



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Ingeniería Eléctrica

Sección de Computación

Algoritmos de Compresión y Descompresión de Imágenes Digitales para Dispositivos Móviles

Tesis que presenta

Francisco López Orozco

para obtener el Grado de

Maestro en Ciencias

en la Especialidad de

Ingeniería Eléctrica

con Opción en Computación

Director de la Tesis

Dr. Arturo Díaz Pérez

México, D.F.

Febrero 2006

Agradecimientos

Agradezco primeramente a mis padres por el gran apoyo, amor, cariño e impulso otorgado durante ya una muy buena parte de mi vida.

Muchas gracias a mi asesor, el Dr. Arturo Díaz Pérez, a quien debo su valiosa ayuda, paciencia y consejos otorgados en todo momento con el fin de guiarme correctamente y hacer posible la culminación de este trabajo.

A tí hermana mía, Sofía, en quien veo un gran ejemplo a seguir y que me impulsa a levantarme después de una caída y seguir siempre adelante.

Por supuesto, a tí Mariana, que me has acompañado en momentos muy importantes de mi vida, siempre de manera incondicional y con quien comparto esta nueva meta alcanzada y pasará el resto de mi vida.

Agradezco a todos los profesores de la Sección de Computación del CINVESTAV por su enorme contribución a mi formación, a través de sus conocimientos y experiencia compartida. En especial, al Dr. Luis Gerardo de la Fraga, al Dr. Guillermo Morales Luna y al Dr. Francisco Rodríguez Henríquez, por sus valiosas observaciones y sugerencias con el fin de mejorar el presente trabajo. A Sofi, por el especial apoyo que de manera desinteresada me brindó durante toda mi estancia en la sección.

Por último, agradezco al Consejo Nacional de Ciencia y Tecnología por el apoyo otorgado a través de una beca-crédito de nivel maestría. Además este trabajo fue parcialmente financiado por el proyecto Conacyt 45306: “Estudio, análisis y desarrollo de algoritmos de muy alto desempeño para arquitecturas hardware/software”.

R e s u m e n

Desde los 90, el uso de una nueva categoría de dispositivos, conocidos como dispositivos móviles se ha extendido. Buenos ejemplos de dispositivos móviles son los asistentes personales digitales (APDs), las computadoras portátiles y los teléfonos celulares. Por otro lado muchas aplicaciones de software han sido desarrolladas para dichos dispositivos, pero los gráficos de alta calidad o imágenes están aún muy por delante de las capacidades computacionales de estos dispositivos, así como los ambientes restringidos con bajos anchos de banda sobre los que trabajan. A pesar de que los dispositivos mencionados cada vez poseen mayores capacidades de cómputo, es importante desarrollar técnicas avanzadas para la transmisión, compresión y descompresión de imágenes digitales para estos dispositivos, ya que dicha problemática no se encuentra completamente resuelta.

Es por ello, que esta tesis propone una solución de compresión y descompresión de imágenes digitales capaz de operar eficientemente en tales dispositivos, que poseen características computacionales limitadas, dentro de un ambiente de comunicación inalámbrica. La solución está basada en la herramienta avanzada para el procesamiento de imágenes, la transformada *wavelet*. El trabajo incluye el estudio, diseño, implementación, evaluación y caracterización de tres técnicas de compresión y descompresión de imágenes basadas en las transformadas *wavelets*: Haar, Daubechies y Coiflets, sobre ambientes de cómputo móvil basados en tres de los estándares más importantes de comunicación inalámbrica, la comunicación infrarroja (IrDA), IEEE802.11b (WLAN) y *Bluetooth*.

Palabras clave: *Coiflets*, compresión de imágenes, *Daubechies*, dispositivos móviles, *Haar*, *wavelets*.

A b s t r a c t

Since 90's, the use of a new category of devices, known as mobile devices have become increasingly widespread. Good examples of mobile devices are the personal digital assistants (PDA's), laptops and cellular phones. On the other hand, a large number of software applications have been developed for these devices. However, the high-quality graphics or images still remain beyond the computational capability of these devices. Also, the mobile devices work in restricted environments and with low bandwidths. Although the mentioned devices every time have greater capacities of computing, it is important to develop efficient techniques for the transmission, compression and decompression of digital images for this kind of devices, since this problem is not completely solved.

Therefore, this thesis proposes a digital image compression and decompression solution able to operate efficiently in such mobile devices with limited characteristics of computing in a wireless environment. The solution is based on advanced tools of digital image processing, the wavelet transform. The work includes the study, design, implementation, evaluation and characterization of three image compression and decompression techniques based on the wavelets: Haar, Daubechies and Coiflets, over communication environments based on three of the most important communication standards, infrared communication (IrDA), IEEE802.11b (WLAN) and Bluetooth.

Key words: Coiflets, image compression, Daubechies, mobile devices, Haar, wavelets.

Índice general

Resumen	III
Abstract	IV
1. Introducción	1
1.1. Estado del arte	3
1.2. Objetivos	4
1.3. Metodología	5
1.4. Organización de la tesis	6
2. Compresión y Descompresión de Datos	7
2.1. Compresión de datos	7
2.1.1. Codificación por largo de recorrido	9
2.1.2. Codificación de Huffman	9
2.1.3. Codificación aritmética	10
2.1.4. Codificación Lempel Ziv Welch (LZW)	11
2.2. Compresión de imágenes digitales	13
2.2.1. Definición de imagen digital	13
2.2.2. Cuantización	15
2.2.3. Espacios de color	15
2.2.4. El estándar JPEG	17
2.3. Comparación de las técnicas de compresión de imágenes	21
3. Compresión de Imágenes Basada en <i>Wavelets</i>	23
3.1. Introducción	23
3.2. Análisis de multiresolución	26
3.3. Transformada <i>wavelet</i> discreta para 1D	27
3.4. Transformada <i>wavelet</i> discreta para 2D	31

3.5.	Compresión de imágenes	32
3.5.1.	Evaluación de la eficiencia de una técnica de compresión	36
3.5.2.	Elección de la transformada <i>wavelet</i>	40
4.	Implantación de la Arquitectura de Cómputo Móvil	43
4.1.	Introducción	43
4.2.	Arquitectura de cómputo móvil	43
4.3.	Estándares IrDA, IEEE802.11b y <i>Bluetooth</i>	47
4.4.	Programación en red	50
4.5.	Módulo de comunicaciones	53
4.5.1.	Descripción del protocolo	53
4.6.	Módulo de compresión/descompresión de imágenes digitales	57
4.7.	Caso de estudio: <i>TRImagen</i>	64
4.7.1.	Características de <i>TRImagen</i>	66
5.	Evaluación y Análisis de Resultados	69
5.1.	Plataforma de experimentación	69
5.2.	Evaluación	70
5.3.	Comunicaciones entre los dispositivos móviles	70
5.3.1.	Resultados experimentales de <i>TRImagen</i> en su versión para IrDA	72
5.3.2.	Resultados experimentales de <i>TRImagen</i> en su versión para <i>Bluetooth</i>	77
5.3.3.	Resultados experimentales de <i>TRImagen</i> en su versión para IEEE802.11b	80
5.4.	Algoritmos de compresión y descompresión de imágenes	88
6.	Conclusiones	103
6.1.	Contribuciones	105
6.2.	Trabajo a futuro	107
A.	Anexos	109
A.1.	Imágenes de prueba	109
A.2.	Otras imágenes de prueba	119

Índice de figuras

2.1. Un sistema típico de compresión de datos.	9
2.2. JPEG Baseline a) Codificador b) Decodificador.	18
2.3. Codificación Diferencial.	19
2.4. Secuencia de Zig-Zag de la codificación entrópica en JPEG.	20
2.5. Tabla de algoritmos de compresión y descompresión de imágenes. . .	22
3.1. Banco de filtros de 2 canales.	27
3.2. (a) Un filtro DWT en el caso de una señal en 2-D. (b) Estructura piramidal de la descomposición <i>wavelet</i>	33
3.3. Diagrama a bloques del esquema de compresión de imágenes basado en la transformada <i>wavelet</i>	34
3.4. Diagrama a bloques del esquema de descompresión de imágenes basa- do en la transformada <i>wavelet</i>	34
3.5. Diagrama a bloques del análisis <i>wavelet</i> de imágenes.	36
3.6. Diagrama a bloques de la síntesis <i>wavelet</i> de imágenes.	37
3.7. Número óptimo de descomposiciones <i>wavelets</i> para diferentes órdenes de filtros.	41
3.8. Resultados de Relación pico señal–ruido en dB para diferentes familias de <i>wavelets</i> y diferentes tasas de compresión. y niveles de descomposi- ción <i>wavelet</i> aplicado a imágenes estándares.	42
4.1. Primer escenario de cómputo móvil.	44
4.2. Segundo escenario de cómputo móvil.	44
4.3. Tercer escenario de cómputo móvil.	45
4.4. Cuarto escenario de cómputo móvil.	45
4.5. Diagrama a bloques de una aplicación cliente-servidor basada en <i>sockets</i> . .	52
4.6. Creación de un circuito virtual mediante <i>sockets</i> en el módulo de co- municaciones.	56

4.7. Diagrama a bloques del módulo de comunicaciones en su versión para IrDA.	57
4.8. Diagrama a bloques del módulo de comunicaciones en su versión para IEEE 802.11b y <i>Bluetooth</i>	57
4.9. Transformada <i>wavelet</i> en 2D.	59
4.10. Transformada <i>wavelet</i> de los componentes de color de una imagen en el modelo RGB.	60
4.11. Función de transferencia del cuantizador lineal utilizado en el formato <i>wavelet</i>	61
4.12. Diagrama a bloques que muestra la interacción entre el módulo de comunicaciones y el de compresión/descompresión de imágenes digitales.	65
4.13. Interfase gráfica de TRImagen en su versión para PC.	66
4.14. Interfase gráfica de TRImagen en su versión para el APD.	67
5.1. Distancia máxima de separación de los dispositivos IrDA vs ángulo de incidencia.	72
5.2. Rendimiento de <i>TRImagen</i> en IrDA.	78
5.3. Rendimiento de <i>TRImagen</i> en IrDA.	79
5.4. Rendimiento de <i>TRImagen</i> en IrDA.	80
5.5. Ancho de banda de <i>TRImagen</i> en IrDA.	81
5.6. Ancho de banda de <i>TRImagen</i> en IrDA.	82
5.7. Ancho de banda de <i>TRImagen</i> en IrDA.	83
5.8. Rendimiento de <i>TRImagen</i> en su versión para <i>Bluetooth</i>	84
5.9. Ancho de banda <i>TRImagen</i> en su versión para <i>Bluetooth</i>	85
5.10. Rendimiento de <i>TRImagen</i> en su versión IEEE802.11b.	86
5.11. Rendimiento de <i>TRImagen</i> en su versión IEEE802.11b.	87
5.12. Ancho de banda de <i>TRImagen</i> en su versión IEEE802.11b.	88
5.13. Ancho de banda de <i>TRImagen</i> en su versión IEEE802.11b.	89
5.14. Ancho de banda de <i>TRImagen</i> en su versión IEEE802.11b.	90
5.15. Ancho de banda de <i>TRImagen</i> en su versión IEEE802.11b.	91
5.16. Imagen tomada como imagen de entrada para llevar a cabo las pruebas 1-7.	92
5.17. Subimágenes, resultado de la primera descomposición <i>wavelet</i> , sin considerar ningún valor de umbral para descartar coeficientes. Basada en la transformada Haar.	93

5.18. Diferencias presentadas entre las tres técnicas de compresión implementadas, respecto al porcentaje de conservación de la energía de la señal de entrada en función del nivel de descomposición <i>wavelet</i>	94
5.19. Variación de la razón de compresión de las técnicas implementadas, en función del nivel de descomposición <i>wavelet</i>	95
5.20. Variación del error cuadrático medio al recuperar la imagen de entrada en función de un valor de umbral, mediante el cual, se descartan ciertos coeficientes resultado de las transformaciones.	96
5.21. Variación de la relación pico-signal a ruido después del proceso de descompresión de la imagen comprimida, en función de un valor de umbral.	97
5.22. Imagen recuperada sin considerar un valor de umbral, a un primer nivel de descomposición <i>wavelet</i> , con 4 bpp. basada en la transformada Daubechies D-4.	97
5.23. Imagen recuperada sin considerar un valor de umbral, a un primer nivel de descomposición <i>wavelet</i> , con 1 bpp. basada en la transformada Coiflets C-6.	98
5.24. Imagen de entrada en formato RGB a 3 bytes/pixel de tamaño 800 x 800 pixeles.	98
5.25. Imágenes recuperadas considerando un valor de umbral = 70, para un primer nivel de descomposición <i>wavelet</i> . De izq. a derecha: Basada en la transformada Coiflets (C-6) y en Duabechies (D-4).	99
5.26. Gráfico de rendimiento del módulo de compresión en su versión para la computadora portátil.	101
5.27. Gráfico de rendimiento del módulo de compresión en su versión para el APD.	102
A.1. Imagen <i>lena</i> en formato BMP a 3 bytes/pixel (RGB) de 256 x 256 pixeles.	109
A.2. Imagen <i>patrón simple</i> en formato BMP a 3 bytes/pixel (RGB) de 450 x 450 pixeles.	110
A.3. Imagen <i>patrón complejo</i> en formato BMP a 3 bytes/pixel (RGB) de 512 x 512 pixeles.	110
A.4. Imagen <i>huella dactilar</i> en formato BMP a 3 bytes/pixel (RGB) de 480 x 480 pixeles.	111
A.5. Imagen <i>piramidal</i> en formato BMP a 3 bytes/pixel (RGB) de 602 x 602 pixeles.	111

A.6. Imagen <i>textura</i> en formato BMP a 3 bytes/pixel (RGB) de 960 x 960 píxeles.	112
A.7. Imagen <i>puesta de sol</i> en formato BMP a 3 bytes/pixel (RGB) de 800 x 800 píxeles.	112
A.8. Imagen <i>eclipse</i> en formato BMP a 3 bytes/pixel (RGB) de 700 x 700 píxeles.	113
A.9. Imagen <i>colina</i> en formato BMP a 3 bytes/pixel (RGB) de 512 x 512 píxeles.	113
A.10. Imagen <i>superficie lunar</i> en formato BMP a 3 bytes/pixel (RGB) de 440 x 440 píxeles.	114
A.11. Imagen <i>montañas</i> en formato BMP a 3 bytes/pixel (RGB) de 1024 x 1024 píxeles.	114
A.12. Imagen <i>con texto</i> en formato BMP a 3 bytes/pixel (RGB) de 684 x 684 píxeles.	115
A.13. Imagen <i>periódico</i> en formato BMP a 3 bytes/pixel (RGB) de 1024 x 1024 píxeles.	115
A.14. Imagen <i>dibujo animado</i> en formato BMP a 3 bytes/pixel (RGB) de 1280 x 1280 píxeles.	116
A.15. Imagen <i>rostro sintético</i> en formato BMP a 3 bytes/pixel (RGB) de 602 x 602 píxeles.	116
A.16. Imagen <i>rostro 1</i> en formato BMP a 3 bytes/pixel (RGB) de 1280 x 1280 píxeles.	117
A.17. Imagen <i>rostro 2</i> en formato BMP a 3 bytes/pixel (RGB) de 500 x 500 píxeles.	117
A.18. Imagen <i>fractal</i> en formato BMP a 3 bytes/pixel (RGB) de 500 x 500 píxeles.	118
A.19. Imagen <i>babuino</i> en formato BMP a 3 bytes/pixel (RGB) de 256 x 256 píxeles.	119
A.20. Imagen <i>pimienta</i> en formato BMP a 3 bytes/pixel (RGB) de 256 x 256 píxeles.	119
A.21. Imagen <i>chapultepec</i> en formato BMP a 3 bytes/pixel (RGB) de 256 x 256 píxeles.	120

Índice de cuadros

4.1. Comparación entre los protocolos <i>Bluetooth</i> y WiFi.	51
5.1. Distancia máxima de separación de los dispositivos IrDA vs ángulo de incidencia.	72
5.2. Envío de imágenes desde el APD hacia la CP a una distancia de separación de 0.20 m.	73
5.3. Envío de imágenes desde el APD hacia la CP a una distancia de separación de 0.60 m.	73
5.4. Envío de imágenes desde el APD hacia la CP a una distancia de separación de 1.0 m.	73
5.5. Envío de imágenes desde el APD hacia la CP a una distancia de separación de 0.20 m. y ángulo de desviación de 45°.	74
5.6. Envío de imágenes desde la CP hacia el PDA a una distancia de separación de 0.20 m.	74
5.7. Envío de imágenes desde la CP hacia el PDA a una distancia de separación de 0.60 m.	75
5.8. Envío de imágenes desde la computadora portátil hacia el PDA a una distancia de separación de 1.0 m.	75
5.9. Envío de imágenes desde la computadora portátil hacia el PDA a una distancia de separación de 0.20 m y un ángulo de separación de 45°.	75
5.10. Transmisión de imágenes entre APD's a una distancia de separación de 0.20 m.	76
5.11. Transmisión de imágenes entre APD's a una distancia de separación de 0.60 m.	76
5.12. Transmisión de imágenes entre APD's a una distancia de separación de 1.0 m.	76
5.13. Transmisión de imágenes entre APD's a una distancia de separación de 0.20 m y un ángulo de incidencia de 45°.	77

5.14. Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante <i>Bluetooth</i> cuando los dispositivos se ubicaron a una distancia de 10m de separación dentro del mismo piso en el mismo edificio.	78
5.15. Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante <i>Bluetooth</i> cuando los dispositivos se ubicaron en pisos distintos en el mismo edificio.	79
5.16. Resultados obtenidos al transferir diferentes tamaños de imágenes mediante IEEE802.11b cuando los dispositivos se ubicaron a una distancia de 10m al mismo nivel del mismo edificio.	81
5.17. Resultados obtenidos al transferir diferentes tamaños de imágenes mediante IEEE802.11b cuando los dispositivos se ubicaron a una distancia de 10m al mismo nivel del mismo edificio.	82
5.18. Resultados obtenidos al transferir diferentes tamaños de imágenes mediante IEEE802.11b con los dispositivos ubicados en diferentes niveles del mismo edificio	83
5.19. Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante IEEE802.11b cuando los dispositivos se ubicaron en pisos distintos en el mismo edificio.	84
5.20. Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante IEEE802.11b cuando los dispositivos se ubicaron en pisos distintos en el mismo edificio.	85
5.21. Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante IEEE802.11b cuando los dispositivos se ubicaron en pisos distintos en el mismo edificio.	86
5.22. Resumen de los resultados obtenidos al probar con las tres técnicas de compresión implementadas.	92
5.23. Resumen de los resultados obtenidos con la 1a. técnica implementada, con un valor de umbral=30.	93
5.24. Resumen de los resultados obtenidos con la 2a. técnica implementada, con un valor de umbral=30.	99
5.25. Resumen de los resultados obtenidos con la 3a. técnica implementada, con un valor de umbral=30.	100
5.26. Tiempos de compresión/descompresión de imágenes para las tres técnicas implementadas.	100

Índice de algoritmos

1.	Construcción del arbol binario	10
2.	Codificación aritmética	11
3.	Proceso de codificación LZW	12
4.	Esquema típico de JPEG	18
5.	Dispositivo cliente	54
6.	Dispositivo servidor	55
7.	Submódulo de compresión	62
8.	Submódulo de descompresión	63

Capítulo 1

Introducción

Desde los años 90, avances en la tecnología de redes inalámbricas y en aplicaciones portables motivaron el surgimiento de un nuevo paradigma de computación, llamado cómputo móvil, en el cuál los usuarios que portan cierto dispositivo tienen acceso a servicios de información a través de una infraestructura común independientemente de su ubicación física o si éstos se encuentran en movimiento. En particular, se ha dado un gran crecimiento en el uso de tecnologías de dispositivos móviles como los teléfonos celulares, APDs (Asistentes Personales Digitales) y computadoras portátiles [2]. Actualmente, éstas siguen siendo áreas de mercado en constante crecimiento y cada vez se incrementa la demanda de transmisión de datos multimedia (imágenes, audio y video) para estos dispositivos.

Los dispositivos móviles poseen características esencialmente limitadas (principalmente en el caso de los teléfonos celulares y los APD's), tales como cantidad de memoria, velocidad de procesamiento, cantidad de energía que pueden almacenar y el ancho de banda utilizado para sus comunicaciones inalámbricas.

A pesar que los dispositivos móviles mencionados poseen cada vez mayores capacidades de cómputo, sigue siendo un gran reto el desarrollo de nuevas técnicas para optimizar su funcionamiento cuando procesan datos de tipo multimedia; ya que los gráficos de alta calidad (o imágenes) están aún muy por delante de las capacidades computacionales de ellos, por lo que una área de interés se refiere al procesamiento de imágenes digitales o gráficos en móviles de manera eficiente.

Por otro lado, es importante señalar que la forma de comunicación entre estos dispositivos es por excelencia en forma inalámbrica y hoy en día existen principalmente tres estándares utilizados para establecer dicha comunicación. El primero de ellos mediante el puerto infrarrojo (IrDA, *Infra Red Data Association*), cuyas caracterís-

ticas principales son: fácil de usar, seguro en cualquier ambiente, inmune al ruido electromagnético, afectado mínimamente por la diafonía. Se requiere que los dispositivos se encuentren alineados y a una distancia máxima de separación entre ellos (típicamente de 1 a 2 metros) [4, 5].

El segundo es la tecnología inalámbrica *Bluetooth*, cuya finalidad es la creación de redes inalámbricas de poco alcance (RAPs, Redes de Área Personal o picoredes) entre dispositivos que se encuentren cercanos. La conexión en este tipo de tecnología se lleva a cabo mediante un enlace de radio de alta frecuencia. Con *Bluetooth* se trata de cubrir algunas de las deficiencias existentes en otras tecnologías, sin embargo, la principal limitante de *Bluetooth* es el corto alcance de sus comunicaciones. A pesar de ello, se espera que *Bluetooth* tenga mayor desarrollo en los siguientes años.

El último es el estándar IEEE 802.11 que permite establecer comunicación multipunto. Éste es el estándar por excelencia utilizado en las redes inalámbricas locales (WLANs) desarrollado por la IEEE. Aunque se han desarrollado varias versiones de dicho estándar, todas cuentan con características comunes tales como: uso del protocolo ethernet y esquemas como CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) para el aprovechamiento óptimo del canal de comunicación, esquemas de modulación como PSK (*Phase Shift Keying*) o más recientes como CCK (*Complementary Code Keying*) que provee una mayor velocidad en la transmisión de datos así como una reducción importante en la vulnerabilidad a interferencias [6].

No obstante lo anterior, las comunicaciones inalámbricas presentan algunos inconvenientes tales como, el bajo ancho de banda de operación, los usuarios deben algunas veces enviar paquetes pequeños de información a fin de agilizar su transmisión, muchas de las aplicaciones necesitan ser reconfiguradas si van a ser usadas en un medio inalámbrico y la mayoría de las aplicaciones cliente-servidor requieren de una conexión persistente, lo cual no siempre se da con las comunicaciones inalámbricas.

Además en la comunicación inalámbrica, se presenta al fenómeno de interferencia, por lo que es inevitable que se presenten errores durante la transmisión de datos [7].

Por último, cuando se establece la comunicación entre dos o más dispositivos es importante adoptar un modelo de comunicación como el de cliente-servidor [1], en donde el dispositivo cliente hace peticiones que son atendidas por el dispositivo servidor. En este esquema es común que el cliente inicie y termine el proceso de comunicación. Normalmente, el dispositivo que actúa como servidor posee mayores capacidades de cómputo que cualquiera de los dispositivos cliente, dado que es posi-

ble que varios clientes envíen peticiones simultáneamente al servidor. Otro esquema sugiere el uso de un tercer dispositivo que actúe como puente o intermediario entre las dos entidades que pretenden comunicarse.

El intercambio de información para dispositivos móviles debe considerar entre varios aspectos la estrechez de los canales de comunicación, las capacidades de almacenamiento, despliegue y procesamiento del dispositivo móvil, y el tipo de información a intercambiar. Particularmente, para el intercambio de imágenes es fundamental destacar que las imágenes contienen mucha información redundante la cual no es perceptible por el ojo humano, los dispositivos móviles tienen capacidades de despliegue muy limitadas por lo que no se requiere enviar imágenes de muy alta resolución. Por lo anterior, al intercambiar imágenes es conveniente aplicar técnicas de compresión de información que permitan hacer un uso eficiente de los canales de comunicación y que a su vez preserven la información semántica de una imagen cuando es vista en un dispositivo móvil.

1.1. Estado del arte

En el trabajo [8] se propone un sistema de transmisión móvil de imágenes basado en un esquema de modulación jerárquica para lograr la transmisión eficiente de imágenes digitales de alta calidad en canales de comunicación deteriorados. Este trabajo se basa en el supuesto que no siempre es necesario para un dispositivo móvil cliente recibir toda la información contenida en una imagen, y que es posible en un momento dado despreocuparse de ella.

En [9] se presentan algunos ejemplos de arquitecturas para la implementación de codificadores basados en las transformadas *wavelets*. Este trabajo muestra cómo se pueden desarrollar implementaciones caracterizadas en una forma paramétrica en términos de características arquitecturales genéricas, tales como: cantidad de memoria y el ancho de banda empleado para las comunicaciones.

En el 2002, se reporta el trabajo [3]. En él se discuten transformaciones que se recomiendan aplicar a los gráficos o imágenes para que sean desplegados en los dispositivos móviles de manera eficiente. El trabajo reporta esencialmente dos tipos de transformaciones, la primera que toma en cuenta únicamente cierta información de la imagen para ajustar y seleccionar la transformación. La segunda categoría de transformaciones toma en cuenta un análisis de la imagen fuente para seleccionar y ajustar los parámetros de la transformación.

En [11], se presenta una técnica de compresión de imágenes de alta calidad para comunicaciones móviles basada en la transformada *wavelet*. Parte de la técnica se basa en el hecho de que no se necesita toda la información contenida en una imagen para reconstruirla, ya que parte de esa información puede aproximarse mediante el uso de dicha transformada. En este trabajo únicamente se hace uso de la TWD (*Transformada Wavelet Discreta*) para un primer nivel. Sin embargo, resultaría interesante probar dicha técnica para niveles superiores con el fin de buscar mayores tasas de compresión.

En [12], se trata el problema de la visualización de imágenes de gran tamaño en dispositivos con capacidades de visualización limitadas como los dispositivos móviles. Este modelo permite calcular una ruta de navegación óptima para simular el comportamiento de navegación de una persona. La evaluación de los resultados del trabajo presentado son completamente experimentales y subjetivos ya que no se cuenta con una medida objetiva para medir el rendimiento o impacto de este tipo de herramientas y se optó por someterlo a la opinión de distintos usuarios. Sin embargo, dichos resultados experimentales mostraron que la tarea de navegación de los usuarios fue mejorada significativamente en una amplia variedad de imágenes.

Como puede observarse en los trabajos mencionados anteriormente han abordado de alguna manera el problema de la transmisión, compresión y descompresión de imágenes digitales para dispositivos móviles, pero cada uno presenta una solución que mejora algún aspecto de esta problemática.

Dadas las características de los dispositivos móviles ligeros y de los medios de comunicación inalámbricos, es necesario desarrollar técnicas adecuadas para el intercambio de información en este ambiente. Considerando las limitaciones de las capacidades gráficas de los dispositivos móviles, de particular importancia abordar el problema para la transmisión de imágenes digitales.

1.2. Objetivos

El presente trabajo pretende proponer una solución de compresión, descompresión y transmisión de imágenes digitales, capaz de operar eficientemente en dispositivos móviles con características limitadas en cuanto a velocidad de procesamiento, cantidad de memoria, capacidad de despliegue de información, espacio disponible para el almacenamiento de información, entre otras.

El objetivo general de este trabajo es hacer un estudio de los algoritmos para

la compresión, descompresión y transmisión de imágenes digitales para su uso entre dispositivos móviles (teléfonos celulares, APD's y computadoras portátiles).

De manera particular, se plantean alcanzar los siguientes objetivos específicos:

- Aplicar técnicas avanzadas de procesamiento digital de señales (posiblemente con autoadaptación) para la compresión y descompresión de imágenes digitales. Utilizando las transformaciones *wavelets*, no consideradas dentro del procesamiento digital de señales clásico.
- Utilizar esquemas de comunicación inalámbrica tales como IrDA, *Bluetooth* e IEEE802.11b.

1.3. Metodología

La metodología seguida para alcanzar los objetivos planteados fue la siguiente:

- Se llevó a cabo la revisión del estado del arte en materia de compresión, descompresión y transmisión de imágenes digitales entre dispositivos móviles y un estudio de las principales características de los dispositivos disponibles en el mercado con el fin de detectar posibles soluciones que serían factibles de implantar en ellos.
 - Se propusieron las soluciones al problema planteado, contemplando para ello los dos aspectos que debían resolverse:
 - Comunicaciones entre los dispositivos. Se desarrolló un protocolo sencillo de comunicación inalámbrica basado en *sockets*, bajo el paradigma cliente-servidor y orientado a conexión.
 - Técnicas de compresión. Se implementaron tres técnicas de compresión y descompresión de imágenes digitales basadas en las transformadas *wavelets Haar*, *Daubechies* y *Coiflets*. Todas con pérdida de información y asimétricas. Se hizo un estudio comparativo entre las técnicas implementadas con el fin de caracterizarlas.
 - Se efectuaron las pruebas y la evaluación de las implementaciones desarrolladas, en la plataforma (o arquitectura) real de cómputo móvil, conformada por una computadora portátil y un asistente personal digital por medio de los estándares de comunicación inalámbrica IrDA, *Bluetooth* e IEEE802.11b
-

El resultado de las pruebas a las que se sometieron las implementaciones desarrolladas, reflejaron buenos resultados y cumplieron con las expectativas planteadas al principio del desarrollo del trabajo, proponiéndose tres soluciones de compresión, descompresión y transmisión de imágenes digitales, capaces de operar eficientemente en los dispositivos móviles mencionados.

1.4. Organización de la tesis

En el capítulo 2, se da una explicación detallada de la compresión y descompresión de datos y en particular cuando esta información es de tipo gráfica como el caso de las imágenes, así como las medidas cuantitativas y cualitativas empleadas para medir su rendimiento. Se presenta también una breve explicación de algunas técnicas clásicas y una breve comparación entre ellas.

El capítulo 3, presenta a las técnicas de compresión y descompresión de datos basadas en las transformadas *wavelets* no consideradas dentro del procesamiento digital clásico basado en la transformada de Fourier. Se hace hincapié en las ventajas que presentan este tipo de técnicas cuando se aplica a datos de tipo gráfico o imágenes en comparación de las técnicas clásicas tratadas en el capítulo 2.

El capítulo 4, explica el diseño de la arquitectura de cómputo móvil empleada así como las técnicas de compresión y descompresión de imágenes digitales basadas en las transformadas *wavelets* que fueron implementadas. El capítulo 5 presenta los resultados obtenidos con las pruebas a las que fueron sometidas las implementaciones realizadas así como un análisis de dichos resultados.

El capítulo 6, menciona las conclusiones y logros alcanzados con el desarrollo de este trabajo de tesis, también da idea de posibles líneas de investigación y estudio para trabajos futuros con el fin de mejorar las soluciones propuestas.

Capítulo 2

Compresión y Descompresión de Datos

2.1. Compresión de datos

La compresión de datos permite representar cierta información en una cantidad menor de espacio en memoria o de almacenamiento, es decir, con una cantidad menor de bytes [13, 14]. La descompresión de datos es el proceso inverso a la compresión, el cual, permite recuperar los datos que se tenían previamente al ser comprimidos. Las técnicas de compresión de datos pueden ser divididas en dos grandes grupos: con pérdida y sin pérdida de información. La compresión de datos con pérdida permite la existencia de mayores tasas o razones de compresión, a cambio de desechar parte de la información que no es posible recuperar durante el proceso de descompresión. La compresión con pérdida presenta un buen rendimiento cuando se le aplica a datos o señales de tipo gráficos (como las imágenes) y a señales de voz. Muchas de las técnicas de compresión con pérdida pueden ser ajustadas a diferentes niveles de calidad, ganando exactitud en la salida a cambio de reducir las tasas de compresión. En cambio las técnicas de compresión sin pérdida consisten en aquellas que garantizan generar un duplicado exacto del flujo de datos después del ciclo completo de compresión y descompresión. Este tipo de compresión es usada cuando se almacenan registros en una base de datos, una hoja de cálculo o archivos de palabras, por ejemplo. En este tipo de aplicaciones, la pérdida de un simple bit de información puede resultar catastrófico.

En general, la compresión de datos consiste en tomar un flujo de símbolos y transformarlos en códigos. Si la compresión es efectiva, el tamaño del flujo resultante

codificado será más pequeño que el del flujo de símbolos original. Se utiliza un modelo para decidir cual código de salida se generará para los símbolos de entrada. El modelo es simplemente una colección de datos y reglas usadas para procesar los símbolos de entrada y determinar cuales códigos serán la salida.

Algunos ejemplos de codificadores son el codificador de Huffman y el codificador por largo de recorrido (CLR). En el caso de la codificación Huffman, la salida real del codificador es determinada por un conjunto de probabilidades. Cuando se usa este tipo de codificación, un símbolo que tiene una alta probabilidad de ocurrencia genera un código de pocos bits. En cambio con un símbolo cuya probabilidad de ocurrencia es baja se genera un código de bits más largo.

La compresión de datos es una de las aplicaciones más importante de la teoría de la información, la cual es la rama de las matemáticas que tuvo sus orígenes a finales de los años 40 con el trabajo de Claude Shannon en los Laboratorios Bell [15].

La compresión de datos explota la información redundante que existe, por ejemplo, en un mensaje. Al eliminar información redundante, el tamaño del mensaje se reduce. La entropía de un mensaje es la medida de qué tanta información es codificada en un mensaje. La entropía de un símbolo es definida como el negativo del logaritmo de su probabilidad. Para determinar este contenido de información de un mensaje en bits, suele expresarse usando el negativo del logaritmo base dos como en la ecuación 2.1.

$$\textit{Entropía} = -\log_2(\textit{probabilidad}) \quad \textit{y} \quad \textit{Entropía} \simeq \textit{No.bits} \quad (2.1)$$

La entropía de un mensaje entero es simplemente la suma de la entropía de todos sus símbolos individuales. La entropía mide cuantos bits de información hay realmente en un mensaje, por ejemplo, si la probabilidad de que el símbolo ‘e’ aparezca en un escrito es 1/16, el contenido de información del carácter es 4 bits, así que la cadena “eeee” tiene un contenido total de 20 bits. Si se estuvieran usando caracteres ASCII estándares a 8 bits para codificar el mensaje, en realidad se usarían 40 bits. La diferencia de 20 bits de la entropía y los 40 bits usados para codificar el mensaje es donde la compresión puede hacer presencia.

A continuación se presentan cuatro técnicas de codificación entrópica, la codificación por largo de recorrido basada en datos repetitivos, la codificación de Huffman, la codificación aritmética, basada en un modelo probabilístico, y finalmente la técnica de Lempel Ziv Welch basada en un esquema de diccionario. En la figura 2.1 se muestra un sistema típico de compresión de datos que muestra claramente en qué parte

se realiza la codificación y decodificación entrópica.

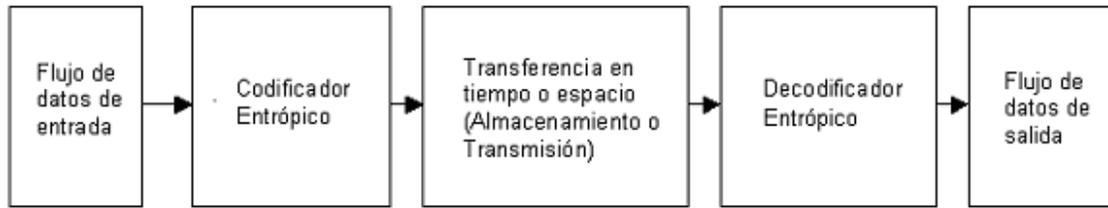


Figura 2.1: Un sistema típico de compresión de datos.

2.1.1. Codificación por largo de recorrido

La codificación por largo de recorrido es una de las técnicas más simples de compresión de datos que aprovecha los datos repetitivos. En el caso de las imágenes, este tipo de codificación puede tomar ventaja cuando se trata de imágenes con áreas de color constante. A los caracteres repetidos se les denominan corridas o *runs*. La técnica de codificación es simple, cada corrida se representa con una cuenta de los octetos de datos originales [13]. Si la cadena origen está dada por:

AAAABBBBBBCCCCCCCCDEEEE

ésta puede representarse por

4A5B8C1D4E

2.1.2. Codificación de Huffman

En 1952, David Huffman publicó un artículo que describía un método de codificación novedoso conocido en la actualidad como la Codificación de Huffman [16]. Se trata de un esquema de compresión óptimo de codificación de longitud variable sin pérdidas. En esa codificación los símbolos más frecuentemente usados son representados con la menor cantidad de bits. La descripción del mensaje resulta más corta y en sentido general no existe otro esquema que produzca una representación más corta siempre que exista la restricción que a cada símbolo debe asignársele una palabra de código discreta. Esta técnica fue el estado de arte hasta casi 1977. El concepto fundamental es que la longitud del carácter codificado es inversamente proporcional a la frecuencia del mismo. Huffman no fue el primero en descubrir esto pero presentó un algoritmo óptimo para asignar los códigos [17]. Los Códigos de Huffman son creados

analizando el conjunto de datos y asignando pocos bits a los datos que ocurren con mayor frecuencia [18]. El algoritmo intenta crear códigos que minimicen el número promedio de bits por carácter.

Durante el proceso de creación de los códigos se crea un árbol binario, esto asegurará que los códigos tengan una propiedad prefija. El primer paso para crear los códigos de Huffman es crear un arreglo de frecuencias de caracteres. El árbol binario puede construirse de manera recursiva agrupando los caracteres de menor frecuencia. El algoritmo 1 muestra el procedimiento.

Algoritmo 1 Construcción del árbol binario

Require: Ordenación de caracteres en forma decreciente según frecuencia de aparición en el mensaje.

- 1: Todos los caracteres inicialmente se consideran nodos libres.
 - 2: Los dos nodos libres con menor frecuencia se asignan a un nodo padre con un peso igual a la suma de los dos nodos hijos.
 - 3: Los dos nodos hijos se eliminan de la lista de nodos libres. El nodo nuevo se agrega a la lista.
 - 4: Se repiten los pasos 2 y 3 hasta que haya un solo nodo libre a la izquierda. Este nodo libre es la raíz del árbol.
 - 5: Finalmente los códigos para cada hoja del árbol se obtienen recorriéndolo desde la raíz y concatenando un 0 por cada movimiento hacia la izquierda o un 1 por cada movimiento hacia la derecha.
-

2.1.3. Codificación aritmética

La técnica de codificación aritmética ha emergido en la última década como una herramienta importante de compresión usada ampliamente para la codificación adaptativa en alfabetos multi-símbolos, gracias a su velocidad, bajos requerimientos de almacenamiento y eficiencia en la compresión [19]. Ideada por Rissanen y Langdon en la década de los 70 [20], esta técnica fue tomada, por varios años, más como una curiosidad que como una técnica aplicable a la codificación y compresión de datos. Comparada con la técnica de Huffman, resultaba menos eficiente para el tratamiento de alfabetos compuestos por muchos símbolos. En el año 1987 Witten [21] propone una implementación del codificador aritmético multi-símbolo que hace que la técnica emerja nuevamente como una solución a los problemas de compresión. La codificación aritmética toma la idea de reemplazar un símbolo de entrada con un código específico. Reemplaza un flujo de símbolos de entrada en un solo número de punto flotante. Se requieren más bits conforme el mensaje es más complejo. La salida de un proceso

de codificación aritmética es un solo número entre 0 y 1. Este número se utiliza en la decodificación para recuperar de manera única el flujo original de símbolos de entrada. La codificación aritmética, al igual que Huffman, es un algoritmo de dos pasos. El primero calcula la frecuencia de caracteres y genera una tabla de probabilidad. El segundo paso hace la compresión. La tabla de probabilidad asigna un rango entre 0 y 1 a cada carácter de entrada. El tamaño de cada rango es directamente proporcional a la frecuencia de caracteres. El orden de asignación de esos rangos no es importante pero debe ser utilizado en el codificador y el decodificador. El rango consiste de un valor alto y un valor bajo. Estos parámetros son muy importantes para el proceso de codificación/decodificación. A los caracteres de mayor ocurrencia se les asigna un rango mayor en el intervalo requerido. A los de menor ocurrencia se les asigna rangos menores, requiriendo más bits. En la codificación aritmética, se comienza con un rango de 0.0 a 1.0 El primer carácter de entrada generará el número de salida con su rango correspondiente. El rango del siguiente carácter fija el siguiente número de salida. Entre más sean los caracteres de entrada más preciso será el número de salida. La codificación aritmética requiere un elevado número de operaciones aritméticas de punto flotante por lo que es significativamente más lenta que la codificación de Huffman [13] tanto en la codificación como en la decodificación. El algoritmo 2 muestra lo anterior.

Algoritmo 2 Codificación aritmética

Require: Cálculo de la frecuencia de aparición de cada símbolo en el mensaje. Se asigna un número comprendido en el intervalo de 0 a 1.

- 1: $LIMITE_SUPERIOR \leftarrow 1, LIMITE_INFERIOR \leftarrow 0$
 - 2: $RANGO \leftarrow LIMITE_SUPERIOR - LIMITE_INFERIOR$
 - 3: $LIMITE_INFERIOR \leftarrow LIMITE_INFERIOR + RANGO * probabilidad_acumulada_inferior(simbolo)$
 - 4: $LIMITE_SUPERIOR \leftarrow LIMITE_INFERIOR + RANGO * probabilidad_acumulada_superior(simbolo)$
 - 5: Se codifica el símbolo en cuestión, asignándole un código (un número cualquiera) comprendido en el $RANGO$ actual.
 - 6: Se repiten los pasos 2 al 5, para cada símbolo del mensaje de entrada.
-

2.1.4. Codificación Lempel Ziv Welch (LZW)

En 1977, Abraham Lempel y Jacob Ziv propusieron un algoritmo de compresión de datos denominado LZ77. Mientras que los códigos de Huffman proporcionaron buenos resultados, éstos se limitaron a codificar un carácter a la vez. Lempel y Ziv

propusieron un esquema de diccionario (adaptativo) para codificar cadenas de datos. Este algoritmo tomó ventaja de la secuencia de caracteres que ocurren frecuentemente como la palabra “the” del idioma inglés. En 1984 Terry Welch publicó para la IEEE el algoritmo LZW (Lempel Ziv Welch), una variante del algoritmo LZ77. En la codificación LZW se busca reemplazar cadenas de caracteres con códigos simples que son almacenados en un diccionario o tabla de cadenas [13]. El algoritmo no analiza para nada el texto de entrada; únicamente añade cada nueva cadena a una tabla de cadenas de caracteres. El código que genera el algoritmo puede ser de cualquier longitud arbitraria, pero debe tener más bits que un carácter simple. Los primeros 256 códigos (cuando se usan caracteres de 8 bits) se asignan por omisión al conjunto estándar de caracteres. Los restantes son asignados a cadenas a medida que el algoritmo realiza su trabajo.

La porción de pseudo-código que aparece en el algoritmo 3 ilustra lo anterior.

Algoritmo 3 Proceso de codificación LZW

Require: Mensaje formado con caracteres

```

1: STRING ← Obtener caracter de entrada.
2: while existan caracteres de entrada do
3:   CHARACTER ← Obtener caracter de entrada.
4:   if STRING + CHARACTER está en la tabla then
5:     STRING ← STRING + CHARACTER.
6:   else
7:     Enviar como salida el código de STRING.
8:     Agregar STRING + CHARACTER a la tabla.
9:     STRING ← CHARACTER
10:  end if
11: end while
12: Enviar como salida el código de STRING

```

En la descompresión LZW se crea la misma tabla de cadenas y se actualiza por cada carácter en el flujo de entrada excepto para el primero. Después de que el carácter ha sido expandido a su cadena correspondiente vía la tabla de cadenas, el carácter final de la cadena es agregado a la cadena previa. Esta nueva cadena se agrega a la tabla en la misma localidad que en la tabla de cadenas del compresor.

2.2. Compresión de imágenes digitales

2.2.1. Definición de imagen digital

El término imagen se refiere a una función bidimensional de intensidad de luz $f(x, y)$, donde x y y denotan las coordenadas espaciales y el valor de f en cualquier punto (x, y) es proporcional al brillo (o nivel de gris) de la imagen en ese punto [22].

Una imagen digital es una imagen $f(x, y)$ que ha sido discretizada en coordenadas espaciales y en brillo. Una imagen digital puede considerarse como una matriz cuyos índices del renglón y columna identifican un punto en la imagen y el correspondiente valor del elemento de la matriz que identifica el nivel de intensidad de luz en ese punto. Los elementos de tal arreglo digital son llamados elementos de imagen, elementos de pintura o píxeles. Para el caso de las imágenes en color es necesario incluir un modelo para el color, que se definirá más adelante.

Para trabajar con números en la computadora, el nivel de brillo, o valor de cada pixel, es cuantizado a códigos binarios enteros positivos. El número de niveles de cuantización está determinado por la relación 2.2.

$$L = 2^B \quad (2.2)$$

donde B representa el número de bits necesarios para cada muestra. Esto es, con 6 bits de cuantización en la intensidad se pueden representar 64 niveles de gris ($2^6 = 64$). Para tener una imagen de buena calidad para el ojo humano es necesario tener como mínimo 64 niveles de gris [23]. Para una apreciación fina se usa una cuantización estándar a 8 bits, esto es, 256 niveles de gris. A una imagen cuantizada de esta manera se le conoce como imagen en tonos de gris. Una imagen binaria es una imagen monocromática cuantizada a 1 bit por pixel, esto es, dos niveles, blanco ó negro.

De igual forma existen dos esquemas de compresión de imágenes: *con pérdida* y *sin pérdida*. Un esquema sin pérdida codifica y decodifica los datos perfectamente, y la imagen resultante coincide exactamente con la original. No hay degradación en el proceso, ni hay pérdida de datos. En el esquema con pérdida se permite que la información redundante y no esencial se pierda. Típicamente en los esquemas con pérdida existe un justo medio entre compresión y calidad de la imagen. Es posible comprimir una imagen a un tamaño increíblemente pequeño pero con muy poca calidad. Aunque no siempre es el caso, los esquemas de compresión con pérdida son más complejos y requieren más cálculos que los esquemas sin pérdida. Estos eliminan datos de la imagen que el ojo humano no puede detectar. La técnica de compresión

a usar depende de los datos de la imagen. Algunas imágenes con un alto grado de detalle, tales como dibujos CAD y especialmente aquellas usadas en diagnósticos médicos no deben ser comprimidas con un algoritmo con pérdida. Los gráficos generados por computadora con grandes áreas del mismo color son comprimibles con algoritmos simples sin pérdida como la codificación por largo de recorrido o LZW. Sin embargo, las imágenes de tono continuo con contornos complejos y sombras requieren una técnica con pérdida para alcanzar una alta razón de compresión. Cuando se elige una técnica de compresión se deben considerar varios factores:

- La razón de compresión.
- El tiempo de compresión/descompresión.
- Complejidad del algoritmo.
- Costo y disponibilidad de recursos computacionales.

Transformaciones lineales

Las transformaciones lineales son la base de muchas técnicas usadas en el procesamiento de imágenes. Tradicionalmente, los codificadores basados en transformaciones lineales son divididos en dos categorías: codificadores de transformada y codificadores de sub-banda. Las técnicas de codificación de transformada están basadas en transformaciones lineales ortogonales. El ejemplo clásico es la Transformada de Fourier Discreta que descompone una señal en componentes de frecuencia sinusoidales. Otros dos ejemplos son la Transformada Coseno Discreta y la Transformada Karhunen-Loeve. Conceptualmente, estas transformadas son calculadas tomando el producto interno de una señal de longitud finita con un conjunto de funciones base ortogonales generando un conjunto de coeficientes que serán cuantizados posteriormente [24]. Las transformaciones de sub-banda son generalmente calculadas convolucionando la señal de entrada con un conjunto de filtros (pasa bajas, pasa altas o pasa banda) y luego son *sub-muestreadas* de tal manera que cada señal filtrada codifica una porción del espectro de frecuencia. En la reconstrucción, las señales de sub-banda son *sobre-muestreadas*, filtradas y combinadas aditivamente. La transformación de sub-banda puede ser utilizada para controlar el error en diferentes partes del espectro de frecuencia.

2.2.2. Cuantización

Para incrementar el número de ceros y reducir la magnitud de los coeficientes de transformación se utiliza un cuantizador [25]. Existen varias técnicas de cuantización [26]: lineales y no lineales. En la práctica conforme se incrementa la eficiencia del cuantizador también lo hace su complejidad y costo computacional. Otros tipos de cuantizadores son, la cuantización escalar y la cuantización vectorial. En la cuantización escalar, cada símbolo de entrada es tratado separadamente produciendo la salida, mientras que en la cuantización vectorial los símbolos de entrada son agrupados en segmentos llamados vectores y procesados para generar la salida. La agrupación de los datos y el tratamiento de los mismos como una unidad simple incrementa la optimalidad del cuantizador vectorial, pero también incrementa su complejidad computacional. La diferencia entre una entrada sin cuantizar y una salida cuantizada se denomina error de cuantización o ruido.

En la ecuación 2.3 se ilustra en ejemplo de cuantizador lineal.

$$ValorCuantizado = \left\lfloor \frac{ValorCoeeficiente \pm \frac{Q}{2}}{Q} \right\rfloor \quad (2.3)$$

donde Q : es el tamaño del paso de cuantización, de \pm : se toma en cuenta el signo negativo para un coeficiente menor que cero y el positivo en caso contrario.

El paso inverso a la cuantización es conocido como descuantización, expresado en la ecuación 2.4.

$$ValorRecuperadoCoeeficiente = Q * ValorCuantizado \quad (2.4)$$

Nótese que no se trata de una función inversa, puesto que el valor original del coeficiente no es en la mayoría de los casos igual al valor recuperado. El factor Q controla la cuantización. Un factor Q elevado proporciona mayor compresión. Un factor Q pequeño mejora la calidad de la imagen. La diferencia entre una entrada sin cuantizar y una salida cuantizada se denomina error de cuantización o ruido.

2.2.3. Espacios de color

Un modelo de color o espacio de color es una forma de representar colores y las relaciones entre ellos [18]. De acuerdo a la teoría tricromática [20], la sensación de color es producida por la excitación de tres receptores en el ojo humano. Existen dos formas básicas de producir esta excitación: por adición y sustracción de color.

En un dispositivo de adición de color como es el Tubo de Rayos Catódicos (TRC) el color se produce excitando tres fósforos: uno por cada color primario (rojo, verde, azul), de tal manera que la luz emitida estimula los tres tipos de receptores del ojo produciendo la percepción de color. Los sistemas de procesamiento de imagen usan diferentes modelos de color [27]. Los monitores de color con TRC y muchos sistemas gráficos de computadora usan el modelo RGB (Red, Green, Blue), los sistemas que manipulan tonos, saturación e intensidad separadamente usan el modelo Hue-Saturation-Intensity (HSI).

El modelo RGB

Este modelo consiste de los tres colores primarios rojo, verde y azul. El modelo RGB es representado por un cubo con un color primario en cada eje. El modelo RGB simplifica el diseño de sistemas gráficos de computadora pero no es ideal para todas las aplicaciones. Muchas veces es necesario convertir una imagen RGB a una imagen en escala de gris, para ello se usa la ecuación 2.5.

$$\text{IntensidadEscalaGris} = 0.299R + 0.587G + 0.114B \quad (2.5)$$

tomada del estándar NTSC para la luminancia.

El modelo YCbCr

YCbCr es otro modelo que separa la componente de luminancia de la información de color. La luminancia es codificada en la componente Y y la componente de azul y rojo en las componentes CbCr.

La conversión del espacio RGB al espacio YCbCr, se indica en la siguiente expresión:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.29900 & 0.58700 & 0.11400 \\ -0.16874 & -0.33126 & 0.50000 \\ 0.50000 & -0.41868 & -0.08131 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

y para convertir de YCbCr a RGB se utiliza:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.00000 & 0.00000 & 1.40200 \\ 1.00000 & -0.34414 & -0.71414 \\ 1.00000 & 1.77200 & 0.00000 \end{pmatrix} \begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix}$$

2.2.4. El estándar JPEG

JPEG es una familia de técnicas de compresión estandarizadas por el Grupo de Expertos Fotográficos Unidos. Este consorcio surge de los esfuerzos cooperativos de la ISO (*International Organization for Standardization*), CCITT (*International Telegraph and Telephone Consultative Committee*) e IEC (*International Electrotechnical Commission*) [27]. El resultado fue el primer estándar de compresión digital de imágenes. En el desarrollo de este estándar se persiguieron los siguientes objetivos:

- La razón de compresión debería ser especificada por el usuario.
- El estándar podría aplicarse a imágenes de tono continuo.
- La complejidad computacional podría ser especificada por el usuario.
- Se presentan cuatro modos de codificación.
 1. Secuencial.- Hace uso de un solo muestreo.
 2. Progresiva.- Hace uso de múltiples muestreos.
 3. Sin pérdida.- La imagen decodificada sería un duplicado exacto de la original.
 4. Jerárquica.- Se generan imágenes en múltiples resoluciones para mostrarlas en dispositivos diferentes.

JPEG tiene tres modos de compresión con pérdida y uno en modo sin pérdida. JPEG toma ventaja de la limitación del ojo humano, el cual puede percibir cambios pequeños de brillantez más que cambios pequeños en color. Esto permite a JPEG eliminar parte de la información de color. Cuando se comprime con JPEG se puede especificar un nivel de calidad Q para la imagen resultante. Un valor alto de Q representa una alta calidad de imagen pero también poca compresión y viceversa.

Conforme la calidad de la imagen disminuye será posible apreciar una estructura de bloques en la imagen. Cada cuadro corresponde al valor promedio de los píxeles que lo componen. JPEG es considerado un algoritmo simétrico puesto que la compresión y descompresión coinciden en el número de operaciones, como se muestra en la figura 2.2, y por tanto en el tiempo de compresión y descompresión.

El esquema típico de JPEG es el *baseline* que consiste de cinco pasos básicos:

Algoritmo 4 Esquema típico de JPEG

Require: Imagen de entrada.

- 1: La imagen se transforma al espacio de luminancia/crominancia YCbCr.
 - 2: Opcionalmente se reducen las componentes de color.
 - 3: La imagen se divide en bloques de 8x8 píxeles y se aplica la Transformada Coseno Discreta (TCD) a cada uno de ellos. La TCD asume que los datos son de periodo $2N$ mientras que la Transformada de Fourier Discreta (TFD) asume que los datos son reflejados para obtener un periodo $2N$.
 - 4: Se cuantizan los coeficientes resultantes de la TCD.
 - 5: Se aplica un codificador entrópico a los coeficientes de transformación.
-

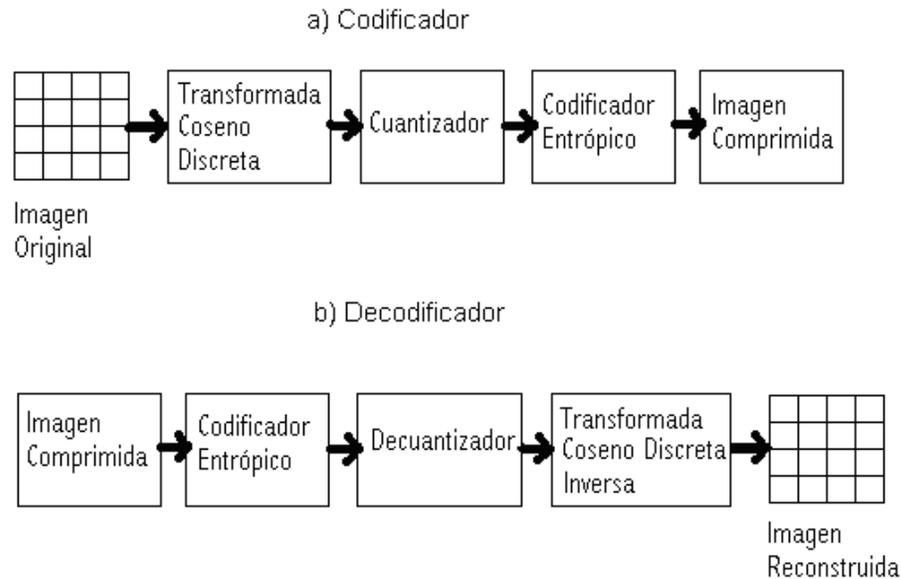


Figura 2.2: JPEG Baseline a) Codificador b) Decodificador.

La razón para separar las componentes de luminancia/crominancia es porque se elimina más información de las componentes de crominancia. El segundo paso es opcional pero estándar en la práctica. Mientras que las componentes de luminancia no se alteran, las componentes de color son sub-muestreadas (1 de cada 2) horizontal

y verticalmente. Este no es el único esquema de sub-muestreo pero es el más popular. El tercer paso consiste en separar las componentes de la imagen y dividir las en arreglos bidimensionales (cuadros) de 8×8 píxeles. Los elementos de cada arreglo son convertidos a enteros con signo. Los arreglos son transformados al dominio de la frecuencia vía la TCD. Al elemento $(0,0)$ de cada arreglo se le denomina componente de DC. Los otros 63 elementos son denominados componentes AC_{xy} , donde la pareja (x,y) es la posición dentro del arreglo. El cuarto paso consiste en cuantizar los coeficientes de transformación. Este es el paso fundamental de pérdida de la información ya que los coeficientes son divididos por su correspondiente coeficiente de cuantización y redondeado al entero más cercano. El valor de Q determina los coeficientes de cuantización. En este paso muchos coeficientes son redondeados a cero. No existen tablas fijas de cuantización.

El factor Q controla los valores de la tabla de cuantización. El quinto paso o codificación entrópica no produce pérdida alguna de información. Los arreglos de coeficientes son codificados mediante la técnica de Huffman. Los valores de DC son codificados como la diferencia del valor de DC del bloque previo. Esta codificación diferencial es posible porque existe una fuerte correlación entre los valores de DC adyacentes [27]. La figura 2.3 ilustra este fenómeno. Los coeficientes restantes son recorridos en orden decreciente a su probabilidad de ocurrencia mediante una técnica de zig-zag, de esta manera el codificador entrópico proporcionará mayores razones de compresión [27]. La técnica de zig-zag se muestra en la figura 2.4.

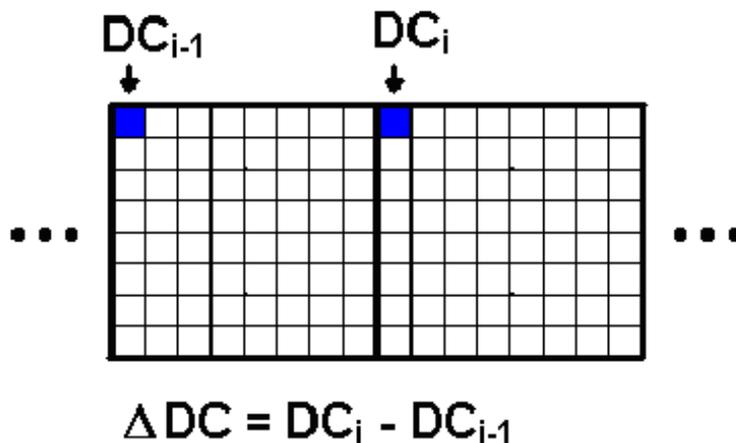


Figura 2.3: Codificación Diferencial.

JPEG también puede utilizar codificación aritmética como un método de codificación entrópica. La codificación entrópica alcanza entre un 5 y 10 por ciento más

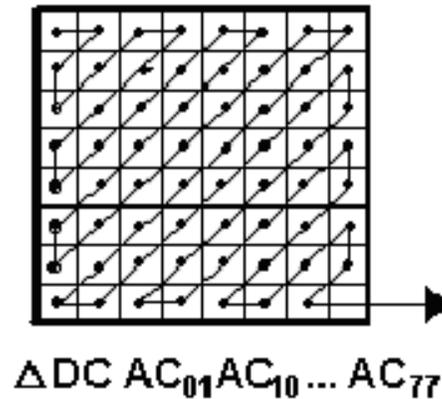


Figura 2.4: Secuencia de Zig-Zag de la codificación entrópica en JPEG.

de compresión que Huffman pero no se usa por dos razones fundamentales:

- Es más difícil de implementar que Huffman.
- Hay varias patentes sobre la codificación aritmética principalmente de AT&T e IBM.

JPEG2000

La siguiente versión del formato de JPEG denominada JPEG 2000 incluye atributos que tratan de tomar lo que la versión actual alcanza en lo que se refiere a las altas razones de compresión y solucionar el problema de degradación sustancial de la imagen debido a la técnica de compresión con pérdida empleada y a la TCD. Entre las nuevas características de esta versión se encuentran las siguientes:

- JPEG 2000 alcanza mayores razones de compresión que JPEG.
- JPEG 2000 reemplaza la TCD por la compresión basada en *wavelets*.
- Mientras que JPEG solo contiene datos del espacio RGB, JPEG 2000 permite manejar modelos de color alternativos como CMYK.
- Como beneficio adicional, a partir de un mismo archivo pueden obtenerse imágenes con diferentes resoluciones.

2.3. Comparación de las técnicas de compresión de imágenes

En las secciones anteriores se han presentado algunas de las técnicas de compresión y descompresión de imágenes digitales más representativas, por lo que en la figura 2.5 se resumen sus principales características. Cabe aclarar, que la mayoría de los algoritmos que se presentan son en realidad algoritmos de codificación de imágenes (o de datos), que no representan a todo el proceso de compresión de datos; sin embargo, constituyen uno de los bloques más importantes del proceso de compresión y descompresión de imágenes. En dicha tabla se presentan características de los algoritmos tales como: la cantidad de memoria que requieren para su operación (una cantidad estimada, baja, media, alta), las razones o tasas de compresión típicas que se consiguen mediante cada algoritmo (baja, media, alta), si su cálculo requiere de efectuar operaciones complejas o que demanden alto poder de procesamiento, como las operaciones entre datos de punto flotante, si presentan un comportamiento aceptable únicamente para imágenes con pocos detalles o con tonos continuos (bajas frecuencias), o incluso para imágenes que contienen muchos detalles (altas frecuencias), si forman parte de técnicas de compresión o descompresión de imágenes con pérdida o sin pérdida de información o ambos casos. Las últimas dos características que se incluyen se refieren a la capacidad de codificación de regiones o secciones de interés de una imagen (CRI) y el manejo de errores durante el proceso de codificación. Estas últimas características son encontradas en técnicas más recientes para la compresión de imágenes, tales como en las herramientas *wavelets*. De la tabla podemos destacar algunos casos como la codificación de Huffman, que en promedio requiere una mayor cantidad de memoria que las otras técnicas, debido a la construcción del árbol binario (con lo que se asegura que los códigos tengan una propiedad prefija), considerada como una operación compleja. Otro caso relevante es el de la codificación aritmética, que si bien no requiere una cantidad alta de memoria para su operación, requiere del manejo de operaciones con datos de punto flotante, lo que demanda un poder de procesamiento extra que si no se tratarán datos de esta naturaleza.

Cabe destacar que las técnicas de compresión y descompresión de datos (o imágenes) basadas en las transformadas *wavelets*, presentan todas las características estudiadas. Por lo general, éstas no requieren grandes cantidades de memoria, y con ellas se logran tasas de compresión altas. La complejidad de sus operaciones en algunos casos son simples y en otros complejas. Es posible conformar técnicas de compresión con/sin

pérdida de información y de tipo asimétricos (que la complejidad de compresión sea diferente a la de descompresión). Todas estas características las convierten en una muy buena alternativa para implantarse en ambientes de cómputo móvil, en donde la mayoría de los dispositivos móviles ligeros poseen capacidades muy limitadas, tales como velocidad de procesamiento y cantidad de memoria, entre otras.

Algoritmo / Características	LZW	Huffman	RLE	Aritmética	Gzip	Basados en TDC	Basada en <i>wavelets</i>
Memoria.	Baja	Alta.	Muy baja	Media	Baja	Media	Baja
Tasa compresión.	Media	Media	Baja-media	Baja-media	Media	Media-alta	Alta
Operaciones complejas.	No	Construcción árbol	No	Punto flotante	No	Algunos casos	Algunos casos
Tipo imágenes	Colores uniformes.	Colores uniformes	Colores uniformes	Tonos continuos	Colores uniformes	No tan importante	Casi cualquier tipo.
Codificación de regiones interés (ROI)	NA	NA	NA	NA	NA	NA	Si
Manejo de errores	NA	NA	NA	NA	Si	NA	Si
Con pérdida de información	NA	NA	NA	NA	NA	Si	Si
Sin pérdida de información	Si	Si	Si	Si	Si	Si	Si

Figura 2.5: Tabla de algoritmos de compresión y descompresión de imágenes.

En este capítulo se explicaron los conceptos relativos a la compresión de imágenes y en general de datos, así mismo se hizo una revisión de las técnicas clásicas más representativas para la compresión de imágenes e introduciendo al estándar JPEG2000, basado en las transformadas *wavelets*, consideradas como herramientas del procesamiento digital moderno de señales. En el siguiente capítulo se presenta un estudio detallado de las técnicas de compresión de imágenes basadas en *wavelets*.

Capítulo 3

Compresión de Imágenes Basada en *Wavelets*

3.1. Introducción

En años recientes, muchos de los esfuerzos orientados a la codificación de imágenes han sido enfocados a las transformadas *wavelets*, las cuales se han convertido en una herramienta estándar, debido a su gran capacidad de compresión de datos [29, 30]. La TWD (Transformada *Wavelet* Discreta) es similar a la TCD (Transformada Coseno Discreta) en cuanto a la descomposición de la señal original en coeficientes asignados a funciones base [37].

La TCD está limitada a funciones coseno que requieren bastantes operaciones aritméticas. Por otra parte, la teoría de *wavelets* usa un rango amplio de funciones base dando como resultado una menor complejidad computacional sin sacrificar la calidad de la imagen. Es importante definir los siguientes conceptos del álgebra lineal, para introducir formalmente el concepto de *wavelets*:

Definición. Un espacio vectorial real V es un conjunto de objetos llamados vectores, junto con dos operaciones llamadas suma y multiplicación por un escalar en R que satisfacen los diez axiomas enumerados a continuación:

1. Si $x \in V$ y $y \in V$, entonces $(x + y) \in V$ (cerradura bajo la suma).
2. $\forall x, y, z \in V, (x + y) + z = x + (y + z)$ (ley asociativa de la suma de vectores).
3. $\exists \vec{0} \in V \mid \forall x \in V, x + \vec{0} = \vec{0} + x = x$ ($\vec{0}$ es el idéntico aditivo).
4. Si $x \in V, \exists (-\vec{x}) \mid x + (-\vec{x}) = 0$ ($(-\vec{x})$ se llama el inverso aditivo de \vec{x}).

5. Si $x, y \in V \rightarrow x + y = y + x$ (ley conmutativa de la suma de vectores).
6. Si $x \in V$ y a es un escalar, entonces $ax \in V$ (cerradura bajo la multiplicación de un escalar).
7. Si $x, y \in V$ y a es un escalar, entonces $a(x + y) = ax + ay$ (primera ley distributiva).
8. Si $x \in V$ y a, b son escalares, entonces $(a+b)x = ax + bx$ (segunda ley distributiva).
9. Si $x \in V$ y a, b son escalares, entonces $a(bx) = abx$ (ley asociativa de la multiplicación por un escalar).
10. $\forall x \in V, 1x = x$ (el escalar 1 se llama idéntico multiplicativo).

Definición. Un subespacio no vacío H de un espacio vectorial V es un subespacio de V si cumple:

1. Si $x, y \in H$, entonces $x + y \in H$
2. Si $x \in H$, entonces $ax \in H$ para todo escalar a .

Sea H un subconjunto no vacío de un espacio vectorial V y suponga que H es en sí un espacio vectorial bajo las operaciones de suma y multiplicación por un escalar definidas en V , entonces se dice que H es un subespacio de V y se denota por $H \subset V$.

Definición. Sean v_1, v_2, \dots, v_n , n vectores en un espacio vectorial V . Entonces se dice que los vectores son linealmente dependientes si existen n escalares c_1, c_2, \dots, c_n no todos cero tales que $c_1v_1 + c_2v_2 + \dots + c_nv_n = 0$. Si los vectores no son linealmente dependientes se dice que son linealmente independientes.

Definición. Un conjunto finito de vectores v_1, v_2, \dots, v_n es una base para un espacio vectorial V si

1. (v_1, v_2, \dots, v_n) es linealmente independiente.
2. (v_1, v_2, \dots, v_n) genera a V .

Definición. Sean u, v dos vectores en R^n . Entonces el producto escalar de u y v , denotado por $\langle u, v \rangle$ está dado por $\langle u, v \rangle = u_1v_1 + u_2v_2 + \dots + u_nv_n$ donde las u_i, v_i son las componentes de los vectores u y v respectivamente.

Definición. Se dice que dos vectores u, v son ortogonales si $\langle u, v \rangle = 0$.

Dado un espacio vectorial U y un sub-espacio vectorial V , siempre existe un subespacio vectorial W tal que:

- Todo elemento de U puede ser escrito como una combinación lineal de un elemento en V y un elemento en W , denotado $U = V + W$.
- La representación anterior es única si y sólo si $U \cap W = \{0\}$, esto es llamado descomposición en suma directa de U y se escribe $U = V \oplus W$.

En general, W no es único, de hecho dado un espacio vectorial U y un subespacio vectorial V es posible escoger W tal que V y W sean ortogonales y se escribe $U = V \oplus W$

Una función de escalamiento Φ es tal que $\Phi(x)$ puede ser escrita como una combinación de $\Phi(2x-k)$, que es una versión de Φ trasladada y escalada. Matemáticamente se expresa así:

$$\Phi(x) = \sum_{k=-\infty}^{\infty} p_k \Phi(2x - k) \quad (3.1)$$

la cual es conocida como la ecuación de escalamiento. La secuencia $\{p_k\}$ es llamada la secuencia de escalamiento de Φ . Considérese,

$$\Phi_{j,k}(x) := \Phi(2^j x - k) \quad j, k \in \mathbb{Z} \quad (3.2)$$

una versión de Φ trasladada en k y escalada en $\frac{1}{2^j}$. Para una j fija se hace referencia al nivel j .

Sea V_j el espacio vectorial generado mediante traslaciones de Φ en el nivel j . Obsérvese que según la ecuación 3.1, $V_0 \subset V_1$. Más aún, se genera una secuencia de subespacios anidados V_j tales que

$$\dots V_{-1} \subset V_0 \subset V_1 \subset V_2 \dots \quad (3.3)$$

Proposición. Toda función suave en R se puede representar en términos de $\Phi_{j,k}$ para algún j suficientemente grande, en otras palabras

$$\text{clos}_{L^2} \left(\bigcup_{j \in \mathbb{Z}} V_j \right) = L^2(R) \quad (3.4)$$

Las propiedades presentadas en las ecuaciones 3.1-3.4 dan lugar a lo que se conoce como Análisis de Multi-Resolución (AMR).

Dada una secuencia anidada de subespacios V_j , existen subespacios W_j que son el complemento ortogonal de V_j en V_{j+1} esto es:

$$V_{j+1} = V_j \oplus W_j, j \in Z \quad (3.5)$$

donde,

$$W_j \perp W_{j'}, \quad j \neq j' \quad (3.6)$$

y puesto que los espacios están anidados según la ecuación 3.3, se tiene

$$V_J = V_j \oplus \bigoplus_{k=0}^{J-j-1} W_{j+k}, \quad j < J \quad (3.7)$$

Proposición. Dada una función de escalamiento Φ en V_j , existe otra función Ψ en W_0 llamada *wavelet* tal que $\{\Psi_{j,k} : k \in Z\}$ genera W_j , donde:

$$\Psi(x) = \sum_{k=-\infty}^{\infty} q_k \Phi(2x - k) \quad (3.8)$$

conocida como ecuación de escalamiento para la *wavelet*. $\{q_k\}$ es llamada la secuencia de escalamiento para la *wavelet*.

Con lo anterior se tiene que una *wavelet* es una función Ψ que genera un subespacio vectorial W_j el cual es el complemento ortogonal de espacios vectoriales V_j anidados $\dots V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots$ quienes generan un análisis de multiresolución (AMR).

3.2. Análisis de multiresolución

Las transformadas *wavelets* llevan a cabo el análisis de las imágenes con multiresolución [34]. El resultado del análisis de multiresolución es la representación de la imagen a diferentes niveles de resolución y calidad [35]. La resolución es determinada por un valor de umbral, debajo del cual todas las fluctuaciones o detalles son ignorados. Las diferencias entre dos resoluciones vecinas representan los detalles. Por lo tanto, una imagen puede ser representada por una imagen en baja resolución (la parte aproximada o promedio) y los detalles en cada nivel más alto de resolución. Si se considera una función unidimensional 1-D, $f(t)$. En el nivel de resolución j , la

aproximación de la función $f(t)$ es $f_j(t)$. En el próximo nivel de resolución $j + 1$, la aproximación de la función $f(t)$ es $f_{j+1}(t)$. Los detalles, denotados por $d_j(t)$, son incluidos en $f_{j+1}(t) : f_{j+1}(t) = f_j(t) + d_j(t)$. Este procedimiento puede ser repetido varias veces y la función $f(t)$ se puede expresar como en la ecuación 3.9.

$$f(t) = f_j(t) + \sum_{k=j}^n d_k(t) \quad (3.9)$$

3.3. Transformada *wavelet* discreta para 1D

Uno de los más grandes descubrimientos para el análisis *wavelet* fue que la reconstrucción perfecta de bancos de filtros puede ser hecha usando las secuencias de coeficientes h_0 y h_1 como se muestra en la figura 3.1.

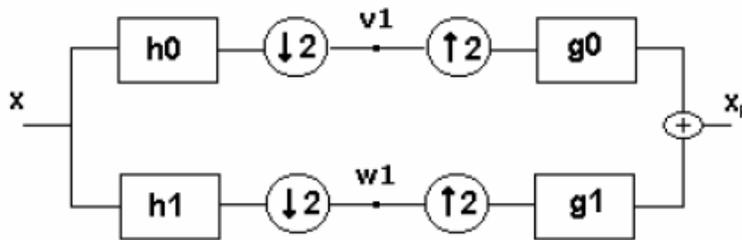


Figura 3.1: Banco de filtros de 2 canales.

La secuencia de entrada x es convolucionada con los filtros pasa-altas h_{1H} y pasa-bajas h_{0L} y submuestreada por dos unidades, resultando las señales transformadas V_1 y W_1 . La señal es reconstruida a través de un sobremuestreo y convolucionando con los filtros de síntesis g_{1H} y g_{0L} . Filtros diseñados correctamente permiten la reconstrucción exacta de la señal x ($x = x_p$).

La elección del filtro no sólo determina si la reconstrucción perfecta es posible, sino también determina la forma en que la *wavelet* es usada para llevar a cabo el análisis.

La aplicación de la TWD da una descomposición multinivel de la señal de entrada en componentes de alta y baja frecuencia en diferentes resoluciones de acuerdo con el número de niveles empleados. A continuación se definen matemáticamente $H(\omega)$ y $G(\omega)$:

$$H(\omega) = \sum_k h_k e^{-jk\omega}, \quad G(\omega) = \sum_k g_k e^{-jk\omega} \quad (3.10)$$

los cuales representan a un filtro pasa-bajas y pasa-altas respectivamente, satisfaciendo la condición de ortogonalidad:

$$|H(\omega)|^2 + |G(\omega)|^2 = 1 \quad (3.11)$$

necesaria para la reconstrucción de la transformada. Los filtros $G(\omega)$ y $H(\omega)$ se conocen también como filtros espejos en cuadratura. Una señal $x[n]$ (en 1D) puede ser descompuesta recursivamente, como a continuación se muestra:

$$c_{(j-1,k)} = \sum_n h_{n-2k} c_{j,n}, \quad d_{(j-1,k)} = \sum_n g_{n-2k} c_{j,n} \quad (3.12)$$

la cual corresponde a las convoluciones que son seguidas por un submuestreo por 2. Las variables h_k y g_k son la respuesta al impulso de los filtros pasa-bajos y pasa-altos respectivamente. El índice j expande el número de niveles de descomposición y va desde $[0, L + 1]$, donde $L + 1$ representa el índice del nivel de resolución alto y 0 representa el índice del nivel de resolución bajo. $c_{L+1,k}$ es igual a la señal de entrada $x[k]$. Los coeficientes $c_{o,k}, d_{o,k}, d_{1,k}, \dots, d_{l-1,k}, d_{l,k}$ son llamados los coeficientes de la TWD de $x[n]$. $c_{o,k}$ es la componente de mínima resolución de $x[n]$ que contiene el pasabajas y $d_{j,k}$ son los coeficientes de los detalles de $x[n]$ en varias bandas de frecuencias. La señal $x[n]$ puede ser reconstruida a partir de sus coeficientes TWD al considerar la fórmula recursiva:

$$C_{(j,n)} = \sum_k h_{n-2k} c_{j-1,k} + \sum_k g_{n-2k} d_{j-1,k} \quad (3.13)$$

La transformada *wavelet* discreta no tiene un conjunto de funciones básicas. Hay muchas familias de *wavelets*, las más conocidas son Haar, Daubechies y Coiflets, cuyas definiciones son las siguientes:

1. **Transformada *wavelet* Haar.** La transformada *wavelet* Haar está dada por:

$$W_1 = (\beta_1, \beta_2, 0, 0, \dots, 0) \quad (3.14)$$

$$W_2 = (0, 0, \beta_1, \beta_2, 0, 0, \dots, 0) \quad (3.15)$$

$$\vdots \quad (3.16)$$

$$W_{N/2} = (0, 0, \dots, 0, \beta_1, \beta_2) \quad (3.17)$$

y la respectiva señal de escalamiento:

$$V_1 = (\alpha_1, \alpha_2, 0, 0, \dots, 0) \quad (3.18)$$

$$V_2 = (0, 0, \alpha_1, \alpha_2, 0, 0, \dots, 0) \quad (3.19)$$

$$\vdots \quad (3.20)$$

$$V_{N/2} = (0, 0, \dots, 0, \alpha_1, \alpha_2) \quad (3.21)$$

donde,

$$\alpha_1 = \frac{1}{\sqrt{2}}, \alpha_2 = \frac{1}{\sqrt{2}}, \beta_1 = \alpha_2, \beta_2 = -\alpha_1,$$

2. Transformada *wavelet* Daubechies (D4). La transformada *wavelet* Daubechies (D4) está dada por:

$$W_1 = (\beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \dots, 0) \quad (3.22)$$

$$W_2 = (0, 0, \beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \dots, 0) \quad (3.23)$$

$$W_3 = (0, 0, 0, 0, \beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \dots, 0) \quad (3.24)$$

$$\vdots \quad (3.25)$$

$$W_{N/2-1} = (0, 0, \dots, \beta_1, \beta_2, \beta_3, \beta_4) \quad (3.26)$$

$$W_{N/2} = (\beta_3, \beta_4, 0, 0, \dots, 0, \beta_1, \beta_2) \quad (3.27)$$

y la respectiva señal de escalamiento:

$$V_1 = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, 0, 0, \dots, 0) \quad (3.28)$$

$$V_2 = (0, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, 0, 0, \dots, 0) \quad (3.29)$$

$$V_3 = (0, 0, 0, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, 0, 0, \dots, 0) \quad (3.30)$$

$$\vdots \quad (3.31)$$

$$V_{N/2-1} = (0, 0, \dots, \alpha_1, \alpha_2, \alpha_3, \alpha_4) \quad (3.32)$$

$$V_{N/2} = (\alpha_3, \alpha_4, 0, 0, \dots, 0, \alpha_1, \alpha_2) \quad (3.33)$$

donde,

$$\alpha_1 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \alpha_2 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \alpha_3 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \alpha_4 = \frac{1-\sqrt{3}}{4\sqrt{2}}, \beta_1 = \alpha_4, \beta_2 = -\alpha_3, \beta_3 = \alpha_2,$$

$$\beta_4 = -\alpha_1$$

3. Transformada *wavelet* Coiflets (C-6).

La transformada *wavelet* Coiflets (C6) está dada por:

$$W_1 = (\beta_3, \beta_4, \beta_5, \beta_6, 0, 0, \dots, 0, \beta_1, \beta_2) \quad (3.34)$$

$$W_2 = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, 0, 0, \dots, 0) \quad (3.35)$$

$$W_3 = (0, 0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, 0, 0, \dots, 0) \quad (3.36)$$

$$\vdots \quad (3.37)$$

$$W_{N/2} = (\beta_5, \beta_6, 0, 0, \dots, 0, \beta_1, \beta_2, \beta_3, \beta_4) \quad (3.38)$$

Y la respectiva señal de escalamiento:

$$V_1 = (\alpha_3, \alpha_4, \alpha_5, \alpha_6, 0, 0, \dots, 0, \alpha_1, \alpha_2) \quad (3.39)$$

$$V_2 = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, 0, 0, \dots, 0) \quad (3.40)$$

$$V_3 = (0, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, 0, 0, \dots, 0) \quad (3.41)$$

$$\vdots \quad (3.42)$$

$$V_{N/2} = (\beta_5, \beta_6, 0, 0, \dots, 0, \beta_1, \beta_2, \beta_3, \beta_4) \quad (3.43)$$

donde,

$$\alpha_1 = \frac{1-\sqrt{7}}{16\sqrt{2}}, \alpha_2 = \frac{5+\sqrt{7}}{16\sqrt{2}}, \alpha_3 = \frac{14+2\sqrt{7}}{16\sqrt{2}}, \alpha_4 = \frac{14-2\sqrt{7}}{16\sqrt{2}}, \alpha_5 = \frac{1-\sqrt{7}}{16\sqrt{2}}, \alpha_6 = \frac{-3+\sqrt{7}}{16\sqrt{2}},$$

$$\beta_1 = \alpha_6, \beta_2 = -\alpha_5, \beta_3 = \alpha_4, \beta_4 = -\alpha_3, \beta_5 = \alpha_2, \beta_6 = -\alpha_1$$

En las expresiones anteriores, N se refiere al número de elementos de un vector o señal f . La transformación *wavelet* directa (subseñal *promedio* y *fluctuación*) de una señal $f = (f_1, f_2, f_3, \dots, f_N)$ puede obtenerse mediante las ecuaciones 3.44 y 3.45.

Subseñal fluctuación:

$$d_m = f \cdot W_m \quad (3.44)$$

Subseñal promedio:

$$a_m = f \cdot V_m \quad (3.45)$$

para $m = 1, 2, \dots, N/2$.

La transformación *wavelet* inversa se obtiene mediante las ecuaciones 3.46a, 3.46b y 3.46c:

$$A = \sum_{m=1}^{N/2} a_m V_m \quad (3.46a)$$

$$D = \sum_{m=1}^{N/2} d_m W_m \quad (3.46b)$$

$$f = A + D \quad (3.46c)$$

El costo computacional para calcular cualquier familia de transformación *wavelet*, la podemos estimar:

Al aplicar el banco de filtros (de orden 2 para el caso de Haar, de orden 4 para el caso de Daubechies-4 y de orden 6 para el caso de Coiflets-6) a una señal de n muestras se requiere a lo mas kn operaciones (k =alguna constante), entonces el numero total de operaciones requeridas para calcular la transformada está dada por:

$$kn + k\frac{n}{2} + k\frac{n}{4} + \dots + k\frac{n}{2^{\log_2 n}} \leq kn \left(1 + \frac{1}{2} + \frac{1}{4} + \dots\right) \leq 2kn = \Theta(n)$$

3.4. Transformada *wavelet* discreta para 2D

La transformada *wavelet* discreta (TWD) de dos dimensiones (2D) y su transformada inversa (TWDI) se pueden obtener mediante la extensión de la transformada

para el caso de una dimensión (1D), ya que se implementan utilizando la TWD y TWDI junto con cada dimensión n y m por separado, como se muestra en la siguiente expresión:

$$TWD_{n \times m}[x[n, m]] = TWD_n[TWD_m[x[n, m]]] \quad (3.47)$$

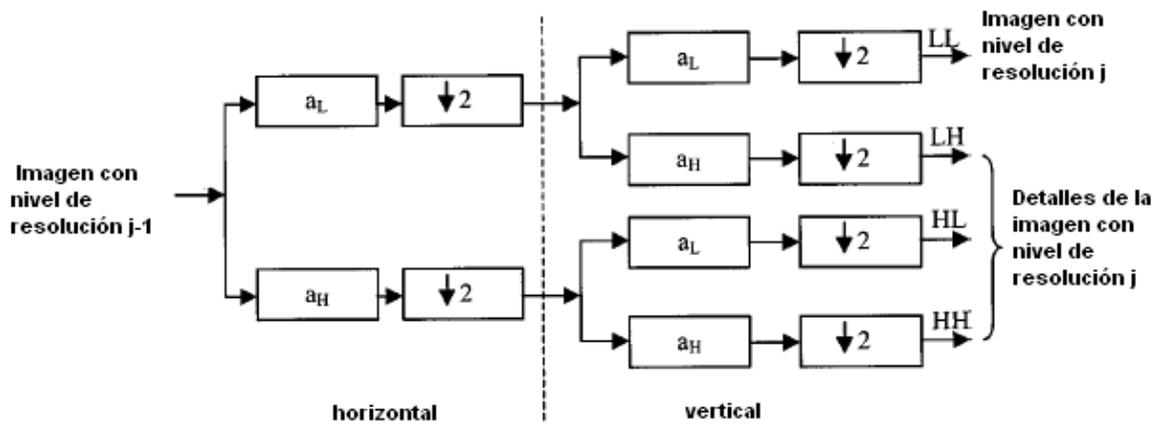
Es decir, la función de escalamiento puede ser obtenida multiplicando dos funciones de escalamiento de 1-D: $\Phi(x, y) = \Phi(x)\Phi(y)$. Las funciones *wavelets* pueden ser obtenidas multiplicando dos funciones *wavelets* o una función *wavelet* con una de escalamiento en el análisis 1-D. Para el caso de 2-D, existen tres funciones *wavelets*, las que exploran los detalles horizontales $\Psi^I(x, y) = \Phi(x)\Psi(y)$, los detalles verticales $\Psi^{II}(x, y) = \Psi(x)\Phi(y)$ y los detalles diagonales $\Psi^{III}(x, y) = \Psi(x)\Psi(y)$. Esto se puede representar como un banco de filtros con reconstrucción perfecta de cuatro canales como se muestra en la figura 3.2 (a). Las cuatro componentes resultantes de la transformación consisten en todas las posibles combinaciones de filtrado pasa-altas y pasa-bajas en las dos direcciones. Usando estos filtros una imagen puede ser descompuesta en cuatro bandas. Para cada resolución se tienen tres tipos de detalle de la imagen: horizontal (HL), vertical (LH) y diagonal (HH). Esta operación puede ser repetida sobre la banda LL, aplicando la transformación *wavelet* para un segundo nivel. En un sistema de compresión de imágenes conformado por la transformación *wavelet* discreta en 2-D, se puede generar la estructura piramidal mostrada en la figura 3.2 (b). En el esquema, el número de descomposiciones (j), describe el número de etapas de filtrado usadas en la descomposición de la imagen.

En el dominio espacial, la imagen puede ser considerada como una composición de información en diferentes escalas. Una transformación *wavelet* mide las variaciones de los niveles de gris de la imagen a diferentes escalas. En el dominio de la frecuencia, la función de sensibilidad del contraste del sistema de visión humana depende de la frecuencia y orientación de los detalles.

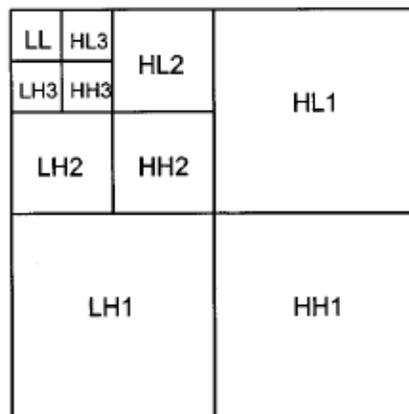
3.5. Compresión de imágenes

En un sistema de compresión de imágenes basado en *wavelets*, la imagen completa es transformada y comprimida como un solo objeto de datos a diferencia de bloque por bloque como en el caso de los sistemas basados en la TCD. Esto permite que se presente una distribución uniforme del error de compresión en toda la imagen.

El esquema de compresión y descompresión de imágenes digitales basado en trans-



(a)



(b)

Figura 3.2: (a) Un filtro DWT en el caso de una señal en 2-D. (b) Estructura piramidal de la descomposición *wavelet*.

formadas *wavelet* discreta se muestran en las figuras 3.3 y 3.4.

Un sistema de compresión de imágenes basado en *wavelets* puede ser creado seleccionando un tipo de función *wavelet*, un cuantizador y un codificador entrópico. Los cuantizadores para los sistemas basados en la TCD y los basados en *wavelets* deben ser adaptados a la estructura de la transformación, ya que existen diferencias entre la TCD y las *wavelets*. Por ejemplo, el cuantizador uniforme es el cuantizador base para el estándar JPEG y el cuantizador *zerotree* es el indicado para su uso con las transformadas *wavelets*[31, 32].

La transformada *wavelet* representa una imagen como una suma de funciones *wavelets*

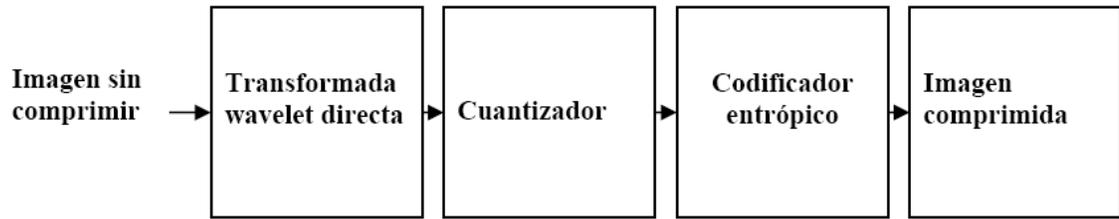


Figura 3.3: Diagrama a bloques del esquema de compresión de imágenes basado en la transformada *wavelet*.

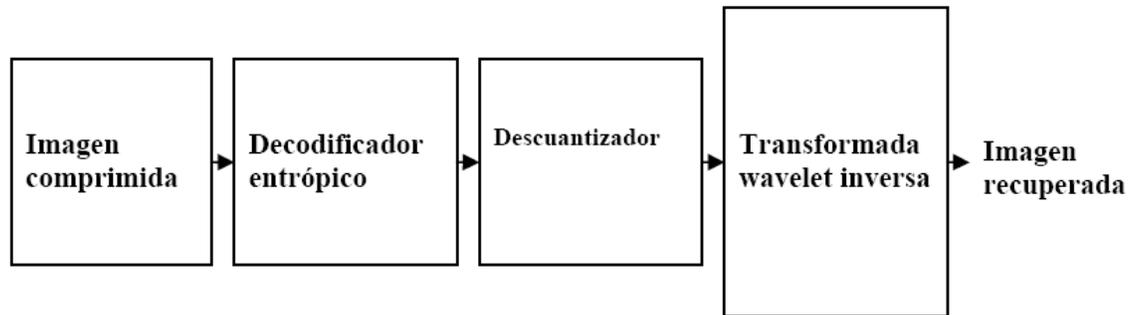


Figura 3.4: Diagrama a bloques del esquema de descompresión de imágenes basado en la transformada *wavelet*.

con diferentes localizaciones y escalas [33]. La descomposición de una imagen en *wavelets* involucra un par de formas de onda (o señales): una que representa las altas frecuencias y que corresponden a los detalles de la imagen (función *wavelet* Ψ) y otra para las bajas frecuencias o partes lisas de la imagen (función de escalamiento Φ).

El primer esquema ilustra el proceso de compresión de imágenes digitales y que consiste en:

1. Se aplica la transformada *wavelet* a la imagen de entrada (sin comprimir) . Este paso tiene como objetivo principal lograr la descorrelación de los píxeles de la imagen. Dependiendo del tratamiento que se le den a los datos obtenidos, este paso puede ser con pérdida, es decir que parte de los valores que se obtienen son despreciados bajo algún criterio por lo que no se asegura la reconstrucción (descompresión) exacta de la imagen que se tenía antes de comprimirla. Sin embargo pueden tomarse en cuenta la totalidad de los valores obtenidos por lo que se trataría de un proceso sin pérdida.
2. A la salida de la transformación anterior, se aplica el cuantizador, cuya finalidad

es la de reducir la cantidad de bits necesarios para representar cada píxel en la imagen de salida. En la mayoría de los casos esta etapa del proceso con lleva pérdida de información, lo que no garantiza que se pueda recuperar fielmente la imagen que se tenía antes del proceso de compresión. Mediante este proceso se aproxima el conjunto de valores continuos de una imagen a un conjunto finito (preferiblemente pequeño) de valores. En este proceso de aproximación, un buen cuantizador es aquel que representa a la señal original con pérdida mínima o poca distorsión.

3. La última etapa del proceso de compresión refiere al uso de un codificador entrópico, el que aprovecha la redundancia de información existente en los datos provenientes del proceso anterior para crear secuencias de bits mas cortas. El principio de su funcionamiento es, utilizar menos bits para las secuencias que aparecen con mayor frecuencia logrando así la compresión de datos. Puesto que es posible recuperar la señal original que se tenía antes de hacer uso del codificador se trata de un proceso sin pérdida de información. Algunos ejemplos de codificadores entrópicos son, Huffman, LZW, aritmético, etc.

Para el caso de la descompresión, es necesario aplicar el proceso inverso, es decir, dada la imagen comprimida se aplicaría primeramente el decodificador entrópico, enseguida el decuantizador y por último la transformada *wavelet* en su forma inversa.

En el caso de las técnicas de compresión y descompresión de imágenes digitales basadas en las transformadas wavelets, el proceso de aplicación de la transformada *wavelet* puede resumirse en lo siguiente:

Si se considera una imagen de tamaño $N \times M$.

1. Al aplicar la transformada *wavelet* a la imagen de entrada, se obtienen cuatro subimágenes de tamaño $\frac{N}{2} \times \frac{M}{2}$. En donde una de las subimágenes corresponde a una versión reducida de la imagen (subimagen promedio o *trend*) y es la que podría transmitirse hacia otro dispositivo, despreciando las tres subimágenes restantes (también de tamaño $\frac{N}{2} \times \frac{M}{2}$). Las otras tres subimágenes corresponden a los detalles de la imagen, tales como detalles horizontales, verticales y diagonales. Este proceso de aplicación de la transformada *wavelet* puede ser aplicado iterativamente, es decir que una vez que se aplica la transformada *wavelet* a la imagen de entrada, la subimagen *trend* es tratada como una nueva imagen de entrada, por lo que a esta subimagen se le aplica la transformada
-

wavelet (y se obtienen subimágenes de tamaño $\frac{N}{4} \times \frac{M}{4}$). Es posible aplicarle a la subimagen *trend* nuevamente la transformada *wavelet* (para obtener subimágenes de tamaño $\frac{N}{8} \times \frac{M}{8}$). En el siguiente esquema se ilustra lo anterior y mostrándose el caso en el que los detalles de la imagen son despreciados (o puestos a cero). A este procedimiento se le conoce como análisis *wavelet* de la imagen y se muestra en la figura 3.5.

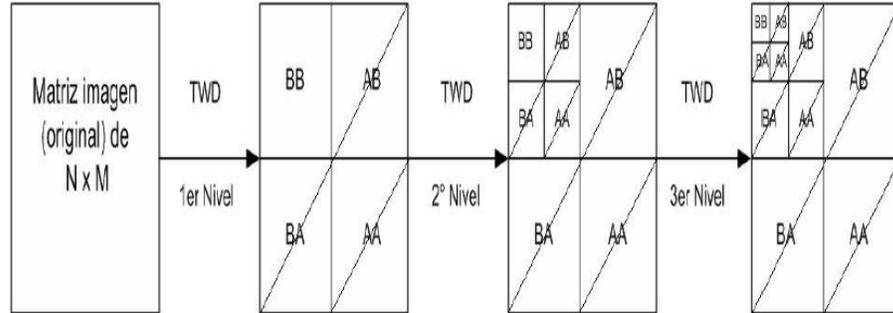


Figura 3.5: Diagrama a bloques del análisis *wavelet* de imágenes.

- Y en el proceso inverso al tenerse como entrada la subimagen *trend* (de tamaño $\frac{N}{8} \times \frac{M}{8}$), se le aplica la transformada *wavelet* inversa y se obtiene una subimagen *trend* de $\frac{N}{4} \times \frac{M}{4}$. Y así sucesivamente hasta obtener una versión aproximada de la imagen que originalmente se tenía, de tamaño $N \times M$. Si se contemplaran la totalidad de los valores generados por la transformada *wavelet* (tomando en cuenta los detalles de la imagen), la reconstrucción de la imagen mediante las transformadas *wavelets* inversas sería exacta, por lo que durante este proceso no habría pérdida de información. A este proceso se le llama síntesis *wavelet* de la imagen y se muestra en la figura 3.6.

3.5.1. Evaluación de la eficiencia de una técnica de compresión

La eficiencia de una técnica de compresión o descompresión de imágenes está en función de factores como: tipo de transformación que se utilice, la naturaleza del cuantizador, el codificador entrópico, así como el tipo de señal o imágenes en cuestión. Desafortunadamente no existe una técnica de compresión y descompresión de imágenes (ni de señales en general) que resulte adecuada para la diversidad de imágenes existentes, ya que la naturaleza de los datos también varía. Dicho de otra forma, no se

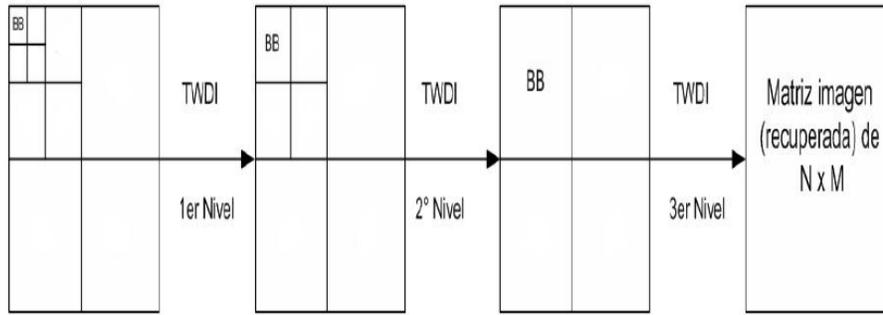


Figura 3.6: Diagrama a bloques de la síntesis *wavelet* de imágenes.

pueden esperar los mismos resultados de una técnica de compresión y descompresión de imágenes digitales cuando es aplicada a imágenes con poco detalle, como una imagen que contenga un cuadrículado (con dos tonalidades) a que si se aplica a imágenes que contienen detalles, como el caso de las huellas dactilares. De ahí la importancia de seleccionar las técnicas (o algoritmos) adecuadas para determinado tipo de imágenes y esperar mejores resultados.

La eficiencia de una técnica de compresión puede ser evaluada de manera objetiva y subjetiva [36]. Los métodos objetivos están basados en medidas de distorsión calculables. Un medida objetiva estándar de la calidad de una imagen es el error cuadrático en la reconstrucción o error cuadrático medio (ECM). Por ejemplo, en un sistema en la cual un bloque de la imagen de entrada $\{x(n)\}, n = 1, 2, \dots, M$ es reproducida como $\{y(n)\}, n = 1, 2, \dots, M$. El ECM es definido como la diferencia entre $x(n)$ y $y(n)$, como se muestra en 3.48.

$$ECM = \frac{1}{M} \sum_{n=1}^M (x(n) - y(n))^2 \quad (3.48)$$

Las varianzas de $x(n)$, $y(n)$ y $r(n)$ son σ_x^2 , σ_y^2 y σ_r^2 . En el caso especial de que las señales promedio sean cero, las varianzas son simplificadas a su respectivo valor cuadrático medio sobre la longitud de secuencia apropiada M,

$$\sigma_z^2 = \frac{1}{M} \sum_{n=1}^M z^2(n), \quad z = x, y, r. \quad (3.49)$$

Otra medida estándar objetiva de la calidad de la imagen es la relación señal a ruido (RSR), que está definida como la relación entre la varianza de la señal y la varianza del error de reconstrucción (Error Cuadrático Medio, o ECM) y usualmente

expresada en decibeles (dB),

$$RSR(dB) = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_r^2} \right) = 10 \log_{10} \left(\frac{\sigma_x^2}{ECM} \right) \quad (3.50)$$

Cuando la señal de entrada es una variable discreta de R bits, la varianza de energía puede ser reemplazada por la energía del símbolo de entrada máximo $(2^R - 1)^2$. Para el caso común de 8 bits por pixel de una imagen de entrada, la relación del pico de la señal a ruido (RPSR) puede definirse como:

$$RPSR(dB) = 10 \log_{10} \left(\frac{255^2}{ECM} \right) \quad (3.51)$$

RSR no es adecuada como una medida significativa de la calidad de la imagen, porque los errores en la reconstrucción en general no tienen la característica de ser ruido aditivo independiente de la señal. Algunas distorsiones pequeñas en una imagen pueden desencadenar en un valor grande de σ_r^2 y consecuentemente, un valor pequeño de RPSR, puede generar un imagen cuya calidad percibida es aceptable.

Otras medidas típicamente utilizadas para medir el rendimiento de una técnica de compresión son: la razón de compresión, el número de bits/pixel y la conservación de la energía de una señal o imagen.

La primera es un buen parámetro para evaluar la efectividad de un técnica de compresión. Por ejemplo, si la relación entre una imagen comprimida con respecto a la imagen sin comprimir, es de 2:1, podría decirse que la nueva imagen (la comprimida) ocupa la mitad del espacio de almacenamiento en disco que la ocupa la imagen sin comprimir, y está dada por:

$$R = \frac{ImagenSalida}{ImagenEntrada} \quad (3.52)$$

donde, R corresponde la razón o tasa de compresión, *ImagenSalida* es el tamaño de la imagen comprimida e *ImagenEntrada* corresponden al tamaño de la imagen sin comprimir, medida en bytes. Se busca que una técnica de compresión presente mayores tasas de compresión, sin permitir mucha degradación en la calidad de la imagen de salida.

Mediante el número de bits/pixel se indica el número de bits que se necesitan para representar cada pixel en la imagen. Al comprimir una imagen se pretende disminuir el número de bits necesarios para representar cada uno de sus pixeles.

Por último, respecto a la conservación de energía, este parámetro da una medida

del parecido que puede tener una imagen comprimida respecto a la versión de la imagen sin comprimir. Normalmente una técnica eficiente de compresión provoca que la imagen comprimida preserve un alto porcentaje de energía de la imagen original. La energía contenida en una imagen A , de tamaño $N \times M$ está dada por:

$$E_A = \sum_{i=1}^N \sum_{j=1}^M I(i, j)^2 \quad (3.53)$$

donde $I(i, j)$ se refiere a la tonalidad de la imagen I en la posición (i, j) .

En los sistemas de compresión de imágenes, la medida real de la calidad de la imagen es la calidad perceptual. La distorsión de una imagen es especificada mediante la escala de opinión media, EOM[38] o por la escala de calidad de la imagen, ECI[39]. Se usa la evaluación subjetiva basada en la percepción, cuantificada por el EOM y la evaluación objetiva también basada en la percepción, que es cuantificada por el ECI. Para obtener los valores del EOM de un conjunto de imágenes distorsionadas, se efectúan experimentos que involucran a cierto número de personas que otorgan una calificación según el nivel de distorsión observado en una imagen.

La metodología de prueba comúnmente usada es el método de escala de distorsión de doble estímulo, que maneja una escala con 5 niveles de distorsión descrita en [40]. El método de escala de distorsión de doble estímulo usa referencias y condiciones de prueba que son organizadas en pares, tal que la primera en el par es una referencia no distorsionada y la segunda es la misma secuencia distorsionada.

La primera imagen o la imagen fuente sin compresión es usada como referencia. Se pide siempre a una persona que otorgue una calificación a la segunda imagen (la versión comprimida), teniendo en mente la imagen original.

El método usa una escala con 5 grados de distorsión, cada uno con su correspondiente descripción: 5–imperceptible, 4–perceptible sin ser molesto, 3–ligeramente molesto, 2–molesto y 1–muy molesto. Después de estas evaluaciones, los valores de EOM para las condiciones e imágenes de prueba son calculados como:

$$EOM = \sum_{i=1}^5 i \cdot p(i) \quad (3.54)$$

donde i es la calificación otorgada por un usuario y $p(i)$ es la probabilidad de ocurrencia de dicha calificación. La evaluación subjetiva de la calidad de una imagen es experimentalmente difícil y los resultados pueden variar dependiendo de las condiciones de prueba. Adicionalmente a la EOM se utiliza la ECI, propuesta en [39]. ECI

fue desarrollado para evaluar la calidad de imágenes comprimidas. Combina varias distorsiones percibidas en una simple medida cuantitativa. ECI es construida mediante regresiones con la EOM. ECI se aproxima por mucho al EOM a la mitad del rango de calidad de la imagen. Para imágenes de muy alta calidad, es posible obtener valores de ECI mayores de 5. En la parte baja de la escala de la calidad de la imagen, ECI puede tener valores negativos (valores carentes de sentido), razón por la que se prefiere en ocasiones la evaluación subjetiva (cuando se trata con imágenes de no muy alta calidad), pero ECI suele ser útil en algunas fases de la evaluación.

3.5.2. Elección de la transformada *wavelet*

La elección de la función *wavelet* óptima en un sistema de compresión de imágenes para distintos tipos de imágenes debe realizarse en varios pasos. Para cada orden de filtro en cada familia de *wavelets*, el número óptimo de descomposiciones debe encontrarse. El número óptimo de descomposiciones es encontrado cuando se presenta el valor más grande para la RPSR en el rango de tasas de compresión para un orden de filtro. Por ejemplo en la figura 3.7, se presentan algunos de los resultados de la aplicación de la transformada *wavelet* a la imagen estándar *Lena*. Se observa que para órdenes de filtro bajos se obtienen mejores resultados con más descomposiciones *wavelets* que para órdenes de filtro altos. Las áreas sombreadas muestran el número óptimo de descomposiciones *wavelets* para un orden de filtro dado, mientras que, el área remarcada con tipografía en negrita indica la combinación óptima del orden un filtro con el número de descomposiciones *wavelets* para la imagen estándar *Lena* (5 descomposiciones con un filtro de orden 5). Resultados similares son obtenidos con las otras familias de *wavelets* y demás imágenes de prueba (incluidas imágenes estándares). En figura 3.8 se presentan los resultados obtenidos para diferentes familias de *wavelets* (como la D:Daubechies y C:Coiflets y con la llamada B: Biortogonal), diferentes órdenes de filtros y con varias imágenes de prueba. Para cada imagen de prueba y familia *wavelet*, la combinación óptima del orden de filtro y el número de descomposiciones se presenta como el área sombreada. Los órdenes de filtro que presentan mejores resultados para RPSR en cada familia de *wavelet* son diferentes para imágenes de prueba diferentes, excepto para la familia de *wavelets* biortogonales en donde los filtros de orden 2, en el proceso de descomposición, así como en el proceso de reconstrucción (BW-2.2) arroja los mejores resultados para todos los tipos de imágenes analizados. La comparación de los valores de RPSR de los filtros óptimos (mostrados en el área sombreada) de cada familia de *wavelet* para diferentes imágenes

de prueba, muestra que la imagen *Peppers* (con baja actividad espectral) tiene los valores más altos de la RPSR y la imagen *Zebra* (con alta actividad espectral) tiene los valores más pequeños de la RPSR. Los valores de la RPSR depende del contenido de las imágenes, por lo que no debe ser usado para comparar imágenes con diferente contenido.

<i>Wavelet</i>		Tasa de compresión				Número de descomposiciones
Familia	Orden	5:1	30:1	50:1	100:1	
TWD	1	35.53	14.76	11.53	9.42	2
		36.29	25.68	23.79	21.59	4
		36.39	25.74	23.94	21.87	6
		36.28	25.75	23.96	21.94	8
	2	36.79	15.09	11.73	9.51	2
		37.25	25.86	23.86	21.36	4
		37.18	25.79	23.89	21.52	6
		37.18	25.72	23.73	21.33	8
		37.07	25.61	23.57	21.04	10
	5	37.04	15.23	11.73	9.51	2
		37.24	25.83	23.38	16.16	3
		37.16	26.21	24.46	22.09	4
		37.05	26.24	24.53	22.18	5
		36.98	26.13	24.37	22.13	6
		36.88	25.91	24.17	21.82	8
	10	35.65	15.15	11.78	9.52	2
		36.00	25.41	23.20	16.46	3
		35.72	25.49	23.85	21.81	4
		35.47	24.13	23.45	21.41	6
		35.08	24.47	22.67	20.38	10

Figura 3.7: Número óptimo de descomposiciones *wavelets* para diferentes órdenes de filtros.

En este capítulo se ha dado una explicación de las transformadas *wavelets* y como éstas son usadas para implementar técnicas de compresión y descompresión de imágenes digitales. Se han examinado los efectos provocados al usar diferentes niveles de descomposición *wavelet*, el contenido de la imagen y las tasas de compresión, entre otros. También se han explicado los parámetros usualmente usados para medir la eficiencia de las técnicas de compresión y descompresión de imágenes digitales, así como

Wavelet	Peppers			Lena			Zebra			Baboon			N
	5:1	50:1	100:1	5:1	50:1	100:1	5:1	50:1	100:1	5:1	50:1	100:1	
D-1	37.98	23.91	21.72	36.28	23.96	21.94	29.16	17.26	15.90	28.01	21.84	21.07	8
D-3	39.31	24.30	21.70	37.33	24.26	22.14	28.06	17.75	16.25	27.98	21.68	20.87	6
D-5	38.75	24.24	21.69	37.05	24.53	22.18	27.72	17.81	16.11	28.21	21.76	20.77	5
D-10	37.09	23.05	16.60	36.00	23.20	16.46	23.02	16.84	14.41	27.59	21.15	12.57	3
C-2	39.67	24.62	22.18	37.80	24.55	22.33	28.30	17.92	16.44	28.22	21.82	20.95	5
C-3	39.65	24.82	22.29	37.73	24.64	22.49	28.10	17.95	16.27	28.24	21.85	20.87	4
C-4	38.96	24.56	21.81	37.52	24.50	22.15	27.96	17.93	16.17	28.23	21.75	20.77	4
C-5	38.92	24.57	21.94	37.46	23.54	22.43	27.90	17.97	16.30	28.24	21.76	20.86	4
B-22	40.20	24.67	22.09	37.78	24.69	22.51	28.22	17.56	16.01	27.71	21.66	20.88	4
B-33	39.15	23.30	20.78	36.54	23.38	21.11	26.24	16.72	15.20	26.39	20.70	20.06	8
B-44	39.62	23.09	16.28	37.98	23.25	16.39	28.10	16.75	14.28	28.18	21.07	12.65	5

Figura 3.8: Resultados de Relación pico señal–ruido en dB para diferentes familias de *wavelets* y diferentes tasas de compresión, y niveles de descomposición *wavelet* aplicado a imágenes estándares.

sus principales limitaciones. En el siguiente capítulo se presenta la arquitectura de cómputo móvil empleada para llevar a cabo el estudio de las técnicas de compresión, descompresión de imágenes digitales basadas en la transformada *wavelet*, así como su transmisión entre los dispositivos móviles.

Capítulo 4

Implantación de la Arquitectura de Cómputo Móvil

4.1. Introducción

En el presente capítulo se presenta a la arquitectura de cómputo móvil empleada para el desarrollo del presente trabajo, describiendo primeramente a los diferentes escenarios en los que se involucraron los dispositivos móviles ligeros. En seguida se hace una revisión que resalta las diferencias y/o similitudes a nivel de capa física entre los estándares de comunicación inalámbrica empleados. Se da la descripción del módulo de comunicaciones y del módulo de compresión/descompresión de imágenes digitales desarrollados y por último se presenta un caso de estudio: *TRImagen*, una aplicación que hace uso de dichos módulos.

4.2. Arquitectura de cómputo móvil

Para llevar a cabo el estudio de los algoritmos para la compresión, descompresión y transmisión de imágenes digitales entre dispositivos móviles, se consideraron los siguientes escenarios:

- Comunicación punto a punto entre una computadora portátil y un APD (Asistente Personal Digital) mediante el uso del puerto infrarrojo, como se muestra en la figura 4.1.

ESCENARIO A



Figura 4.1: Primer escenario de cómputo móvil.

ESCENARIO B



Figura 4.2: Segundo escenario de cómputo móvil.

- Comunicación punto a punto entre dos APD's mediante el uso del puerto infrarrojo, como se muestra en la figura 4.2.
- Comunicación inalámbrica mediante el estándar IEEE 802.11 entre la computadora portátil y un APD y entre dos APD's, como se muestra en la figura 4.3.
- Comunicación inalámbrica entre dos APD's mediante el estándar *Bluetooth*, como se muestra en la figura 4.4.

En los cuatro escenarios mencionados anteriormente, cada uno de los dispositivos involucrados tiene la capacidad de actuar como una entidad cliente o como una

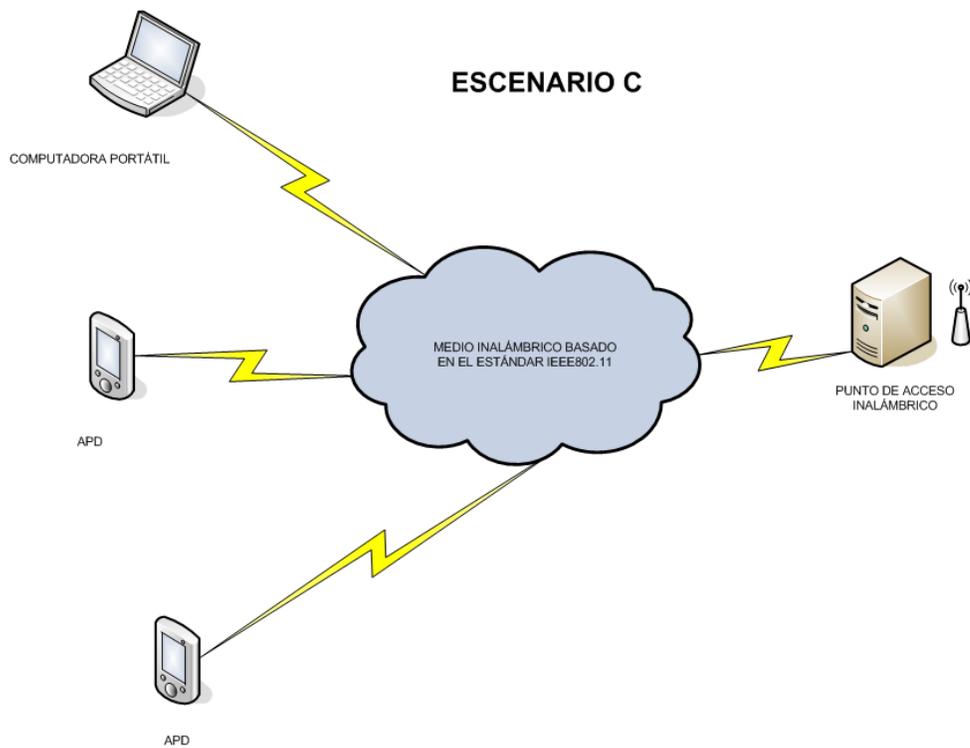


Figura 4.3: Tercer escenario de cómputo móvil.



Figura 4.4: Cuarto escenario de cómputo móvil.

entidad servidor, es decir, pueden transmitir o recibir imágenes desde/a cualquier dispositivo. Cada uno de los escenarios propuestos posee características que los hacen diferentes, entre las que pueden citarse las siguientes:

- En el caso del primer escenario, se establece comunicación en forma inalámbrica entre la computadora portátil y el APD mediante el uso del puerto infrarrojo. Este medio de comunicación limita la libre ubicación de los dispositivos, los

cuales deben encontrarse alineados y una distancia de separación no mayor de 1 ó 2 metros. Además la línea imaginaria que los une debe estar completamente libre (sin obstáculo alguno). En este escenario las diferencias de capacidades de cómputo entre los dos dispositivos son tomadas en cuenta, ya que por ejemplo, se debe evitar enviar una imagen (desde la computadora) de gran tamaño o de muy alta calidad al APD (hacia el APD), que posee capacidades de cómputo reducidas (tamaño de pantalla reducido, velocidad de procesamiento limitada, etc.) y en el caso contrario, esta situación es menos crítica, ya que las capacidades de cómputo del dispositivo destino (computadora portátil), son mayores que las del dispositivo origen (APD).

- El segundo escenario involucra a los dos dispositivos móviles ligeros de igual capacidad de cómputo (pero de menor que la computadora portátil), los dos APD's, usando como medio de comunicación inalámbrica, el estándar IrDA. En este esquema no existen diferencias computacionales entre los dispositivos, sin embargo, sigue siendo importante el considerar esquemas eficientes para la compresión, descompresión y transmisión de imágenes entre estos dispositivos de manera eficiente, ya que ambos poseen las mismas limitaciones. Al considerarse de nuevo la comunicación infrarroja, sus limitaciones son tomadas en cuenta, tal como el bajo ancho de banda a la que opera.

 - El tercer escenario contempla el esquema de comunicación inalámbrica basado en el estándar IEEE802.11b, con la interacción entre todos los dispositivos móviles ligeros contemplados en el desarrollo del presente trabajo, una computadora portátil, los dos APD's y el uso de un punto de acceso inalámbrico que sirve como intermediario para establecer comunicación entre cualquier par de dispositivos. A diferencia de los dos escenarios anteriores, en éste el radio de cobertura se ve beneficiado por la construcción de una red inalámbrica de área local sobre la que los dispositivos anteriormente mencionados operan. En este escenario, nuevamente las diferencias de capacidad de cómputo de los dispositivos es tomada en cuenta y a diferencia de los casos anteriores en donde los enlaces de comunicación eran dedicados única y exclusivamente para el intercambio de información entre los dispositivos, en este nuevo escenario al hacer uso de una infraestructura ya existente de comunicación inalámbrica, la compartición del canal con dispositivos ajenos al escenario de cómputo móvil es inherente.
-

- El cuarto y último escenario contempla la comunicación entre los APD's mediante el estándar *Bluetooth*. A diferencia de la comunicación infrarroja, con *Bluetooth* no es necesario que los dispositivos que pretenden comunicarse se encuentren alineados, ya que existe un radio de cobertura dentro del cual los dispositivos pueden establecer dicha comunicación (típicamente a un máximo de 10m). Este radio de cobertura es menor al que se presenta en con el esquema de comunicación inalámbrica basado en el estándar IEEE802.11b, en donde es posible alcanzar distancias de hasta 100m. El ancho de banda mediante el cual se da la transferencia de datos entre los dispositivos *Bluetooth*, es mayor al de la comunicación infrarroja pero menor que la basada en IEEE802.11b.

4.3. Estándares IrDA, IEEE802.11b y *Bluetooth*

La comunicación infrarroja IrDA (Infrared Data Association) o RI (radiación infrarroja) es un estándar de comunicación inalámbrica que se basa en la propagación de ondas de luz infrarroja para transmitir datos. La propagación de las ondas es similar a la luz visible, con una longitud de onda corta, por lo que el fenómeno de difracción no juega un papel importante. La RI toma ventaja de un rango espectral completamente no regulado, y su uso no está restringido. La RI no viaja a través de las paredes, así que los sistemas ubicados en recintos diferentes no interfieren entre sí, y presenta una alta seguridad, haciendo imposible la intervención de terceros. Ya que las paredes y techos típicamente reflejan el 60-90 % de la luz que incide en ellos, la transmisión puede ser posible aún cuando exista un obstáculo que obstruya la línea de mira formada entre los dispositivos, lo cual no siempre es posible ya que está en función de las condiciones del recinto. A pesar de que múltiples rutas de propagación contribuyen a la señal que se recibe, ésto no es un problema, como lo es en el caso de otras formas de comunicación inalámbrica (como la basada en radio-frecuencia), ya que los detectores ópticos presentan un área grande comparada con la longitud de onda de la luz infrarroja; por lo tanto un fotodiodo (un detector de intensidad de luz), realiza intrínsecamente una recepción de la diversidad espacial. Por otro lado, la interferencias de campo son usualmente prevenidas porque los tiempos de coherencia común de los transmisores ópticos son mucho más pequeños que la duración de los bits; la superposición de las componentes de la señal óptica toma lugar con respecto a la intensidad (potencia) y no respecto a la amplitud.

Sin embargo la RI presenta algunas desventajas, entre las que destacan, el rango

límite de distancias soportado por los dispositivos IrDA (relativamente corto de 1 a 2 metros) y la potencia de transmisión requerida para establecer comunicación, ambos determinados por la sensibilidad del dispositivo transmisor-receptor. La transmisión es del tipo *half duplex* (o de tipo unidireccional a la vez), lo que imposibilita que la transmisión se de en forma de multipunto.

Contrariamente a la RI, donde se lleva a cabo la detección directa y la modulación de intensidad de la señal de luz, en la radio-frecuencia RF se utiliza una parte del espectro electromagnético, basándose en la detección de una señal heterodina (señal a una frecuencia producida por la mezcla de dos o más señales en un dispositivo no lineal), para hacer posible la comunicación inalámbrica mediante el uso de un transmisor-receptor. Dentro de los dispositivos que se comunican por RF, existe una división, los que no cumplen ningún protocolo estándar y las que cumplen con un protocolo estándar, y en las normativas sobre sus distintas frecuencias de trabajo, que a la vez definen velocidad de transmisión o ancho de banda y campo de aplicación. Ejemplo de las que cumplen con un protocolo son, IEEE802.11b y *Bluetooth*, tratadas en el presente trabajo.

En el caso de *Bluetooth*, su especificación está establecida para el enlace entre dispositivos de voz y datos a corto alcance, de forma fácil y simple. *Bluetooth* opera en la banda ISM (*Industrial Scientific Medical*) de 2.4-2.5GHz permitiendo la transmisión de voz y datos, de forma rápida y segura con un rango de hasta 10 m a 1 mW o 100m si se usa un amplificador a 100 mW. Se puede transmitir en forma simétrica a 721 Kbps y simétricamente a 432 Kbps. Se puede transmitir voz, datos e incluso video, a diferencia de los esquemas basados en RI. *Bluetooth* minimiza la interferencia potencial al emplear saltos rápidos en frecuencia (1600 veces por segundo).

La arquitectura *Bluetooth* se organiza en pico-redes, formadas por dos o más dispositivos compartiendo un canal; una de las terminales actúa como maestro de la pico-red, mientras que el resto actúan como esclavos. Varias pico-redes superpuestas con áreas de cobertura superpuestas forman una *scatternet*. La versión 2.0 de *Bluetooth* soporta velocidades de 4, 8 y 12 Mbps, dependiendo del dispositivo, pero todas compatibles entre sí. Por otro lado, se proporciona un nivel de acceso al medio más eficiente que garantiza los tiempos de respuesta de aplicaciones de audio y video en tiempo real. La distancia sigue siendo de 10 metros y consume el doble de potencia. Una de las características más importantes de la versión 2.0 es que evita los problemas cuando se cae el maestro de una pico-red. En la nueva versión, cualquier dispositivo de la pico-red puede ser el supervisor o maestro de las comunicaciones

cuando algún otro desaparece o falla.

En el caso de la comunicación basada en el estándar IEEE802.11, ésta es utilizada para redes de computadoras y periféricos. La transmisión de datos trabaja en modo bidireccional mediante el protocolo CSMA/CD, que evita colisiones monitorizando el nivel de señal en la red, con las siguientes características

- 802.11 a 2Mb/s, 2.4 GHz
- 802.11b a 11Mb/s, 2.4 GHz
- 802.11g a 55Mb/s, 2.4 GHz
- 802.11a a 55Mb/s, 5.7 GHz

La versión más conocida actualmente es la 802.11b y se conoce con el nombre comercial de WiFi (Wireless Fidelity).

En el caso de estos dos últimos estándares, es posible destacar algunas similitudes y/o diferencias importantes: Ambos esquemas usan la técnica de separación del espectro en la banda de los 2.4 GHz, con rangos de 2.4 a 2.4835 GHz, para un ancho de banda total de 83.5 MHz. Ambos protocolos permiten diferentes niveles de protección al ruido: WiFi usa la modulación, codificación y la técnica de multiplexación correspondiente a las señales que van de 1 a 54 Mbp/s, mientras que *Bluetooth* usa una señal cuyo rango es fijo a 1Mbp/s y algunos esquemas de codificación.

Respecto a la interferencia, ambas tecnologías sufren el fenómeno de la interferencia de otros dispositivos que operan en la misma banda. La banda de 5 GHz usada por IEEE802.11a es también usada por los teléfonos inalámbricos de 5 GHz, mientras que la banda usada por *Bluetooth* e IEEE802.11g coincide con la utilizada por los hornos de micro-ondas, dispositivos de RF caseros y algunos teléfonos. Mientras ambos estándares son inherentemente resistentes a la interferencia, su mismo éxito ha llevado a empeorar el problema mas que cuando apareció.

Referido al consumo de potencia, ambos estándares definen limitaciones de potencia para dispositivos de acuerdo a los límites impuestos por organizaciones regulatorias de telecomunicaciones. En el caso de *Bluetooth* se especifica un promedio máximo de potencia de salida de 100 mW (clase 1), de 2.5 mW (clase 2) y de 1 mW (clase 3). Sin embargo para el caso WiFi, se exige que todos los dispositivos permitan una configuración para diferentes niveles de potencia. Muchos dispositivos del mercado, manejan un rango de consumo de potencia de 30-100 mW. Algunos dispositivos permiten ajustar su consumo de potencia.

En la tabla 4.1 se resumen las principales características entre estos dos estándares de RF, en donde destacan los siguientes puntos:

Relacionado a los mecanismos de adaptación al ruido (inmunidad al ruido), con *Bluetooth*, éstos son llevados a cabo a nivel de la capa de enlace de datos, al contrario a como se implementan en el caso de IEEE802.11b, a nivel de su capa física, por lo que éste último puede ser mayormente afectado por el ruido. La reducción del consumo de potencia entre un dispositivo *Bluetooth* y uno para WiFi es del orden de 30:1 ó 100:10, lo que hace a la tecnología *Bluetooth* adecuada para trabajar en entornos de cómputo móvil, donde la mayoría de los dispositivos tienen restricciones en cuanto a la energía que pueden almacenar. Esta situación hace ver la marcada diferencia en cuanto al radio de cobertura de los dispositivos IEEE802.11 (alrededor de 100m) respecto a los de *Bluetooth* (alrededor de 10m), así también en cuanto a los anchos de banda para la transferencia de datos en donde los basados en WiFi, alcanzan una velocidad máxima de 54 Mb/s (en IEEE802.11g y un máximo de 16 Mb/s en el caso de IEEE802.11b) y de 1 Mb/s para *Bluetooth*. En cuanto a la forma de acceso al canal de comunicación, con *Bluetooth*, se lleva a cabo mediante la exploración de dispositivos compatibles dentro del radio de cobertura, mientras que en IEEE802.11 mediante CSMA/CD (*Carrier Sense Multiple Access with Collision Detect*) para el aprovechamiento óptimo del canal de comunicación. Con la arquitectura IEEE802.11 se pueden conectar una gran cantidad de dispositivos a fin de formar "redes" (máximo de 2007), caso contrario en *Bluetooth* con un máximo de 255. Ambos esquemas involucran aspectos de seguridad como la autenticación y el cifrado de datos, necesarios por la naturaleza de sus comunicaciones por RF. Por último se resalta el aspecto de consumo de energía, en donde nuevamente la tecnología *Bluetooth* cobra gran importancia.

4.4. Programación en red

Para el caso de la transmisión de datos entre los dispositivos móviles, se hace uso de técnicas de programación de aplicaciones en red del tipo cliente-servidor, tal es el caso de los *sockets*. La interfase de *sockets* es un conjunto de subrutinas de programación que permiten que la comunicación entre procesos (o aplicaciones) sea posible. Estos procesos o aplicaciones pueden ejecutarse en un mismo dispositivo (local) o en diferentes entidades (entre cliente y servidor en forma remota). Estas rutinas son usadas para crear una canal de comunicación entre las aplicaciones que

CARACTERÍSTICA	Bluetooth	IEEE802.11
Banda de frecuencia	2.4 GHz	2.4 GHz, 5 GHz
Mecanismo de coexistencia	Frec. adaptiva	Selección frec. dinam.
Adaptación al ruido	Capa de enlace	Capa física
Potencia disipada típica	1-10 mW	30-100 mW
Rango nominal	10 m	100 m
Max. transf. unidireccional	732 kb/s	31.4 Mb/s
Célula básica	Pico-redes	Estaciones WiFi
Extensión célula básica	Scatternet	Estaciones WiFi agrupadas
Topologías	82.7962	83.6112
Max. disp. / cel. básica	8-255	Ilimitada 0 2007.
Max. tasa de señal	1 Mb/s	54 Mb/s
Métodos de acceso al canal	Exploración	CSMA/CD
Eficiencia del canal	Constante	Disminuye con el tráfico
Capacidad espacial	0.1-400 kb/s.m ²	Aprox. 15 kb/s.m ²
Protección de datos	CRC 16 bits	CRC 32 bits
Procedimiento config. red	Exploración	Exploración, autenticación
Veloc. prom. config. s/interf.	5s+exclavos*(1.28s)	exclavos*(1-13)*(1.35ms)
Autenticación	Secreto compartido (pares)	Secreto compartido (reto)
Encriptación	Cifrado flujo E0	Cifrado flujo RC4, RES
Mecanismos QoS	Tipos de enlace	Funciones de coordinación
Corriente típica consumida	1-35 mA	100-350 mA
Modos ahorro energía	Dormido	Dormido

Tabla 4.1: Comparación entre los protocolos *Bluetooth* y WiFi.

desean comunicarse. Típicamente se pretende que una aplicación cliente establezca comunicación en forma remota con una aplicación servidor. Un socket puede verse como el extremo de un canal de comunicación bidireccional. Los *sockets* pueden usarse para crear el canal de comunicación y usar dicho canal para transferir datos entre las aplicaciones. En la figura 1.4, se muestra un diagrama simple de dos aplicaciones que se comunican mediante el uso de *sockets*.

La interfase para el manejo de *sockets* provee de 6 funciones básicas. Es posible:

1. **Crear un socket.** Esto se logra mediante la rutina `socket()`, especificando el tipo de canal de comunicación que se usará. Esta especificación está dada por:
 - a) La familia de direcciones del canal de comunicación. Por ejemplo, especificar el formato de las direcciones (UDP y TCP usan el mismo formato de direcciones por lo que pertenecen a la misma familia de direcciones).

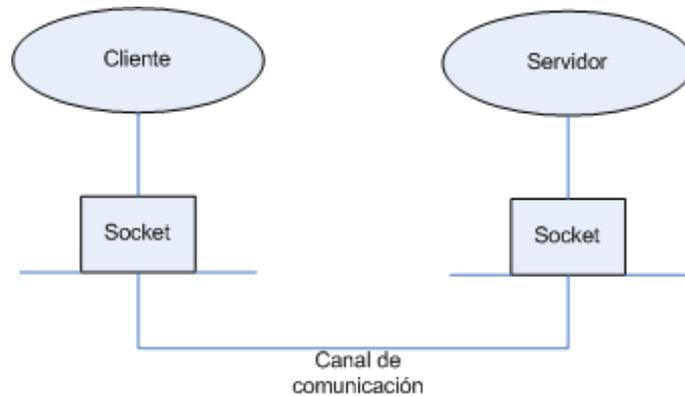


Figura 4.5: Diagrama a bloques de una aplicación cliente-servidor basada en *sockets*.

- b) El tipo de servicio del canal de comunicación. Se especifica la semántica de comunicación.
 - c) El protocolo a usar dada la familia de direcciones y el tipo de servicio.
2. **Asociar un *socket* con una dirección.** A este proceso también se le conoce como nombrar un *socket*. El formato de la dirección depende de la familia de direcciones del *socket*. Este paso debe llevarse a cabo antes de que la transferencia de datos sea posible. Si se usa un canal de comunicación por circuitos virtuales, este proceso equivale a asignar un número de teléfono a un teléfono. En cambio si se usa un canal de comunicación de tipo datagrama, nombrar a un *socket* equivale a asignar una dirección postal a un buzón de correo. Este proceso hace uso de la rutina `bind()`.
 3. **Conectarse con otro *socket*.** Esto se logra especificando la dirección de un *socket* remoto. Después de que la conexión ha sido establecida, es posible enviar datos al *socket* al cual ha sido conectado. Una analogía de este proceso es realizar una llamada telefónica y esperar a que sea contestada. Este proceso hace uso de la rutina `connect()`.
 4. **Aceptar una petición de conexión por parte de un *socket*.** Esto puede llevarse a cabo sólo del lado de la aplicación servidor cuando se está usando un canal de comunicación que provee de un circuito virtual. Cuando una aplicación remota intenta conectarse a un *socket*, la aplicación del servidor puede aceptar dicha petición. Si la petición no es aceptada el circuito virtual no es establecido.

Aceptar la petición de conexión equivale a contestar una llamada telefónica. Este proceso hace uso de la rutina `accept()`.

5. **Transferencia de datos.** La interfase *sockets* provee de varias rutinas para la transferencia de datos. Por ejemplo, pueden usarse las siguientes rutinas cuando un *socket* ha sido conectado:

- a) Las llamadas a sistema mediante `read()` y `write()`.
- b) Mediante las rutinas `send()` y `recv()`. Estas rutinas proveen de mayor funcionalidad que `read()` y `write()`. Con `send()` y `recv()` se pueden enviar y recibir datos. Existen otras rutinas que pueden usarse tales como, `sendto()`, `recvfrom()`, `sendmsg()`, `recvmsg()` entre otras.
- c) Terminar con las operaciones de *sockets*. Después de que un *socket* ha sido apagado, no es posible seguir transfiriendo datos mediante él.

4.5. Módulo de comunicaciones

Se desarrolló un módulo que permite la comunicación entre los dispositivos involucrados en cada uno de los escenarios propuestos, siguiendo el modelo cliente-servidor [20], en donde el dispositivo cliente hace peticiones que son atendidas por el dispositivo servidor. Este módulo es ejecutado en cada uno de los dispositivos móviles, tales como los APD's y la computadora portátil y permite al dispositivo sobre el que se ejecuta actuar como cliente o como servidor. El protocolo de comunicaciones del módulo desarrollado, se explica en la siguiente sección.

4.5.1. Descripción del protocolo

En el lado del dispositivo cliente:

1. Inicia la aplicación en modo escucha (*listening*) y al momento que se pretende enviar una imagen (o archivo):
 - a) Se crea un socket.
 - b) El dispositivo entra en modo de exploración, para localizar a algún dispositivo IrDA compatible (o el dispositivo con la IP especificada, para el caso de la versión de IEEE 802.11 y *Bluetooth*) aproximadamente 10 seg., si no lo encuentra se aborta la operación y se continúa en modo escucha.
-

En caso contrario solicita confirmación al usuario para enviar la imagen al dispositivo encontrado, y si éste acepta se intenta conectarse al dispositivo encontrado.

- c) Establecida la comunicación con el dispositivo encontrado, se envía el numero de bytes del archivo (o la imagen que se haya seleccionado con este fin), y enseguida los datos del archivo. Se utilizan las primitivas *send* y *recv* en modo bloqueantes.
- d) Al concluirse el envío del archivo (o de la imagen), el dispositivo cliente termina la conexión y pasa de nueva cuenta al modo *listening*.

Y el pseudo-código correspondiente:

Algoritmo 5 Dispositivo cliente

```

1: while aplicación en modo escucha y se solicita el envío de una imagen do
2:   Entra en modo exploración aproximadamente 10 seg (para IrDA, IEEE y Bluetooth).
3:   if SERVIDOR es hallado y USUARIO confirma envío then
4:     if Comunicación establecida then
5:       Enviar número de bytes de la imagen.
6:       Enviar la imagen.
7:       CLIENTE termina la conexión.
8:     end if
9:   end if
10: end while

```

En el lado del dispositivo servidor:

- a) Inicia la aplicación en modo escucha (*listening*).
 - b) En el momento que un dispositivo (cliente) solicita establecer comunicación con él (servidor), éste acepta la petición eventualmente si corresponde a un nombre de servicio (o la IP que éste posee) preestablecido.
 - c) Establecida la comunicación con el dispositivo cliente, el dispositivo servidor recibe el número de bytes del archivo (o de la imagen que se haya seleccionado con este fin), y enseguida los datos del archivo. Se utilizan las primitivas *send* y *recv* en modo bloqueantes.
 - d) Al concluirse la recepción del archivo (o de la imagen), el dispositivo cliente termina la conexión y el dispositivo servidor pasa de nueva cuenta al modo *listening*.
-

Su pseudo-código:

Algoritmo 6 Dispositivo servidor

```
1: while aplicación en modo escucha (listening) do
2:   if Dispositivo CLIENTE solicita establecer comunicación con el
   SERVIDOR then
3:     if CLIENTE es válido then
4:       Recibir número de bytes de la imagen.
5:       Recibir la imagen.
6:       CLIENTE termina la conexión.
7:     end if
8:   end if
9: end while
```

El protocolo de comunicación implementado hace uso del concepto de circuitos virtuales, es decir se crea un enlace (o canal de comunicación exclusivo) previo a la transferencia de los datos. La creación de estos circuitos virtuales es similar a lo que sucede cuando una persona hace una llamada telefónica (dispositivo cliente), y otra persona contesta (dispositivo servidor). En la figura 1.5 se muestra el diagrama a bloques del proceso que se sigue para establecer un circuito virtual mediante *sockets* en el módulo desarrollado.

Sin embargo, dada la naturaleza de los esquemas de comunicación inalámbrica empleados en el desarrollo del presente trabajo, fue necesario adaptar el módulo de comunicaciones, para operar bajo IrDA, *Bluetooth* e IEEE802.11. En las figuras 4.6 y 4.7 se muestran los diagramas a bloques de las versiones del módulo desarrolladas para el modo IrDA, IEEE 802.11 y *Bluetooth*, respectivamente.

El módulo de comunicaciones toma en cuenta las diferencias en cuanto a velocidad de procesamiento y cantidad de memoria se refiere, existente entre los dispositivos móviles involucrados en los escenarios propuestos, considerando principalmente dos casos:

- Envió de una imagen de un dispositivo de mayor capacidad de cómputo a uno de menor capacidad.
- Envió de una imagen de un dispositivo de menor capacidad de cómputo a uno de mayor capacidad.

La solución propuesta para el primer caso y dado que la comunicación fue basada en *sockets*, mediante esquemas en modo bloqueantes (en donde una vez que el dispositivo origen envía un byte de información o de datos de la imagen, éste no procede

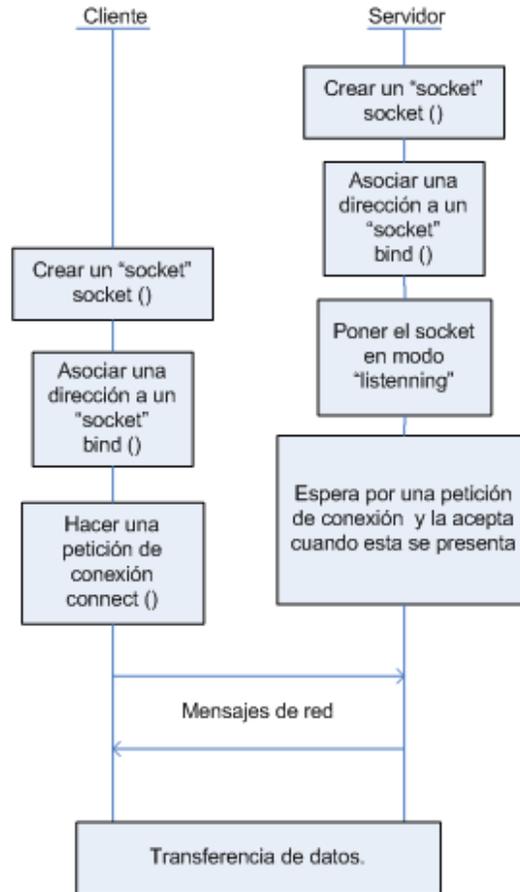


Figura 4.6: Creación de un circuito virtual mediante *sockets* en el módulo de comunicaciones.

a enviar el siguiente byte antes de que el dispositivo destino haya recibido ese primer byte) el dispositivo origen (el de mayor capacidad) envía bloques de información pequeños (byte por byte), para permitirle al dispositivo destino recibir sin ningún problema dicha información. Sin embargo en el segundo caso, el dispositivo origen envía bloques de información (de 10-50 bytes aprox.) a la vez al dispositivo destino, ya que por sus gran capacidad de cómputo, este no se ve afectado como en el caso de la primera situación, si se le enviarán ráfagas de información a algún dispositivo de capacidades de cómputo limitadas.

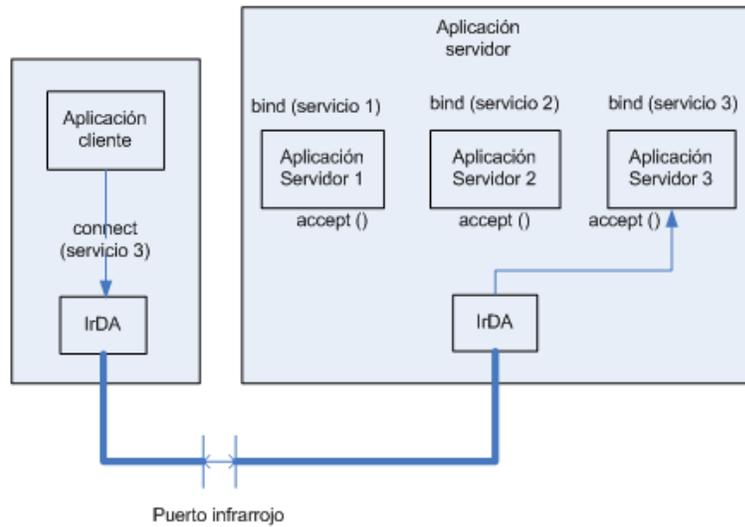


Figura 4.7: Diagrama a bloques del módulo de comunicaciones en su versión para IrDA.

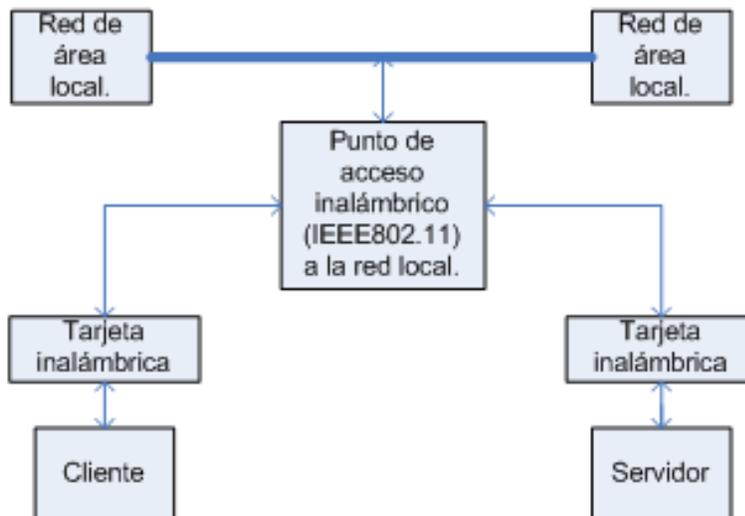


Figura 4.8: Diagrama a bloques del módulo de comunicaciones en su versión para IEEE 802.11b y *Bluetooth*.

4.6. Módulo de compresión/descompresión de imágenes digitales

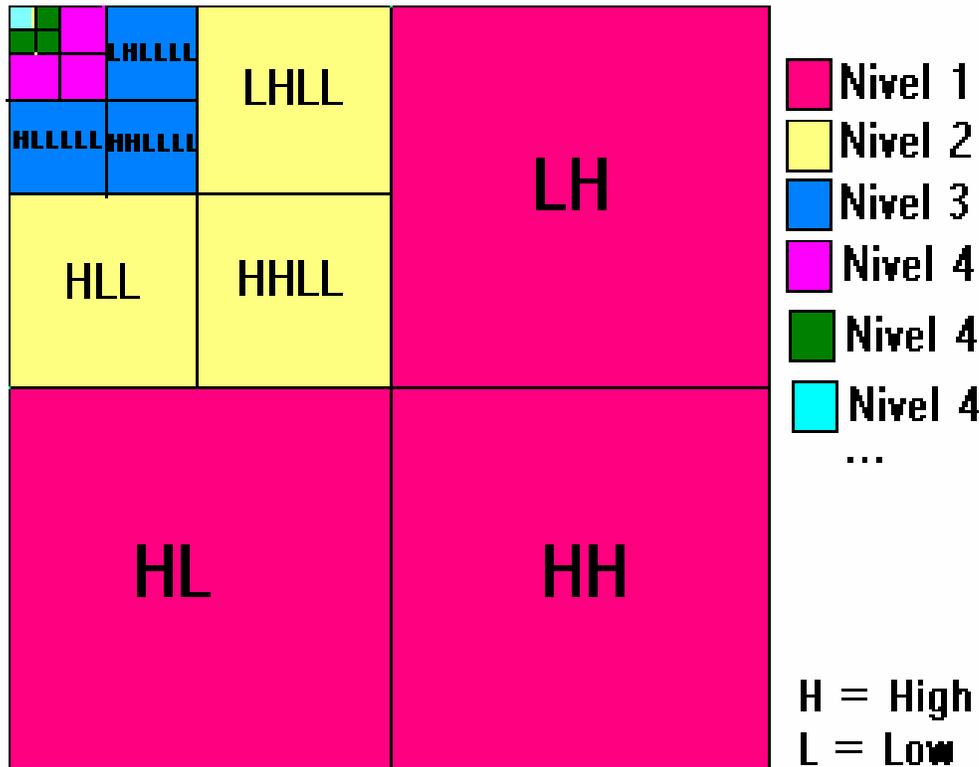
Este es el módulo encargado de la compresión y descompresión de imágenes digitales. Tal y como se mencionó en el capítulo anterior fueron implementadas tres

técnicas, basadas en las transformadas *wavelets* Haar, Daubechies y Coiflets respectivamente. A pesar de que no existe una respuesta absoluta para elegir qué tipo de transformada *wavelet* es la más adecuada, la mayoría de las veces esta elección depende del tipo de aplicación que requiere hacer uso de ellas. En este trabajo, la elección de esta familia de *wavelets* estuvo motivada principalmente entre otras por las siguientes razones:

- Son desde el punto de vista computacional, fáciles de calcular, aunque cabe hacer mención que la transformada Haar es aún más simple de calcular y de comprender que las restantes. La baja complejidad de los cálculos necesarios para computar las transformaciones *wavelets* se ve reflejada en el número de operaciones y la cantidad de memoria utilizada para llevar a cabo dichas operaciones.
- El algoritmo para calcular la transformada *wavelet* Daubechies presenta un ligero incremento en su complejidad respecto al caso de la transformada Haar, pero no más que la transformada Coiflets.
- Tanto la transformada Daubechies así como la transformada Coiflets, presentan un traslape entre iteraciones en el paso de transformación, mostrado en las ecuaciones 3.22-3.27, 3.28-3.33, 3.34-3.38 y 3.39-3.43, por la ubicación de los coeficientes α y β . Este traslape permite que las transformadas Daubechies y Coiflets preserven más detalles, en comparación que la transformada Haar. En cualquiera de los tres casos, las transformaciones *wavelets* comprimen la imagen de entrada.

El módulo desarrollado es capaz de comprimir (antes de enviar una imagen) y descomprimir (una vez recibida) una imagen digital utilizando cualesquiera de las tres técnicas implementadas. Dada que la transformación *wavelet* es una transformación lineal y la naturaleza de las imágenes digitales corresponden a una señal bidimensional, su transformación *wavelet* es calculada primero en la dirección horizontal y luego en la dirección vertical o viceversa. De esta manera al aplicar la transformada *wavelet* a una imagen se obtiene una pirámide de n niveles como se muestra en la figura 4.9. El componente de más baja frecuencia (II, III, etc.) se descompone en el siguiente nivel de la pirámide.

Por otro lado, y dada la naturaleza del color de las imágenes consideradas, en el modelo RGB (Red-Green-Blue), el proceso de aplicación de la transformada *wavelet*

Figura 4.9: Transformada *wavelet* en 2D.

se llevó a cabo para cada uno de los planos, R, G y B, que se muestra en la figura 4.10, así como el proceso de cuantización y codificación. Sin embargo, en el formato de compresión *wavelet* se retoma la idea del estándar JPEG de eliminar parte de la información de color a fin de lograr una mayor razón de compresión. Esto se hace aprovechando que el ojo humano percibe cambios pequeños de brillos más que cambios pequeños en color [27]. Puesto que el origen de los datos del codificador fueron archivos BMP (espacio RGB) fue necesario hacer una conversión al espacio YCbCr para separar los componentes de color de las de crominancia. Entonces la transformada es aplicada a cada componente Y, Cb, Cr por separado, lo que genera 3 matrices (pirámides) con $2^n \times 2^m$ coeficientes de transformación, denominadas respectivamente pyr1, pyr2, pyr3. En el descompresor sucede lo contrario, fue necesario convertir del espacio YCbCr al espacio RGB para escribir el archivo de salida BMP.

Para el caso de la cuantización, el módulo cuantiza de manera independiente las pirámides obtenidas de la transformación de cada componente YCbCr. El tamaño del paso de cuantización es variable y está determinado por el factor de compresión

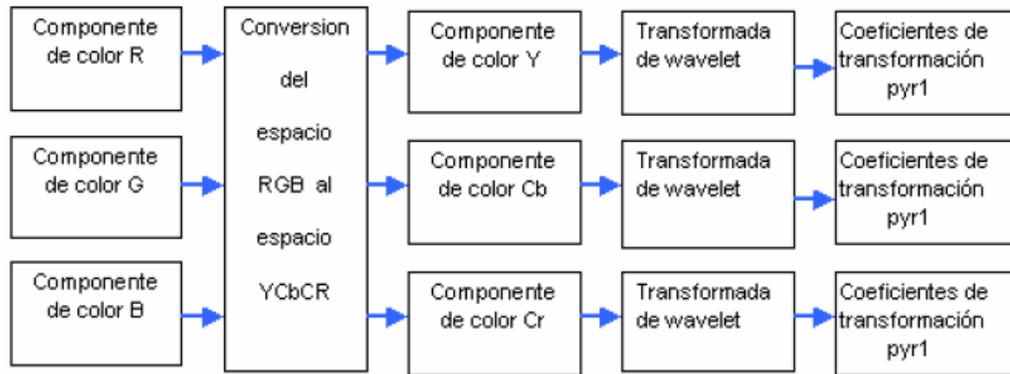


Figura 4.10: Transformada *wavelet* de los componentes de color de una imagen en el modelo RGB.

mediante la ecuación 4.1.

$$PasoCuantizacion = FactorQDeCuantizador = FactorCompresion \quad (4.1)$$

Conforme el paso de cuantización se incrementa, la calidad de la imagen empeora. Sin embargo, existe la posibilidad de que al utilizar un paso de cuantización relativamente pequeño, el rango del cuantizador sea tal que trunque una cantidad considerable de coeficientes de transformación y por consiguiente se tenga una pésima calidad de imagen. Los coeficientes que no cumplan la condición siguiente serán igualados con el valor máximo o mínimo del cuantizador según sea el signo del coeficiente. Véase la función de transferencia del cuantizador en la figura 4.11.

Las pirámides de coeficientes cuantizados correspondientes a los componentes Y, Cb y Cr son almacenadas en `qpyr1`, `qpyr2` y `qpyr3`, respectivamente. Asociado a cada pirámide cuantizada está un arreglo de bytes en el cual se almacena el factor de compresión utilizado por cada nivel.

A fin de lograr una mayor razón de compresión, se aplica un factor de compresión mayor para los primeros niveles de la pirámide que para los últimos, de hecho, se mantiene una razón 2:1 del nivel anterior al siguiente, aprovechando que en los primeros niveles de la pirámide existe una mayor cantidad de ceros y disminuye a medida que aumenta el número de nivel.

Una vez cuantizados los coeficientes de transformación correspondientes a cada componente YCbCr, se reúnen en una zona de memoria única para ser codificados entrópicamente. En la codificación entrópica se utiliza el algoritmo *deflate* de la

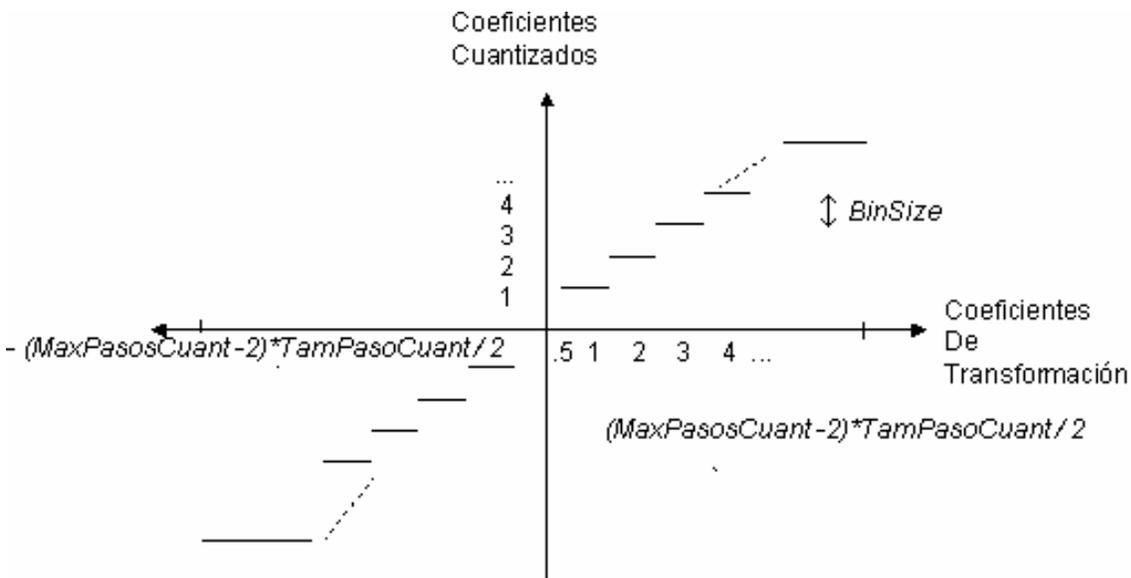


Figura 4.11: Función de transferencia del cuantizador lineal utilizado en el formato *wavelet*.

biblioteca Zlib , que implementa una variante de LZ77 (antecesor de LZW). Por tratarse de un codificador entrópico este paso no introduce pérdida de datos. Zlib es una biblioteca para compresión de datos de propósito general desarrollada con el objetivo de ser libre de patentes y es parte esencial en PNG (Portable Network Graphics). El procedimiento anterior es descrito en los algoritmos 7 y 8.

Dado que el desarrollo del presente trabajo tomó en cuenta una plataforma de cómputo móvil como la descrita anteriormente, es por ello que para su construcción se tomaron en cuenta las limitaciones computacionales de los dispositivos móviles, tales como velocidad de procesamiento, cantidad de memoria, tamaño de pantalla, entre otras. Así como también las marcadas diferencias de dichas características entre ellos. Así entonces,

- La carga mayor de trabajo es llevada a cabo siempre por el dispositivo que posee mayor capacidad de cómputo. El trabajo menor es delegado al dispositivo de menor capacidad. Cualquiera de los dispositivos puede actuar como una entidad servidor (origen) o como una entidad cliente (destino). De manera particular, en los escenarios tomados en cuenta en el desarrollo del presente trabajo y dado

Algoritmo 7 Submódulo de compresión

Require: En el *SERVIDOR*, selección de imagen I , en formato BMP (RGB) que se pretende enviar al *CLIENTE*.

- 1: Transformación modelo de color RGB a $YCbCr$.
- 2: **if** resolución en color(I) > resolución en color (pantalla *CLIENTE*) **then**
- 3: Adecuar(I) a la resolución en color de pantalla *CLIENTE*
- 4: **end if**
- 5: Aplicar transformación *wavelet* a cada plano Y, Cb y Cr .
- 6: **if** tamaño(I) > 4 x tamaño (pantalla *CLIENTE*) **then**
- 7: **while** tamaño(subimágenes *wavelets*) > tamaño (pantalla *CLIENTE*) **do**
- 8: Aplicar transformación *wavelet* a subimágenes *trend* (de Y, Cb y Cr), resultado de la transformación anterior.
- 9: **end while**
- 10: Cuantización no uniforme de cada plano (Y, Cb y Cr), resultando $qpyr1, qpyr2$ y $qpyr3$.
- 11: Se lleva a cabo la codificación de los coeficientes cuantizados con Zlib.
- 12: **else**
- 13: Cuantización no uniforme de cada plano (Y, Cb y Cr), resultando $qpyr1, qpyr2$ y $qpyr3$.
- 14: Se lleva a cabo la codificación de los coeficientes cuantizados con Zlib.
- 15: **end if**
- 16: La imagen resultante (I' , en formato BMP) es procesada por el módulo de comunicaciones, quien se encarga de transmitirla al dispositivo destino (*CLIENTE*).

que se utilizó un APD y un computadora portátil, la computadora portátil fue el dispositivo con mayor capacidad de cómputo, en lo que se refiere a la cantidad de memoria existente, espacio de almacenamiento, velocidad de procesamiento y el tamaño de su pantalla o visualizador, por lo que la solución propuesta siempre opta por encargarle una mayor cantidad de trabajo a la computadora portátil que al APD, para cada uno de los escenarios propuestos.

- Una imagen de gran tamaño o con una alta resolución de colores no es enviada a un dispositivo cuyas capacidades de visualización son menores, por lo que está se adecua en cuanto a tamaño y resolución de colores se refiere, posteriormente se comprime y finalmente se transmite. El dispositivo destino se encarga de descomprimirla y desplegarla.
 - Si se trata del caso contrario, donde se tiene un dispositivo destino con mayores capacidades que las que la imagen que se pretende enviar demanda, entonces la imagen simplemente se comprime en el dispositivo origen, se trasmite y el dispositivo destino es el que se encarga de descomprimirla, posiblemente
-

Algoritmo 8 Submódulo de descompresión

Require: En el *CLIENTE*, una vez recibida la imagen I' , ésta es procesada,

- 1: Se lleva a cabo la decodificación de los coeficientes para cada uno de los planos Y, Cb y Cr con Zlib.
- 2: Decuantización no uniforme de cada plano (Y, Cb y Cr), resultando $pyr1$, $pyr2$ y $pyr3$.
- 3: Aplicación de la transformación *wavelet* inversa a cada uno de los coeficientes $pyr1$, $pyr2$ y $pyr3$, de cada uno de los planos.
- 4: **while** tamaño(imagen recuperada (I) < tamaño (pantalla *DISPOSITIVOACTUAL*) **do**
- 5: Aplicar transformación *wavelet* inversa a I
- 6: **end while**
- 7: En caso de ser necesario, la imagen de salida es redimensionada.
- 8: Transformación modelo de color $YCbCr$ a RGB .
- 9: La imagen de salida es almacenada en el formato BMP.

redimensionarla y por último de desplegarla.

- Para llevar a cabo el redimensionamiento de las imágenes, se aprovechan las propiedades de las transformaciones *wavelets*, la *wavelet* en su forma directa, para el caso en que el tamaño de una imagen deba reducirse (tal y como sucede en el proceso de análisis *wavelet*) y la *wavelet* en su forma inversa para el caso en el que el tamaño de la imagen deba ampliarse (tal y como sucede en el proceso de síntesis *wavelet*). Entonces el número de niveles de descomposición y composición *wavelet* está en función de las capacidades de visualización del dispositivo destino. Sin embargo mediante este proceso es posible únicamente aproximar el tamaño original de la imagen de entrada al requerido, por lo que la adecuación exacta del tamaño es llevada a cabo mediante la técnica de replicación de píxeles (en el caso de la ampliación) y la de reducción de píxeles (en el caso de la reducción).

Consideremos como un primer caso particular, que se pretende enviar una imagen (posiblemente de gran tamaño) de un dispositivo A a un dispositivo B, en donde el dispositivo A (como el caso de la computadora portátil) posee mayores capacidades de velocidad de procesamiento y de memoria que el dispositivo B (como el caso del APD). Entonces, la carga de trabajo a desarrollar (o complejidad computacional) debe ser menor en el caso del dispositivo B que en el dispositivo A. Para determinar el nivel de descomposición *wavelet* (o nivel de compresión), se toma en cuenta el tamaño de la imagen de entrada (original) y las capacidades de visualización del

dispositivo destino (tamaño de pantalla y capacidad de resolución), así primeramente se adecuan las imágenes a la resolución en color (bits de profundidad) del dispositivo destino y después el nivel de descomposición *wavelet* es determinado considerando que si se tiene una imagen de tamaño $N \times M$, después de un nivel de compresión *wavelet* se obtienen subimágenes de $N/2 \times M/2$ y para un segundo nivel, de tamaño $N/4 \times M/4$ y así sucesivamente; entonces se aplica el número de niveles necesarios tal que en el proceso de descompresión, al aplicar una vez el proceso de transformación *wavelet* inversa, una vez, la imagen resultante se aproxime al tamaño de la pantalla del dispositivo destino (en este caso del APD). En el dispositivo destino (APD), una vez aplicado el proceso de descompresión a la imagen recibida, el módulo es capaz de redimensionar la imagen según sea conveniente. Si la imagen que se pretende enviar es más pequeña que la pantalla del dispositivo destino, la imagen es comprimida a un sólo nivel y en el dispositivo destino, la imagen es descomprimida y redimensionada si se hace necesario. Coincidentemente cuando se tratará de enviar una imagen (que se presume, es de dimensiones reducidas y de poca resolución) del dispositivo B al dispositivo A (cuyas capacidad de cómputo son mayores), se aplica un nivel de descomposición *wavelet*, y una vez recibida la imagen en el dispositivo destino, ésta es descomprimida y redimensionada en caso de ser necesario. Se aplica el mismo criterio si la imagen que se pretende enviar de B a A es de alta resolución y de dimensiones grandes, se aplica un nivel de compresión *wavelet* y el resultado se envía al dispositivo destino (A). Sólo si tratará de una imagen que está almacenada en el APD y cuya resolución y tamaño rebasan a las capacidades de visualización de la computadora portátil, el APD adecuaría la imagen en resolución y aplicaría el número de niveles de descomposición *wavelet* necesarios, con el fin de aproximar el tamaño de la imagen de salida a las capacidades de visualización de la computadora portátil. Cabe hacer mención que si en el proceso de compresión *wavelet* se aplican n descomposiciones, en el proceso de descompresión pueden aplicarse m composiciones *wavelets*, donde por lo general $n \neq m$. Así se busca que si el dispositivo origen tiene menores capacidades que el dispositivo destino, entonces al dispositivo origen se le permite aplicar n niveles de compresión y el dispositivo destino m niveles de descompresión, con $n \leq m$

4.7. Caso de estudio: *TRImagen*

Con el fin de utilizar los módulos desarrollados en el presente trabajo, se desarrolló una aplicación como caso de estudio, *TRImagen*: Transmisión y Recepción

de Imágenes, en donde ambos módulos interactúan, ejecutándose en alguno de los dispositivos móviles.

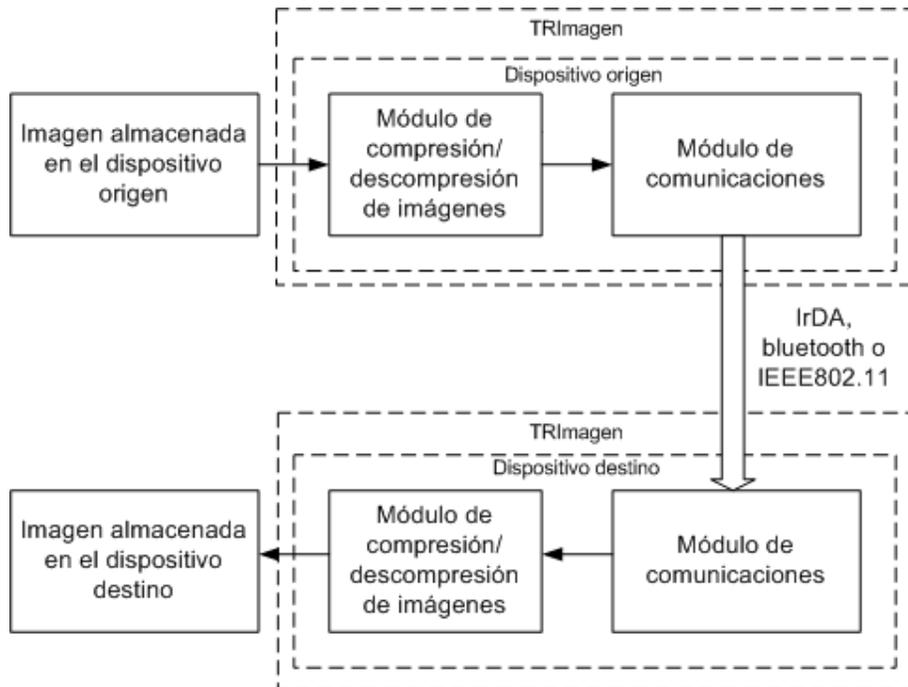


Figura 4.12: Diagrama a bloques que muestra la interacción entre el módulo de comunicaciones y el de compresión/descompresión de imágenes digitales.

En *TRImagen* antes de enviarse una imagen de un dispositivo origen al destino, ésta es comprimida empleando alguna de las tres técnicas implementadas (la basada en la transformada *wavelet* Haar, Daubechies y Coiflets), después es enviada mediante alguno de los esquemas de comunicación inalámbricos tratados en el presente trabajo y en el dispositivo destino, la imagen es descomprimida, para después visualizarla. Lo anterior se muestra en el diagrama a bloques de la figura 4.12.

Las tres técnicas de compresión anteriormente mencionadas, fueron implementadas en dos plataformas. En la primera de ellas, las técnicas se codificaron en lenguaje ANSI C, compiladas con gcc ver. 2.96 bajo ambiente Linux (Suse 9.2). Y en la segunda, se empleó el entorno de programación Visual C++ ver. 6.0 en el ambiente Windows XP profesional, para *TRImagen* en su versión para PC y Microsoft Embedded Visual C++ 3.0, para *TRImagen* para los APD's.

4.7.1. Características de *TRImagen*

Las interfaces gráficas de las aplicaciones desarrolladas, se muestran en las figuras 4.13 y 4.14. Algunas de las características de *TRImagen* son:

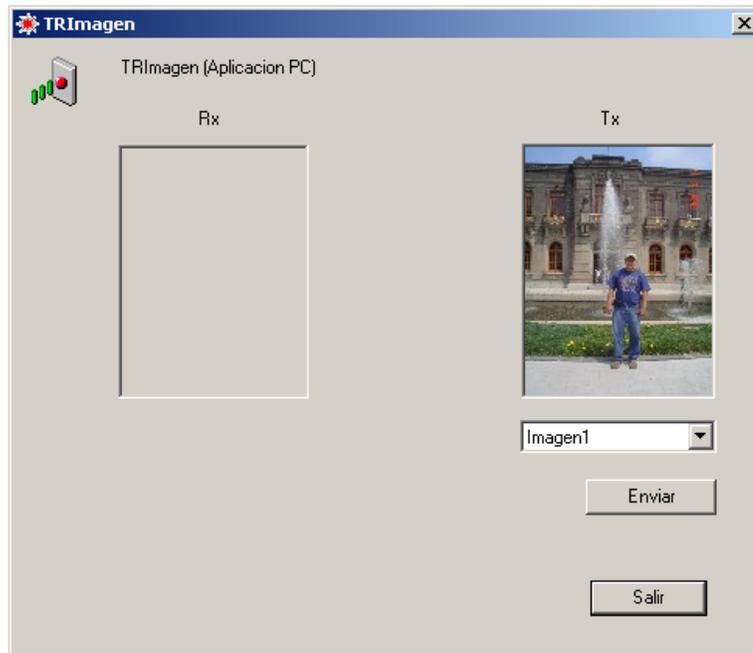


Figura 4.13: Interfase gráfica de TRImagen en su versión para PC.

1. Aplicación cliente-servidor y orientada a conexión. Se necesita que se establezca comunicación entre las entidades para que se inicie el proceso de envío y recepción de datos.
 2. Las aplicaciones que se ejecutan en cualquiera de los dispositivos móviles, le permiten al dispositivo actuar como una entidad cliente o una entidad servidor.
 3. TRImagen está basada en Winsocks (biblioteca de funciones desarrollada para el manejo de *sockets* bajo ambiente Windows).
 4. Basada en el paradigma de programación orientada a objetos (C++).
 5. El cliente envía una petición para establecer comunicación con el servidor.
 6. El servidor acepta (eventualmente) esa petición.
 7. El cliente envía datos (o una imagen) al servidor.
-

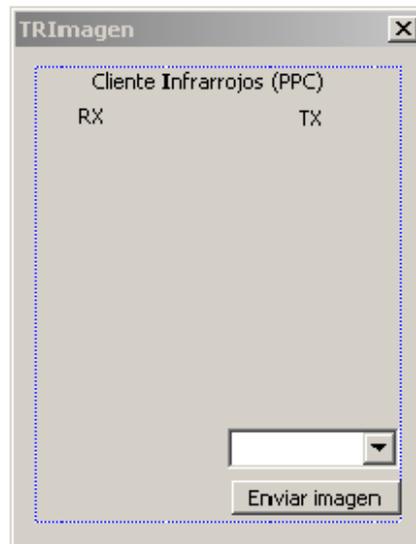


Figura 4.14: Interfase gráfica de TRImagen en su versión para el APD.

8. El cliente termina la conexión.
 9. Tanto la versión de TRImagen para la computadora portátil así como la del APD, fueron desarrolladas en la computadora portátil mediante los entornos de programación previamente mencionados. Una vez generadas las versiones ejecutables de tales aplicaciones, en la computadora portátil puede ejecutarse inmediatamente, sin embargo, en el caso del APD, este archivo ejecutable debe transferirse al dispositivo, empleando algunas de sus formas de comunicación, tales como IrDA, IEEE802.11 o *Bluetooth*, para que pueda ser ejecutada desde el mismo APD. Es importante que los modos de comunicación antes mencionados estén activos para que la comunicación entre los dispositivos móviles sea posible, mediante TRImagen.
 10. Algunos datos técnicos relevantes:
 - Cantidad de memoria consumida en la versión "servidor" (CP): 479 Kb
 - Cantidad de memoria consumida en la versión "cliente" (APD): 25 Kb
 - Cantidad de memoria/espacio de almacenamiento disponible en el APD: 128 Mb RAM, 48 Mb Flash ROM (con 17 Mb disponibles).
 - Velocidad del procesador en el APD: Intel XScale PXA255 a 400 MHz.
-

- Capacidad de despliegue del APD: pantalla a color de 240 x 320 pixeles, TFT a 65536 colores.

En el presente capítulo se ha dado una explicación detallada de la arquitectura de cómputo móvil empleada para el desarrollo del presente trabajo, contemplando los dos aspectos fundamentales incluidos en su desarrollo: la comunicación entre los dispositivos móviles así como las técnicas de compresión y descompresión de imágenes digitales. En el siguiente capítulo se presentan las pruebas a las que sometieron las implementaciones desarrolladas, así como el análisis de los resultados más importantes obtenidos de las mismas.

Capítulo 5

Evaluación y Análisis de Resultados

5.1. Plataforma de experimentación

Dado que se desarrolló un proyecto que involucra aspectos de innovación tecnológica, se utilizaron dispositivos móviles actuales, como los que se citan a continuación:

1. Computadora portátil, DELL Latitude D600, con procesador Mobile Technology Centrino a 1.5 GHz, 768 Mb RAM, 80 Gb HD, con características inalámbricas integradas (WLAN 802.11b e IrDA 1.1). Sistema Operativo Windows XP Profesional y Red Hat Linux 9.0.
2. Dos HP iPAQ Pocket PC h5550, con procesador Intel XScale PXA255 a 400 MHz. Sistema Operativo Microsoft Windows CE 4.2, 128 Mb RAM, 48 Mb Flash ROM (con 17 Mb disponibles), pantalla a color de 240 x 320 pixeles, TFT a 65536 colores, con comunicaciones WiFi, IrDA y *Bluetooth* integradas.

Sin embargo, únicamente con el fin de someter a pruebas exhaustivas las implementaciones desarrolladas, se utilizaron dispositivos no tan actuales para obtener pruebas bajo condiciones aún más limitadas.

En cuanto a las herramientas de software empleadas, podemos citar, compiladores de C/C++, JAVA, los entornos de desarrollo Visual C++ 6.0, Embedded Visual C++ 3.0. Herramientas de cálculo numérico como Matlab 6.0 (con Digital Image Processing Toolbox) o Maple 9.0. Editores y depuradores, etc.

5.2. Evaluación

Dado que en el presente trabajo se involucraron esencialmente dos aspectos, el de comunicaciones entre los dispositivos móviles mediante los estándares de comunicación inalámbrica (IrDA, IEEE802.11b y *Bluetooth*), y por otro lado, las técnicas de compresión y descompresión de imágenes digitales, las pruebas y evaluación que se presentan fueron llevadas a cabo en dos fases. La primera fase, orientada a la caracterización y evaluación del protocolo de comunicaciones implementado con el fin de hacer posible la transmisión y recepción de imágenes entre los dispositivos móviles, en cada uno de los escenarios propuestos para tal fin. La segunda fase, contempla la evaluación de las técnicas de compresión y descompresión de imágenes digitales basadas en las transformadas *wavelets*, con el fin de localizar, la(s) mejor(es) combinación(es) y ajuste de parámetros óptimo, para lograr una efectividad aceptable de cada una de las técnicas.

5.3. Comunicaciones entre los dispositivos móviles

Las pruebas llevadas a cabo con el fin de evaluar y caracterizar las comunicaciones entre los dispositivos móviles involucrados en cada uno de los escenarios propuestos, son las siguientes:

Para el caso de IrDA

1. **Medición de la distancia máxima de separación entre los dispositivos IrDA en función del ángulo de incidencia (o punto de mira).** El objetivo de la prueba fue determinar la distancia máxima de separación a la que podían ubicarse los dispositivos IrDA (y establecer comunicación), cuando se encontraban completamente alineados (a 0°) así como cuando este ángulo se variaba.
 2. **Medición del rendimiento de TRImagen al enviar y recibir imágenes desde/hacia la computadora portátil hacia/desde el APD (o entre APDs), en función del tamaño de las imágenes.** Estas pruebas fueron llevadas a cabo con el fin de medir los tiempos consumidos para la transmisión/recepción de distintos tamaños de imágenes entre los dispositivos móviles, cuando se variaba la distancia de separación entre ellos así como el ángulo de incidencia que formaban sus puntos de mira.
-

Para el caso de *Bluetooth*

1. **Medición del rendimiento de TRImagen al enviar y recibir imágenes desde/hacia la computadora portátil hacia/desde el APD (o entre APDs), en función del tamaño de las imágenes.** Estas pruebas fueron llevadas a cabo con el fin de medir los tiempos consumidos para la transmisión/recepción de distintos tamaños de imágenes entre los dispositivos móviles *Bluetooth*.

- Con los dispositivos a 10m de separación dentro del mismo nivel de un edificio.
- Con los dispositivos a 10m de separación en diferentes niveles de un edificio.

Para el caso de IEEE802.11

1. **Medición del rendimiento de TRImagen al enviar y recibir imágenes desde/hacia la computadora portátil hacia/desde el APD (o entre APDs), en función del tamaño de las imágenes.** Estas pruebas fueron llevadas a cabo con el fin de medir los tiempos consumidos para la transmisión/recepción de distintos tamaños de imágenes entre los dispositivos móviles IEEE802.11.

- Con los dispositivos a 10m de separación del punto de acceso inalámbrico dentro del mismo nivel de un edificio.
- Con los dispositivos a 10m de separación del punto de acceso inalámbrico dentro de diferentes niveles de un edificio.

En la siguientes secciones se presentan los resultados obtenidos de las pruebas llevadas a cabo, en donde se hace uso de la siguiente nomenclatura, **C**:Dispositivo cliente, **S**: Dispositivo servidor, **d**:Distancia de separación entre los dispositivos, **W**: Ángulo de desviación entre los dispositivos.

5.3.1. Resultados experimentales de *TRImagen* en su versión para IrDA

Ángulo	Distancia
0°	1.5 m
5°	1.3556 m
10°	1.2111 m
15°	1.0667 m
20°	0.9222 m
25°	0.7778 m
30°	0.6333 m
35°	0.4889 m
40°	0.3444 m
45°	0.2 m

Tabla 5.1: Distancia máxima de separación de los dispositivos IrDA vs ángulo de incidencia.

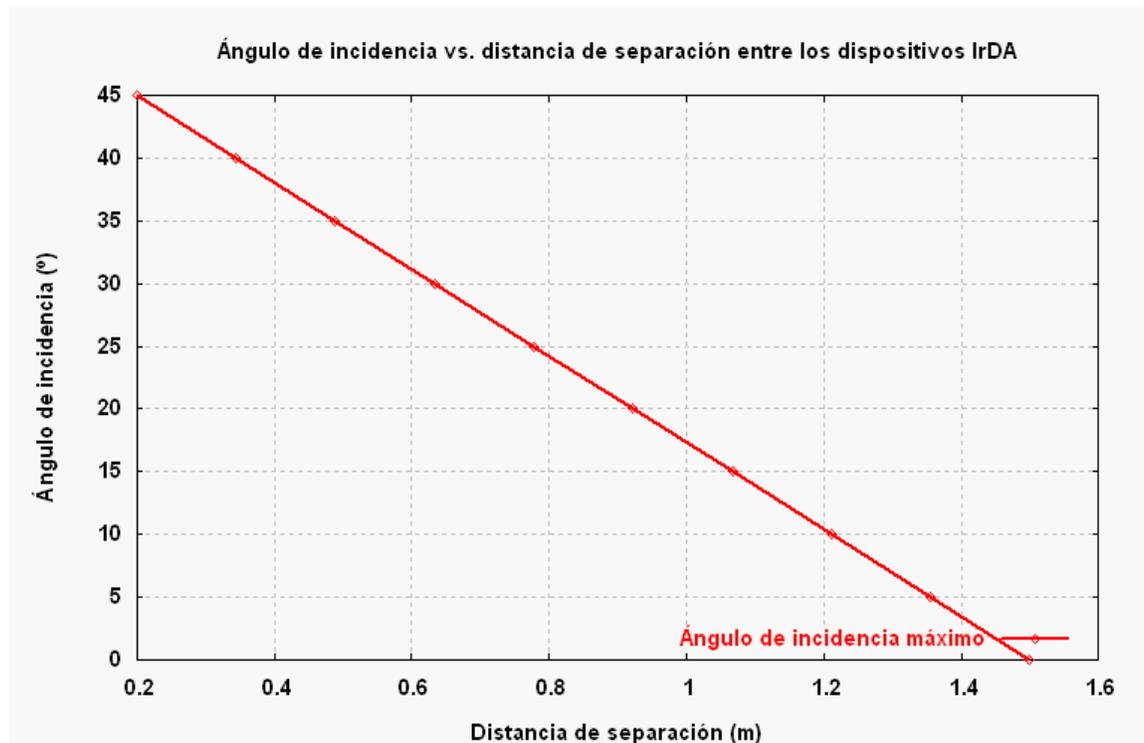


Figura 5.1: Distancia máxima de separación de los dispositivos IrDA vs ángulo de incidencia.

- Caso I. $C=PC$, $S=PDA$, $d=0.20m$, $W=0^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	3.63	0.90
18.44	5.17	1.32
36.88	13.41	3.52
73.76	17.04	4.25
149.62	20.57	5.12
209.48	29.76	7.40
1027.33	140.22	36.29

Tabla 5.2: Envío de imágenes desde el APD hacia la CP a una distancia de separación de 0.20 m.

- Caso II. $C=PC$, $S=PDA$, $d=0.60$ m, $W=0^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	4.44	1.19
18.44	7.63	2.25
36.88	12.35	3.63
73.76	19.16	5.31
149.62	20.75	5.60
209.48	30.13	7.53
1027.33	146.76	35.92

Tabla 5.3: Envío de imágenes desde el APD hacia la CP a una distancia de separación de 0.60 m.

- Caso III. $C=PC$, $S=PDA$, $d=1.0$ m, $W=0^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	4.76	0.89
18.44	8.97	1.82
36.88	14.64	2.66
73.76	21.17	3.84
149.62	22.36	5.10
209.48	30.19	7.15
1027.33	153.48	35.06

Tabla 5.4: Envío de imágenes desde el APD hacia la CP a una distancia de separación de 1.0 m.

- Caso IV. **C=PC**, **S=PDA**, **d=0.20 m**, **W=45°**.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	3.59	0.92
18.44	7.23	1.80
36.88	10.53	2.81
73.76	15.36	3.78
149.62	20.70	5.01
209.48	29.15	7.28
1027.33	143.08	35.88

Tabla 5.5: Envío de imágenes desde el APD hacia la CP a una distancia de separación de 0.20 m. y ángulo de desviación de 45°.

- Caso V. Para **C=PC**, **S=PDA**, **d=0.60 m** y **1.0 m**, **W= 45°**, no se logró establecer comunicación entre los dispositivos.
- Caso VI. **C=PDA**, **S=PC**, **d=0.20 m**, **W=0°**.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	3.22	0.80
18.44	5.66	1.41
36.88	14.11	3.52
73.76	17.16	4.29
149.62	24.17	6.04
209.48	33.1	8.27
1027.33	162.32	40.55

Tabla 5.6: Envío de imágenes desde la CP hacia el PDA a una distancia de separación de 0.20 m.

- Caso VII. $C=PDA$, $S=PC$, $d=0.60$ m, $W=0^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	4.06	1.01
18.44	7.77	1.92
36.88	12.31	3.07
73.76	19.17	4.79
149.62	24.51	6.12
209.48	34.62	8.62
1027.33	169.78	42.27

Tabla 5.7: Envío de imágenes desde la CP hacia el PDA a una distancia de separación de 0.60 m.

- Caso VIII. $C=PDA$, $S=PC$, $d=1.0$ m, $W=0^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	5.81	1.45
18.44	9.03	2.25
36.88	15.08	3.77
73.76	20.14	5.03
149.62	21.92	5.48
209.48	31.10	7.75
1027.33	152.52	38.00

Tabla 5.8: Envío de imágenes desde la computadora portátil hacia el PDA a una distancia de separación de 1.0 m.

- Caso IX. $C=PDA$, $S=PC$, $d=0.20$ m, $W=45^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	4.41	1.11
18.44	7.59	1.89
36.88	10.63	2.63
73.76	15.03	3.75
149.62	20.06	5.15
209.48	28.95	7.23
1027.33	142.76	35.45

Tabla 5.9: Envío de imágenes desde la computadora portátil hacia el PDA a una distancia de separación de 0.20 m y un ángulo de separación de 45° .

- Caso X. Para $C=PDA$, $S=PC$, $d=0.60$ m y 1.0 m, $W=45^\circ$, no se logró establecer comunicación entre los dispositivos.

- Caso XI. $C=APD$, $S=APD$, $d=0.20m$, $W= 0^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	3.63	4.26
18.44	5.17	7.82
36.88	13.41	16.01
73.76	17.04	17.1
149.62	20.57	22.31
209.48	29.76	29.88
1027.33	140.22	145.31

Tabla 5.10: Transmisión de imágenes entre APD's a una distancia de separación de 0.20 m.

- Caso XII. $C=APD$, $S=APD$, $d=0.60m$, $W= 0^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	3.63	4.33
18.44	5.17	7.92
36.88	13.41	16.11
73.76	17.04	17.14
149.62	20.57	23.01
209.48	29.76	30.19
1027.33	140.22	146.17

Tabla 5.11: Transmisión de imágenes entre APD's a una distancia de separación de 0.60 m.

- Caso XIII. $C=APD$, $S=APD$, $d=1.0m$, $W= 0^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	3.63	4.3
18.44	5.17	8.06
36.88	13.41	16.47
73.76	17.04	18.61
149.62	20.57	23.95
209.48	29.76	29.96
1027.33	140.22	145.88

Tabla 5.12: Transmisión de imágenes entre APD's a una distancia de separación de 1.0 m.

- Caso XIV. $C=APD$, $S=APD$, $d=0.20$ m, $W= 45^\circ$.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	4.41	4.2
18.44	7.59	7.98
36.88	10.63	16.0
73.76	15.03	17.21
149.62	20.06	22.96
209.48	28.95	29.44
1027.33	142.76	144.15

Tabla 5.13: Transmisión de imágenes entre APD's a una distancia de separación de 0.20 m y un ángulo de incidencia de 45° .

- Caso XV. Para $C=APD$, $S=APD$, $d=0.60$ m y 1.0 m, $W= 45^\circ$, no se logró establecer comunicación entre los dispositivos.

En la figura 5.2 se presentan la gráficas comparativas con los datos de las tablas anteriores.

Y el gráfico que muestra la tasa de transferencia se presenta en la figura 5.3.

Los resultados obtenidos en las pruebas anteriores son satisfactorios, ya que en primera instancia se refleja un incremento en el tiempo de transmisión y/o recepción de imágenes desde/hacia algún dispositivo, al incrementarse el tamaño de imágenes. Otros factores que contribuyeron en el incremento de éste tiempo fueron, la distancia de separación entre los dispositivos, el ángulo de incidencia formado entre los dispositivos, y en menor medida, pero no menos importante, se vió afectado por el tipo de dispositivo con el que en algún momento se pretendía transmitir ó recibir una imagen dada, mostrándose un incremento pequeño de tiempo cuando se enviaba una imagen del APD a la computadora portátil, más que si se hacía en sentido inverso. Los tiempos consumidos fueron mejorados con la inclusión de las técnicas de compresión y descompresión de imágenes.

5.3.2. Resultados experimentales de *TRImagen* en su versión para *Bluetooth*

Los resultados se muestran en las tablas 5.14 y 5.15 respectivamente. Y el gráfico comparativo de rendimiento y ancho de banda se muestra en la figuras 5.8 y 5.9 respectivamente.

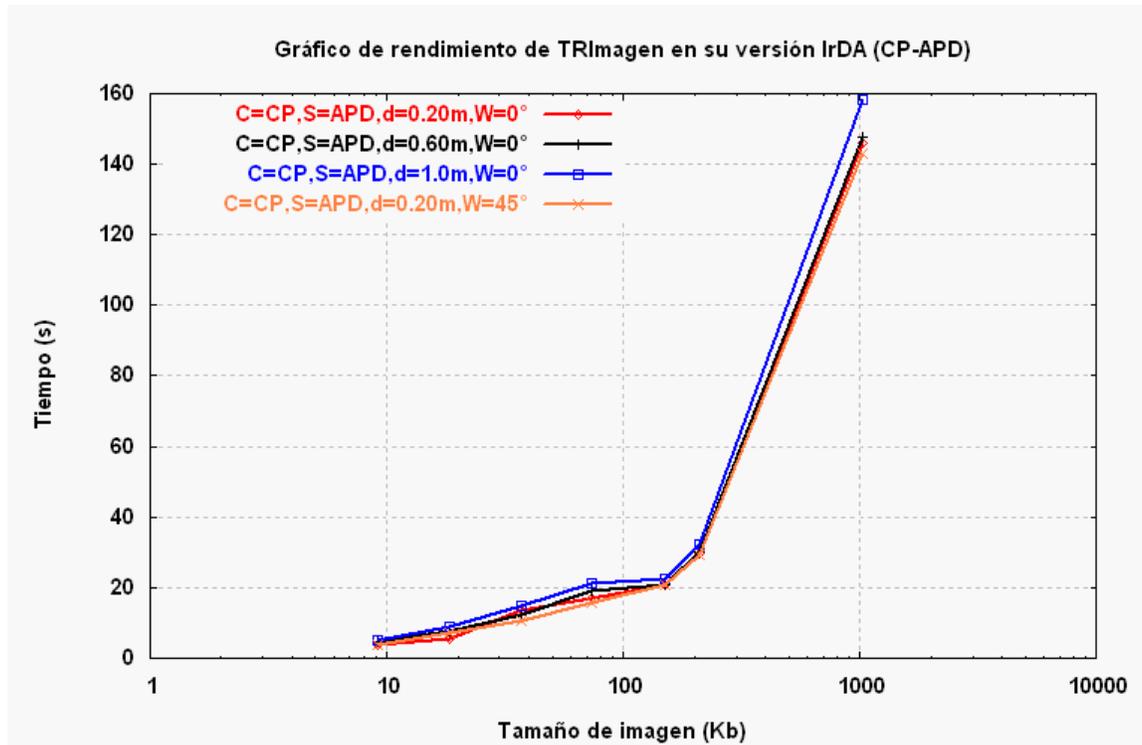


Figura 5.2: Rendimiento de *TRImagen* en IrDA.

- Caso I. Para $C=APD$, $S=APD$, $d=10m$ en el mismo nivel del edificio.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	0.078	0.019
18.44	0.306	0.0764
36.88	0.837	0.209
73.76	1.898	0.474
149.62	2.810	0.702
209.48	3.461	0.865
1027.33	18.416	4.604

Tabla 5.14: Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante *Bluetooth* cuando los dispositivos se ubicaron a una distancia de 10m de separación dentro del mismo piso en el mismo edificio.

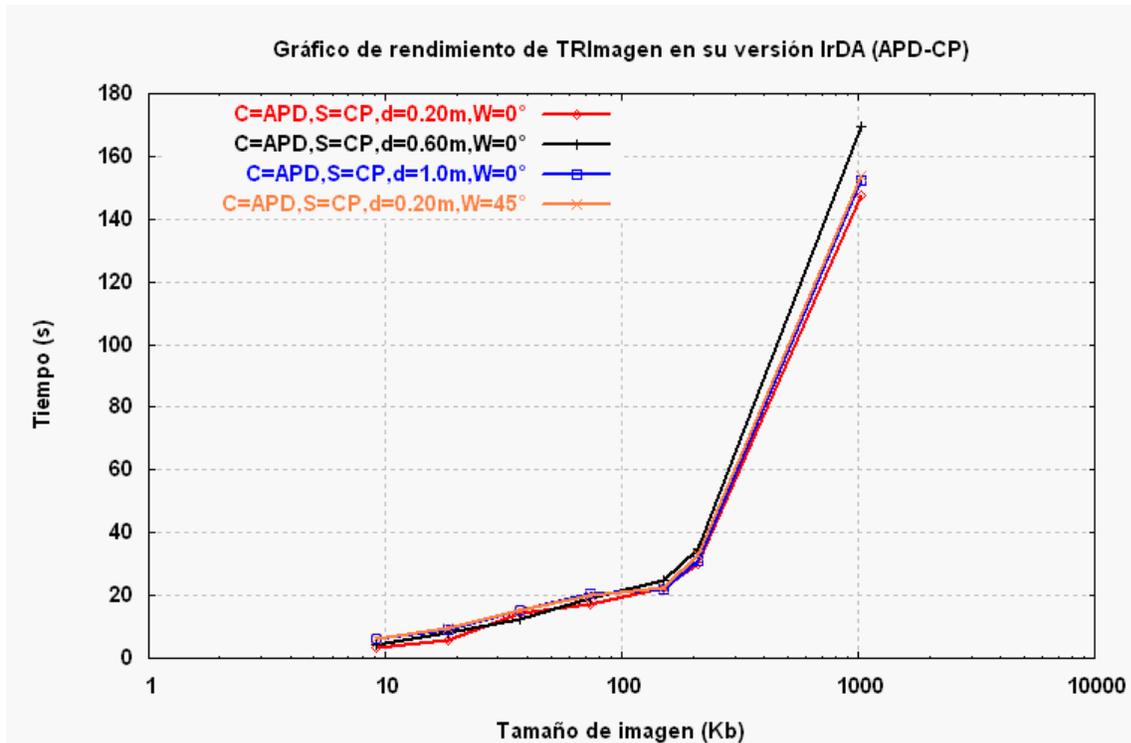


Figura 5.3: Rendimiento de *TRImagen* en IrDA.

- Caso II. Para $C=APD$, $S=APD$, $d=10m$ en diferentes niveles del edificio.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	0.978	0.244
18.44	0.959	0.243
36.88	1.737	0.433
73.76	2.898	0.724
149.62	5.896	1.474
209.48	8.261	2.065
1027.33	38.416	9.604

Tabla 5.15: Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante *Bluetooth* cuando los dispositivos se ubicaron en pisos distintos en el mismo edificio.

En el caso de las pruebas llevadas a cabo con *Bluetooth*, los resultados fueron también aceptables, ya que en primera instancia se mejoraron los resultados obtenidos con la versión IrDA (tal y como era de esperarse). Los incrementos en tiempo se dan de nueva cuenta al incrementarse el volumen de datos (con imágenes más grandes) que se transmiten/reciben entre un par de dispositivos así como por la distancia a

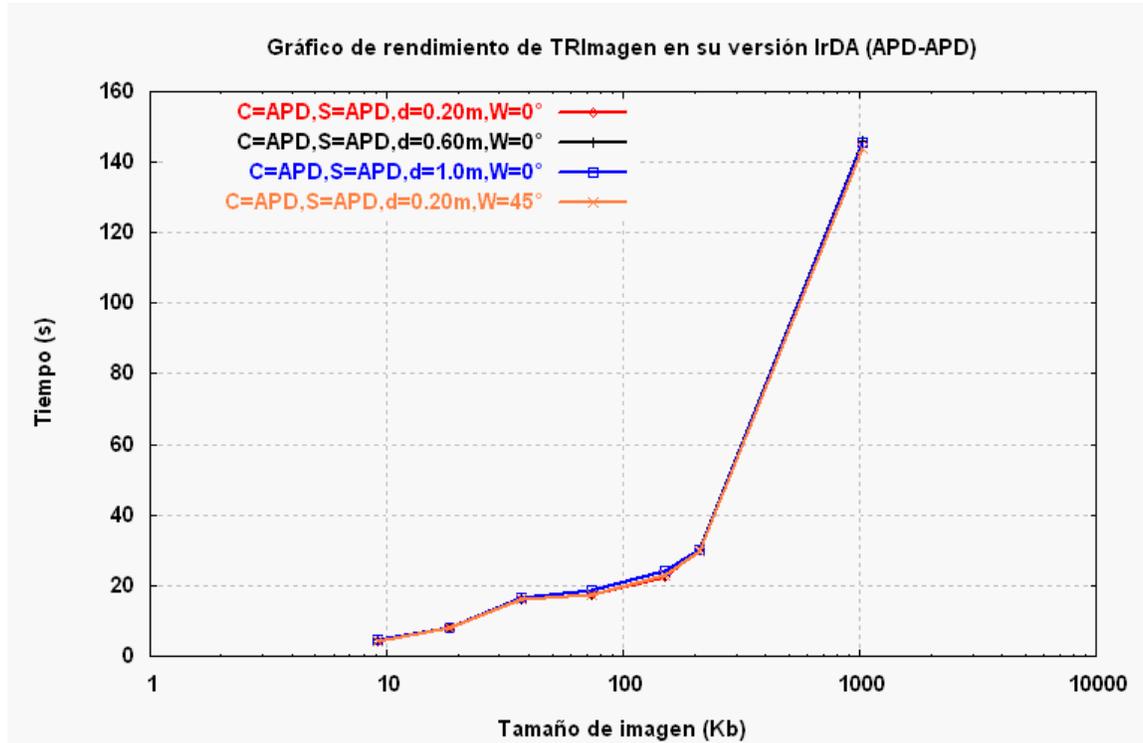


Figura 5.4: Rendimiento de *TRImagen* en IrDA.

la cual se ubicaban dichos dispositivos, así entonces si se intercambiaba una imagen entre dos dispositivos *Bluetooth*, ubicados en el mismo recinto, el tiempo consumido es menor que si los dispositivos se ubican en recintos o pisos diferentes.

5.3.3. Resultados experimentales de *TRImagen* en su versión para IEEE802.11b

Los resultados se muestran en las tablas 5.14-5.21 respectivamente. Y el gráfico comparativo de rendimiento y ancho de banda se muestra en la figuras 5.10 y 5.11 respectivamente.

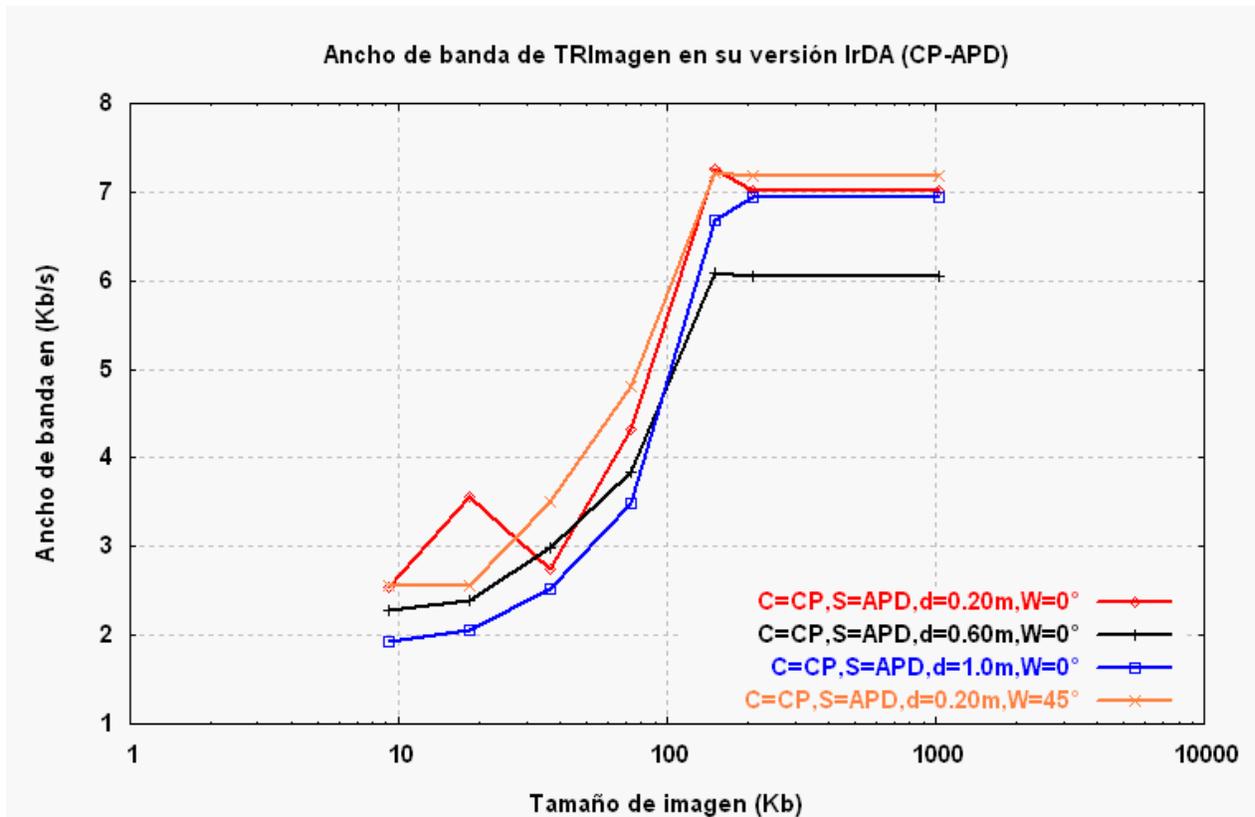


Figura 5.5: Ancho de banda de *TRImagen* en IrDA.

- Caso I. Para $C=CP$, $S=APD$, $d=10m$ en el mismo nivel del edificio.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22		0.166
18.44	0.195	0.051
36.88	0.245	0.061
73.76	0.735	0.179
149.62	1.47	0.367
209.48	2.036	0.509
1027.33	9.95	2.487

Tabla 5.16: Resultados obtenidos al transferir diferentes tamaños de imágenes mediante IEEE802.11b cuando los dispositivos se ubicaron a una distancia de 10m al mismo nivel del mismo edificio.

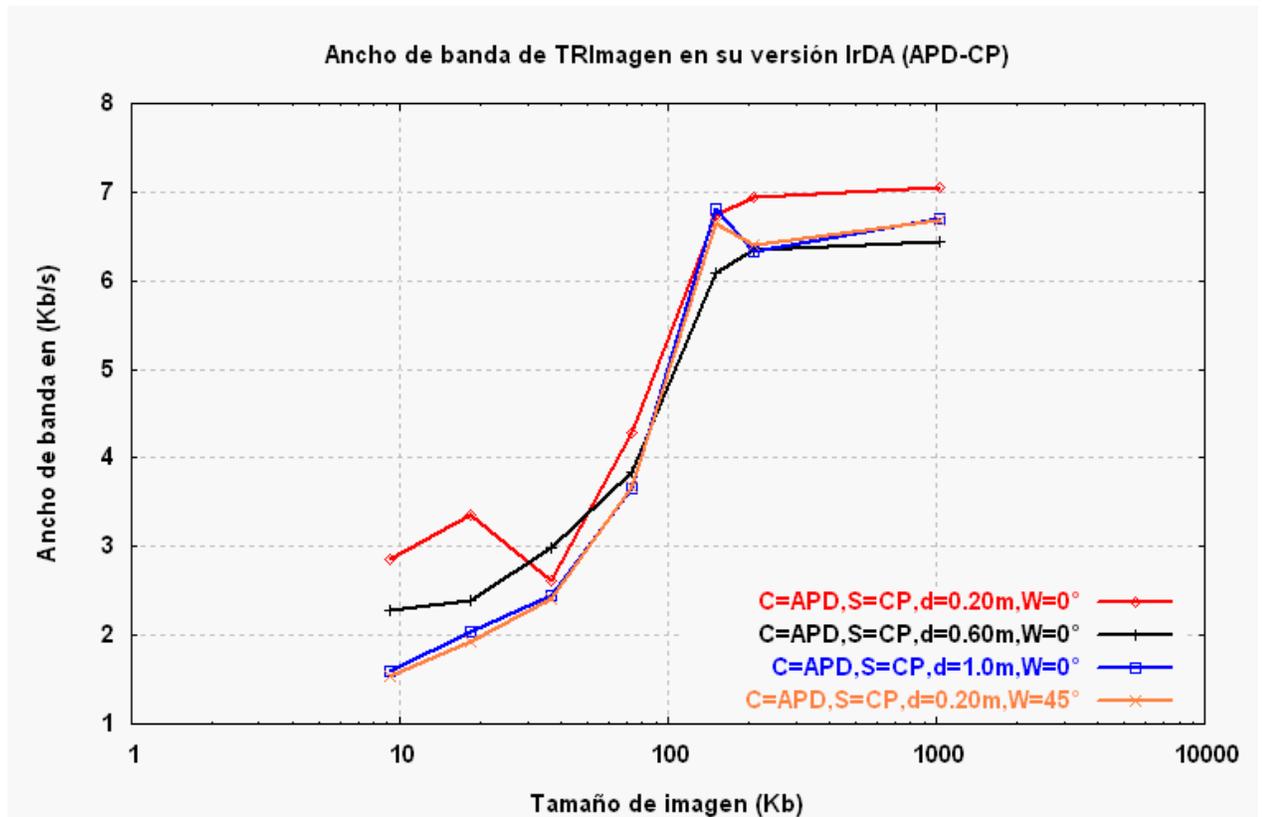


Figura 5.6: Ancho de banda de *TRImagen* en IrDA.

- Caso II. Para $C=APD$, $S=CP$, $d=10m$ en el mismo nivel del edificio.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	0.135	0,03375
18.44	0.254	0.063
36.88	0.409	0.102
73.76	0.801	0.200
149.62	1.564	0.391
209.48	2.065	0.516
1027.33	7.54	1.885

Tabla 5.17: Resultados obtenidos al transferir diferentes tamaños de imágenes mediante IEEE802.11b cuando los dispositivos se ubicaron a una distancia de 10m al mismo nivel del mismo edificio.

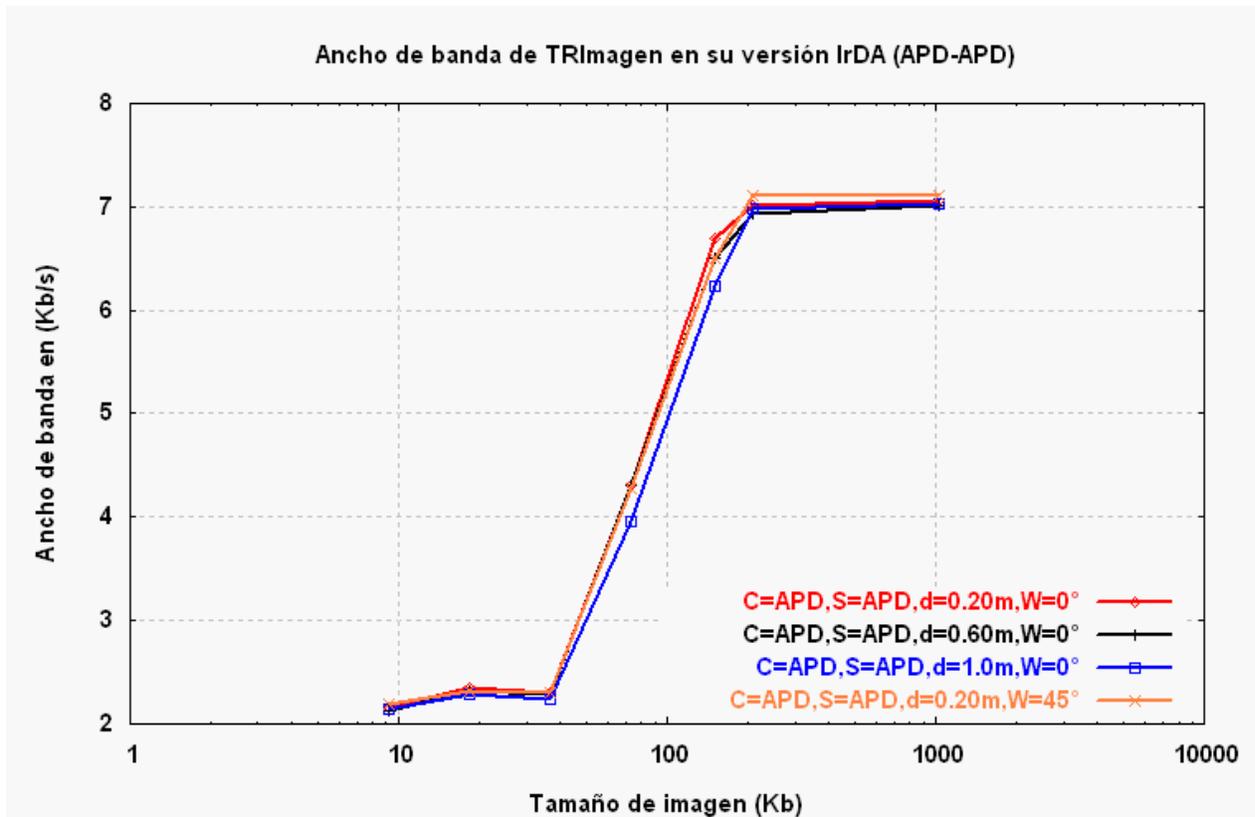


Figura 5.7: Ancho de banda de *TRImagen* en IrDA.

- Caso III. Para $C=CP$, $S=APD$, $d=10m$ (diferentes niveles del edificio).

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	1.944	0.487
18.44	3.152	0.789
36.88	7.186	1.792
73.76	12.971	3.242
149.62	16.667	4.063
209.48	25.363	6.503
1027.33	124.371	31.249

Tabla 5.18: Resultados obtenidos al transferir diferentes tamaños de imágenes mediante IEEE802.11b con los dispositivos ubicados en diferentes niveles del mismo edificio

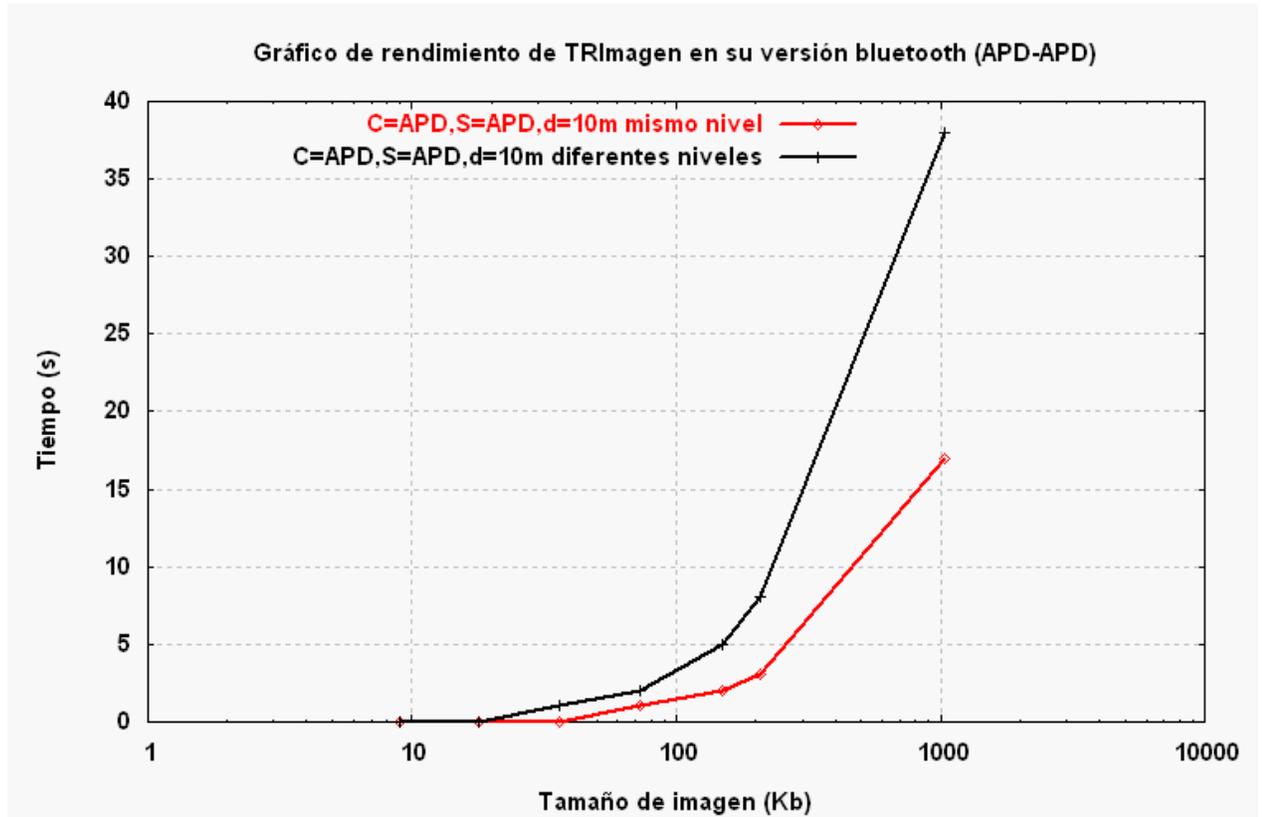


Figura 5.8: Rendimiento de *TRImagen* en su versión para *Bluetooth*.

- Caso IV. Para $C=APD$, $S=CP$, $d=10m$ (diferentes niveles del edificio).

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	0.94	0.98
18.44	3.15	3.23
36.88	7.18	7.41
73.76	15.87	16.26
149.62	16.66	17.81
209.48	25.31	27.31
1027.33	124.11	133.93

Tabla 5.19: Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante IEEE802.11b cuando los dispositivos se ubicaron en pisos distintos en el mismo edificio.

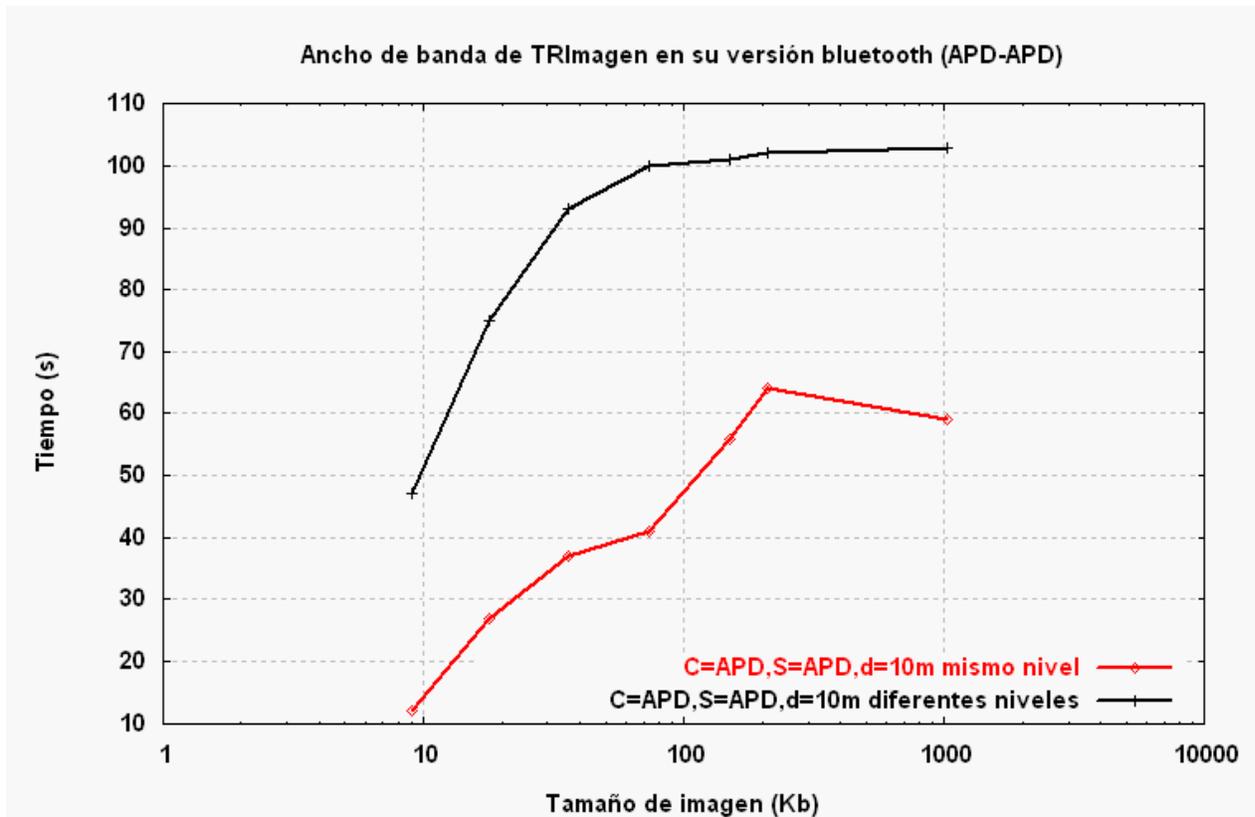


Figura 5.9: Ancho de banda *TRImagen* en su versión para *Bluetooth*.

- Caso V. Para $C=APD$, $S=APD$, $d=10m$ en el mismo nivel del edificio.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	0.245	0.06125
18.44	0.3908	0.100
36.88	0.72358	0.180
73.76	1.45715	0.355
149.62	2.47596	0.604
209.48	3.07436	0.788
1027.33	12.05	2.939

Tabla 5.20: Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante IEEE802.11b cuando los dispositivos se ubicaron en pisos distintos en el mismo edificio.

- Caso VI. Para $C=APD$, $S=APD$, $d=10m$ en diferentes niveles del edificio.

Tamaño de imagen (kb)	Tiempo s/compresión (s)	Tiempo c/compresión (s)
9.22	2.889	0.740
18.44	4.0965	1.001
36.88	8.6878	2.119
73.76	20.7897	5.197
149.62	23.3496	5.695
209.48	30.2936	7.389
1027.33	160.78	40.175

Tabla 5.21: Resultados obtenidos al transferir diferentes tamaños de imágenes entre los dispositivos mediante IEEE802.11b cuando los dispositivos se ubicaron en pisos distintos en el mismo edificio.

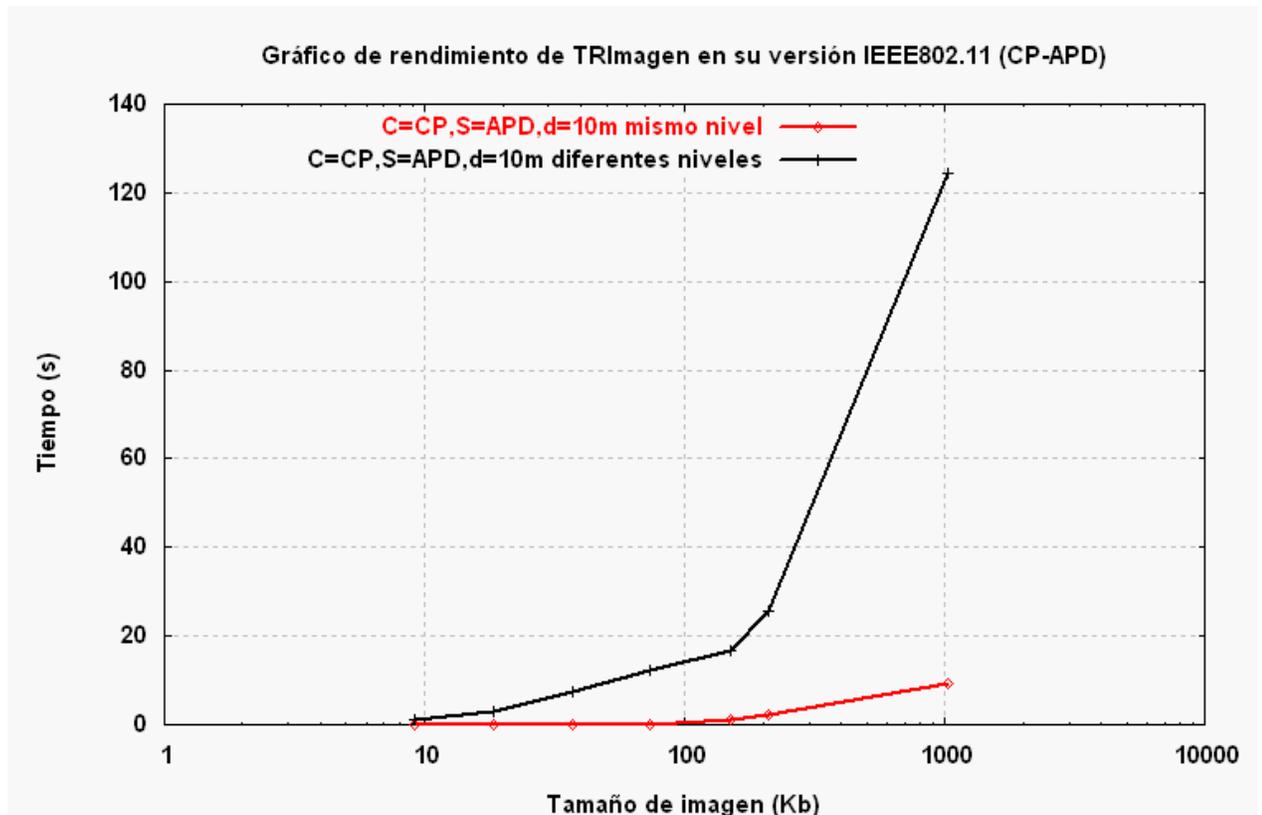


Figura 5.10: Rendimiento de *TRImagen* en su versión IEEE802.11b.

En este caso en particular, el rendimiento de la aplicación se vio afectada por la compartición del canal de comunicación con otros dispositivos vecinos, ya que la red inalámbrica empleada deja de ser un canal 100 por ciento dedicado para el

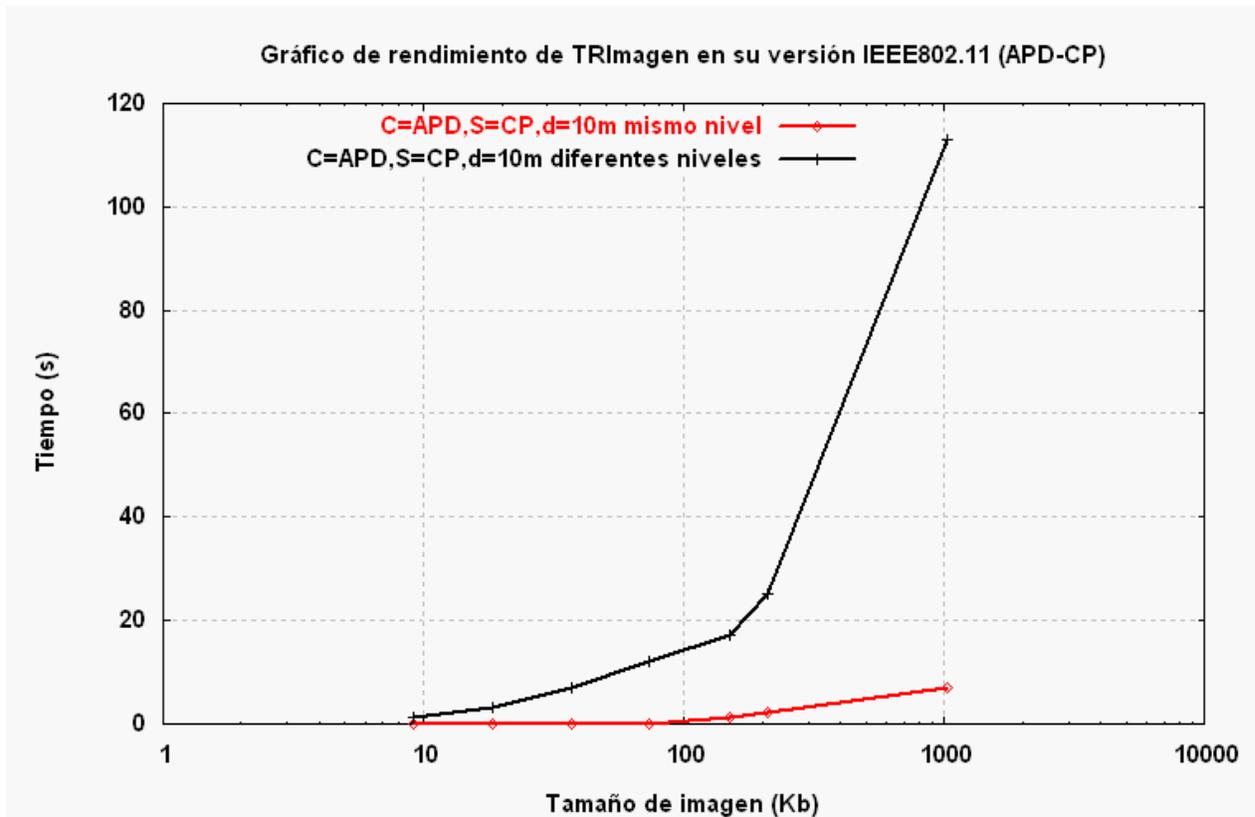


Figura 5.11: Rendimiento de *TRImagen* en su versión IEEE802.11b.

uso de los dispositivos que pretenden comunicarse, además que se hace uso de un punto de acceso inalámbrico que se encuentran a cierta distancia de los dispositivos que pretenden comunicarse. De nueva cuenta el uso de las técnicas de compresión mejoró los resultados.

Comparando las medidas de rendimiento entre la versión para IrDA, *Bluetooth* e IEEE 802.11, éste último presentó un mejor rendimiento a pesar de la compartición del canal de comunicación (o del ancho de banda) y la ubicación del punto de acceso inalámbrico ya que este se ve compensado por la velocidad de transmisión de datos a la que puede operarse bajo el estándar a diferencia de la comunicación infrarroja y de *Bluetooth*, que si bien mostró un mejor rendimiento que la versión para IrDA, no lo hizo así con la versión para WiFi.

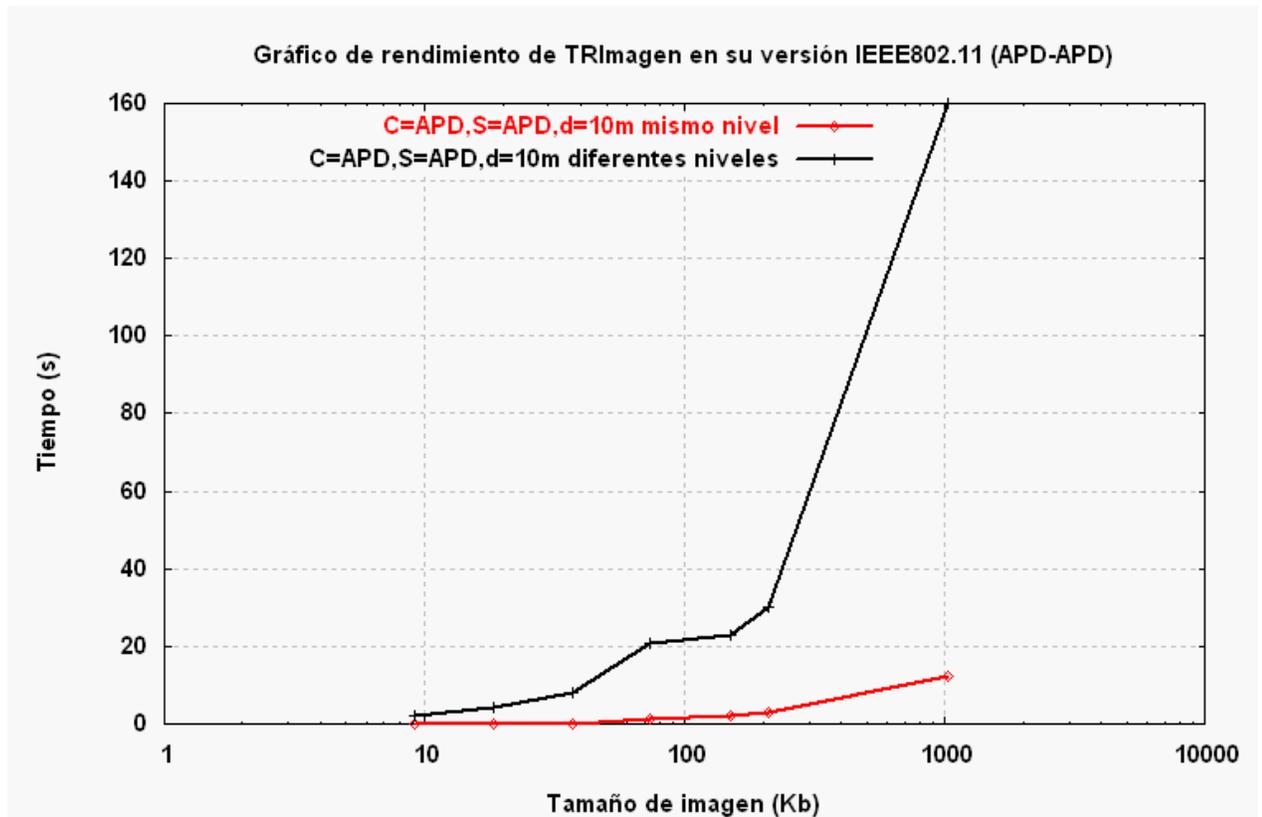


Figura 5.12: Ancho de banda de *TRImagen* en su versión IEEE802.11b.

5.4. Algoritmos de compresión y descompresión de imágenes

Las pruebas llevadas a cabo con el fin de evaluar la efectividad de las técnicas de compresión y descompresión de imágenes implementadas fue la siguiente:

1. **Variación del nivel de descomposición *wavelet*, para 1, 2, 3 y 4 niveles.**
El objetivo de esta prueba, consistió en calcular los parámetros de compresión obtenidos al aplicar consecutivamente la transformada *wavelet* a la imagen de entrada.
2. **Variación del valor de umbral para descartar coeficientes por debajo de este umbral.** En este punto, se despreciaron ciertos valores (coeficientes) cuyo valor absoluto estaba por debajo del valor de umbral. Esta consideración se llevó a cabo para de cada una de las descomposiciones *wavelets*, con el fin de conseguir mayores tasas de compresión, al eliminar detalles que son

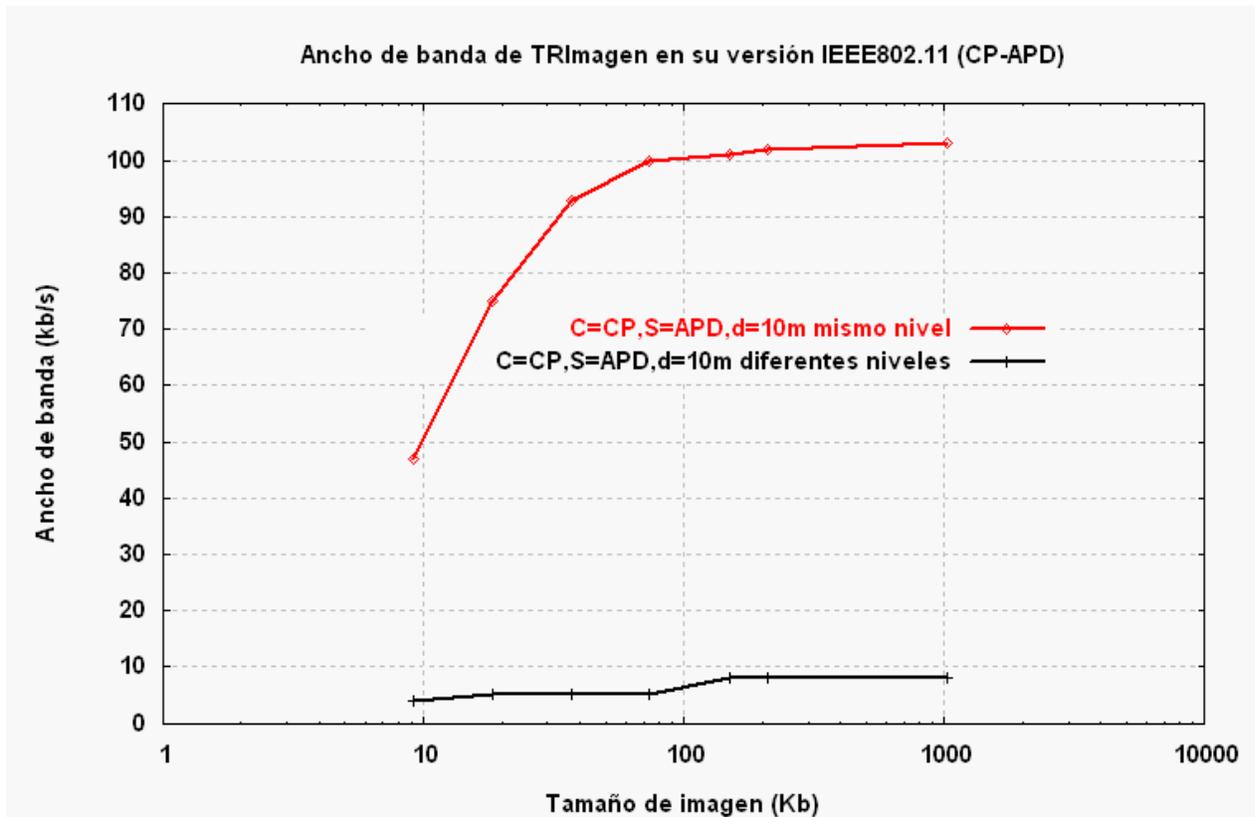


Figura 5.13: Ancho de banda de *TRImagen* en su versión IEEE802.11b.

imperceptibles a nuestra vista y que son representados por coeficientes cuyo valor es despreciado.

3. **Eliminación de los coeficientes de detalles, resultado de la aplicación de la transformada *wavelet*.** Se evaluó el rendimiento cada una de las técnicas implementadas cuando se eliminaban las subimágenes de detalles, resultado de las transformaciones *wavelets*.
4. **Disminución del número de bits de profundidad para representar cada pixel en la imagen de salida.** Su objetivo fue caracterizar a las técnicas de compresión implementadas cuando se especificaba el número de bits con que se representaría cada pixel en la imagen de salida.
5. **Variación del tamaño de la imagen de salida.** Se calcularon los parámetros de compresión obtenidos, cuando se especifica el tamaño deseado de la imagen de salida después del proceso de descompresión.

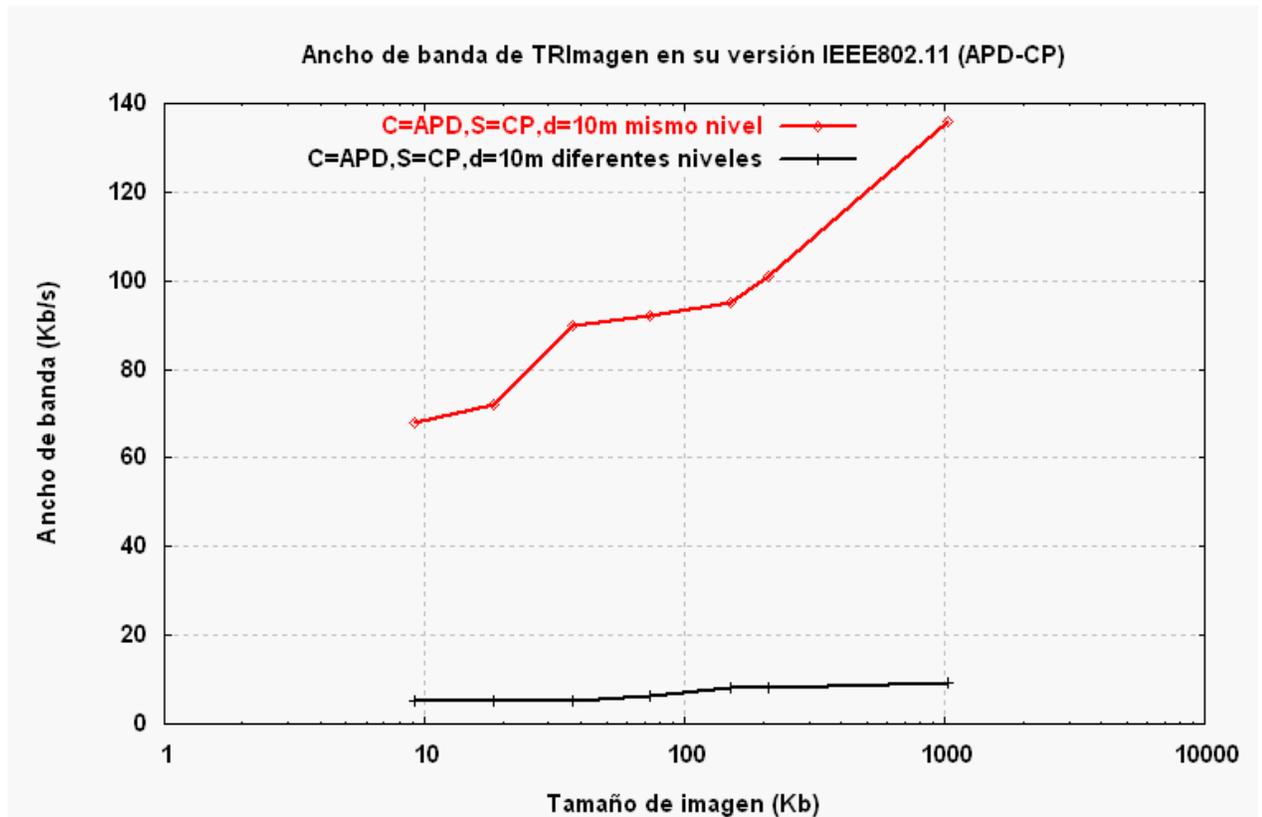


Figura 5.14: Ancho de banda de *TRImagen* en su versión IEEE802.11b.

6. **Variación del nivel de descomposición *wavelet*, del valor de umbral, así como del número de bits de profundidad para representar cada pixel en la imagen de salida.** Se caracterizó a los algoritmos de compresión y descompresión cuando se variaba más de un parámetro durante el proceso de compresión.

7. **Evaluación de la técnica de compresión cuando se le aplica a distinto tipo de imágenes.** La finalidad de esta última prueba, fue lograr la caracterización de la técnica de compresión y descompresión de imágenes digitales, cuando se le aplica a diferentes tipos de imágenes. Es decir, a imágenes de distinta naturaleza, tal es el caso de imágenes de paisajes naturales, de huellas dactilares, de rostros humanos, de tipo sintéticas, de tipo astronómicas, de tipo médicas, entre otras. A partir de esta caracterización, fue posible decidir cuál de las técnicas compresión de imágenes implementadas presentó un mejor rendimiento para un determinado grupo o tipo de imágenes y bajo qué paráme-

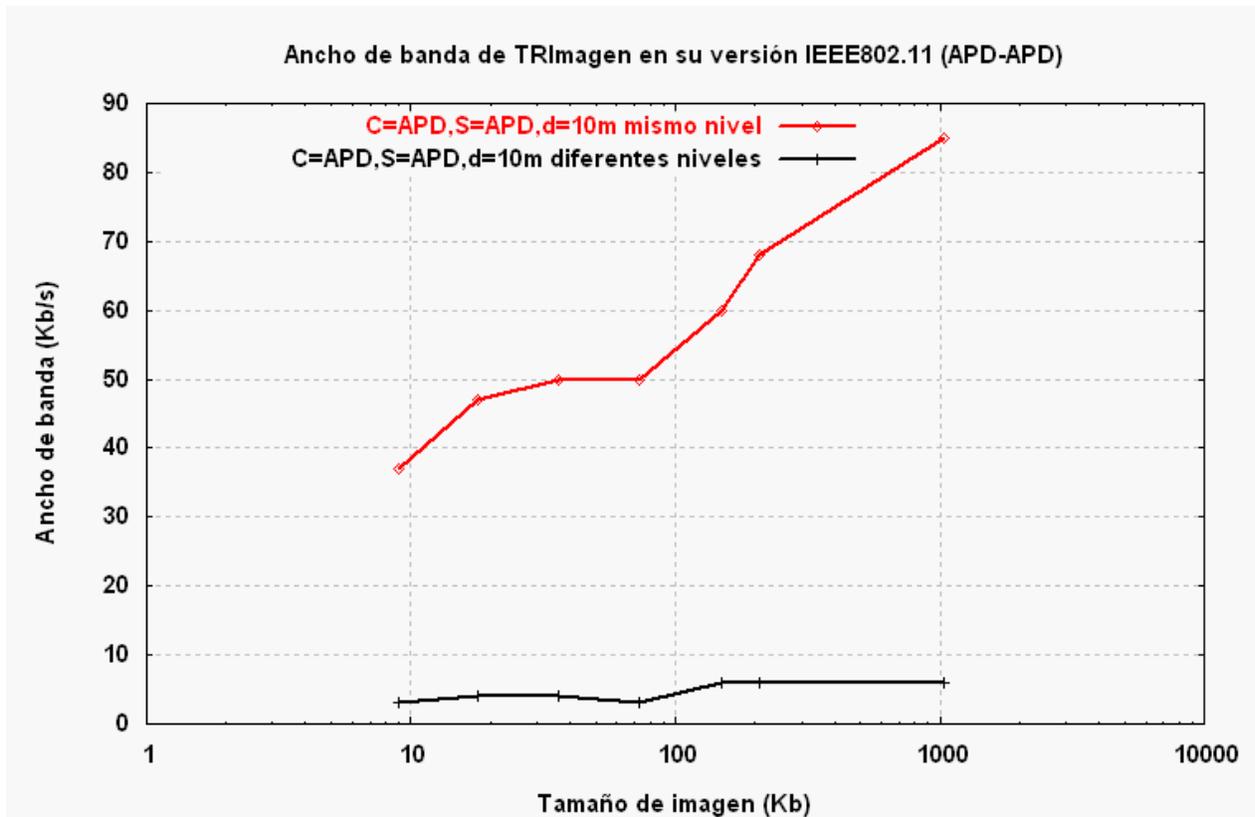


Figura 5.15: Ancho de banda de *TRImagen* en su versión IEEE802.11b.

tros de compresión.

8. **Medición de los tiempos requeridos para llevar a cabo la compresión y descompresión de imágenes tanto en el dispositivo cliente, como en el dispositivo servidor.** Esta última prueba fue llevada a cabo con el fin de medir los tiempos requeridos por cada una de las técnicas de compresión/descompresión de imágenes para comprimir y descomprimir distintos tamaños de imágenes, tanto del lado del cliente como del servidor.

Se tomó como imagen de entrada una versión de *lena* (una imagen de prueba considerada un estándar en materia de procesamiento digital de imágenes, ver anexo 1) en formato BMP a 24 bits/píxel (en el modelo RGB, con 1 byte por componente), mostrada en la figura 5.16. Los resultados de dichas pruebas se resumen en las tablas 5.22, 5.23 y 5.24. En las figuras 5.18-5.21 se presentan los gráficos comparativos correspondientes. Otros resultados e imágenes obtenidas, se muestran en seguida.

Niveles de compresión.	CE (%)	RC	ECM	RPSR
1a. técnica				
1	30.27	5.26	100.14 (U=10)	950.52
2	11.72	28.15	133.47 (U=30)	627.11
3	3.66	63.17	427.79 (U=60)	225.17
4	1.17	190.11	924.12 (U=120)	50.61
2a. técnica				
1	62.21	8.18	58.43 (U=10)	1123.83
2	31.19	43.56	66.14 (U=30)	836.36
3	6.42	71.10	218.43 (U=60)	401.02
4	3.33	410.28	624.12 (U=120)	188.25
3a. técnica				
1	90.05	10.26	50.02 (U=10)	1300.12
2	13.14	55.61	60.94 (U=30)	932.18
3	7.79	117.15	158.76 (U=60)	618.29
4	4.09	523.47	529.17 (U=120)	380.76

Tabla 5.22: Resumen de los resultados obtenidos al probar con las tres técnicas de compresión implementadas.

donde **CE**: porcentaje de conservación de energía, **RC**: razón de compresión, **ECM**: error cuadrático medio y **RPSR**: relación pico-señal a ruido.



Figura 5.16: Imagen tomada como imagen de entrada para llevar a cabo las pruebas 1-7.

donde TIE: tamaño medido en pixeles de la imagen de entrada, TIPO: refiere al tipo de imagen procesada.

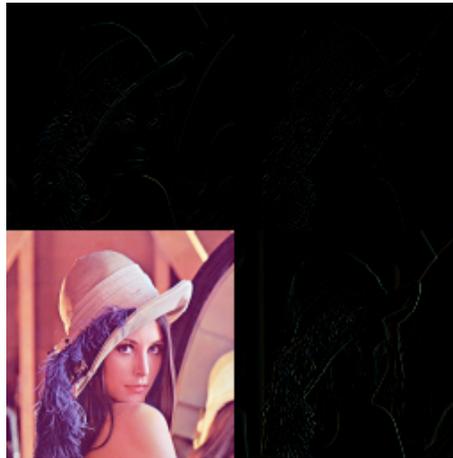


Figura 5.17: Subimágenes, resultado de la primera descomposición *wavelet*, sin considerar ningún valor de umbral para descartar coeficientes. Basada en la transformada Haar.

TIPO	DIE	CE	ECM	RPSR	RC
Patrón simple	450x450	25.96	0.443	135.860	28.17
Patrón complejo	512x512	24.15	831.024	73.467	15.43
Huella dactilar	480x480	34.47	33.017	91.296	10.11
Piramidal	602x602	25.56	77.736	85.742	20.73
Textura	960x960	24.31	330.429	72.791	18.10
Puesta de sol	800x800	25.09	7.364	93.585	27.68
Eclipse	700x700	31.22	27.618	89.280	28.04
Gold hill	512x512	29.32	25.744	84.796	23.49
Superficie lunar	440x440	24.85	44.058	75.072	19.15
Montañas	1024x1024	24.99	20.825	81.216	21.24
Con texto	684x684	24.51	200.796	61.812	10.07
Periódico	1024x1024	24.49	82.434	69.435	11.27
Caricatura	1280x1280	24.94	25.371	79.887	26.38
Rostro sintético	602x602	24.91	13.772	88.727	22.15
Rostro 1	1280x1280	25.01	17.571	94.034	14.71
Rostro 2	500x500	24.91	20.257	97.411	14.60
Fractal	500x500	20.39	2.442	99.883	12.17

Tabla 5.23: Resumen de los resultados obtenidos con la 1a. técnica implementada, con un valor de umbral=30.

Por último en la tabla 5.26 se incluyen los resultados obtenidos al medir los tiempos de compresión/descompresión de imágenes para cada una de las técnicas implementadas, para distintos tamaños de imagen de entrada, donde TIPO 1: 120 x 160 pixeles, TIPO 2: 240 x 320 pixeles, TIPO 3: 480 x 640 pixeles, TIPO 4: 960

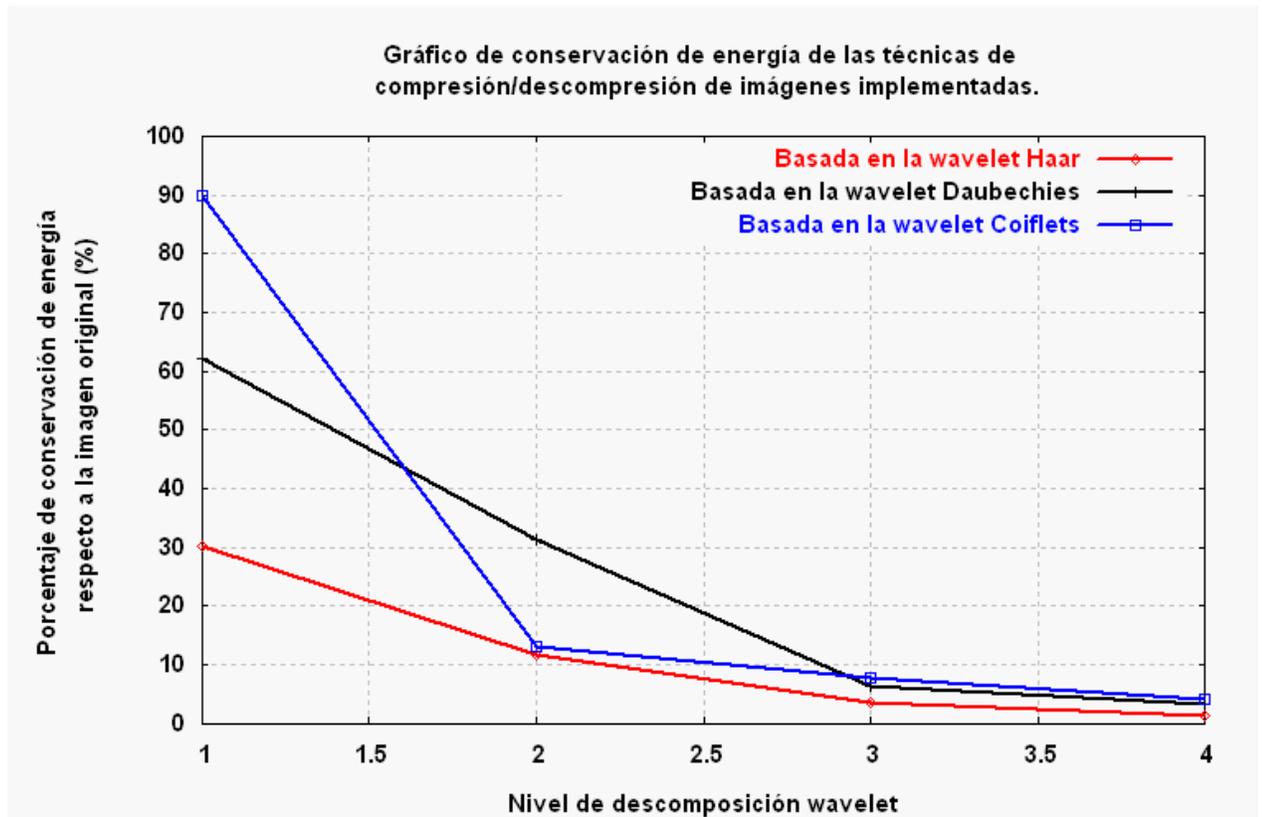


Figura 5.18: Diferencias presentadas entre las tres técnicas de compresión implementadas, respecto al porcentaje de conservación de la energía de la señal de entrada en función del nivel de descomposición *wavelet*.

x 1280 pixeles y TIPO 5: 1920 x 2560 pixeles. Los gráficos correspondientes en las figuras 5.26 y 5.27, respectivamente.

De acuerdo a los resultados obtenidos, es posible destacar los siguientes puntos:

- De las 3 técnicas implementadas y como era de esperarse, la técnica basada en la transformada *wavelet* Haar presentó un mejor rendimiento, cuando es aplicada a señales o imágenes que poseen poco detalle (sin altas frecuencias), a diferencia de cuando se le aplicó a imágenes con más detalles. Lo anterior puede observarse al comparar las tablas 5.23, 5.24 y 5.25, en donde para una misma imagen (patrón simple, una imagen sin detalles), se obtuvo un valor para ECM de 0.443 y un RPSR de 135.60; sin embargo conforme se sometían a prueba imágenes con más detalle, como el caso de la *superficie lunar*, dichos valores se vieron fuertemente afectados, de forma tal que el ECM fue de 44.058 y la RPSR de 75.072. El caso extremo se presentó en la imagen *Con texto*, ya que al con-

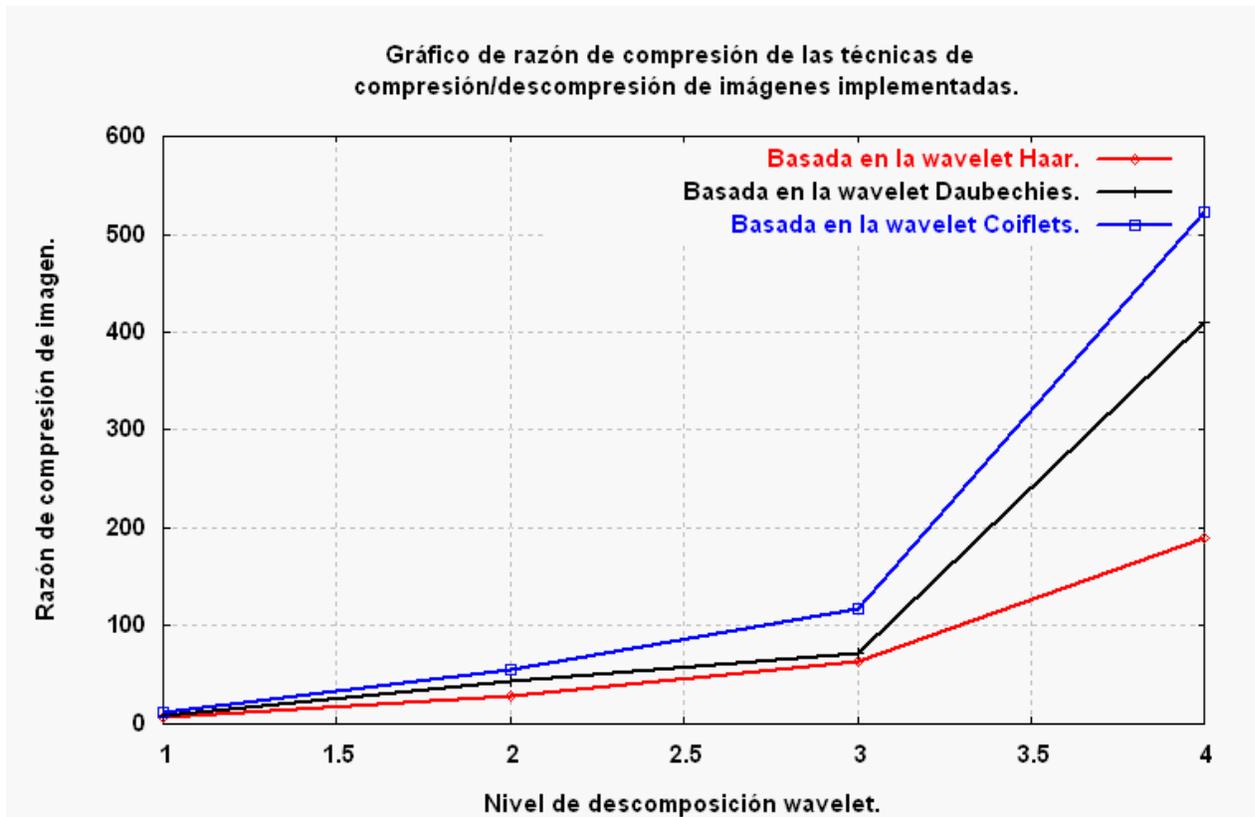


Figura 5.19: Variación de la razón de compresión de las técnicas implementadas, en función del nivel de descomposición *wavelet*.

tener, dicha imagen una gran cantidad de detalles y perderse los mismos en el proceso de compresión/descompresión, el valor de ECM fue de 200.796 (un valor muy alto para esta prueba) y su correspondiente valor de RPSR de 61.812 (un valor muy bajo para esta prueba). En los casos anteriores, Daubechies presentó un mejor rendimiento, ya que dicha transformada permite la existencia de un suavizamiento. Por lo anterior, mientras se procesen imágenes con mayor número de detalles, las transformadas Daubechies mostrarán un mejor desempeño. Sin embargo, en el caso de la transformada Coiflets, ésta mejora en buena medida el rendimiento de la transformada Daubechies, ya que esta última se ve afectada cuando se incrementa la tasa de compresión. Dado los resultados obtenidos en la evaluación de las tres técnicas de compresión y descompresión de imágenes digitales, se tiene que la técnica que presenta un mejor rendimiento es la basada en la transformada Coiflets, seguida de Daubechies y por último, se ubica la transformada *wavelet* Haar, que si bien es cierto que

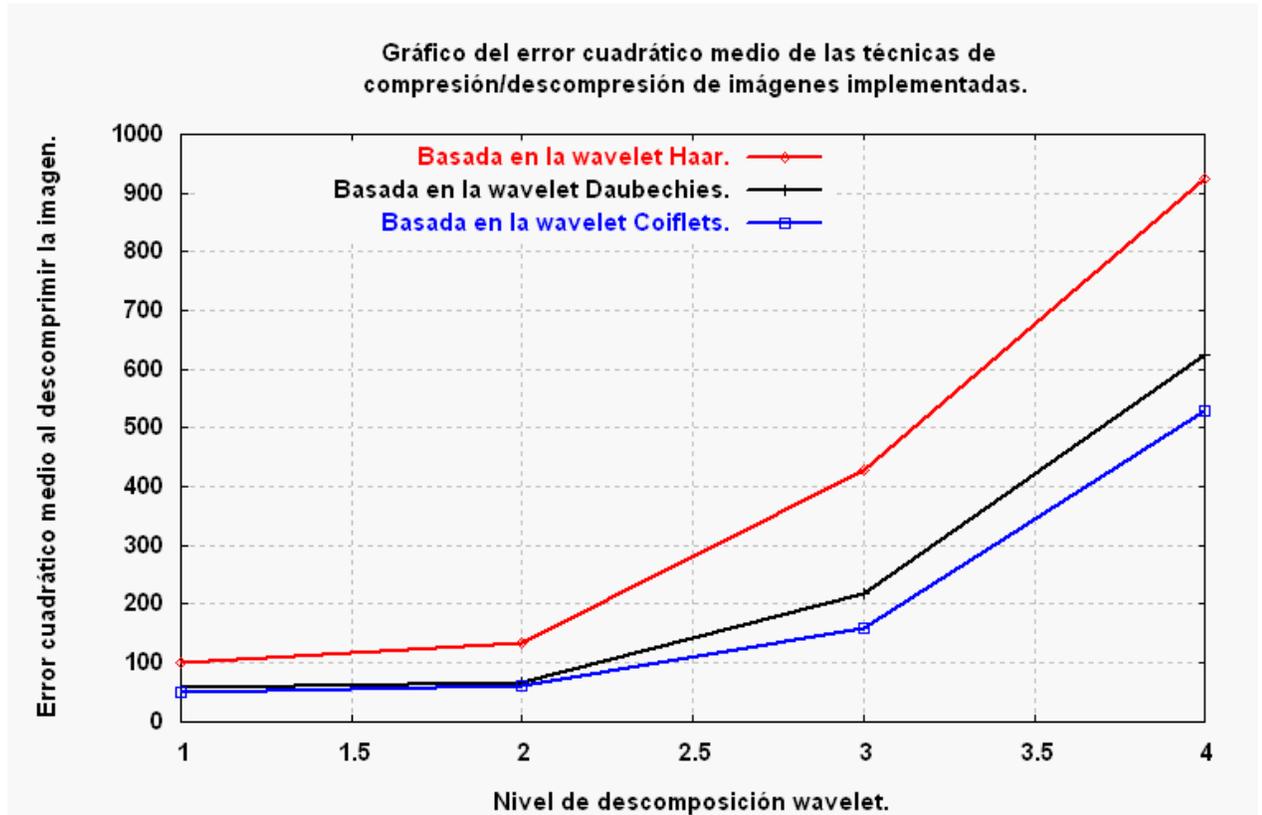


Figura 5.20: Variación del error cuadrático medio al recuperar la imagen de entrada en función de un valor de umbral, mediante el cual, se descartan ciertos coeficientes resultado de las transformaciones.

ofrece el peor rendimiento de la familia diversa de *wavelets*, es la más simple y sencilla de implementar, por lo que es una buena elección para usarse sobre dispositivos móviles que como se ha remarcado a lo largo del presente trabajo poseen características computacionales muy limitadas, por lo que su uso queda justificado. Cabe mencionar que conforme se incrementaba el número de niveles de la descomposición *wavelet*, la mejoría en los parámetros de compresión (ECM, RPSR, conservación de la energía, etc.) se hacía cada vez menos importante, razón por la que se decidió aplicar el proceso de descomposición *wavelet* hasta un segundo nivel (máximo de 3 niveles).

- Respecto a la capacidad de preservar un alto porcentaje de energía después de la aplicación de las transformadas *wavelets*, como se puede observar en la figura 5.18, que para una primera aplicación de la transformación *wavelet* la técnica que conserva un mayor porcentaje de energía, respecto a la imagen

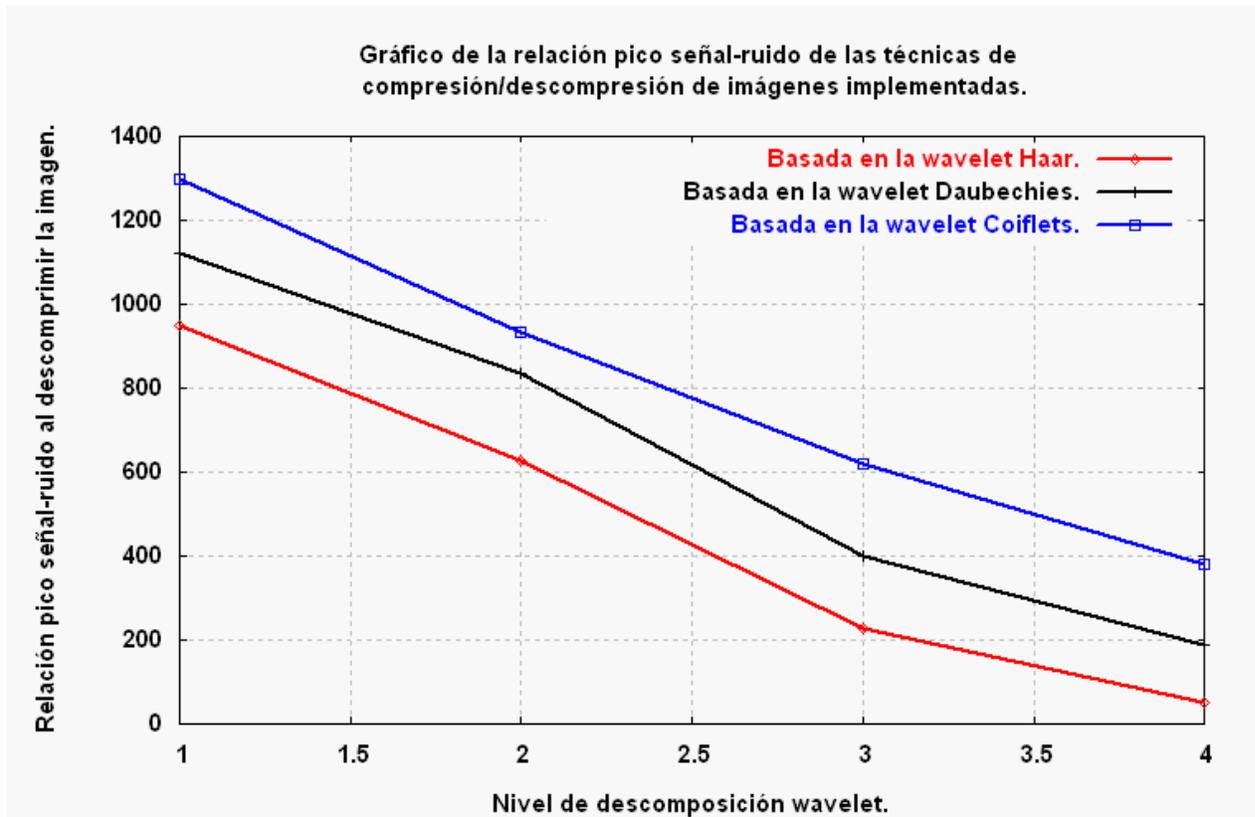


Figura 5.21: Variación de la relación pico-señal a ruido después del proceso de descompresión de la imagen comprimida, en función de un valor de umbral.

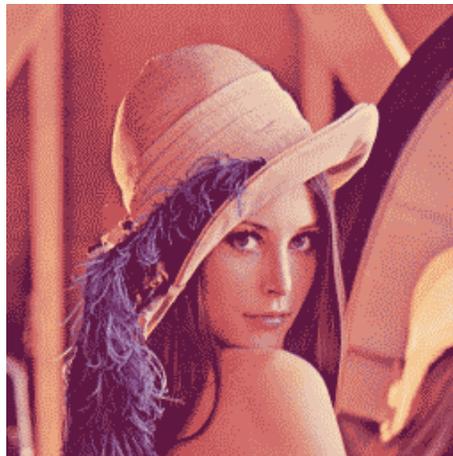


Figura 5.22: Imagen recuperada sin considerar un valor de umbral, a un primer nivel de descomposición *wavelet*, con 4 bpp. basada en la transformada Daubechies D-4.



Figura 5.23: Imagen recuperada sin considerar un valor de umbral, a un primer nivel de descomposición *wavelet*, con 1 bpp. basada en la transformada Coiflets C-6.

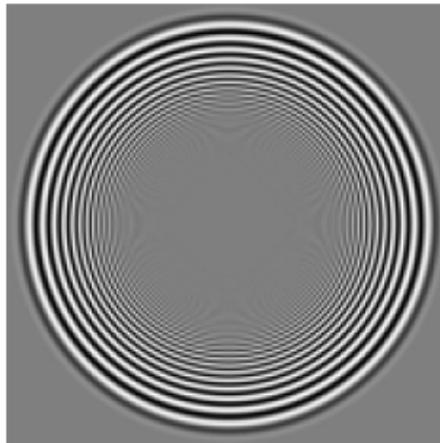


Figura 5.24: Imagen de entrada en formato RGB a 3 bytes/píxel de tamaño 800 x 800 píxeles.

original, es la técnica basada en la transformada *wavelet* Coiflets, seguida de la basada en Daubechies y al último la basada en Haar; sin embargo éstas diferencias se hacen mínimizan conforme se incrementa el número de niveles de descomposición *wavelet*.

- En materia de las tasas de compresión conseguidas con las 3 técnicas implementadas y como se muestra en la figura 5.19, para un primer nivel transformación las razones o tasas de compresión son equiparadas, sin embargo, para un segundo o tercer nivel, las diferencias, empiezan a ser mas notorias y así para

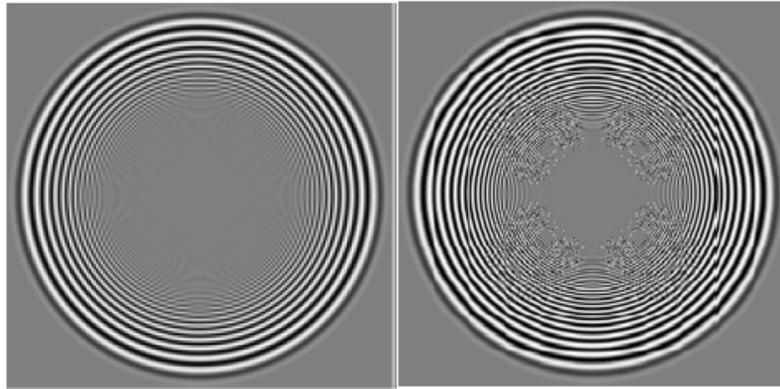


Figura 5.25: Imágenes recuperadas considerando un valor de umbral = 70, para un primer nivel de descomposición *wavelet*. De izq. a derecha: Basada en la transformada Coiflets (C-6) y en Daubechies (D-4).

TIPO	DIE	CE	ECM	RPSR	RC
Patrón simple	450x450	36.95	0.3430	145.7702	34.23
Patrón complejo	512x512	35.14	731.0247	84.4571	17.10
Huella dactilar	480x480	45.43	23.0174	100.1760	12.39
Piramidal	602x602	36.55	67.7364	107.5524	25.35
Textura	960x960	35.30	230.4293	84.5300	22.19
Puesta de sol	800x800	36.08	6.3643	108.1233	28.92
Eclipse	700x700	42.21	17.6182	103.6703	29.00
Goldhill	512x512	39.31	15.7446	95.6773	26.14
Superficie lunar	440x440	35.84	34.0585	86.1524	21.17
Montañas	1024x1024	35.98	10.8250	90.1765	23.78
Con texto	684x684	35.50	100.7962	70.9622	12.02
Periódico	1024x1024	35.48	72.4340	77.1359	11.99
Caricatura	1280x1280	35.93	15.3715	90.8774	29.81
Rostro sintético	602x602	35.90	3.7725	99.1279	26.26
Rostro 1	1280x1280	36.00	7.5716	101.0345	15.25
Rostro 2	500x500	35.90	10.2578	106.4116	13.41
Fractal	500x500	23.91	1.546	114.357	15.11

Tabla 5.24: Resumen de los resultados obtenidos con la 2a. técnica implementada, con un valor de umbral=30.

el caso que se aplican 4 transformaciones *wavelets*, la técnica basada con la transformada Coiflets, presentó mayores tasas de compresión que la basada en la transformada Daubechies y más aún, respecto a la basada a la transformada Haar, lo que nos hace situar nuevamente a la técnica basada en la transformada Coiflets, como la más eficiente, y dado que a pesar de estar conformada con

TIPO	TIE	CE	ECM	RPSR	RC
Patrón simple	450x450	66.95	0.2430	147.6602	38.17
Patrón complejo	512x512	65.1471	0.0247	84.5681	23.81
Huella dactilar	480x480	75.43	20.0174	99.4710	14.25
Piramidal	602x602	66.55	60.7364	115.6414	27.98
Textura	960x960	65.30	210.4293	84.5601	26.31
Puesta de sol	800x800	66.08	4.3643	105.3356	30.27
Eclipse	700x700	72.21	12.6182	100.1513	33.42
Goldhill	512x512	69.31	10.7446	107.8754	28.02
Superficie lunar	440x440	65.84	24.0585	96.2734	24.39
Montañas	1024 x 1024	65.98	8.8250	104.1145	26.00
Con texto	684x684	65.50	82.7962	83.6112	15.03
Periódico	1024x1024	65.48	62.4340	99.3379	14.29
Caricatura	1280x1280	65.93	10.3715	119.8874	34.21
Rostro sintético	602x602	65.90	1.7725	118.7279	29.74
Rostro 1	1280x1280	66.00	5.5716	124.0345	17.45
Rostro 2	2500x500	65.90	7.2578	127.4116	19.15
Fractal	500x500	23.91	1.546	114.357	16.27

Tabla 5.25: Resumen de los resultados obtenidos con la 3a. técnica implementada, con un valor de umbral=30.

Imagen.	1a técnica	2a técnica	3a técnica
1	5.163/5.106 ms	93.284/20.423 ms	107.836/31.750 ms
2	27.474/20.705 ms	403.427/37.136 ms	408.403/40.200 ms
3	97.613/126.730 ms	2221.401/148.137 ms	2229.501/162.652ms
4	174.031/169.034 ms	16296.698/583.776 ms	16354.69/642153,6 ms
5	743.676/719.829 ms	138848.061/9443.558 ms	179762.638/ 6655.702 ms

Tabla 5.26: Tiempos de compresión/descompresión de imágenes para las tres técnicas implementadas.

filtros de mayor longitud (de orden 6, a diferencia de la basada en Daubechies de orden 4 y Haar de orden 2), su cálculo es simple por lo es una buena elección para ser implantada en una plataforma de cómputo móvil.

- Además de considerarse medidas cuantitativas, para evaluar la calidad de las imágenes, tal es el caso del ECM y la RPSR, se tomaron en cuenta aspectos subjetivos basados en la percepción visual de la calidad de la imagen, y entonces para una imagen cuya baja calidad era eminentemente notoria, la técnica de compresión que generaba dicha imagen es descalificada como una buena técnica, y por el contrario, cuando se generaba una imagen de alta calidad (o por lo

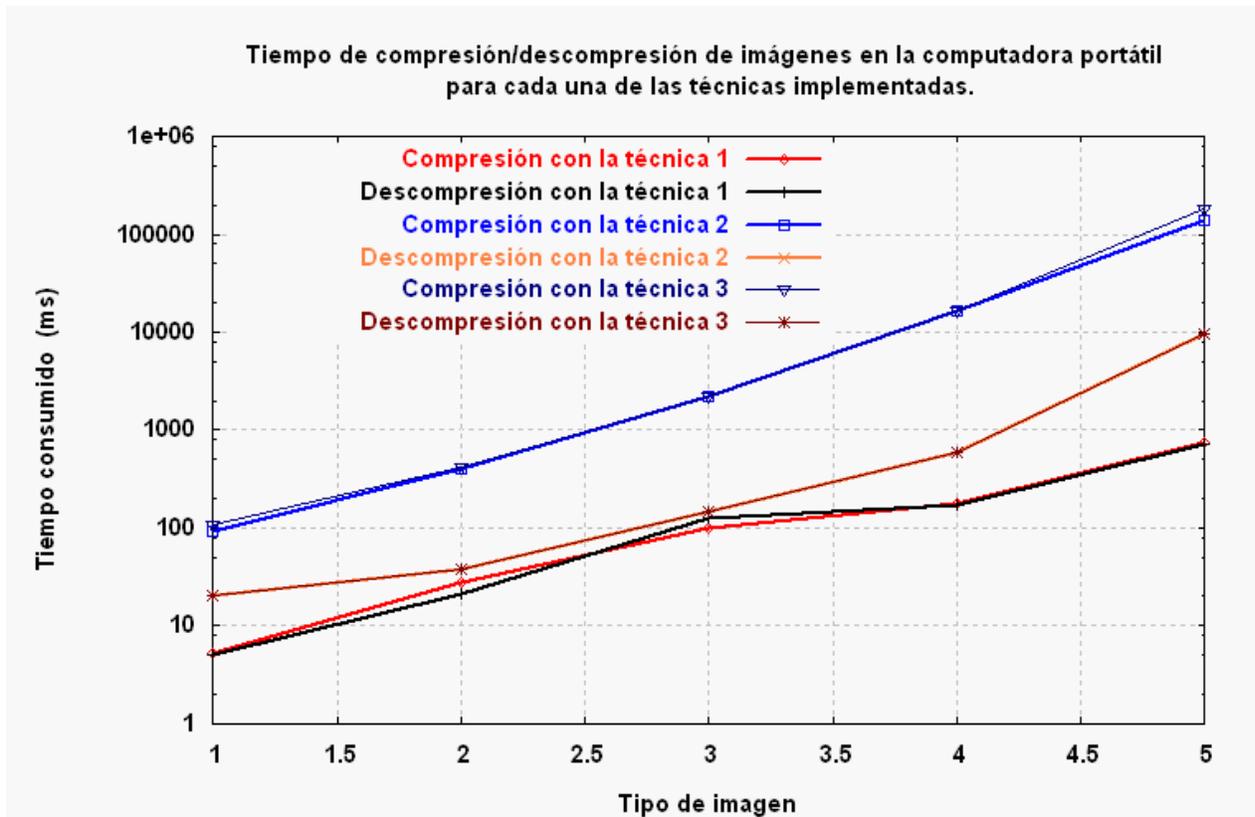


Figura 5.26: Gráfico de rendimiento del módulo de compresión en su versión para la computadora portátil.

menos aceptable), la técnica que la generó es considerada, como una técnica eficiente. Con base a este criterio, nuevamente la técnica que presentó un menor rendimiento fue la basada en la transformada Haar, sin embargo las técnicas basadas en las transformadas *wavelets* Daubechies y Coiflets mostraron un comportamiento muy similar, pero en ambos casos mejor que Haar.

- Los resultados han mostrado que para poder determinar la eficiencia de una técnica de compresión y descompresión de imágenes, es importante considerar el tipo de transformada que se utilice. Así como también, la calidad del cuantizador, el tipo de codificador entrópico, y finalmente la naturaleza de las imágenes, a las cuales la técnica le es aplicada.

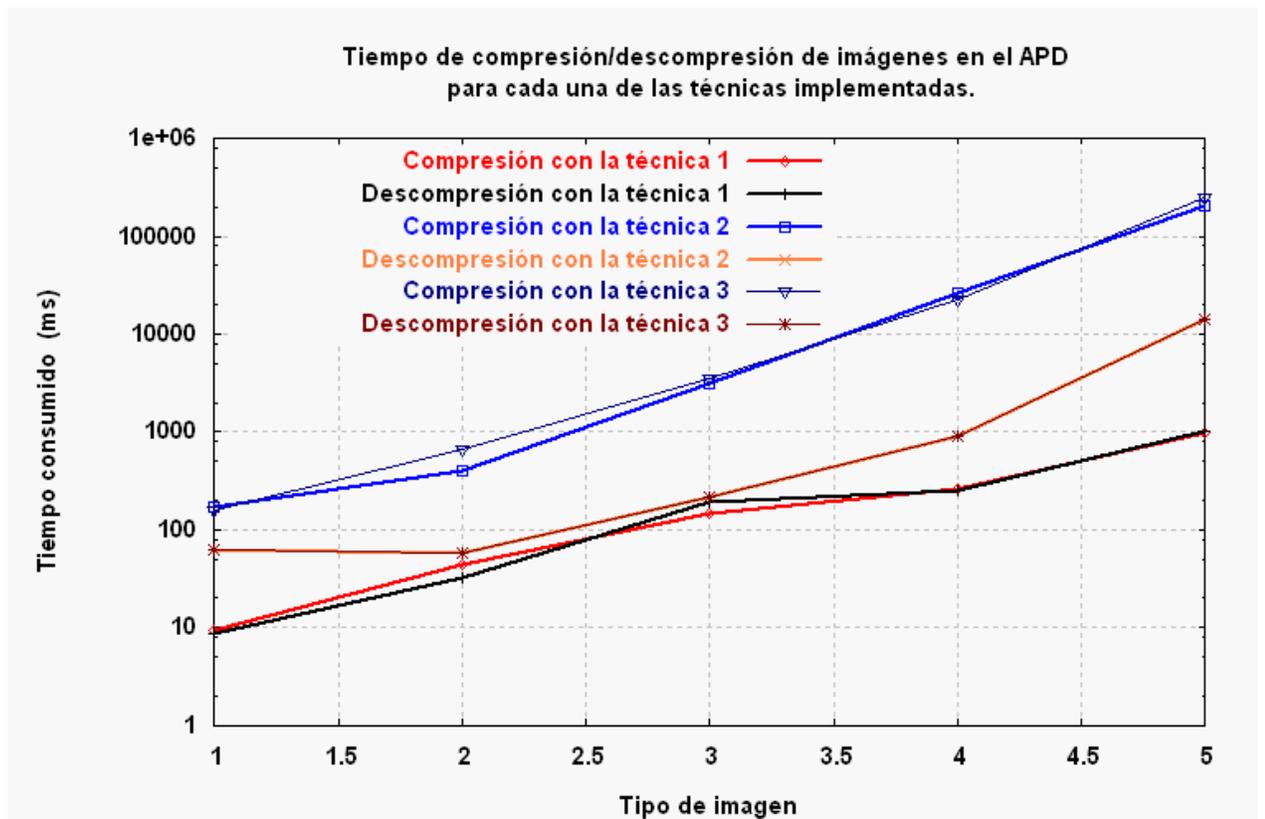


Figura 5.27: Gráfico de rendimiento del módulo de compresión en su versión para el APD.

Capítulo 6

Conclusiones

El desarrollo del presente trabajo consistió en el estudio, implementación, evaluación y caracterización de tres técnicas de compresión y descompresión de imágenes digitales basadas en las transformadas *wavelets*: Haar, Daubechies y Coiflets, para su transmisión en una plataforma de cómputo móvil conformado por una computadora portátil y un asistente personal digital. La comunicación entre los dispositivos involucrados en cada uno de los escenarios estudiados, fue en base a los estándares de comunicación inalámbrica, IrDA (comunicación infrarroja), *Bluetooth* e IEEE802.11b (comunicación WiFi).

Tomando como base los resultados de las pruebas orientadas a evaluar el aspecto de comunicaciones de las implementaciones desarrolladas, es claro que en cada una de las configuraciones propuestas como escenarios de cómputo móvil, el mejor rendimiento siempre fue el ofrecido por los que involucraban al estándar IEEE802.11b, a pesar de que en las configuraciones que hacían uso de este esquema de comunicación inalámbrica siempre se utilizaba un punto de acceso inalámbrico, lo que imposibilitaba que se dedicara el canal de comunicación al cien por ciento para la comunicación de los dispositivos móviles involucrados. Sin embargo dado el ancho de banda que es capaz de alcanzarse con esta tecnología inalámbrica, este hecho fue superado. En segundo término, la que ofreció un mejor rendimiento fue el esquema basado en *Bluetooth*, en dónde si bien no mejoró el rendimiento del caso anterior, fue capaz de mejorar los resultados obtenidos con el esquema basado en la comunicación infrarroja.

Fue importante considerar las diferentes configuraciones entre los dispositivos móviles (entre la computadora portátil y el APD) con los distintos esquemas de comunicación inalámbrica citados previamente. De esa manera se cubrieron todas

las posibles opciones de escenarios de cómputo móvil (entre estos dos dispositivos) que en aplicaciones reales podrían ser requeridos por los usuarios.

Por otro lado es claro que la revolución multimedia ha traído consigo el uso exhaustivo del espacio de almacenamiento y recursos de transmisión de datos, principalmente de tipo gráfico como las imágenes, entonces para hacer práctico el uso de imágenes digitales para ser procesadas en forma eficiente en dispositivos móviles, es necesario introducir técnicas de compresión de datos, tal y como se plantea en el presente trabajo.

Se optó por el uso del análisis *wavelet* para desarrollar el marco de compresión, ya que dada las limitadas características computacionales (principalmente del APD) de los dispositivos móviles ligeros y los bajos anchos de banda con que trabajan sus comunicaciones, se requerían de esquemas de compresión eficientes, que consiguieran altas tasas de compresión con una alta preservación del contenido de las imágenes (poca degradación de las imágenes) y que computacionalmente fueran simples de calcular. Las herramientas *wavelets*, poseen todas estas características. Respecto a las tres técnicas implementadas, la basada en la transformada Haar presentó un mejor rendimiento, cuando se le aplicó a imágenes que poseían poco detalle, a diferencia de cuando se le aplicó a imágenes con mucho detalle. En estos casos, la basada en Daubechies presentó un mejor rendimiento, aunque aún mejor fue el rendimiento de Coiflets. El análisis *wavelet* promete seguir siendo en los próximos años, una idea competente acorde las nuevas necesidades por parte de las nuevas aplicaciones, arquitecturas y usuarios. Sin embargo, poder determinar la eficiencia de una técnica de compresión de imágenes, es también importante considerar la calidad del cuantizador, el tipo de codificador entrópico, y finalmente la naturaleza de las imágenes, a las cuales, la técnica le es aplicada.

Es importante evaluar la eficiencia de una técnica de compresión en base a medidas objetivas, como el error cuadrático medio y la relación pico de la señal a ruido, porque ciertamente dan una idea cuantitativa de qué tanta distorsión (o qué tan eficiente) resulta ser una técnica. Sin embargo, las medidas subjetivas, basadas en la percepción visual también son importantes ya que a partir de ella (mediante la opinión de un usuario), es posible decidir si una técnica de compresión es eficiente o no, en función de qué tanta distorsión visual le agrega a una imagen comprimida o una vez descomprimida. Otra razón por la que deben considerarse siempre medidas subjetivas es que no siempre para el caso en que se comprime/descomprime una imagen, y la medida del error cuadrático medio es bajo, necesariamente la imagen

en su calidad perceptual será muy buena, tal y como se esperaba.

Acorde a los resultados obtenidos en la evaluación del módulo de comunicaciones entre los dispositivos móviles, los tiempos de ejecución fueron mejorados al hacer uso de las técnicas de compresión y descompresión de imágenes, ya que se enviaban imágenes adecuadas (comprimidas y/o ajustadas en niveles de resolución y/o redimensionadas) al dispositivo destino, y así se evitaba enviar una imagen de muy alta resolución o de gran tamaño a un dispositivo cuyas características de visualización gráfica son pobres. Ciertamente al involucrar a las técnicas de compresión, se llevaba a cabo un trabajo adicional que en algunos casos es considerable (principalmente cuando se comprime/descomprime una imagen de gran tamaño en el APD), reflejándose esta carga adicional de trabajo en los tiempos consumidos. Sin embargo, este proceso de compresión en el dispositivo origen, únicamente podría ser llevado a cabo la primera ocasión en que cierta imagen se eligiera para enviarse al dispositivo destino. De igual forma, en el dispositivo destino el proceso de descompresión podría efectuarse únicamente en la primera ocasión.

Con este trabajo se facilita la transmisión de datos de tipo multimedia (particularmente imágenes) entre dispositivos móviles y como la incursión de datos de tipo multimedia son cada vez más demandado en servicios como WEB y las videoconferencias, se contribuye en el desarrollo de la WEB. Con lo anterior, afirmamos que los objetivos planteados en esta tesis, fueron cumplidos.

6.1. Contribuciones

El trabajo desarrollado consiste de un producto de aplicación tecnológica acorde a las necesidades de comunicaciones inalámbricas actuales entre dispositivos móviles ligeros, por lo que toma muy en cuenta los bajos ancho de banda a los que operan y las limitadas características computacionales de tales dispositivos. Con esto se contribuye con uno de los retos vigentes en materia de procesamiento digital de señales, y en particular de imágenes, desde el punto de vista de dispositivos móviles ligeros.

A continuación se presentan las características más importantes las aplicaciones desarrolladas:

- La transmisión y recepción eficiente de imágenes digitales entre los dispositivos móviles es posible, mejorando el rendimiento de la arquitectura de cómputo móvil, al procesar imágenes digitales comprimidas (con menos cantidad de
-

bytes), con el fin de aprovechar de mejor forma el ancho de banda del canal de comunicación inalámbrico.

- Dada la baja complejidad computacional de las técnicas implementadas, la exigencia de recursos computacionales requeridos para su funcionamiento es severamente disminuida, situación acorde para la plataforma de cómputo considerada.
 - Las aplicaciones desarrolladas consideran la diferencias de poder computacional entre los dispositivos, principalmente en el caso del APD. Diferencias tales como, velocidad de procesamiento, capacidad de visualización, cantidad de memoria, etc. Así se evita que una imagen de gran tamaño y/o de alta resolución, sea enviada sin ser previamente comprimida y/o redimensionada, a un dispositivo cuya capacidad de visualización de información gráfica y demás características son limitadas.
 - Se garantiza la transmisión y recepción de las imágenes de un dispositivo origen al dispositivo destino, mediante el protocolo de comunicaciones implementado bajo el paradigma cliente-servidor y orientado a conexión, con lo que se asegura que los dos dispositivos crean un canal de comunicación común, antes de que la transferencia de datos se inicie.
 - Se ha desarrollado un módulo que implementa tres técnicas de compresión y descompresión de imágenes digitales basadas en las transformadas *wavelets*, cuya reutilización o extensión es posible gracias a su naturaleza modular orientada a objetos. Lo mismo sucede con el módulo de comunicaciones. Se presenta TRImagen, como una aplicación ejemplo, que hace uso de los módulos desarrollados.
 - Se ha desarrollado el trabajo con el fin de mejorar el intercambio de imágenes entre dispositivos móviles mediante el uso de estándares por excelencia de comunicación inalámbricos. El trabajo fue encaminado a la búsqueda del mejor aprovechamiento del canal de comunicación, sin embargo, otro posible enfoque podría ser abordar la misma problemática tratada en la presente tesis, pero considerando aspectos orientados a la optimización del aprovechamiento de la energía consumida por los dispositivos móviles, por citar un ejemplo.
-

6.2. Trabajo a futuro

A pesar de que con el desarrollo del presente trabajo, se ha cumplido con los objetivos planteados al principio, un posible trabajo que podría desarrollarse sería abordar la misma problemática, pero desde otros puntos de vista, es decir, resolver el problema de la transmisión, compresión y descompresión de imágenes digitales en forma eficiente en un ambiente de cómputo móvil, pero en vez de buscar el aprovechamiento óptimo del canal de comunicación, minimizar el consumo de energía, la velocidad de procesamiento, etc., de cada uno de los dispositivos móviles ligeros.

Otra posible línea sería llevar a cabo una exploración exhaustiva con otro tipo de transformaciones *wavelets* de la gran diversidad de familias existentes. Adicionalmente, como se espera que cada día más los dispositivos móviles posean mayores capacidades de cómputo, las grandes limitaciones en cuanto a la utilización de técnicas más robustas (y complejas) se verán disminuidas. Otra mejoría sería orientada a la adaptación (o autoadaptación) automática de las implementaciones llevadas a cabo, con el fin de hacer posible el ajuste automático de cada parámetro de una técnica de compresión, tales como, el tipo de transformada a utilizar, niveles de descomposición *wavelet*, ajuste y tipo de cuantizador y codificador entrópico, entre otros; ya que como se mencionó anteriormente, este ajuste debe estar en función de la naturaleza de la señal a comprimir (o de la imagen, para el caso 2D) y tomando en cuenta la gran diversidad de imágenes que en un dispositivo móvil van a procesarse, el desarrollar una herramienta con dicha autoadaptación es un reto vigente. Otra posibilidad de extensión al trabajo, sería procesar secuencias de imágenes, tales como las secuencias de video.

El tema de la compresión de imágenes podría ser abordado desde el punto de vista de *splines* en vez de elegir alguna de las familias de *wavelets*, y aprovechar sus propiedades tales como, la simetría, y su excelente capacidad de aproximación [41]. Además mediante el uso de la codificación piramidal se puede sacar ventaja respecto al trabajo adicional requerido para llevar a cabo el procesamiento mediante *wavelets*, situación que haría a los *splines* apropiados para elegirse en ambientes donde se pretenden efectuar un número reducido de operaciones.

Un último punto que podría agregarse, sería la exploración de otro tipo de esquemas de compresión de imágenes pero desde el punto de vista de la morfología matemática aplicada a imágenes, ya que es una opción prometedora también, en materia de compresión de imágenes y desde el punto de vista de complejidad computa-

cional para los dispositivos móviles.

Apéndice A

Anexos

A.1. Imágenes de prueba

En esta sección se presentan las imágenes de prueba a las que se hace referencia en el capítulo 4, de la presente tesis.



Figura A.1: Imagen *lena* en formato BMP a 3 bytes/píxel (RGB) de 256 x 256 píxeles.

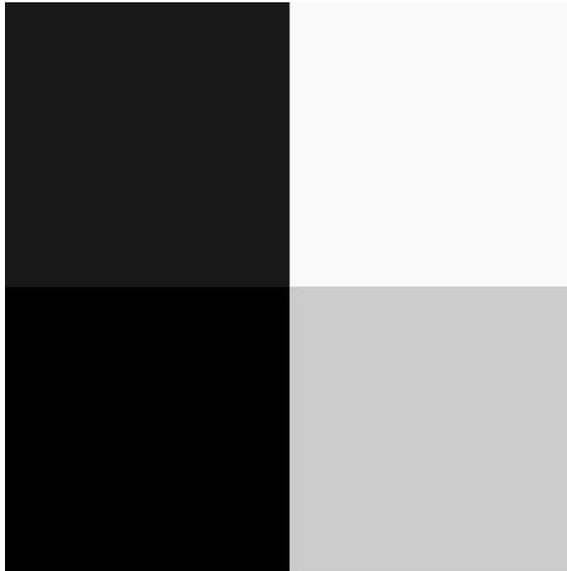


Figura A.2: Imagen *patrón simple* en formato BMP a 3 bytes/píxel (RGB) de 450 x 450 píxeles.

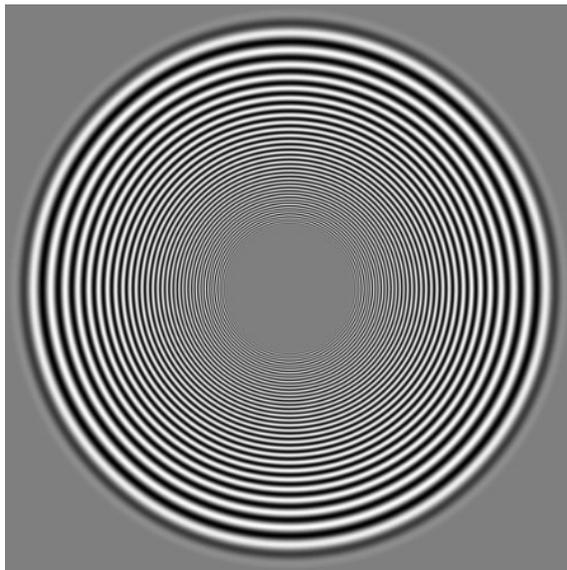


Figura A.3: Imagen *patrón complejo* en formato BMP a 3 bytes/píxel (RGB) de 512 x 512 píxeles.



Figura A.4: Imagen *huella dactilar* en formato BMP a 3 bytes/pixel (RGB) de 480 x 480 píxeles.

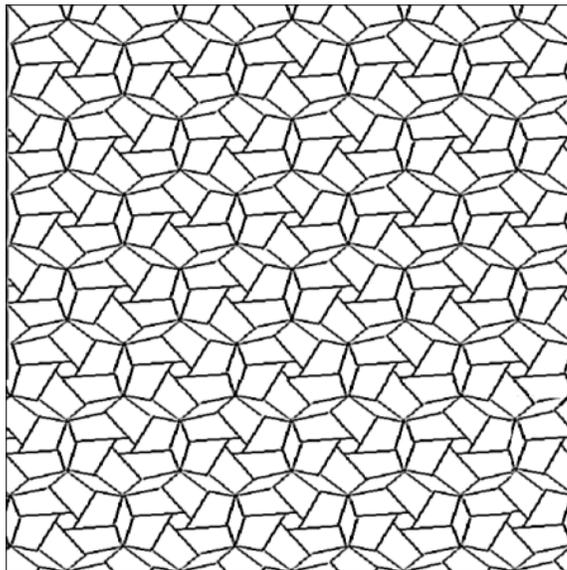


Figura A.5: Imagen *piramidal* en formato BMP a 3 bytes/pixel (RGB) de 602 x 602 píxeles.

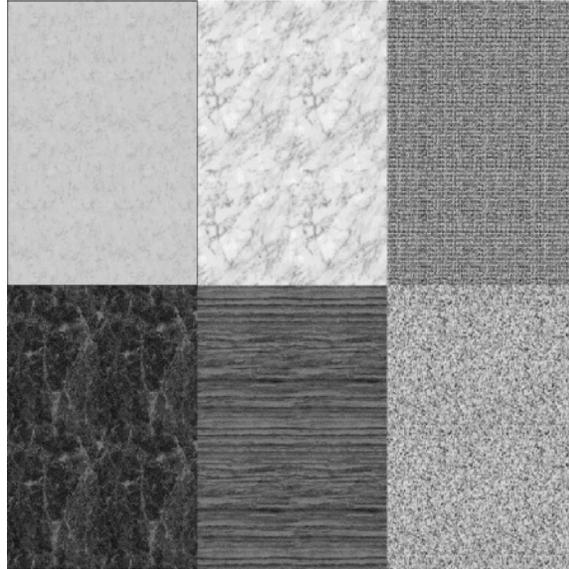


Figura A.6: Imagen *textura* en formato BMP a 3 bytes/pixel (RGB) de 960 x 960 píxeles.



Figura A.7: Imagen *puesta de sol* en formato BMP a 3 bytes/pixel (RGB) de 800 x 800 píxeles.

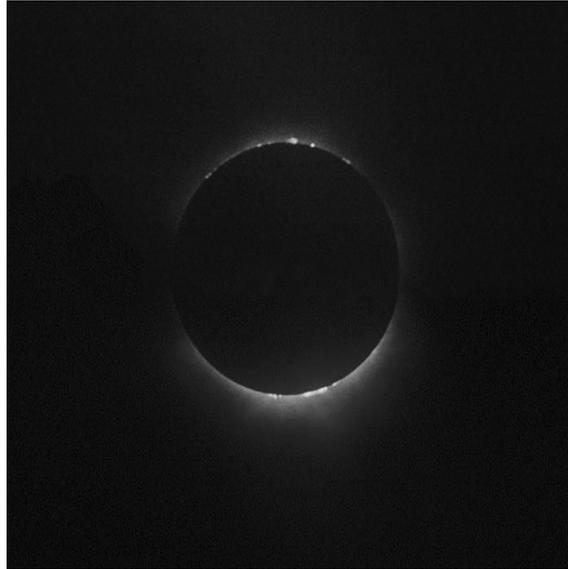


Figura A.8: Imagen *eclipse* en formato BMP a 3 bytes/pixel (RGB) de 700 x 700 píxeles.



Figura A.9: Imagen *colina* en formato BMP a 3 bytes/pixel (RGB) de 512 x 512 píxeles.

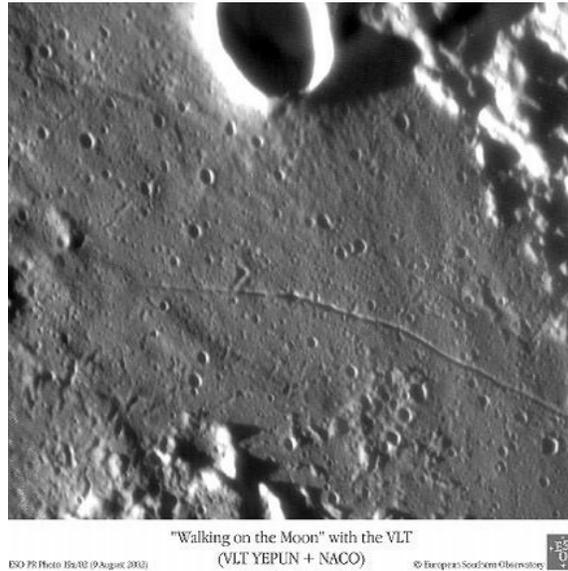


Figura A.10: Imagen *superficie lunar* en formato BMP a 3 bytes/pixel (RGB) de 440 x 440 pixeles.



Figura A.11: Imagen *montañas* en formato BMP a 3 bytes/pixel (RGB) de 1024 x 1024 pixeles.

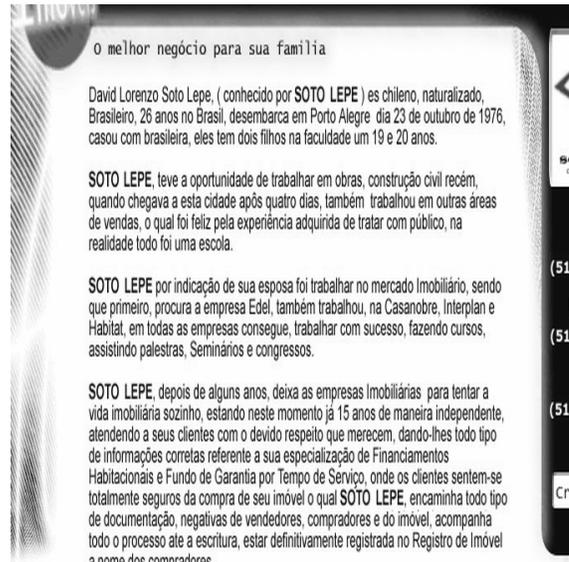


Figura A.12: Imagen *con texto* en formato BMP a 3 bytes/pixel (RGB) de 684 x 684 pixeles.



Figura A.13: Imagen *periódico* en formato BMP a 3 bytes/pixel (RGB) de 1024 x 1024 pixeles.



Figura A.14: Imagen *dibujo animado* en formato BMP a 3 bytes/píxel (RGB) de 1280 x 1280 píxeles.



Figura A.15: Imagen *rostro sintético* en formato BMP a 3 bytes/píxel (RGB) de 602 x 602 píxeles.



Figura A.16: Imagen *rostro 1* en formato BMP a 3 bytes/píxel (RGB) de 1280 x 1280 píxeles.



Figura A.17: Imagen *rostro 2* en formato BMP a 3 bytes/píxel (RGB) de 500 x 500 píxeles.

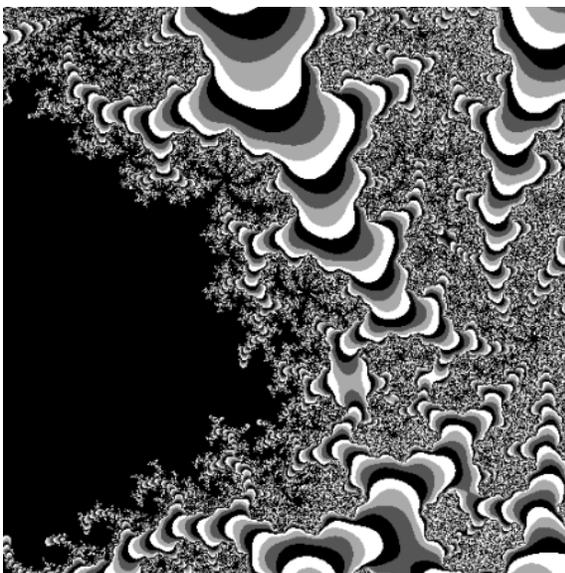


Figura A.18: Imagen *fractal* en formato BMP a 3 bytes/pixel (RGB) de 500 x 500 píxeles.

A.2. Otras imágenes de prueba

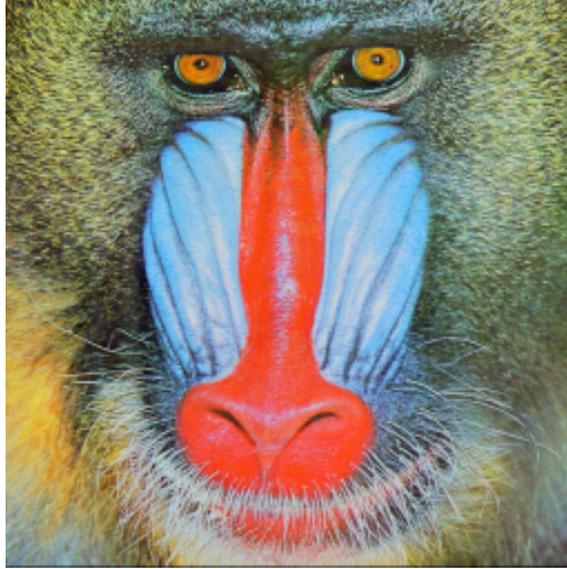


Figura A.19: Imagen *babuino* en formato BMP a 3 bytes/pixel (RGB) de 256 x 256 píxeles.



Figura A.20: Imagen *pimienta* en formato BMP a 3 bytes/pixel (RGB) de 256 x 256 píxeles.



Figura A.21: Imagen *chapultepec* en formato BMP a 3 bytes/píxel (RGB) de 256 x 256 píxeles.

Bibliografía

- [1] Jing, Helal, Elmagarmid, *Client-Server Computing in Mobile Environments*. ACM Computing Surveys, Vol. 31 Number 2, pages 117-157, 1999.
- [2] Lamberti, F., Zunino, C., et. al., *An Accelerated Remote Graphics Architecture for PDAs*. Proceedings of ACM/SIGGRAPH Web3D 2003 Symposium, pp. 55-61:203.
- [3] Rist, T., Brandmeier, P., *Customizing Graphics for Tiny Displays of Mobile Devices*. Personal and Ubiquitous Computing (2002)6: 260-268.
- [4] Megowan, P., Suvak, D., et. al., *IrDA Infrared Communications: An Overview*. Counterpoint Systems Foundry, Inc.
- [5] Williams, S., *IrDA: Past, Present and Future*. Personal Communications, IEEE Vol. 7, FEB 2000.
- [6] Brenner, P., *A Technical Tutorial on the IEEE802.11 Protocol*. Spread Spectrum Scene Online, Breezecom Wireless Communications, July 1997.
- [7] Zhang, L., Gao, C., *Application of RS-Coded MPSK Modulation Scenarios to Compressed Image Communication in Mobile Fading Channel*. Proceedings 52nd IEEE Vehicular Technology Conference, VTS-Fall VTC.2000, Volume: 3, 2000 pp: 1198 -1203.
- [8] Morimoto, M., Okada, M., et. al., *A Hierarchical Image Transmission System for Multimedia Mobile Communication*. First International Workshop on Wireless Image/Video Communications, September 1997.
- [9] Ravasi, M., Mattavelli, M., et. al., *Wavelet Image Compression for Mobile Portable Applications*. IEEE Transactions on Consumer Electronics, Vol. 45, No. 3, August 1999.

-
- [10] Lei, Y., Jiang, Z., et. al., *Image-Based Walkthrough over Internet on Mobile Devices*. International Workshop on Visualization and Visual Steering, VVS 2004.
- [11] Kondo, H., Ishikawa, T., et. al., *High Picture Quality Image Compression Technique for Mobile Communication*. IEEE International Conference on Industrial Technology, Bangkok, Thailand, 2002.
- [12] Liu, Xie, et. al., *Automatic Browsing of Large Pictures on Mobile Devices*. International Multimedia Conference, Proceedings of the eleventh ACM international conference on Multimedia, Berkeley, CA, USA, 2003.
- [13] Nelson, M., *The Data Compression Book*. ISBN 1-55851-214-4. M&T Publishing, Inc, 1995.
- [14] Gonzalez, R., Woods, R., *Tratamiento Digital de Imágenes*. Addison-Wesley, 1994.
- [15] Shannon, C. E., *A mathematical theory of communication*. Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
- [16] Huffman, D. A., *A method for the construction of minimum-redundancy codes*. Proc. Inst. Radio Eng. 40, 9 (.), 1098-1101, Sept 1952.
- [17] Held, G., Marshall, T. R., Wiley, J., *Data and Image Compression*. Tools and Techniques, Fourth Edition, USA, 1996.
- [18] Crane, R., *Simplified Approach to Image Processing*. New Jersey, Prentice Hall, 1997.
- [19] Mofat, A., Neal, R. and Witten, I., *Arithmetic coding revisited*. ACM Transactions on Information systems, 16(3): 256-294, July, 1998.
- [20] Rissanen, J., Langdon, G., *Arithmetic coding*. IBM J. Res. Dev. 23, 2 , (Mar 1979). Hunt R.W.G., *Measuring Colour*, (New York: Halsted Press 1987).
- [21] Witten, I., Neal, R., Cleary, J., *Arithmetic coding for data compression*. Commun. ACM 30, 6, 520-540, June, 1987.
- [22] Rosenfeld, A., KaK, C., *Digital Picture Processing*. Academic Press, 1982.
- [23] K. Pratt, W., *Digital Image Processing*. John Wiley & Sons, 1991.
-

-
- [24] Simoncelli E. P., *Orthogonal Sub-band Image Transforms*. Master's Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, (Cambridge, May, 1998).
- [25] Ambardar, A., *Analog and Digital Signal Processing*. PWS Publishing Company, (USA, 1995).
- [26] Couch, Leon W., *Digital and Analog Communication Systems*. Fourth Edition, Macmillan Publishing Company (1993).
- [27] Pennebaker, W., Michael, J., Reinhold, J. *JPEG, Still Image Data Compression Standard*. ITP Inc. (1993).
- [28] Grgic, S., Grgic, M., *Performance Analysis of Image Compression Using Wavelets*. IEEE Transactions on Industrial Electronics, vol. 48, no. 3, June 2001.
- [29] Lewis, A. S., Knowles, G., *Image compression using the 2-D wavelet transform*. IEEE Transactions on Image Processing, vol. 1, pp. 244–250, April 1992.
- [30] Antonini, M., Barland, M., *Image coding using the wavelet transform*. IEEE Transactions on Image Processing, vol. 1, pp. 205–220, April 1992.
- [31] Shapiro, J. M., *Embedded image coding using zerotrees of wavelet coefficients*. IEEE Trans. Signal Processing, vol. 41, pp. 3445–3463, December 1993.
- [32] Said, A., Pearlman, W. A., *A new fast and efficient image codec based on set partitioning in hierarchical trees*. IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 243–250, June 1996.
- [33] Daubechies, I., *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
- [34] Mallat, S., *A theory of multiresolution signal decomposition: The wavelet representation*. IEEE Trans. Pattern Anal. Machine Intell., vol. 11, pp. 674–693, July 1989.
- [35] Mallat, S., *Multifrequency channel decomposition of images and wavelet models*. IEEE Trans. Acoust., Speech, Signal Processing, vol. 37, pp. 2091–2110, Dec. 1989.
- [36] Cosman, P. C., Gray, R. M., Olshen, R. A., *Evaluating quality of compressed medical images: SNR, subjective rating and diagnostic accuracy*. Proc. IEEE, vol. 82, pp. 920–931, June 1994.
-

-
- [37] Yves M., *Wavelets Algorithms and Applications*. Society for Industrial and Applied Mathematics, SIAM Press, USA, 1994.
- [38] Allnatt, J., *Transmitted-Picture Assessment*. New York: Wiley, 1983.
- [39] Miyahara, M., Kotani, K., *Objective picture quality scale (PQS) for image coding*. [Online] Available: <http://info.cipic.ucdavis.edu/scripts/reportPage?96-12>
- [40] *Methods for the subjective assessment of the quality of television pictures*. ITU, Geneva, Switzerland, ITU-R Recommendation. BT-500.7, Aug. 1998.
- [41] Unser, M., *Splines: A perfect fit for signal and image processing*. IEEE Signal Processing Magazine, vol. 16, no. 6, pp. 22–38, November 1999.
- [42] Ahlberg, J.H., Nilson, E.N. and Walsh, J.L., *The Theory of Splines and Their Applications*, New York: Academic Press, 1967.
- [43] Bartels, R.H., Beatty, J.C. and Barsky, B.A., *A Splines for Use in Computer Graphics*, Los Altos, CA: Morgan Kaufmann, 1987.
-