Colom, 2001] Colom Palero R. J. “*Estudio e implementación de la transformada wavelet para la compresión de imágenes y vídeo*”. Tesis doctoral. Universidad Politécnica de Valencia. Valencia 2001.

[Cruz, 2005] Cruz-Chávez, M.A. (2005). Cooperación de Procesos para el Problema de Job Shop Scheduling Aplicando Recocido Simulado, Tesis de Doctorado, Marzo 2005.

[Cruz y Martínez, 2010] Cruz-Chávez M. A., Martínez-Oropeza A., Serna Barquera S. A., “Neighborhood Hybrid Structure for Discrete Optimization Problems”, Electronics, Robotics and Automotive Mechanics Conference, CERMA2010, IEEE-Computer Society, ISBN 978-0-7695-4204-1, pp. 108 – 113, Oct. 2010.

[Deever y Hemami, 2000] Deever A., Hemami S. S., “What’s your sign? Efficient Sign Coding for Embedded Wavelet Image Coding” Data Compression Conference, 2000. Proceedings. DCC 2000 pp. 273 – 282.

[Deever y Hemami, 2003] Deever A. T. and Hemami S. S., “Efficient Sign coding and Estimation of Zero-Quantized Coefficients in Embedded Wavelet Image Codecs”, IEEE Trans. On Image Procs. pp. 420 – 430. Apr. 2003.

[Effelsberg y Steinmetz, 1998] Effelsberg Wolfgang and Steinmetz Ralf, “Videos Compression Techniques”, dpunkt.verlag, ISBN: 3-920993-13-6, 1998.

[Fernández, 2007] Fernández Sarría A. *Estudio de técnicas basadas en la transformada wavelet y optimización de sus parámetros para la clasificación*

por texturas de imágenes digitales. Tesis doctoral. Universidad Politécnica de Valencia. Valencia 2007.

[García, 2010] García Gutiérrez R., “Implementación de un algoritmo Genético para la Búsqueda de una Solución Óptima o Subóptima, Asociada a la Codificación del Signo de los Coeficientes de la Transformada Wavelet 2D”, Trabajo para obtención del Diploma de Estudios Avanzados por la Universidad Miguel Hernández dentro del programa de doctorado Tecnologías Industriales y de Telecomunicación, del Departamento de Ciencia de Materiales, Óptica y Tecnología Electrónica, Oct. 2010

[García *et al.*, 2011] García Ricardo, López Otoniel, Martí Antonio and Malumbres Manuel P., “On the Use of Genetic Algorithms to Improve Wavelet Sign Coding Performance”, Lecture Notes in Computer Science, 2011, Volume 6691/2011, 505-512, DOI: 10.1007/978-3-642-21501-8_63.

[González y Woods, 1996] González Rafael C., Woods Richard E., “Tratamiento digital de imágenes”, Addison-Wesley/Díaz de Santos, ISBN: 0-201-62576-8, 1996.

[Hillier y Lieberman, 2010] Hillier, F. S., Lieberman, G. J. (2010). Introducción a la Investigación de Operaciones, Novena Edición, ISBN: 978-607-15-0308-4.

[Hirst y Selman, 1987] Hirst G. and Selman B., “Parsing as an Energy Minimization Problem”, en Davis, pp. 141 – 154, 1987

[ISO/IEC 15444-1] ISO/IEC 15444-1: “JPEG2000 image coding system”, 2000.

[IEEE Standard 610.4, 1990] IEEE Standard 610.4-1990. IEEE Standard Glossary of Image Processing and Pattern Recognition Terminology.

[Kirkpatrick *et al.*, 1983] Kirkpatrick, S., Gelatt, C.D., Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, Vol. 220, pp 671-680. 1983.

[Laarhoven *et al.*, 1992] Van Laarhoven P. J. M., Aarts, E. H. L., Lenstra, J. K. (1992). Job Shop Scheduling by Simulated Annealing, *Operations Research*, Vol. 40, No. 1, pp. 113-125.

[Lewis y Knowles, 1992] Lewis A. And Knowles G., "Image Compression using the 2-D Wavelet Transform", *IEEE Trans. On Image Proc.* pp. 244 – 250, Apr 1992.

[López, 2010] López Granado Otoniel M., "Fast and Efficient Coding Tools for Digital Image and Video Signals", Tesis Doctoral, Alicante España, Mayo 2010.

[López *et al.*, 2009] Lopez O., Martinez M., Pinol P., Malumbres M.P., Oliver J., "E-LTW: An enhanced LTW encoder with sign coding and precise rate control", 16th IEEE International Conference on Image Processing (ICIP), 2009 ISBN: 978-1-4244-5653-6, pp. 2821 – 2824, Nov. 2009

[LoPresto *et al.*, 1997] LoPresto S. M., Ramchandran K., and Orchard M. T., "Image coding base don mixture modeling of wavelet coefficients and a fast estimation-quantization framework", *Proc. Of 1997 Data Compression Conference*, pp. 221-230. Mar. 1997.

[Martínez, 2010] Martínez, O. A., "Solución al Problema de Máquinas en Paralelo no Relacionadas mediante un Algoritmo de Colonia de Hormigas", Tesis de Maestría, Agosto 2010.

[Mbainabeye y Noureddine, 2007] Mbainabeye Jérôme and Noureddine Ellouze, "Optimal Image Compression Base don sign and Magnitude Coding of Wavelet Coefficients", *World Academy of Science, Engineering and Technology*, 2007.

[Metrópolis *et al.*, 1953] Metrópolis, N., Rosenbluth, A. W., Rosenbluth, M., Teller, A.H., Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *J.Chem.Phys.* Vol.21, pp. 1087-1092, 1953.

[Morton, 1966] Morton G. M., "A computer oriented geodetic data base and a new technique in file sequencing", Tech. rep., LMB Ltd. (1966).

[Oliver y Malumbres, 2006] Oliver J. and Malumbres M. P., "*Low-complexity multiresolution image compression using wavelet lower tress*", IEEE Transactions on Circuits, Systems and Video Technology 16 (11) 2006 pp. 1437-1444. Nov. 2006.

[Osman y Kelly, 1996] Osman, I.H., Kelly, J.P. (1996). Meta-Heuristics: Theory and Applications, 39 Kluwer Academic Publishers. ISBN: 0792397002. USA.

[Papadimitriou y Steiglitz, 1998] Papadimitriou, C. H., Steiglitz, K. (1998). Combinatorial Optmization. Algorithms and Complexity. ISBN: 0-486-40258-4. USA.

[Said y Pearlman, 1996] Said A., Pearlman A., "*A new, fast and efficient image codec based on set partitioning in hierarchical tress*", IEEE Transactions on Circuits and Systems for Video Technology. pp. 243-250 June 1996.

[Shannon 1948] Shannon. A mathematical Theory of Communication. (1948)

[Shapiro, 1993] Shapiro, J. M., "*Smart compression using the embedded zerotree wavelet (EZW) algorithm*", Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, pp. 486 - 490 vol.1 Nov. 1993.

[Shapiro, 1996] Shapiro J. M., "*A fast technique for identifying zerotress in the EZW algoritm*", Proc. IEEE Int. Conf. Acoust,. Speech, Signal Processing 3 (1996) 1455-1458.

[Stuart y Geman, 1984] Stuart Geman, Donald G., "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images " IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 721 – 741, Nov. 1984.

[Sungjei *et al.*, 2009] Sungjei Kim, Jinwoo Jeong, Yong-Goo Kim, Yungho Choi and Yoonsik Choe, "DIRECTION-ADAPTIVE CONTEXT MODELING FOR SIGN CODING IN EMBEDDED WAVELET IMAGE CODER", IEEE International Conference on Image Processing (ICIP), ISBN: 978-1-4244-5653-6, pp. 2809 – 2812, Nov. 2009.

[Taubman, 2000] Taubman D., "High performance scalable image compression with EBCOT ", IEEE Trans. On Image Procs. pp. 1158 – 1170. Jul. 2000

[Welch, 1984] Welch T. A., "A Technique for High-Performance Data Compression", IEEE Computer, ISSN : 0018-9162, pp. 8 – 19, June 1984.

[Wetzel, 1983] Wetzel, A. (1983). Evaluation of the Effectiveness of Genetic Algorithms in Combinatorial Optimization. University of Pittsburgh, Pittsburgh (unpublished). 1983.

[Wu, 1997] Wu X., "*High-order context modelling and embedded conditional entropy coding of wavelet coefficients for image compression*", in: Proc. Of 31st Asilomar Conf. on Signal, System, and Computers, 1997, pp.1378-1382

[Xiong *et al.*, 1997] Xiong Z, Ramchandran K., and Orchard M. T., "Space frequency quantization for wavelet image coding", IEEE Trans. Image Processing. pp. 677 – 693 May 1997.

[Yves, 1994] Yves M., "*Wavelets Algorithms and Applications*", Society for Industrial and Applied Mathematics, SIAM Press, USA, 1994.

[Zandi *et al.*, 1995] Zandi A., Allen J. D., Schwartz E. L., and Boliek M. "*CREW: Compression with Reversible Embedded Wavelets*". Data Compression Conference, 1995. DCC '95. Proceedings. pp. 212 – 221, ISBN:0-8186-7012-6 IEEE Press, 1995.

[Ziv y Lempel, 1977] Ziv J. and Lempel A., "A Universal Algorithm for sequential Data Compression", IEEE Trans. On Information Theory, ISSN : 0018-9448, pp. 337 – 343, May 1977.

<http://www.kodak.com/digitalImaging/samples/classic.shtml>

www.lenna.org

http://www.tecnick.com/public/code/cp_dpage.php?aiocp_dp=testimages

<http://www.grc.upv.es/Software/ltw.html>

Apéndice A

Estructuras de Datos Utilizadas en el Algoritmo de Recocido Simulado.

Para dar una solución al problema por medio de un algoritmo de Recocido Simulado, fue necesaria la utilización de estructuras de datos que permitieran ordenar la información necesaria de las imágenes y del vector predictor para hacer uso eficiente de la información.

Para cada combinación a analizar y para una vecindad, se define una estructura común que almacenará los datos relativos a cada combinación predicha.

```
typedef struct{
    char signo[TAM5]
    int costo;
    int aciertos[TAM5];
    int numOcuurrencias[TAM];
}st_solucion;
```

Esta estructura contiene el mapa de signos para cada predicción, el valor de costo para la función objetivo, el número de aciertos por combinación y el número de ocurrencias por combinación.

Se realiza un tratamiento previo sobre el mapa de signos de una imagen dada, abrimos el fichero txt que contiene los mapas de signos de la imagen a analizar, se analiza dicho fichero y se almacena en la estructura siguiente:

```
typedef struct {
    int *datos
    int numFilas
    int numColumnas
}st_imagenes;
```

Los datos (signos) de la imagen y el tamaño de esta, en filas por columna

Apéndice B

Complejidad Temporal por pasos.

De acuerdo a [Laarhoven 1985], la complejidad computacional del algoritmo de Recocido Simulado se presenta en la siguiente expresión

$$O(\tau L \ln |R|) \quad (1)$$

donde τ es el tiempo para generar y evaluar una posible solución,

$$\tau = O(n) + O(n^2)$$

donde n es la cantidad de signos generados en el vector con respecto al patrón de signos vecinos (NSP), es decir, cada elemento del vector predictor generado de forma aleatoria (ver tabla 2-2) y n^2 es la complejidad de evaluar cada elemento del vector NSP en la función que me permite obtener el signo de los vecinos de la matriz de símbolos de los signos de los coeficientes wavelet.

La longitud de la cadena de Markov representada por la letra L esta denotada por 3^m , donde m es el número de vecinos que serán utilizados para realizar la predicción de signo, por lo tanto tenemos que $n = 3^m$.

El espacio de soluciones del problema que se evalúa se representa con la letra R , donde es igual a 2^n , por lo tanto la complejidad de R es $O(2^n)$ siendo esta exponencial.

Por lo tanto la complejidad Asintótica del algoritmo de Recocido Simulado para el problema de la predicción de los signos de los coeficientes wavelet sustituyendo las variables de las expresiones en la función (1) se presenta de la siguiente forma:

$$O((n^2+n)(3^m)\ln(2^n))$$

El $\ln(2^n)$ es igual $n \ln 2$, por lo tanto la complejidad asintótica del algoritmo de Recocido Simulado es:

$$O((n^3+n^2)(3^m n) \ln 2)$$

La figura B-1 muestra la complejidad de ejecución del algoritmo, donde n representa la cantidad de signos generados con respecto al NSP y la $f(n)$ representa la complejidad asintótica del problema tratado, podemos observar que al aumentar el tamaño de n el número de instrucciones también aumenta.

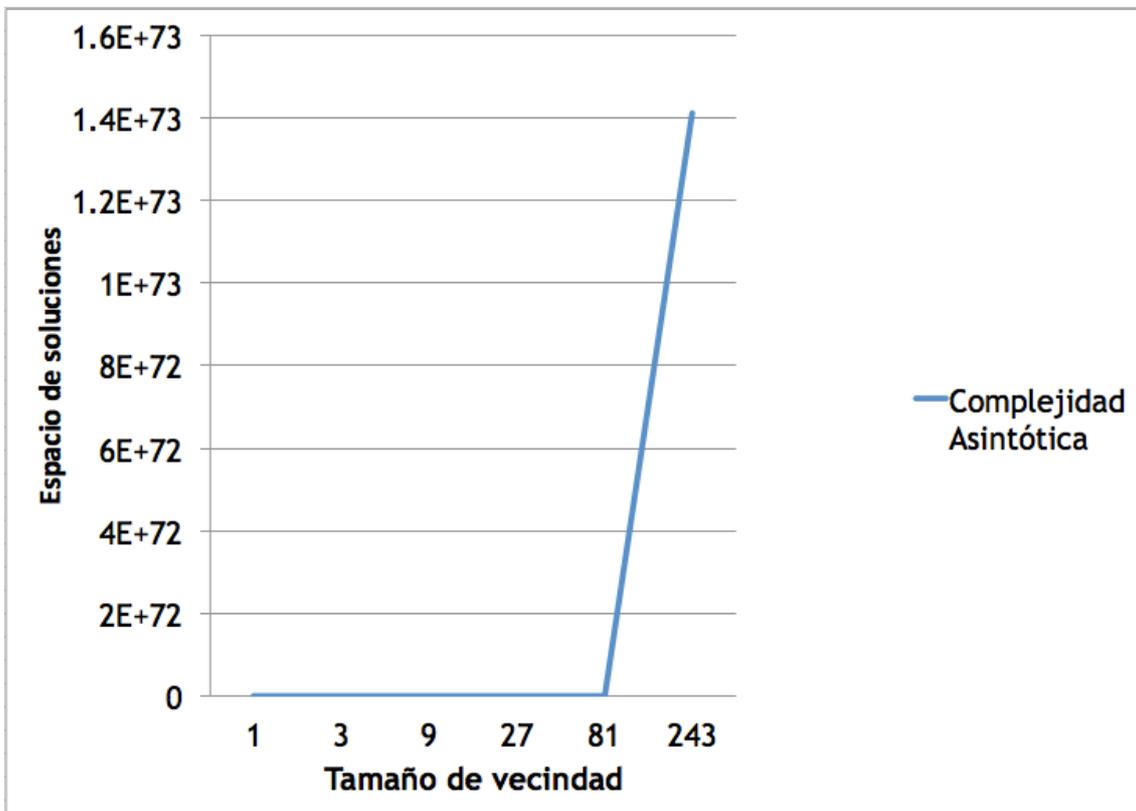


Figura B-1. Complejidad Asintótica para el algoritmo de Recocido Simulado

Apéndice C

Código fuente del Algoritmo de Recocido Simulado.

A continuación se presenta el código fuente correspondiente a la función que realiza el algoritmo de Recocido Simulado para la predicción de signo de los coeficientes wavelet utiliza una estructura de vecindad simple. Recibimos los parámetros, T_0 representa la temperatura inicial, T_f representa la temperatura final, β es el factor de decremento y $st_solucion$ representa la configuración de la solución inicial.

Mientras la temperatura inicial sea menor o igual que la temperatura final, generamos un estado S' (vecino) por medio de una perturbación de S (actual), evaluamos la función objetivo de S' y si dicha función objetivo es mayor o igual que la función objetivo de S , el estado se acepta como el estado actual, en caso contrario, se acepta o rechaza de acuerdo al criterio de aceptación de la función de probabilidad de Boltzmann

```
int f_recocidoSimulado(int iteraRS,st_solucion *actual,st_solucion *vecino,st_solucion
*mejorRS, int subbanda, int nivelWavelet,double tempFin,double _beta,double
templni){

    int i=0,cMarkov = TAM
    int costo_RS =0;
    int costo_actual =0
    int costo_vecino =0
    double temp_ini=templni;
    double alfa=0.0,probAcepta=0.0;
    double temp_fin = tempFin;
    double beta = _beta;

    for(temp_ini = templni ;temp_ini >= temp_fin;temp_ini *= beta){
        for(i = 0;i < longM ;i++){
            f_perturbarSolucionActual(actual,vecino);

            f_obtenerSigno5VecinosMatriz(subbanda,p_imagen,vecino,nivelWavelet);
            costo_vecino = f_costoFncObjetivo(vecino);
            costo_actual = f_costoFncObjetivo(actual);

            if ((costo_vecino - costo_actual) >= 0){

                f_copiarEstructuras5(actual,vecino);
                if(costo_vecino > costo_RS){
                    costo_RS=costo_vecino;
                    f_copiarEstructuras5(mejorRS,vecino);
                }
            }
        }
    }
}
```

```
    } else{
        probAcepta = exp((double)(costo_vecino - costo_actual)/temp_ini);
        alfa = ((double)rand()/ RAND_MAX);
        if(alfa < probAcepta){
            f_copiarEstructuras5(actual,vecino);
        }
    }
}
}
return costo_RS;
}
```

Para el criterio de aceptación generamos una variable α con la siguiente restricción $0 \geq \alpha < 1$, comparamos en la función de probabilidad de Boltzmann, si $\alpha < \text{probabilidad de aceptación}$, aceptamos el estado S' como una solución posible del problema hasta alcanzar el equilibrio.

Glosario de Términos.

Análisis de Sensibilidad: Es un componente importante en la construcción de modelos matemáticos, computacionales y simulación.

Complejidad Espacial: Cantidad de memoria que un algoritmo consume o utiliza durante su ejecución.

Complejidad Temporal: Tiempo que necesita el algoritmo para ejecutarse.

Costo: Para el problema de la predicción de signo de los coeficientes wavelet es el valor máximo que representa la cantidad de aciertos en el vector predictor.

Estructura de Vecindad: Tipo de movimiento (permutación, inserción o eliminación) utilizado para explorar el espacio de soluciones de un problema de optimización.

Heurística: Procedimiento intuitivo bien definido que proporciona buenas soluciones aproximadas a problemas difíciles de resolver sin garantizar la optimalidad, en un tiempo computacional razonable [Wetzel, 1983].

Metaheurísticas: Métodos aproximados que mejoran procedimientos heurísticos, los cuales son diseñados para ser aplicados a problemas considerados difíciles de resolver, donde las heurísticas no son eficientes [Osman y Kelly, 1996].

Recocido Simulado (SA): Es una metaheurística de búsqueda aleatoria utilizada en la solución de problemas de optimización combinatoria [Kirkpatrick et al., 1983]. Se basa en la analogía del proceso de recocido, es cuando un material se somete a un calentamiento a temperatura muy alta llegando al punto de fusión y después es enfriado gradualmente, sus moléculas se acomodan de tal forma que la energía potencial de la configuración de las

moléculas es mínima.

Óptimo Global: Es la mejor solución de un espacio de soluciones $f(x)$.

Óptimo Local: Representa la mejor solución de $f(x)$ en un entorno x

Sintonización: Es la proporción adecuada en cuanto a los valores obtenidos mediante el análisis de sensibilidad aplicado a los parámetros de control, tomando en cuenta el problema y el método de optimización utilizado, de modo que el algoritmo muestre una mejora tanto en eficiencia como en eficacia.

Vecindad: Es el conjunto de soluciones que pueden ser alcanzadas desde una solución dada por medio de un movimiento.

Problema clase P: Es un conjunto de todos los problemas de decisión que pueden ser resueltos por un algoritmo determinístico en tiempo polinomial.

Instancia: Es un problema de prueba que puede ser definido como el tamaño de entrada de los datos del problema.

Tiempo polinomial: Se dice que un algoritmo puede ser resuelto en tiempo polinomial si su función de complejidad es de orden $O(n)$, es decir, que puede ser representado por un polinomio.