

**UNIVERSIDAD CATÓLICA "NUESTRA SEÑORA
DE LA ASUNCIÓN"**

**TEORIAS Y APLICACIONES DE LA
INFORMÁTICA 2**

J P E G 2 0 0 0

FÁTIMA FLORES POMPA

SETIEMBRE - 2004

JPEG 2000

1. INTRODUCCIÓN

Dado que cada día se ha vuelto más común e incluso necesario el manejo de imágenes, se ha debido desarrollar ciertas técnicas para reducir el tamaño de las mismas, ya que de otra manera, al almacenarlas o transmitir las de una máquina a otra, se requeriría de una gran capacidad de memoria y se tardaría un tiempo considerable.

Para tener una idea de porqué la compresión de imágenes es tan necesaria, se puede considerar una imagen en color de solo 3x4 pulgadas, digitalizada con una resolución de 500 dpi (puntos por pulgada). Siendo la resolución una medida de la calidad de la imagen, a medida que esta se incrementa también lo hace su calidad. Si cada pulgada cuadrada de la imagen tiene $500 \times 500 = 250,000$ puntos (píxeles), la imagen poseerá un total de 3.000.000 píxeles, y cada uno requerirá 3 Bytes de datos para guardar los diferentes colores de la imagen, por lo que esta necesitará 9.000.000 Bytes de memoria, los cuales no entrarían en un diskette común. Es por esto que la compresión resulta tan necesaria, ya que para guardar varias imágenes se necesitaría de mucha memoria; y para transmitir las via *Internet* haría falta un tiempo de transferencia muy largo. Esto ha fomentado el desarrollo de varias técnicas de compresión de imágenes.

Actualmente, la forma más común de compresión es el standard conocido como JPEG, desarrollado por el *Joint Photographic Experts Group*, al final de los años 80. Cerca del 80% de las imágenes utilizadas en sitios Web se encuentra codificado en ese standard. Sin embargo, en 1997, debido al incremento de la necesidad de la utilización de aplicaciones que contengan imágenes, el comité del JPEG se dio cuenta que era necesario desarrollar un nuevo standard con mayor capacidad de compresión de la imagen. Ese nuevo standard ahora se conoce como JPEG 2000.

2. QUÉ ES EL JPEG2000

JPEG2000 o jpeg2k, es el nuevo standard ISO/ITU-T para imágenes digitales fijas y para su compresión. Está basado en una compresión de tipo *wavelet*, solo que al comprimirla, se produce una pequeña pérdida de datos. En inglés esto se conoce como *lossy compression*. Estos conceptos se aclaran más adelante.

Debido a que la *Internet* está basada en código HTML, esto dificulta el manejo de imágenes de alta calidad.

JPEG2000 fue diseñado justamente en respuesta a esa necesidad de procesamiento de imágenes de alta calidad pero de baja tasa de bits, y para corregir las limitaciones del formato

JPEG en *Internet* y en otras aplicaciones.

2.1. ESTRUCTURA DEL JPEG2000

Todo el núcleo de la estructura del JPEG2000 (o jpeg2k), es una metodología de compresión basada en *wavelets* (pequeña onda), que consiste en la compresión de una imagen convirtiéndola en un conjunto de funciones/expresiones matemáticas que representan los datos de la imagen.

Esta metodología provee varios beneficios sobre el método de la **Transformación Discreta del Coseno (DCT)**, que es utilizado por el formato JPEG.

El **DCT** comprime una imagen en bloques de 8x8 bits y los coloca de forma consecutiva en el archivo. En este proceso, los bloques se comprimen individualmente, sin hacer referencia a los bloques contiguos. Esto resulta en un efecto llamado *blockiness* (**figura 1**), que está asociado a los archivos JPEG comprimidos. A altos niveles de compresión, solo la información más importante se utiliza para obtener lo esencial de la imagen, pero, mucha de la sutileza que hace a la imagen placentera y continua, se pierde.



Figura 1. a) imagen comprimida con JPEG en la que se nota el efecto de *blockiness*.
b) imagen comprimida con JPEG2000

En contraste, JPEG 2000 utiliza la DWT (*Discrete Wavelet Transform*) o *transformada discreta de wavelets*, implementada por el algoritmo EBCOT (*Embedded Block Coding with Optimized Truncation*).

3. CÓMO SE REALIZA LA COMPRESIÓN JPEG2000

El proceso de compresión consta de tres pasos principales (**Figura 2**):

- 1) A la imagen fuente se le aplica un proceso llamado **Transformación Forward**
- 2) El resultado de la etapa anterior pasa por un proceso de **cuantización**
- 3) Finalmente se realiza una **codificación entrópica** de los datos cuantizados, para obtener el archivo de imagen comprimido.

La primera etapa (**Transformación Forward**), se subdivide en tres etapas:

- A1) A la imagen se aplica el proceso de *Tiling*.
- A2) Luego se le aplica el *DC level shifting*.
- A3) Finalmente se aplica la DWT a cada tile obtenido en el paso A1.

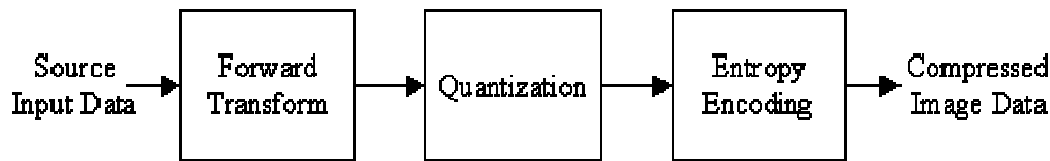


Figure 4 JPEG 2000 Block Diagram

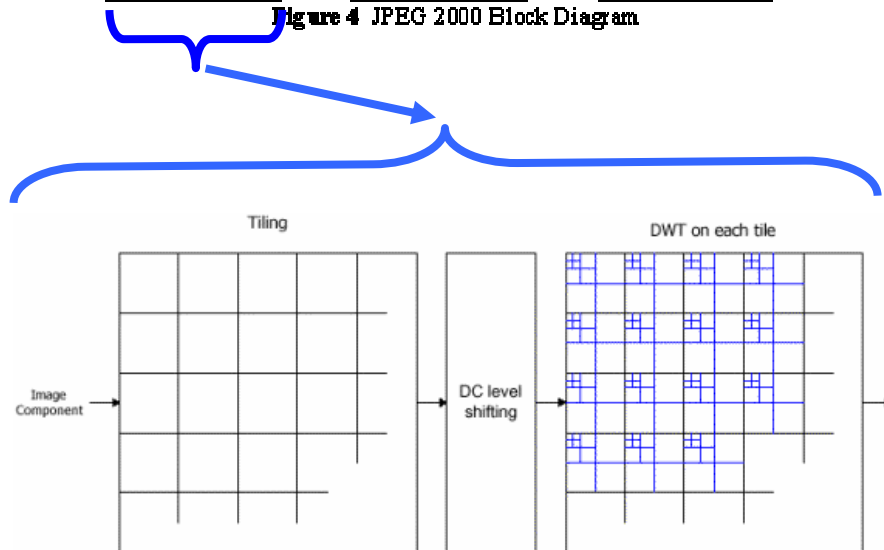


Figura 2. Proceso de compresión de JPEG2000

3.1. ETAPA DEL FORWARD TRANSFORM

La compresión DWT primero se aplica a los datos de la imagen fuente.

La imagen fuente se particiona en bloques rectangulares no superpuestos. Este proceso se llama *tiling*.

Cada bloque (*tile*) está conformado por array de elementos de tipo entero con signo.

$$\mathbf{A} = \{a_1, \dots, a_n\}.$$

Este array se describe por medio de un número de planos de bits (*bit planes*), que son secuencias de arrays binarios con un bit de cada coeficiente del array entero.

El primer plano de bits contiene el bit más significativo (MSB) de todas las magnitudes.

$$\mathbf{A1} = \{a_1', \dots, a_n'\}$$

El segundo array, contiene el siguiente MSB. $\mathbf{A2} = \{a_1'', \dots, a_n''\}$, y así sucesivamente hasta el array final, que consiste en los bits menos significativos de todas las magnitudes.

Antes de realizar la DWT a cada *tile*, se le aplica un proceso llamado *DC Level Shifting* que consiste en sustraer a cada *tile* una misma cantidad específica, por ejemplo, el

componente de *profundidad* de cada muestra de imagen, lo que significa mover la imagen del *tile* a un bit *plane* deseado.

Luego, utilizando el algoritmo **EBCOT** se realiza la transformada DWT a cada *tile* para subdividir la energía de la imagen original en subbandas, en las cuales se diferencia la información en baja frecuencia (la más importante para la visión humana), de la información en alta frecuencia. La imagen se divide en cuatro imágenes más pequeñas que son las mencionadas subbandas.

Estas subbandas son los componentes de los diferentes niveles de resolución, y contienen un número de coeficientes que describen las características horizontales y verticales del bloque original.

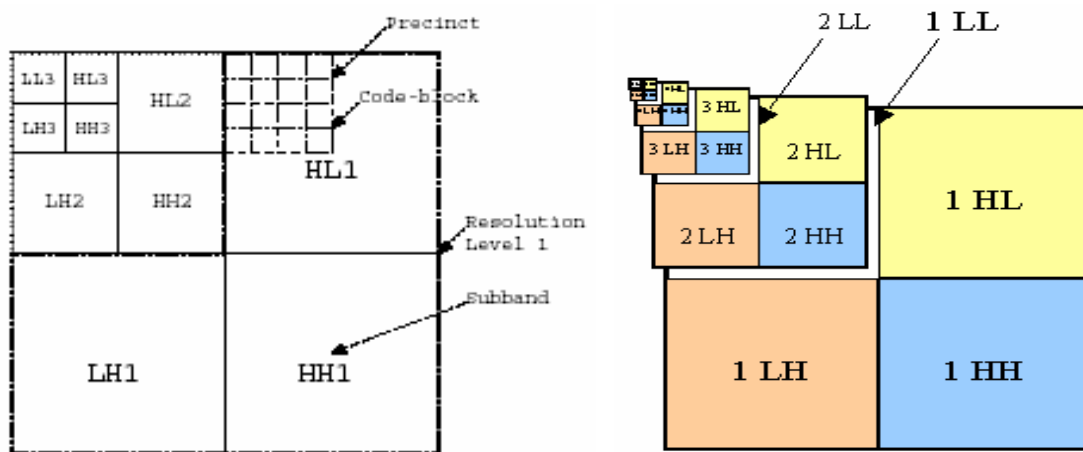


Figura 3. Descomposición diádica.

El primer bloque, luego de usar un filtro *low-pass*, contiene las bajas frecuencias del archivo original, la otra subbanda contiene alta resolución vertical y baja horizontal (HL), la otra con baja resolución vertical y alta horizontal (LH), y la cuarta con ambas resoluciones altas (HH). Estas tres últimas subbandas utilizan un filtro *high-pass*.

Este proceso se repite un número de veces en el bloque de imagen de baja resolución. El proceso descrito se llama **descomposición diádica** (dyadic decomposition), (**figura 3**), en la que las subbandas 1HL, 1LH, 1HH, 1LL son el resultado de aplicar la descomposición de *wavelets* al bloque completo, las subbandas 2HL, 2LH, 2HH, 2LL resultan de la descomposición aplicada a la subbanda 1LL, y así sucesivamente. La combinación de dos letras permite las cuatro combinaciones de filtros bidimensionales.

Las muestras *low-pass* representan una versión de baja resolución de la imagen original, mientras que las *high-pass* representan una versión residual, que será necesaria para reconstruir el conjunto original a partir del conjunto de muestras *low-pass*.

Un ejemplo de *descomposición diádica* en subbandas con la misma imagen tratada como un solo *tile*, se observa en la **figura 4**.



Figura 4. Descomposición diádica en subbandas de la misma imagen tratada como un solo *tile*.

El proceso descompone la imagen fija en una organización jerárquica compuesta por múltiples niveles de resolución, subbandas, precintos, y bloques de código (**figura 3**)

3.2. ETAPA DE CUANTIZACIÓN

Luego de estas transformaciones, en la **fase de cuantización**, todos los coeficientes son cuantizados, es decir, se reduce la precisión de los coeficientes dividiendo la magnitud de cada uno entre un factor de cuantización y redondeando el resultado.

Esta operación es de tipo *losy*, a menos que los coeficientes sean enteros. En este caso el factor de cuantización es 1.0, no se realiza cuantización, y los coeficientes no sufren ningún cambio, permitiendo que la operación sea de tipo *lossless*.

Luego de la cuantización, cada subbanda es particionada en bloques de código (rectángulos no superpuestos) (**Figura 3**), que contienen los coeficientes que serán codificados.

3.3. ETAPA DE CODIFICACIÓN ENTRÓPICA.

Estos bloques de código se utilizan para realizar la ultima fase del proceso, llamada **codificación entrópica**, que se realiza independientemente en cada bloque de código. Esta codificación se logra a través de una codificación binaria aritmética de los *bit planes*, la cual se realiza escaneando cada *bit* en un conjunto de tres pasos *llamados pasos de código*:

El paso en el que un bit dado está codificado depende de la significancia de la ubicación de ese bit y de la significancia de las ubicaciones vecinas de éste. Una ubicación se considera significativa si en el plano de bits previo o en el actual, se ha codificado un 1 para esa ubicación.

Pasos de código:

- 1) **Paso de propagación de significancia:** Un bit será codificado en este primer paso, si su ubicación no es significativa, pero si al menos uno de sus vecinos sí lo es.
- 2) **Paso de refinamiento de magnitud:** en este paso, se codifican todos los bits de ubicaciones que se volvieron significantes en un plano de bits previo.
- 3) **Paso de limpieza (clean-up):** se encarga de todos los bits que no fueron codificados en los dos primeros pasos.

Al finalizar la codificación entrópica, se almacena una versión comprimida de la imagen original.

4. JPEG2000 vs JPEG

Para describir mejor el formato estudiado, lo compararemos con el JPEG, al cual podemos ver como el padre del JPEG 2000, sin embargo, éste posee varias ventajas sobre el anterior.

El standard JPEG fue desarrollado para imágenes en color y blanco y negro. El intento original se debió al deseo de procesar imágenes fotográficas a una pequeña tasa de bits, y superar las limitaciones de sistemas de compresión tales como el GIF, cuyos archivos de imágenes son de mayor tamaño. Sin embargo, el JPEG pronto demostró poseer sus propias limitaciones. Se dice que su sistema de compresión es de tipo *lossy compression*, es decir, la imagen sufre de una pérdida de datos luego del proceso de compresión/descompresión, lo cual, eventualmente afecta su calidad resultando en una distorsión de la imagen.

Otra limitación de JPEG es su baja habilidad para manejar bordes agudos en una imagen. Estos bordes creados por texto y figuras geométricas aparecen más borrosos a medida que aumenta la tasa de compresión.

JPEG2000 se creó para manejar estos problemas:

✂ Utiliza transformaciones de tipo *wavelet*, es decir, fórmulas matemáticas que representan los datos de la imagen, de modo a lograr una compresión de mayor grado con poca cantidad de datos. Estas imágenes son de menor tamaño que las codificadas en JPEG y pueden ser transferidas y descargadas a velocidades mayores. Además, poseen mayor calidad, manejan mejor los bordes agudos, y reducen la pérdida de datos de la compresión.

Aclaremos que existen dos clases de compresión de datos: con y sin pérdida de datos (o en inglés: *lossy* y *lossless compression* respectivamente). En la compresión con pérdidas, parte de los datos se pierde durante el proceso de compresión, resultando en una imagen final de menor calidad que la original. De esta manera se obtiene altos grados de compresión. Por otro lado, la compresión sin pérdidas vuelve a obtener la imagen original con la misma calidad luego de comprimirla, sin embargo su grado de compresión es menor que el obtenido por la compresión *lossy*.

✂ Otra ventaja es que JPEG 2000, gracias a las transformaciones de tipo *wavelet*, provee de ambos tipos de compresión (*lossy* y *lossless*) en el mismo archivo de imagen, mientras que JPEG utiliza normalmente la compresión *lossy* para un archivo.

✂ Como el formato JPEG 2000 incluye más información que los archivos JPEG actuales, el primero maneja archivos mucho más pequeños que sin embargo contienen el mismo nivel de detalle que los archivos JPEG originales de mayor tamaño.

✂ Los archivos JPEG 2000 soportan imágenes a dos tonos, en escalas de grises, a 256 colores, a millones de colores, en standard RGB, PhotoYCC, CIELAB, CMYK en contraste con los JPEG que están limitados a los tres canales del RGB.

✂ Otra ventaja a favor del JPEG2000, consiste en que este formato ofrece mayor grado de compresión para la compresión con pérdidas, ya que puede comprimir imágenes de un 20% a un 200% más que el JPEG.

✂ Para la compresión sin pérdidas, JPEG 2000 también provee de tasas de compresión de 2.5:1, mientras que JPEG, a pesar de poder realizar este tipo de compresión, no logra hacerlo de manera satisfactoria alguna. Las tasas de compresión *lossless*, se miden según cuánta menor cantidad de memoria usa la imagen comprimida.

✂ Una siguiente ventaja de JPEG 2000 es su habilidad de desplegar imágenes a diferentes resoluciones y tamaños de aquellas del archivo de imagen original. Con JPEG, un archivo de imagen solo podía desplegarse de una manera y una resolución determinadas.

✂ Debido a que JPEG 2000, como ya se explicó, se basa en *wavelets*, éstas pueden ser solo parcialmente descomprimidas si el usuario solo desea una imagen de baja resolución, o bien presentarle a éste la imagen con resolución completa. Con JPEG, para obtener una imagen de baja resolución, se necesitaría generar un archivo diferente para cada imagen, mientras que con JPEG 2000, se utiliza el mismo archivo de imagen, y el usuario solo elige con qué resolución desplegarla.

✂ Otro beneficio más del JPEG 2000 es su capacidad **ROI (Region of interest)**. El uso de *wavelets* también permite seleccionar una cierta área de una imagen para verla con alta calidad, pero dejando el resto de la imagen con baja calidad. Esto permite al usuario ver solo la porción deseada de la imagen en vez de verla completa, lo cual reduce de gran manera la cantidad de memoria que requiere la imagen, y el tiempo requerido para acceder a esta.

✂ JPEG 2000 también es superior en cuanto al *error resilience*¹⁾, es decir, la medición de la capacidad de un método de compresión, para evitar que errores introducidos al archivo de imagen afecten la calidad de ésta. Por ejemplo, cuando se descarga una imagen de *Internet*, el "ruido" a causa de la transmisión de esta puede introducirse al archivo de imagen, causando pequeños errores. Otra manera de que ocurra esto, es la conversión del archivo de imagen para utilizarlo en otra aplicación. JPEG2000 obtiene mayor *error resilience* que JPEG, por lo tanto, hay menor posibilidad de que la imagen se corrompa y que su calidad se vea afectada.

✂ JPEG2000 también es superior al formato JPEG ya que provee de una mejor calidad de imagen para archivos del mismo tamaño; y para archivos con la misma calidad de imagen, ofrece un ahorro de espacio del 20-30%.

✂ Para archivos de imágenes escalables, es decir, cuyo tamaño puede variar, no se necesita reformatear la imagen. Con JPEG 2000, la imagen más parecida a la original se puede extraer de un solo archivo comprimido. El usuario puede acceder a tamaños de imágenes desde muy pequeñas escalas hasta su tamaño real, a una escala de grises de 3

¹⁾ Resilience: capacidad de recuperar la forma o condición original luego de ser dañado.

canales de color y a una calidad de imagen desde muy baja a *lossless* (sin pérdidas, idéntica a la imagen original).

✂ Otra característica del JPEG2000, comprende el renderizado y transmisión progresivos, a través de una estructura de archivos de imágenes por capa. Por ejemplo, de una imagen original de 512x512 se puede extraer un archivo de imagen de 100 Kbytes, y de éste se puede transmitir una imagen a pequeña escala con una resolución baja de 32x32 píxeles enviando solo 10 Kbytes. Enviando 15 Kbytes adicionales se incrementa la resolución a 64x64 píxeles, y así sucesivamente. Otras capas proveen de renderizado y transmisión progresivos basados en la calidad, en componentes del color y en la ubicación espacial en la imagen.

5. APLICACIONES DEL JPEG2000

Una de las áreas de aplicación del formato estudiado es la **Internet**, cuyos usuarios normalmente deben descargar imágenes de gran tamaño y de alta calidad, por lo que los proveedores de la imágenes deben almacenar varias versiones de éstas.

JPEG2000 soluciona estos problemas, ya que las imágenes se pueden codificar de modo que al transmitir las, gradualmente vayan incrementando su resolución y su calidad. Así, el usuario puede detener la descarga cuando la imagen tenga el suficiente nivel de detalle deseado por él.

Si una imagen puede ser desplegada en tamaños diferentes, actualmente por cada tamaño desplegado se debe poseer un archivo diferente, mientras que con JPEG2000 se puede almacenar un solo archivo de imagen, y acceder a éste a la resolución deseada. Con una conexión rápida a la *Internet* se podría optar por una resolución completa, mientras que alguien con conexión lenta, puede elegir ver la imagen a pequeña escala.

Lo explicado arriba también se aplica al **e-commerce** a través de la *Internet* para las empresas que ofrecen imágenes de los productos que ofrecen al público.

Para lograr estas mejoras, JPEG2000 permite crear un sistema de archivos virtual, que crea dos tipos de estructuras de archivos. La estructura de *resolución progresiva* que permitirá que la imagen se presente al usuario en una versión de baja resolución, y que progresivamente ésta vaya incrementándose. Y la de *calidad progresiva*, que inicialmente se presentará con total resolución pero con una mínima cantidad de colores por píxel, y gradualmente irá agregando más bits por píxel según sea necesario.

Acceder a la imagen a diferentes resoluciones también permitirá ahorrar ancho de banda, ya que si un usuario hace *click* en una imagen para agrandar su tamaño, y esta fue diseñada con acceso de tipo *resolución progresiva*, se presentará primero con baja resolución, hasta que, como ya dijimos, el usuario detenga su descarga al lograr el grado de resolución deseado.

Además, se han agregado ciertas características de seguridad al standard para

evitar que las imágenes sean copiadas y distribuidas de manera ilegal.

Una de estas características es el **JPEG2000 secured (JPSEC)**, que utilizando técnicas de firma digital, *watermarking*, y generación de claves, provee de:

- Encriptación parcial o total del contenido de una imagen.
- Autenticación del origen de la imagen.
- Verificación de la integridad de los datos, de la imagen completa, u opcionalmente de ciertas zonas específicas de esta.
- Acceso condicional: un usuario puede tener permiso para acceder a una versión de baja resolución de una imagen, pero no a una de alta resolución.
- Protección de derechos de autoría, a través de mecanismos de identificación de ataques maliciosos y procesamiento indebido de la imagen.

Otra característica es el **Jpeg2000 interactive protocol (JPIP)**, que consiste en un protocolo cliente-servidor que puede ser implementado sobre HTTP, pero está abierto a otros protocolos. Maneja diferentes formatos para las imágenes devueltas por el servidor, por ejemplo, archivos JPEG y JPEG2000, y por supuesto, el concepto de imágenes en bloques del que ya se ha hablado, con el cual obtiene ventaja de las propiedades de escalabilidad de la imagen. Esto puede ser aplicado a cualquier formato de archivo, no solo a formatos JPEG2000. También se define un nuevo indexado para los archivos, que a través de un acceso randómico permite acceder a estos incluso si la versión de HTTP del servidor no soporta JPIP.

La publicación del standard de esta característica se prevé para antes de finales de este año.

También cuenta con el **JPEG2000 3D (JP3D)**, que aún se encuentra en una etapa de desarrollo, y trata de la codificación de datos tridimensionales, lo que sería la extensión de imágenes planas de JPEG2000 a imágenes volumétricas. Un aspecto esencial de la codificación volumétrica será la utilización de grillas no uniformadas para concentrar los datos más significantes de cada región de la imagen.

Otra característica importante del standard es el **Jpeg2000 wireless (JPWL)**, que será apto para aplicaciones *wireless multimedia* ofreciendo calidad de servicio (QoS) a los operadores de redes. Soporta las siguientes funcionalidades:

- Protección de la imagen codificada contra errores de transmisión.
- Descripción de los grados de sensibilidad de diferentes partes de la imagen codificada ante los errores de transmisión.
- Descripción de la ubicación de errores residuales en la imagen codificada.

Para lograr esta protección contra los errores, el proceso de codificación modifica la codificación de la imagen para darle más capacidad de tipo *error resilience*, agregándole códigos de redundancia e incluyendo corrección de errores a través del **Forward Error Correcting (FEC)**.

Por los aspectos de seguridad dentro de una red, esta parte del standard está ligada con el **JPSEC** y el **JPIP**.

Una última característica soportada por el standard JPEG2000 es el **ISO BASE media file format**, que es un formato de archivo que tiene partes en común con el standard MPEG-4. Es una unión de **JPEG** y **MPEG** para crear un formato base de archivos para futuras aplicaciones. Se trata de un formato general para secuencias temporizadas de datos multimedia. Utiliza la misma arquitectura que el formato de archivos de Apple's QuickTime.

Otra aplicación del JPEG2000 se da en el campo de la **medicina**. Como ya dijimos, la compresión de tipo *wavelet* permite codificar con mayor fidelidad las imágenes, las cuales generalmente corresponden a partes del cuerpo del paciente para determinar ciertas anomalías. El estándar estudiado permite grandes niveles de detalle, además permite utilizar la característica de *región de interés*. El médico puede desplegar la imagen completa a una resolución dada, escalar ciertas partes de esta con una resolución que mantiene la calidad de la imagen original.

Gracias a su capacidad de Error Resilience puede enviar datos a través de ambientes perturbados, como por ejemplo, la **telefonía móvil**.

También será muy útil en la **industrias**, por ejemplo en las fábricas, en las cuales dependiendo del proceso aplicado a los productos se necesitará diversos niveles de resolución de la imagen. En una fábrica de autos, por ejemplo, la fabricación de una cierta pieza pequeña requerirá de una imagen escalada pero que mantenga un buen nivel de claridad.

También será utilizado en el campo de la **vigilancia**, para la detección de movimiento, y para proveer a las cámaras de seguridad de mayor capacidad de aumentar las imágenes con el máximo nivel de claridad y resolución.

JPEG2000 también es deseado en el campo de la **imagen geoespacial**, ya que facilita la compresión, distribución y escalado de los datos utilizados en los mapas. Debido a esto la Agencia de Imágenes y Mapas de los Estados Unidos (NIMA) y la OTAN, han dado el visto bueno al JPEG2000 como formato de compresión para los nuevos sistemas de imágenes.

JPEG2000 también ofrece a los **archivos digitales** los beneficios ya mencionados de escalabilidad de la imagen manteniendo las resoluciones deseadas, y la posibilidad de transferir lotes de archivos de imágenes de una institución a otra en menor tiempo.

Por supuesto, otra aplicación es la de la **fotografía digital**, y el **cine digital**, que precisan comprimir las imágenes sin pérdida de datos y almacenarlas hasta llenar la memoria de la cámara, o bien almacenarlas con compresión con pérdidas aumentando la cantidad de imágenes almacenadas.

Otras aplicaciones generales comprenden el **scaneo** de la imagen, guardándola en formato JPEG2000, y reproducirla a diferentes resoluciones dependiendo si se la quiere **imprimir** o desplegar en un **monitor**, recordando que siempre se trabaja con el mismo archivo de imagen, y solo se procesan versiones de éste dependiendo de la resolución deseada.

5.1. SOFWTARES QUE UTILIZAN EL JPEG2000

En estos momentos no hay muchas aplicaciones que hagan uso de JPEG2000, aunque está previsto que las empresas de software empiecen a hacer uso de él en todo tipo de aplicaciones. Los usos para los que está previsto este formato gráfico son mucho más avanzados que el resto de los formatos, y aunque su desarrollo en las empresas es aún experimental, no falta mucho para verlo en aplicaciones comerciales. Algunas aplicaciones actuales son:

- ✗ **JJ2000:** implementación en Java. Su versión actual es la 5.1. incluye un *applet* para visualizar imágenes JP2. Es de libre distribución.
- ✗ **Jasper:** librerías y aplicaciones para codificar y manipular imágenes. Está programado en C. Es de libre distribución.
- ✗ **Aware:** *toolkits* y SDKs para el desarrollo de JPEG2000.
- ✗ **ImagePower:** Software de tratamiento de imágenes de JP2.
- ✗ **Migrator 2000:** su objetivo es desarrollar, demostrar y diseminar herramientas para la norma de imágenes digitales JPEG2000. El proyecto comprende el archivado y distribución de imágenes, así como su utilización en aplicaciones médicas, incluyendo la implementación de protección de los derechos de autor (IPR), descripción de contenido y marcas de agua.
- ✗ **Microimages:** aplicación de tratamiento de imágenes espaciales.
- ✗ **Morgan Multimedia:** aplicaciones de video (Motion JPEG2000). *Plugin* para navegadores.
- ✗ **Kakadu:** conjunto de herramientas optimizadas para desarrolladores de JPEG2000. Implementada en C++.
- ✗ **Elysium:** *plugin* para uso en los navegadores más comunes (Netscape, Opera, Explorer). Basado en la distribución de software Kakadu.
- ✗ **LeadTools:** *plugin* para manejo de imágenes JP2 en Photoshop. Software propio para compresión y descompresión de imágenes JP2.
- ✗ **Algo Vision LuraTech:** ofrece varias herramientas para el manejo de JP2. Plugins para Photoshop, Netscape, aplicación de compresión y SDKs. Una de sus herramientas es el **LuraWave**, que comprime imágenes usando el método de la DWT, pero sus archivos utilizan la extensión ".LWF", en vez de la ".J2K" de JPEG2000. En la **figura 5** se compara una imagen comprimida con este software y una con el tradicional JPEG.



Figura 5. imagen .tif comprimida con LuraWave. a) imagen original.
 b) imagen .LWF (versión de JPEG200) comprimida con LuraWave.
 c) Imagen comprimida en JPEG tradicional.

6. CONCLUSIONES

Debido a los varios beneficios ofrecidos por el JPEG2000, como la habilidad de realizar las compresiones de tipo *lossy* y *lossless*, obtener una mayor calidad de imagen, mayor grado de compresión, visualizar el mismo archivo con múltiples resoluciones, examinar una sola área de interés de la imagen, etc., se presenta como el siguiente standard de compresión de imágenes.

Estas habilidades serán muy necesarias en varios ámbitos que ya mencionamos como el de la medicina, fotografía, etc., siendo que su desempeño es superior a los estándares actuales, por lo tanto, es muy probable que ciertos fabricantes de tecnologías tiendan a crear software y hardware que lo soporten.

Sin embargo, actualmente, softwares muy utilizados como el *Microsoft Internet Explorer* y el *Netscape Navigator*, no soportan archivos con la extensión ".j2k" así que dependerá de grandes compañías como estas observar la amplia gama de aplicaciones de este formato, y sus características mejoradas con respecto a los actuales, de modo a aceptar su uso e introducir nuevos dispositivos y softwares que hagan uso de él.

7. BIBLIOGRAFÍA

<http://jj2000.epfl.ch>

<http://www.ece.uvic.ca/~mdadams/jasper>

<http://www.aware.com/products/compression/jpeg2000.html>

<http://www.kakadusoftware.com>

<http://www.imagepower.com>

<http://www.algovisionluratech.com>

<http://www.leadtools.com>

<http://public.migrator.org/main.xalter>

http://www.elysium.ltd.uk/jpeg_2000.html

<http://microimages.com/gallery/jp2>

<http://www.morgan-multimedia.fr/JPEG2000/index.html>

http://www.aware.com/products/compression/jpeg2000_geosp.html

http://www.aware.com/products/compression/jpeg2000_med.html

<http://www.aware.com/products/compression/seispact.html>

http://www.cerc.wustl.edu/~jefritts/papers/weiyu_icip02.pdf

http://www.gvsu.edu/math/wavelets/student_work/EF/applications.html

http://www.gvsu.edu/math/wavelets/student_work/EF/background.html

http://www.gvsu.edu/math/wavelets/student_work/EF/background.html

http://www.gvsu.edu/math/wavelets/student_work/EF/comparison.html

http://www.gvsu.edu/math/wavelets/student_work/EF/conclusion.html

http://www.gvsu.edu/math/wavelets/student_work/EF/how-works.html

http://www.gvsu.edu/math/wavelets/student_work/EF/index.html

<http://www.jpeg.org/jpeg2000/j2kpart10.html>

<http://www.jpeg.org/jpeg2000/j2kpart11.html>

<http://www.jpeg.org/jpeg2000/j2kpart12.html>

<http://www.jpeg.org/jpeg2000/j2kpart8.html>

<http://www.jpeg.org/jpeg2000/j2kpart9.html>

<http://www.jpeg2000-image-compression.com/>

<http://www.jpeg2000-image-compression.com/jpeg2000--the-new-standard.html>

<http://www.jpeg2000-image-compression.com/what-is-jpeg2000-image-compression.html>

<http://www.remotesensing.org/jpeg2000/>

<http://www.webopedia.com/TERM/W/wavelet.html>

Oxford Advanced Learner's Dictionary of Current English. Oxford university Press.
3ª ed. 1986

ÍNDICE

1. INTRODUCCIÓN.....	1
2. QUÉ ES EL JPEG2000.....	1
2.1. ESTRUCTURA DEL JPEG2000.....	2
3. CÓMO SE REALIZA LA COMPRESIÓN JPEG2000.....	2
3.1. ETAPA DEL FORWARD TRANSFORM.....	3
3.2. ETAPA DE CUANTIZACIÓN.....	5
3.3. ETAPA DE CODIFICACIÓN ENTRÓPICA.....	5
4. JPEG2000 VS JPEG.....	6
5. APLICACIONES DEL JPEG2000.....	8
5.1. SOFTWARES QUE UTILIZAN EL JPEG2000... ..	11
6. CONCLUSIONES.....	12
7. BIBLIOGRAFÍA.....	13