

UNIVERSIDAD LATINOAMERICANA DE CIENCIA Y TECNOLOGIA

FACULTAD DE INGENIERIA INFORMATICA

Seminario de graduacion

La compresion de datos

Teirlynck Nicolas

Pas: EE748744

Profesor: Miguel Perez

Índice

<i>Índice</i>	<i>II</i>
1. Introducción	III
2. Los logaritmos y el concepto de teoría de la información	V
a. La maquina a compresión.....	VII
3. Algunas Técnicas básicas	VIII
b. Compresión irreversible de los textos.....	IX
c. Run-Length Encoding.....	IX
d. Cuantificación escalar:.....	IX
e. El método estadístico:.....	X
4. Compresión de datos multimedia	XI
a. Las normas de compresión para las imágenes numéricas.....	XI
b. Las normas de compresión para el vídeo numérico.....	XIX
c. Las normas de compresión para el audio numérico.....	XXII
5. Conclusión	XXVI
6. Bibliografía	XXVII

1. Introducción.

La compresión, veló un tema que desde hace mucho tiempo se hablaba. El concepto nace de las necesidades de simplificar la información con el fin de ganar tiempo en su transmisión, en la lengua es representado por abreviaturas: ok, asap, etc... El Morse o el Braille, son unas técnicas de compresión de información que encuentran sus utilidades en nuestras vidas. La compresión de datos es sin duda la expresión fundamental de la teoría de la información, rama de matemáticas que nace en los años 40, en el estudio de Bell Labs. El padre de la teoría de la información Claude Shannon (www.bell-labs.com) es manija de cuestiones con respecto a la información, entre otras cosas la manera de almacenar y transmitir dicha información. Sus teorías sobre la redundancia sirven de base a la compresión de datos tal como se utiliza hoy día. La información redundante en un mensaje es la que utiliza más bits que lo necesario para codificar la información, la idea es poder eliminar esta información problemática y así reducir el número de bits utilizados sin cambiar el sentido del mensaje. La teoría de la información utiliza el término entropía como unidad de medida para determinar cuántas informaciones esta codificada en un mensaje. La informática y la termodinámica comparten la palabra entropía donde tiene el mismo significado. Cuanto más la entropía de un mensaje esta elevada, más contiene informaciones. La entropía de un símbolo se define como el logaritmo negativo de su probabilidad. Para determinar el contenido de la información de un mensaje en bits, se utiliza un logaritmo de base dos para representar la entropía.

Números de bits = - Log base 2 (probabilidad)

La entropía de un mensaje completo, es simplemente la suma de la entropía de todos los símbolos. En resumen, la entropía es una medida del número de redundancia de un símbolo en un mensaje dado el cual nos permite codificar los símbolos que son los mas utilizados en un mensaje con valores inferior en bits que los que son menos utilizados. La compresión de datos es también

llamada codificación de fuente. Uno se imagina que el símbolo de entrada (como bits, código ASCII, bytes, o valores de píxeles) está emitido por una determinada fuente de información y debe ser codificado antes de ser enviado a su destino. La fuente puede no tener memoria, o puede tener una. En el primer caso, cada bit es independiente de su antecesor. En el segundo caso cada símbolo depende de sus antecesores o también de los que siguen. Existe diferente método de compresión de datos, se basan sobre distintas ideas, son útiles para diferentes datos y producen diferentes resultados, pero todos se basan en el mismo principio, comprimen los datos eliminando la redundancia de la información fuente. Toda colección de datos que no sea aleatoria, tiene obligatoriamente una cierta estructura, y esta estructura puede explotarse para alcanzar una representación más pequeña, una representación donde no se puede distinguir ninguna estructura. En Inglés, por ejemplo, la letra “E” aparece de manera recurrente, mientras que la letra “Z” es más bien escasa, llamamos eso la redundancia alfabética, la idea es de asignar un código de tamaño variable a la letra, la letra “E” recibiendo el código más corto y a la letra “Z” recibiendo el código más largo. Otro tipo de redundancia, llamada la redundancia conceptual, es ilustrado por el hecho de que por ejemplo la letra “Q” casi siempre va seguida por la letra “U”. La redundancia en la imagen es ilustrada por el hecho de que en una imagen no aleatoria, los píxeles adyacentes tienden a tener un color similar.

La compresión de datos se hace cambiando una representación ineficaz por una eficaz, la compresión es posible en informática por el hecho de que los datos son representados en un formato que es más largo que lo necesario, para facilitar su comprensión. Tomando como ejemplo el código ASCII para los caracteres, es un buen ejemplo de representación de datos que es más largo que lo necesario. Utiliza un código fijo de 7 bits por la simple razón que un código fijo es más fácil de utilizar. Una representación a código variable habría sido más eficaz dado que determinados caracteres son utilizados más a menudo que otros.

2. Los logaritmos y el concepto de teoría de la información.

Sabemos intuitivamente lo que es la información. Recibimos constantemente y enviamos información en forma de texto, palabra, e imágenes. Consideramos también que la información es una cantidad no matemática evasiva que no puede definirse, capturarse y medirse con precisión. Las definiciones de los diccionarios normales son:

- El conocimiento derivado del estudio, experiencia, o instrucción.
- El conocimiento de un acontecimiento o de una situación específica (inteligencia.)
- Una colección de hechos o datos.
- El acto de informar o el estado de estar informado (comunicación del conocimiento.)

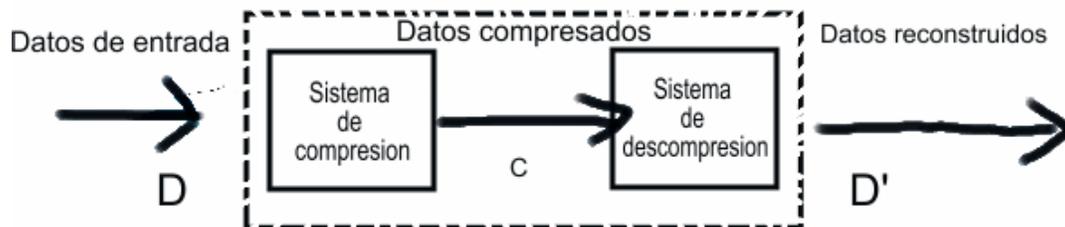
Imagínese a una persona que no sabe que es la información. ¿Estas definiciones lo ayudaría él a entenderlo? Probablemente no. La importancia de la teoría de la información es que mide la información. Muestra cómo medir la información, de modo que podemos responder a la cuestión "¿se incluye cuánto información en este pedazo de datos?" ¡Con un número preciso! La medida de la información se basa en la observación que el contenido en información de un mensaje es equivalente a la cantidad de "sorpresas" en el mensaje. Si les digo algo que ya saben (por ejemplo, "tú y mí trabajamos aquí"), no les proporcioné ninguna información. Si les digo algo nuevo (por ejemplo, "tenemos ambos un aumento"), lo proporcioné información. Si les digo algo que les asombro realmente (por ejemplo, "soy yo, el único que recibirá un aumento"), lo proporcioné más información, independientemente del número de palabras que empleé, y de lo que piensan de mi información. Comenzamos por un acontecimiento simple, familiar y fácil a analizar, es decir el lanzamiento en el aire de una moneda. Hay dos resultados, así el resultado de cualquier lanzamiento en el aire está al principio dudoso. Debemos realmente lanzar la moneda con el fin de solucionar la incertidumbre. El resultado es cara o pilas, que pueden también expresarse como sí o no, o como uno 0 o 1, un bit. Un simple bit soluciona la incertidumbre en el lanzamiento en el aire de una moneda. Lo que hace este ejemplo importante es el hecho de que se puede generalizar fácilmente. Muchos problemas reales pueden solucionarse, y sus

soluciones Expresarse, por medio de varios bits. La razón de esta técnica, consiste en encontrar el número mínimo de cuestiones binario (cuya respuesta puede ser sí o no) que deben ser respondidas con el fin de llegar al resultado. Puesto que la respuesta a una cuestión binaria puede expresarse con 1 bit, el número de cuestiones igualará el número de bits que toma para expresar la información contenida en el resultado. Un ejemplo ligeramente más complejo es un juego de 64 tarjetas. Para la simplicidad ignorarán sus nombres y números tradicionales y simplemente lo numeran de 1 a 64. consideran el acontecimiento de "A" que lanzará una tarjeta y de "B" debe conjeturar lo que es. ¿La posibilidad es un número entre 1 y 64. cuál es el número mínimo de cuestiones binario necesario para responder a la conjetura? Los que están informados de la técnica de la investigación dicotómica saben la respuesta. ¿ Al utilizar esta técnica, "B" debería dividir la gama 1-64 en dos, y debería comenzar por preguntarse si el resultado se encuentra entre 1 y 32? Si la respuesta es no, entonces el resultado está en la gama 33-64. Esta gama entonces es dividida por dos y la próxima cuestión de "B" debería ser: "el resultado se encontraría entra 33 y 48?" Este proceso sigue hasta que la gama empleada por "B" se reduzca a un número simple. No es necesario mucho tiempo para darse cuenta de que exactamente seis cuestiones son necesarias para responder a la cuestión. Es porque 6 es el número de vez que 64 puede dividirse en dos. Matemáticamente, es equivalente a la escritura $6 = \log_2 64$. por eso el logaritmo es la función matemática que mide la información. Otro enfoque al mismo problema es plantear la cuestión: "Dado un número entero no negativo "N", cuántas cifras son necesarias para expresarlo? ". La respuesta, naturalmente, depende de "N", más "N" es grande, más un gran número de cifras son necesarias. Los 100 primeros números enteros no negativos (0 a 99) pueden ser expresados por dos cifras decimales. Los 1000 primeros números enteros no negativos pueden ser expresados por tres cifras. De nuevos, no es necesario mucho tiempo para ver la relación. El número de cifras requeridas para representar "N", igual aproximadamente $\log "N"$. La base del logaritmo es la misma que la base de las cifras. Para las cifras decimales, se emplean la base 10, y para los binarios (bit), se emplean la base 2. Si convenimos que el número de cifras necesario para expresar "N" es

proporcional al contenido en información de "N", de nuevo, el logaritmo es la función que nos proporciona una medida de la información.

a. La máquina a compresión.

En un sentido, podemos decir que la compresión de datos es un método que toma datos de entrada "D" y produce una representación de "D" llamada "C" esta representación, aunque expidiendo la misma información, contiene menos bits que "D". Al contrario, la descompresión es el proceso que a partir de "C" reconstruye los datos "D". El conjunto del sistema de compresión (codificación) y descompresión (desciframiento) es llamado un "CODEC".



En función de las condiciones de reconstrucción, los datos reconstruidos en "D'" pueden o ser idénticos, o ser una aproximación de los datos originales en "D". Si los datos reconstruidos "D'" son una reproducción exacta de los datos originales "D", llamamos el algoritmo aplicado a esta compresión "lossless" o compresión sin ruidos. Por otra parte, decimos que los algoritmos son "lossy" o compresión con ruidos cuando no es una reproducción exacta de "D". Por lo tanto a lo que se refiere a la reversibilidad de los datos originales, los algoritmos de la compresión de datos pueden ampliamente clasificarse en dos categorías "lossless" y "lossy". Habitualmente debemos aplicar técnicas llamadas "lossless" sobre textos o datos científicos. Uno se espera a que sea reconstruido el mismo texto después de un proceso de descompresión ya que un pequeño error en un texto reconstruido puede implicar un cambio de significado completo con relación al texto original. Del mismo modo, por ejemplo, si comprimimos un fichero ASCII conteniendo un programa escrito en lenguaje C, debemos obviamente recuperar el mismo código C después de la descompresión. Las técnicas de compresión llamada "lossy" son habitualmente aplicables a los datos donde no se exige la elevada fidelidad de los datos

reconstruidos. Es el caso, en particular, para los sonidos, las imágenes, el vídeo, los gráficos, la palabra, y todos los tipos de datos que aprovecha de las propiedades limitadas del sistema sensorio humano. Algunas aplicaciones de tratamiento de imagen pueden exigir de la compresión de ser “*loss/ess*”. Las imágenes médicas son un buen ejemplo de tal aplicación donde compresiones de radiografías numéricas de tipos “*lossy*” podrían implicar la inexactitud del diagnóstico. La estructura de la compresión puede ser estática o dinámica. Para los métodos estáticos, el cociente flujo de información sobre tiempo (o paso de un conjunto de mensaje (datos o señal) al conjunto de datos comprimido) es siempre fijo. Para los métodos dinámicos, este paso es variable, los códigos se adaptan a los cambios de características con el tiempo. Por ejemplo, si las probabilidades de frecuencias de los símbolos de la fuente no son fijas en el tiempo, podemos de manera adaptativa formular los códigos binarios de los símbolos, de modo que el tamaño del fichero comprimido pueda, para una mejor eficacia de compresión, cambiar durante el tiempo.

3. Algunas Técnicas básicas.

La compresión de datos se realiza reduciendo la redundancia, pero esto vuelve también los datos menos fiables, más propensos a los errores. Volver datos más fiables, por otra parte, se hace añadiendo bits de control y bits de paridad, un proceso que aumenta el tamaño de los códigos, aumentando por lo tanto la redundancia. La compresión de datos y la fiabilidad de datos pues se opone, y es interesante tener en cuenta que este último es un sector de estudio relativamente reciente, mientras que el primero existe desde antes de la llegada de las computadoras. El telégrafo, el código Braille, y el código Morse forman tanto ejemplo de este primero. Hoy estos métodos son la mayor parte del tiempo de interés histórico, puesto que son generalmente ineficaces y no pueden competir con los métodos modernos de compresión desarrollados en los últimos 15 ó 20 años.

b. Compresión irreversible de los textos.

A veces es aceptable "comprimir" el texto omitiendo simplemente información. Esto se llama compresión irreversible de los textos o "asentamiento". El texto, una vez decodificado no será idéntico al original. No se adaptan tales métodos a todo uso; pueden solamente emplearse en casos especiales. Una secuencia consecutiva de espacios vacíos puede ser sustituida por un espacio simple. Esto puede ser aceptable para textos literarios y en la mayoría de las lenguas máquina, pero no deberían emplearse cuando los datos están con forma de cuadros. En casos extremos pueden omitirse todos los caracteres de los textos excepto las letras y los espacios, y convertir las letras en mayúscula o minúscula. Esto dejará exactamente 27 símbolos (26 letras capitales y el espacio), pudiendo ser cifrado en 5 en vez de los 8 bits habituales. Se obtiene así un informe de compresión de $5/8 = 0.625$, no malo, pero la pérdida puede ser demasiado importante para una buena comprensión del texto.

c. Run-Length Encoding.

La idea detrás de este enfoque a la compresión de datos es esto: Si un dato "d" se produce "n" vez de manera consecutiva en un flujo de datos, se sustituyen las ocurrencias "n" con el simple par "nd". Este enfoque a la compresión de datos se llama en inglés: Run-Length Encoding, o RLE. Este principio es aplicable a la vez a la compresión de texto así como a la compresión de imagen.

d. Cuantificación escalar:

La definición del término "cuantificación" tal como lo define el diccionario es la siguiente: "limitar una cantidad variable a los valores discretos más bien que en un conjunto continuo de valores." En el ámbito de la compresión de datos, la cuantificación se emplea de dos maneras:

- Si los datos que se debe comprimir se presentan en forma de grandes números, la cuantificación se emplea para convertirlos en números pequeños. Los pequeños números toman menos espacio que los grandes, así la cuantificación produce compresión. Por otra parte, los

pequeños números contienen generalmente menos informaciones que los grandes, así cuantificación tienen como consecuencia una compresión “*lossy*”.

- Si los datos que deben comprimirse son análogos (Ej. : un voltaje que cambia con el tiempo), se emplea la cuantificación para digitalizarlos en pequeños números. Más los números son pequeños, mejor es la compresión, pero también mayor es la pérdida de información. Este aspecto de la cuantificación es empleado por varias técnicas de compresión de la voz.

e. El método estadístico:

Los métodos discutidos hasta ahora tienen en conjunto que; ellos asignan códigos a tamaño fijo a los símbolos (de los caracteres o el píxel) que operan. En cambio, los métodos estadísticos emplean códigos de tamaño variable, con los códigos más cortos asignados a los símbolos o a los grupos de símbolos que aparecen más a menudo en los datos (teniendo una probabilidad más elevada de ocurrencia.) Existen dos problemas cuando se trata de asignar código de tamaño variable. En primer lugar, asignar los códigos para que puedan descifrarse de manera clara y en segundo lugar, asignar código cuyo tamaño es una media mínima. Samuel Morse empleó código de tamaño variable cuando ha concebido su código de telégrafo muy conocido. Es interesante tener en cuenta que la primera versión de su código, desarrollada durante un viaje transatlántico en 1832, era más compleja que la versión que tiene publicada en 1843. La primera versión empleaba pequeños y largos guiones recibidos y dibujados sobre una banda de papel, el orden en el cual eran enviados estos guiones representaba números. A cada palabra (y no cada letra) se afectaba un número de código, Morse produjo un libro, un diccionario de estos códigos en 1837. Esta primera versión era así una forma primitiva de compresión. Morse más tarde abandonó esta versión en favor de sus puntos y guiones famosos.

4. Compresión de datos multimedia.

La compresión de datos multimedia se convirtió en una obligación de los sistemas de telecomunicación, la telefonía digital, el telefax, la comunicación celular, la videoconferencia, Internet, la radio difusión, los servicios de mensajería vocal, los sistemas archivísticos de imágenes, las bibliotecas numéricas, el DVD, la distribución de película y vídeo, el grafismo, la industria de películas y muchos otros utilizan la compresión bajo una forma u otra. Nuevos productos, utilizando el concepto de la compresión surgen cada día en todo el mundo y el número de aplicaciones seguirá sin cesar desarrollándose. En consecuencia, es necesario definir una norma de compresión de datos para que los sistemas sean perfectamente compatibles. Mencionamos aquí algunas de las normas de la compresión de datos para distintos tipos de multimedia: imagen, vídeo, palabra, texto, etc.

a. Las normas de compresión para las imágenes numéricas.

Los dos principales cuerpos internacionales en el sector de compresión de imagen son: ISO, "*Organization Standardization International*" e ITU-T, "*International Telecommunication Union - Sector Telecommunications*" antes conocido bajo el nombre de CCITT. ISO trata los problemas relativos al tratamiento de la información como el almacenamiento y la recuperación de imagen, mientras que ITU-T trata de la transmisión de la información. JPEG (*Joint Photographic Expert Group*) es la norma conjuntamente desarrollada por ISO e ITU-T en 1992 para imágenes a tono continuo de niveles de gris y de color. La idea común entre mucha gente es que el JPEG es un simple algoritmo para la compresión de las imágenes fijas. En realidad, la norma JPEG define cuatro métodos de operaciones que son el método secuencial, el método "lossless" secuencial, el método progresivo, y el método jerárquico. El algoritmo más extendido para la compresión de imagen en método secuencial se llama el JPEG. El formato JPEG comprime imágenes con una compresión de 0.25-2 bit por Píxel. El Grupo de Trabajo 1 en ISO está ocupado a definir la segunda generación de JPEG, el JPEG2000 para realizar compresiones inferiores a una más alta calidad. Las aplicaciones modernas utilizan intensivamente las

capacidades gráficas de los computadores. ¡Las imágenes son más importantes ahora que diez años atrás, y tienden a ser cada vez más voraz en recurso! Sabiendo que el material moderno puede mostrar muchos colores, es común representar un píxel como un número 24-bit (o 32-bit), donde los porcentajes de rojo, de verde, y el azul ocupan 8 (o 12) bits cada uno. Tal píxel (24-bit) puede representar uno de los 16.78 millones de colores. En consecuencia, una imagen teniendo una resolución de 512×512 que consta de tal píxel ocupa 786.432 octetos. A una Resolución de 1024×1024 eso exige 3.145.728 octetos. Se emplean también muchas películas en computadores, conduciendo a un aún mayores necesidad de tratamiento de las imágenes. Esta es la razón por la que la compresión de imagen es tan importante. Un dispositivo de compresión de imagen es generalmente de tipo “*lossy*” (donde es posible una pérdida de calidad.) Una imagen, después de todo, existe para que las personas lo observen, así, cuando se comprime, es aceptable perder la información de imagen a la cual el ojo no es sensible. Es una de las ideas principal detrás de los numerosos métodos de compresión de imagen “*lossy*”. En general, la información puede comprimirse si es superflua. Se mencionó varias veces que la compresión de datos corresponde a la reducción de la redundancia, con la compresión “*lossy*”, sin embargo, nosotros tenemos un nuevo concepto. Una imagen puede comprimirse gracias a la eliminación de la información no pertinente. Es necesario mencionar que una imagen sin redundancia no es siempre aleatoria. La definición de la redundancia nos indica que una imagen donde cada color aparece con la misma frecuencia no tiene ninguna redundancia (estadísticamente), sin embargo ella no es necesariamente aleatoria y puede incluso ser interesante y/o útil. La idea de perder la información de imagen se vuelve más agradable cuando consideramos cómo se crean algunas imágenes numéricas. He aquí tres ejemplos:

- Una imagen real puede barse a partir de una fotografía o de una pintura y ser digitalizada (convertida en píxel.)
- Una imagen puede ser registrada por una cámara fotográfica visual que crea píxel y la almacena directamente en la memoria.

- Una imagen puede pintarse sobre la pantalla por medio de un programa de pintura.

Cada vez, se pierde la información cuando se numera la imagen. El hecho de que el telespectador esté dispuesto a aceptar esta pérdida sugiere que aún más pérdida de información podría ser tolerable, si esta hecha correctamente. La digitalización de una imagen implica dos etapas: exacción y cuantificación. El muestreo de una imagen es el proceso de dividir la imagen original bidimensional en pequeñas regiones, los píxeles. La cuantificación es el proceso de asignar un valor de número entero a cada píxel. Se debe tener en cuenta que la digitalización del ruido implica las mismas dos etapas, pero tiene como diferencia que el ruido es unidimensional.

¿Cómo debería comprimirse una imagen? Las técnicas de compresión discutidas en capítulos anteriores son RLE, cuantificación escalar, métodos estadísticos. Por ellos mismos, ningún método resulta satisfactorio para imágenes de color o niveles de gris (aunque puedan emplearse en combinación con otros métodos.) RLE puede emplearse para la compresión de una imagen. Se emplea para la compresión JPEG, especialmente en su método "lossless". Sin embargo, es un conjunto de técnicas utilizadas por JPEG que permite tal compresión, y la RLE solo permite realmente una compresión mínimo de la información de una imagen. La cuantificación escalar puede emplearse para comprimir imágenes, pero su ejecución es mediocre. Imagínesse una imagen con el píxel de 8 bits. Puede comprimirse con la cuantificación escalar recortando los menos significativos de cada píxel. Esto nos da un informe de compresión de 0.5, no muy impresionante, y trae al mismo tiempo el número de colores (o de niveles de gris) de 256 a 16. Tal reducción deteriora no solamente la calidad global de la imagen reconstruida, pero puede también crear bandas de distintos colores, de un efecto aparente y aburrido. Los métodos tradicionales son pues insuficientes para la compresión de imagen, así la compresión de imagen utiliza un diferente enfoque según el siguiente principio: Si elegimos un píxel en la imagen al azar, hay fuertes oportunidades que los píxeles vecinos tengan el mismo color o unos colores muy similares. La compresión de imagen se basa pues en el hecho de que se correlacionan mucho los píxeles vecinos entre sí. Esta correlación se llama también la redundancia espacial. He aquí un ejemplo simple que ilustra lo que

puede hacerse con los píxeles correlacionados. La siguiente orden de valores da las intensidades de 24 píxeles adyacentes en una hilera de una imagen a tono continuo:

12, 17, 14, 19, 21, 26, 23, 29, 41, 38, 31, 44, 46, 57, 53, 50, 60, 58, 55, 54, 52, 51, 56, 60.

Solamente dos de los 24 píxeles son idéntico. El valor medio es de 40.3. La sustracción de los pares de píxeles adyacentes tiene como consecuencia el orden:

12, +5, -3, +5, +2, +4, -3, +6, +12, -3, -7, +13, +4, +11, -4, -3, +10, -2, -3, +1, -2, -1, +5, +4.

Los dos órdenes se ilustran sobre el esquema siguiente.



Esta secuencia de número define tres propiedades que permiten determinar su potencial de compresión:

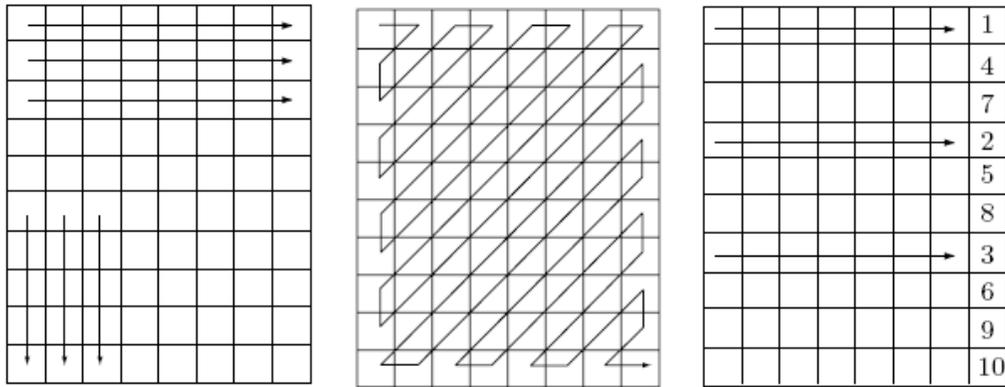
- Los valores que diferencian los píxeles entre ellos son más pequeños que los valores originales de píxeles.
- Su media es 2.58, se repiten.
- Solo hay 15 valores distintos, así en principio pueden ser cifrados por cuatro bits cada uno. Ellos no presentan ninguna relación entre sí a estos niveles: los valores adyacentes tienden a ser diferentes.

Existe un otro aspecto a este principio de la compresión de imagen. Se correlaciona también el valor de luminancia de los píxeles vecinos. Dos píxeles adyacentes pueden tener distintos colores. Pueden ser la mayor partes del tiempo rojo o, la mayor parte del tiempo verde. Si el componente rojo del

primero es luminoso, el componente verde de su vecino, en la mayoría de los casos, será también luminoso. Esta propiedad puede explotarse convirtiendo las representaciones de píxel de RGB a tres otros componentes, cuyos uno es el resplandor, y los otros dos representan el color. Tal formato (o espacio de color) es YCbCr, donde Y (el componente de brillo) representa el resplandor de un píxel, y CB y Cr definen su color. Es fácil definir su ventaja. El ojo es sensible a pequeños cambios de resplandor pero no a pequeños cambios de color. Así pues, la información dispuesta en los componentes de CB y Cr comprime la imagen presentando al mismo tiempo deformaciones a las cuales el ojo no es sensible. La información que se perdió en el componente de Y, por otra parte, es muy aparente al ojo. Se concibe normalmente un método de compresión de imagen para un tipo específico de imagen. Existen diferentes tipos de enfoques, los cuales son discutidos a continuación:

Enfoque 1: Esto se emplea para imágenes a dos niveles; Un píxel en tal imagen está representado por un bit. La aplicación del principio de la compresión de imagen a una imagen a dos niveles significa pues que los vecinos inmediatos de un píxel "P" tienden a ser idénticos a "P". Así, se basta de emplear la codificación RLE para comprimir tal imagen. Un método de compresión para tal imagen puede:

- Barrerlo en el orden de trama (guardada por hilera) y calcular la RLE del píxel blancos y negros. Las longitudes se cifran entonces con códigos de tamaño variable. Un ejemplo de tal método es compresión de facsímil. Se debería tener en cuenta que es un enfoque que se adapta solamente para imagen monocromo.
- Barrer una imagen columna por columna o en zigzag, puede convertir la imagen subdividiéndola en cuatro o puede barrerla región por región utilizando una curva (x, y)



Enfoque 2: Por otro lado para imágenes a dos niveles. El principio de la compresión de imagen nos indica que los vecinos de un píxel tienden a ser similares a este píxel. Podemos continuar este principio y concluir que si el píxel actual tiene el color “c” (dónde “c” es negro o blanco), entonces el píxel del mismo color visto anteriormente y los que serán encontrados en el futuro tienden a tener los mismos vecinos inmediatos. Este enfoque toma en cuenta el píxel vecino, y le asigna un número “n”. Este número es el “contexto” del píxel. Algunos contextos serán comunes mientras otros serán raros. El codificador hace la cuenta de cuántas veces se encontraron cada contexto para un píxel de color “c”, y asigna probabilidades a los contextos en consecuencia. Si el píxel actual tiene el color “c” y su contexto tiene la probabilidad “p”, el codificador puede emplear la codificación aritmética adaptativa para cifrar el píxel con esta probabilidad. Volvamos hacia imágenes a distintos niveles de gris. Un píxel en tal imagen está representado por el número “n” y puede tener valores “2n”. la aplicación del principio de la compresión de imagen a una imagen de distintos niveles de gris implica que los vecinos inmediatos de un píxel “P” tienden a ser similares a “P”, pero no son necesariamente idénticos. Así pues, RLE no debería emplearse para comprimir tal imagen. En vez de eso se discuten dos enfoques.

Enfoque 3: Separan la imagen a distintos niveles de grises en imágenes a dos niveles de “n” y comprimen cada uno con RLE y códigos de prefijo. El principio de la compresión de imagen parece implicar intuitivamente que dos píxeles adyacentes que son similares en la imagen a distintos niveles de gris serán

idénticos en la mayoría de las imágenes a dos niveles de “n”. Sin embargo, no es verdadero, como lo demostró el ejemplo siguiente. Imaginémosnos una imagen a distintos niveles de grises con “n” = 4 (es decir, píxel 4-bit, o 16 matices de gris.) La imagen puede separarse en cuatro imágenes a dos niveles. Si dos píxeles adyacentes en la imagen original a distintos niveles de gris tienen los valores 0000 y 0001, entonces son similares. Son también idénticos en tres de las cuatro imágenes a dos niveles. Sin embargo, dos píxeles adyacentes con los valores 0111 y 1000 son también similar en la imagen a distintos niveles de gris (sus valores son 7 respectivamente y 8) pero difieren en cada una de las cuatro imágenes a dos niveles. Este problema se produce porque los códigos binarios de los números enteros adyacentes pueden diferir por varios bits. Los códigos binarios de 0 y de 1 difieren por un bit, los de 1 y de 2 difieren por dos bits, y los de 7 y de 8 difieren por cuatro bits. La solución consiste en concebir códigos binarios especiales como los códigos de todos los números enteros consecutivos, l e $l + 1$ diferirán por un bit solamente.

Enfoque 4: Emplean el contexto de un píxel para prever su valor. El contexto de un píxel es los valores de algunos de sus vecinos. Podemos examinar algunos vecinos de un píxel “P”, calcular una media “A” de sus valores, y prever que “P” tendrá el valor “A”. El principio de la compresión de imagen nos indica que nuestra previsión será correcta en la mayoría de los casos, casi correcta en muchos casos, y para nada correcta en otros casos. Podemos decir que el valor previsto del píxel “P” representa la información superflua en “P”.

Enfoque 5: Transforman los valores del píxel y cifran los valores transformados. La compresión es realizada por redundancia reductora. La redundancia de una imagen es causada por la correlación entre los píxeles, pues transformar la representación de los píxeles de manera que no tengan ninguna relación entre ellos elimina la redundancia. Es también posible pensar en la transformación en términos de entropía de la imagen. En una imagen muy correlacionada, los píxeles tienden a tener valores probables, que tienen como consecuencia la entropía máxima. Si se transforma los píxeles de manera que no tengan ninguna relación, algunos valores de píxel se vuelven comunes,

teniendo por lo tanto grandes probabilidades, mientras que otros son raros. Esto tiene como consecuencia la pequeña entropía. La cuantificación de los valores transformados puede producir la compresión eficaz de imagen de tipo "lossy". Queremos que los valores transformados sean independientes porque la codificación de los valores independientes es más simple para construir un modelo estadístico.

Enfoque 6: Veamos ahora hacia imágenes de color. Un píxel en tal imagen consta de tres componentes de color, como el rojo, el verde, y el azul. La mayoría de las imágenes de color son de tono continuo. El principio de este enfoque es separar una imagen a tono continuo de color en tres imágenes a distintos niveles de gris y comprimir cada uno de los tres separadamente, utilizando los enfoques 3, 4, o 5. Para una imagen a tono continuo, el principio de la compresión de imagen implica que el píxel adyacentes es similar, aunque quizá no idéntico a los niveles de los colores. Sin embargo, los colores similares no significan valores similares de píxel. Consideran, por ejemplo, los valores del píxel de 12-bit donde se expresa cada componente de color en cuatro bits. Así pues, los 12 bits 1000 0100 0000 representan un píxel cuyo color es una mezcla de ocho unidades de rojo (alrededor del 50%, puesto que el máximo es 15 unidades), cuatro unidades del verde (alrededor del 25%), y ningún azul. Se imaginan ahora dos píxeles adyacentes con los valores 0011 0101 0011 y 0010 0101 0011. Tienen los colores similares, puesto que solamente sus componentes rojos difieren, y solamente por una unidad. Sin embargo, una vez considerados como números 12-bit, los dos números 001101010011 y 001001010011 son muy diferentes, puesto que difieren en uno de los bits más significativos. Una importante propiedad de este enfoque es que se debe emplear una representación de color de chrominance (CbCr) en vez del RGB más común. La ventaja de esta representación es que el ojo es sensible a pequeños cambios en el brillo pero no en el color representado por CbCr. Esto permite la pérdida de datos considerables en los componentes CbCr, permitiendo al mismo tiempo para descifrar la imagen sin pérdida evidente significativa de calidad.

b. Las normas de compresión para el vídeo numérico.

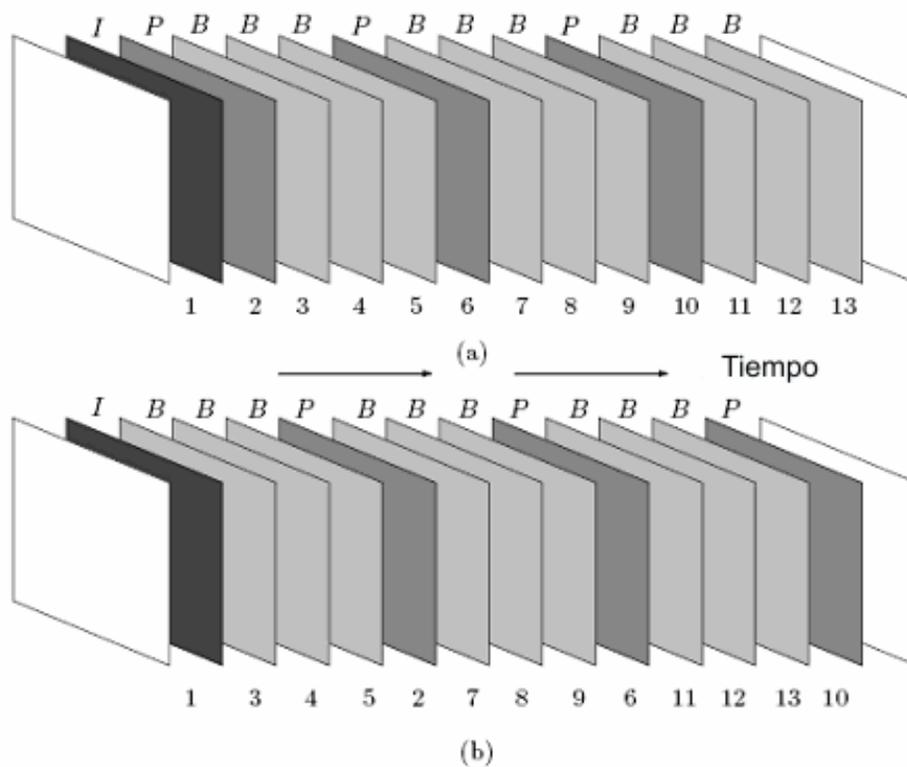
MPEG (*Moving Picture Expert Group*) es la norma ISO del sistema numérico de compresión de imágenes móviles (vídeo) y acústica asociada. MPEG-1 (oficialmente conocido bajo el nombre ISO 11172) se normalizó en 1994 y fue la primera generación de MPEG que ofrecía una compresión vídeo y un audio estéreo de dos vías a una velocidad de cerca de 1.5 Mbps (Méga bits por segundo) para el almacenamiento sobre CD-ROM. ISO desarrolló la segunda generación, el MPEG-2 (oficialmente conocido bajo el nombre de ISO 13818) en 1995 ofreciendo unas velocidades de 6 a 10 Mbps es la norma actual de los DVD. ISO define ahora la segunda generación de estándares MPEG-4 para afrontar los nuevos retos de la codificación visual llamada DIVX o XVID. El Comité MPEG trabaja actualmente sobre una nueva norma llamada MPEG-7. No se trata de una nueva norma visual de compresión, el MPEG-7 no definirá ningún nuevo algoritmo de compresión. Normaliza simplemente el formato del fichero vídeo y su descripción de metadata con el fin de definir una norma para la información multimedia cifrados con los codecs. Se ha lanzado recientemente un otro grupo de Trabajo en el Comité de MPEG con el fin de desarrollar el MPEG-21. El Mpeg-21 debería "permitir una utilización mayor y con mas transparencia de los recursos multimedia a través de un amplio abanico de redes y dispositivos." En paralelo en la ISO, el ITU define las normas de compresión para aplicaciones como la teleconferencia. H261 es una norma desarrollada en 1990 para la parte visual de codificación. H.320 para transmitir el vídeo a las producciones de 56 Kbps hasta 2 Mbps. H.263 es la norma desarrollada en 1995 para la codificación visual que permite la transmisión vídeo a un velocidad inferior a 28.8 Kbps por el canal de telecomunicación. H.263L es la próxima norma que debería responder a las nuevas exigencias de la telecomunicación. La compresión vídeo se basa en dos principios. El primero es la redundancia espacial que existe en cada imagen. El segundo es el hecho de que la mayor parte del tiempo, una imagen es muy similar a la imagen que la precede o a la que la sigue. Esto se llama la redundancia temporal. Un método tradicional de compresión vídeo prevé entonces de empezar por cifrar la primera imagen utilizando un método idéntico a la que se utiliza para la compresión de imagen fija, y seguir cifrando cada

imagen sucesiva identificando las diferencias entre ellas y cifrando estas diferencias en el medio. Si la imagen es muy diferente de su antecesor (como para la primera imagen de un nuevo plan), deberían cifrarle independientemente de cualquier otra imagen. En la literatura de compresión video, una imagen que es cifrado sobre la base de su antecesor se llama "*inter frame*" (o *inter*), mientras que una imagen que se cifra independientemente se llama "*intra frame*" (o "*intra*".)

La compresión vídeo está normalmente "*lossy*". La codificación de una imagen en función de un antecesor presenta algunas deformaciones. En consecuencia, la codificación de la imagen siguiente aumenta aún más la deformación. Incluso en la compresión video "*lossless*", una imagen puede perder una serie de bits, por ejemplo, durante la transmisión. Si una imagen perdió una serie de bits, entonces todas las imágenes siguientes, hasta la próxima *intra*, serán decodificadas incorrectamente. Esta es la razón por la que las imágenes *intra* deben ser intercalar de vez en cuando dentro de la serie de imagen, no simplemente a su principio. Se señala una imagen *intra* I, y se señala una imagen *inter* P (para profético.) Una vez esta idea entendida, es posible generalizar el concepto de una imagen *inter*. Tal imagen puede cifrarse sobre la base de uno de sus antecesores y también sobre uno de sus sucesores. Normalmente, un codificador no debería nunca emplear información que no esté disponible al decodificador, pero la compresión video es especial debido a las grandes cantidades de datos implicados.

La velocidad del codificador no es generalmente importante, pero el decodificador, debe ser rápido. Un caso típico es una película DVD jugada al revez, debe jugarlo al revez a una tasa correcta de imágenes por segundo. Esta es la razón por la que un decodificador trabaja simultáneamente sobre varias imágenes. Así se puede ahora imaginar una situación donde el codificador cifra la imagen 2 basada en las dos imágenes 1 y 3, y escribe las imágenes sobre el DVD en el orden 1, 3, 2. El decodificador las lee en este mismo orden, descifra las imágenes 1 y 3 en paralelo, produce la imagen 1, luego descifra la imagen 2 basada en las imágenes 1 y 3. Las imágenes deberían, naturalmente, etiquetarse claramente I o P. Se señala una imagen que se cifra basándose en las dos imágenes pasada y futura como B (para bidireccional.)

Prever una imagen basado en su sucesor toma todos su sentido cuando el movimiento de un objeto en una imagen hace que se descubre poco a poco parte de la imagen de fondo. Este sector de la imagen puede ser en parte conocida en la imagen actual, pero puede ser más detallado en la imagen siguiente. De hecho, la mayoría de las imágenes cifradas son de este tipo. La figura siguiente permite incluir este principio donde (a) es el orden en el cual es cifrado las series de imágenes por el codificador y recibidas por el decodificador, y (b) es el orden en el cual el decodificador produce las imágenes sobre la pantalla de televisión:



A continuación algunos ejemplos de técnicas utilizadas para la compresión de imágenes vídeo.

Exacción (SubSampling): El codificador elige una imagen sobre dos y lo escribe en el medio. Esto aporta un factor de compresión de 2. El decodificador entre una imagen y lo reproduce para crear dos imágenes.

Diferenciación: Se compara una imagen a su antecesor. Si la diferencia entre ellos es pequeña (solo algunos Píxeles), el codificador cifra los Píxeles que son diferentes escribiendo tres números en el soporte para cada Píxeles: sus

coordenadas en la imagen, y la diferencia entre los valores del Píxel en las dos imágenes.

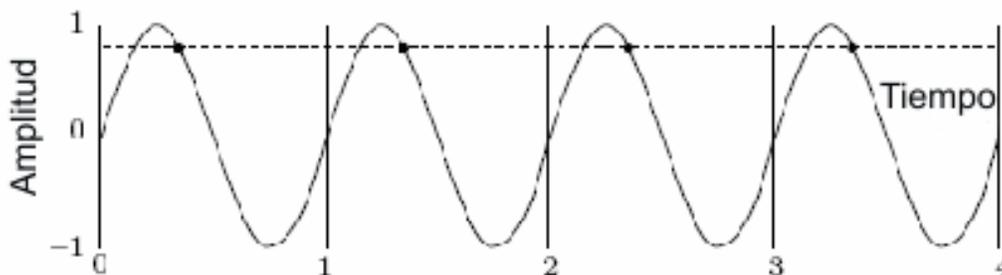
Diferenciación de bloque: se trata de un método basado en la diferenciación donde se divide la imagen en bloques de Píxeles, y se compara cada bloque B en la imagen corriente al bloque P correspondiente en la imagen anterior. Si los bloques difieren más que por unos determinadas cantidades, entonces B se comprime escribiendo sus coordenadas en la imagen, seguido de los valores de todos sus Píxeles (expresados como diferencias) en el soporte. La ventaja es que las coordenadas de bloques son números reducidos (más pequeños que las coordenadas de un Píxel), y estas coordenadas deben escribirse solamente una vez para todo el bloque. En cambio, los valores de todos los píxeles deben ser inscritos en el soporte, pero sabiendo que estos valores son representados como diferencias, esos no toman muchos espacios.

Compensación de Movimiento: En una película video, la diferencia entre las imágenes consecutivas son reducidas. Eso se da gracias a las propiedades de la cinematografía (*zooming, traveling, panning, tilling*, etc). Estas características pueden ser explotadas para obtener una mejor compresión. Si el codificador descubre que se desplazó rígidamente una parte P de la imagen anterior a un diferente lugar en la imagen corriente, entonces P puede comprimirse escribiendo las tres propiedades siguientes en el soporte: su dirección anterior, su dirección actual, y la información que identifican las fronteras de P.

c. Las normas de compresión para el audio numérico.

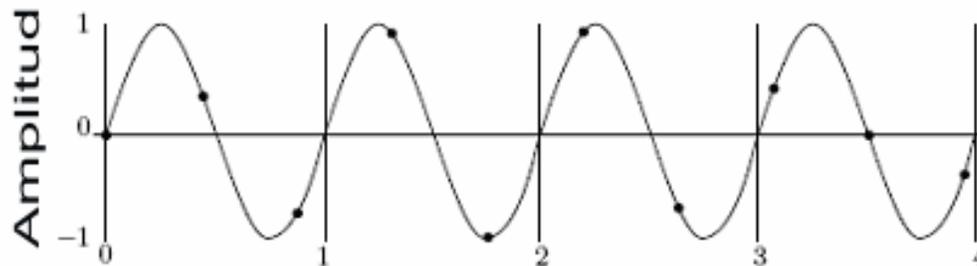
El grupo MPEG lanzó el principio de la normalización de la norma acústica. Las normas de MPEG-1 y MPEG-2 son las primeras normas internacionales en el ámbito de la compresión audio numérico de alta calidad. El sistema de codificación MPEG-1 audio funciona en métodos estéreo a las frecuencias de 32 kilohercios, de 44.1 kilohercios, y de 48 kilohercios. El sistema fue subdividido en tres capas para distintas producciones. MPEG-1 capa I proporciona una calidad a 192 Kbps por canal, mientras que la capa II proporciona una calidad de 128 Kbps, y la capa III de 64 Kbps. El sistema audio de codificación avanzada por MPEG-2 (MPEG-2 AAC) funciona a una frecuencia entre 8 y 96

kilohercios y sostiene hasta 48 canales audio. MPEG-2 AAC se emplea para la norma MPEG-4 ofreciendo transferencias iguales o superior a 16 Kbps por canal. Existen otras herramientas audio de codificación de gran interés, el dolby CA-3, el Philips DCC, etc. El MP3 es el término que los usuarios de Internet emplean generalmente como norma audio. Sin embargo, MP3 no es uno, se basa en MPEG-1 capa III. Una forma de onda típica tal como se representa un sonido en un Osciloscopio, no es una simple ola de seno que tiene un período regular y que puede ser descrito como función matemática simple. En vez de eso, es una combinación de distintas frecuencias a distintas amplitudes y fases. Así como cualquier otra ola, el ruido tiene tres atributos importantes, su velocidad, su amplitud, y su período. La frecuencia de una ola no es un atributo independiente; es el número de períodos que se producen en una unidad de tiempo (un segundo.) Al igual que las imágenes, los sonidos puede digitalizarse y almacenarse. Adonde un Píxel esta transformado en coordenadas, los sonidos lo pueden también. El esquema siguiente muestra un ejemplo típico de una onda sonora empezando a cero y oscilando varias veces.



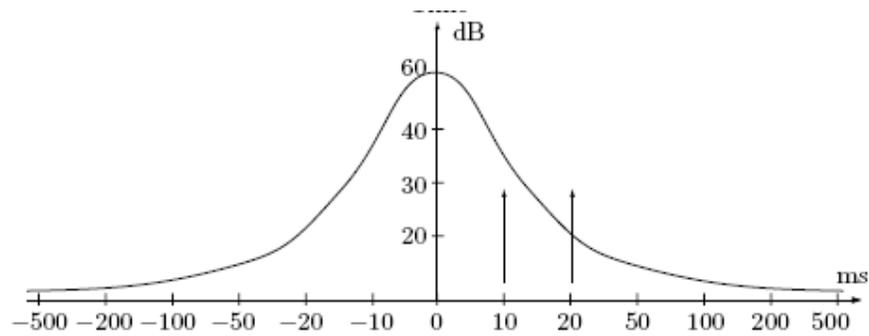
Tal esquema es una representación analógica del sonido. La digitalización del sonido se hace midiendo el voltaje durante un tiempo una serie de veces, traduciendo cada medida en número. Este proceso se llama la exacción, el "*sampling*". El dispositivo utilizado para esta exacción es un convertidor analógico-digital (CDA.). Al contrario, se utiliza un convertidor digital-analógico (DAC) para jugar los sonidos sobre altos parlantes. Queda claro que un elevado tipo de exacción implica como consecuencia una mejor reproducción, pero también mucho más muestras y en consecuencia archivos de mayores tamaño. Así pues, el problema principal en la exacción consiste en saber cuántas veces tomar un sonido. Reanudemos el ejemplo del esquema preceden, en este esquema, la exacción se hace cuatro veces, en el mismo momento. Si se debían escuchar este sonido escucharíamos un "buzz" idéntico

al de un despertador. El problema es que la exacción es demasiado baja para obtener una representación fiel de los sonidos. La solución a este problema consiste en tomar el sonido a una tasa superior al de Nyquist, el cual es dos veces la anchura de banda del sonido (la anchura de banda es la diferencia entre el máximo y el mínimo de la onda sonora.) Así pues, si un sonido contiene frecuencias entre 500 hertzios y 2 kilohercios (la anchura de banda de la voz humana), debería ser tomada a un poco más de 3 kilohercios. Tal tipo de exacción garantiza la reproducción correcta del sonido. Esto se ilustra en el esquema siguiente, que muestra 10 exacción equidistantes tomada sobre cuatro períodos. Tenga en cuenta que las muestras no deben tomarse al máximo o al mínimo de la onda; pueden tomarse de cualquier punto.



La gama de la audición humana es típicamente de 16-20 Hz. a 20.000-22.000 Hz., según la edad y la persona. Cuando el sonido está digitalizado a alta fidelidad, se debe entonces tomar una medida superior a la tasa de Nyquist a $2 \times 22000 = 44000$ hertzios. Esta es la razón por la que el sonido numérico de alta calidad se basa en un tipo de exacción de 44.100 hertzios. La existencia de limitaciones en el sistema auditivo humano sugiere para la compresión acústica un enfoque “lossy”. Así pues, el papel del codificador es de determinar las frecuencias que no serán audibles al ser humano y eliminarlas de la gama. Además de esto, se emplean dos propiedades suplementarias del sistema auditivo humano en la compresión audio. Estas son el enmascaramiento de frecuencias y el enmascaramiento temporal. El enmascaramiento de frecuencias se produce cuando un ruido que tiene una frecuencia humanamente audible es encubierto por otro ruido que tiene una frecuencia vecina. Un buen método de compresión audio debería definir este tipo de problemas y suprimir este tipo de señal. El enmascaramiento temporal puede producirse cuando un ruido muy fuerte A va precedido o va seguido de un ruido más escaso B a una frecuencia muy cercana. Si el intervalo de tiempo entre los

ruidos es corto, B puede no ser audible. El esquema siguiente ilustra este ejemplo:



El límite máximo de audición del ser humano desciende de un golpe debido a un sonido muy potente al tiempo 0, primero precipitadamente, luego mas lentamente. En este ejemplo, un ruido más escaso que 30 decibelio no será audible al tiempo 10 o si se produce antes que el limite del sistema auditivo humano bajo suficientemente.

5. Conclusión

La compresión de los datos esta en el camino de tomar un papel importante debido al desarrollo de las redes y al multimedia. Su importancia sobre todo se debe al desfase que existe entre las existencias materiales de los dispositivos que utilizamos y las necesidades que expresan los usuarios. Aun si este desfase no existiría, la compresión permite de todas formas hacer economías. Los sofisticados métodos que se utilizan actualmente son generalmente eficaces y utilizan teorías bastante complejas, los métodos emergentes son aun mas prometedores pero somos lejos de haber agotado todas las pistas de investigación. Los métodos del futuro sabrán seguramente adaptarse a la naturaleza de los datos a comprimir y utilizarán la inteligencia artificial.

6. Bibliografía

Fuentes de Internet:

LZW and GIF Explained,

“<http://www.ece.uiuc.edu/~ece291/class-resources/gpe/gif.txt.html>”.

Mp3licensing.com

“<http://www.mp3-tech.org/>”

Fraunhofer-Gesellschaft

“<http://www.iis.fraunhofer.de/amm/>”

“<http://www.id3.org/>”

mp3 info

“<http://www.mp3.com/>”

Mpeg

“<http://www.mpeg.org/MPEG/index.html>”

nyquist theorem

“<http://cnx.rice.edu/content/m10791/latest/>”

“<http://www.vqf.com/>”

ISO

“[http://www.iso.ch/.](http://www.iso.ch/)”

Jpeg

“[http://www.jpeg.org/jbighomepage.html.](http://www.jpeg.org/jbighomepage.html)”

JPEG 2000 Organization

“[http://www.jpeg.org/JPEG2000.htm.](http://www.jpeg.org/JPEG2000.htm)”

Fuentes de e-book:

“JPEG2000 standard for image compression : concepts, algorithms and VLSI”,
Acharya, T., Wiley-Interscience publication, ISBN 0-471-48422-9

“PDF (2001) Adobe Portable Document Format Version 1.4, third edition,
Reading, MA”

“Addison-Wesley, December.”

“Sayood, Khalid (2000) Introduction to Data Compression, 2nd Ed., San
Francisco, CA,
Morgan Kaufmann.”

“Salomon, David (2005) Data Compression The Complete Reference, third
edition, Springer-Verlag”