

# COMPARACIÓN ENTRE COMPRESIÓN DE AUDIO EN DIFERENTES FORMATOS DE IMÁGENES EQUIVALENTES Y EL FORMATO DE COMPRESIÓN MP3

Daniel Barzola, Roberto Cabrera, Ing. Patricia Chávez, Ing. Juan Carlos Avilés

Facultad de ingeniería en electricidad y computación

Escuela Superior Politécnica del Litoral (ESPOL)

Km. 30.5 Vía Perimetral, 09015863, Guayaquil, Ecuador

[danibarzo@hotmail.com](mailto:danibarzo@hotmail.com), [dcabrera@fiec.espol.edu.ec](mailto:dcabrera@fiec.espol.edu.ec), [pchavez@fiec.espol.edu.ec](mailto:pchavez@fiec.espol.edu.ec),  
[jcaviles@fiec.espol.edu.ec](mailto:jcaviles@fiec.espol.edu.ec)

## RESUMEN

Este trabajo tiene por objeto enseñar la técnica de conversión de archivos de audio en imágenes equivalentes comprimidas, la cual es relativamente nueva y poco investigada, y luego compararla con el formato de compresión MP3. Esta novedosa técnica se basa en obtener imágenes equivalentes de un archivo de audio en formato WAV mediante el procesamiento digital del mismo. Las imágenes se pueden comprimir en diferentes formatos de imagen (JPG, TIFF, PNG) obteniendo así una reducción significativa de la cantidad de bytes. En el receptor mediante un proceso de descompresión y decodificación se recupera la voz o audio a partir de las imágenes comprimidas que se reciben. Mediante el uso de Matlab se aplican como parte de la técnica la transformada rápida de Fourier fft, diferentes tipos de formatos de compresión de imágenes y la transformada de Wavelet aplicada para la reducción de ruido blanco. Después de que esta técnica es explicada, se la comparará cuantitativa y cualitativamente con el ya conocido formato de compresión MP3.

*Palabras claves: Imágenes equivalentes comprimidas, JPG, TIFF, PNG, MP3, FFT, WAV.*

## ABSTRACT

The objective of this work is to teach the technique to convert audio files to equivalent images compressed which is relatively new and little investigated and then to compared it with the format of compression MP3. This new technique is based on obtaining equivalent images of a file of audio in format WAV by means of the digital processing of the same one. The images can be compressed in different formats of image (JPG, TIFF, PNG) then it obtains a significant reduction of the quantity of bytes. In the receiver by means of a decompression and decoding processing recovers the voice or audio from the images compressed that they are received. By means of the use of Matlab they apply as part of the technique: the Fast Fourier transform (FFT), different images compression formats types, and the Wavelet transform applied for the reduction of white noise. After this technique is explained, it will be compared quantitative and qualitatively with the already compression format acquaintance MP3.

*Keywords: Equivalent images compressed, JPG, TIFF, PNG, MP3, FFT, WAV.*

## INTRODUCCIÓN

En la actualidad sabemos que existen muchas técnicas digitales de compresión de

audio, una de ellas y la más utilizada es la técnica de compresión mediante el formato MP3 que proporciona una reducción significativa en el tamaño de los archivos de

audio, por ejemplo los archivos WAV y a la vez mantiene en cierto grado la calidad del sonido. Estas características del formato Mp3 hace que el tamaño del archivo sea pequeño en comparación a un archivo de audio que no ha sido comprimido con éste formato o sencillamente no ha pasado por un proceso de compresión. Sin embargo, la técnica de compresión en formato MP3 no es la única, pues últimamente se han investigado sistemas de compresión que permiten un considerable porcentaje de reducción del tamaño de los archivos de audio. Se sabe que estas técnicas o sistemas de compresión requieren de dos algoritmos, uno para comprimir los archivos en el origen o fuente y el otro para descomprimir los archivos en el destino. En teoría estos algoritmos de compresión y descompresión se conocen como codificadores y decodificadores respectivamente. Una comparación cualitativa y cuantitativa entre la técnica de compresión del formato MP3 y las nuevas técnicas o sistemas de compresión podría hacer notar ventajas y desventajas entre ambas. Es importante indicar, que en cualquier procesamiento digital de señales de audio, existe una pérdida de la señal original con respecto a la recuperada, pues no siempre es posible obtener una señal perfecta. Una de las características de las señales que pasan a través de un codificador en un procesamiento digital de señales, es la pérdida de ciertos parámetros como amplitud en ciertas frecuencias de la señal, y también adición de ruido blanco; este codificador puede usar cualquier algoritmo de compresión, por ejemplo un algoritmo de compresión de imágenes JPEG. La decodificación puede resultar un poco difícil en el caso que se quiera mantener características muy similares a la señal original, es por eso que debe ser necesario utilizar técnicas que permitan disminuir el ruido de la señal recuperada con el decodificador, y a la vez mantener amplitud en todo el espectro de frecuencias de la señal muy parecida a la original, por ejemplo se puede citar una de las aplicaciones de la transformada de wavelet para la reducción del ruido blanco o cualquier otro

ruido que se obtenga en la señal recuperada. [1 - 5]

## **EXPLICACION DE LA COMPARACION**

Para la comparación se utilizará cualquier archivo de audio o voz en formato WAV ya que es uno de los más utilizados para almacenar sonidos y es un formato sin ningún tipo de compresión de datos.

Con el uso de un software libre se convertirá los archivos WAV a formato MP3 y viceversa y mediante un algoritmo de compresión que se ha diseñado se convertirá los archivos WAV en imágenes equivalentes, a las que se les aplica tres tipos de formatos de compresión de imágenes como lo son jpeg, png y tif. Mediante otro algoritmo, pero esta vez de descompresión se hará el proceso contrario es decir a partir de las imágenes comprimidas se recuperará un archivo de audio equivalente al archivo original. Por lo tanto se hará una comparación entre la señal original, la recuperada de cualquiera de los formatos de imagen y la recuperada desde el formato MP3 al formato WAV. [5 - 7]

## **PROCEDIMIENTO DE COMPARACIÓN**

Comparación Cuantitativa

Paso 1:

Se compara cuantitativamente los archivos de audio en los formatos que fueron comprimidos (MP3 y formatos de imagen). En esta prueba el parámetro a medir y comparar será el tamaño de los archivos en estos dos tipos de compresión. Lógicamente el archivo con menor tamaño podría considerarse como el más apropiado para la transmisión, sin considerar todavía un análisis cualitativo.

Paso 2:

Comparación cualitativa

Se reproduce el archivo de audio original, los archivos de audio WAV recuperados a partir del formato MP3 y los

archivos de audio recuperados de las imágenes comprimidas para compararlos cualitativamente, para ello se realizaron encuesta a un grupo de personas con preguntas relacionadas a la percepción de calidad de audio en los humanos.

En la figura 1 se muestra el procedimiento a seguir; en que momento se realiza la comparación cuantitativa y cualitativa

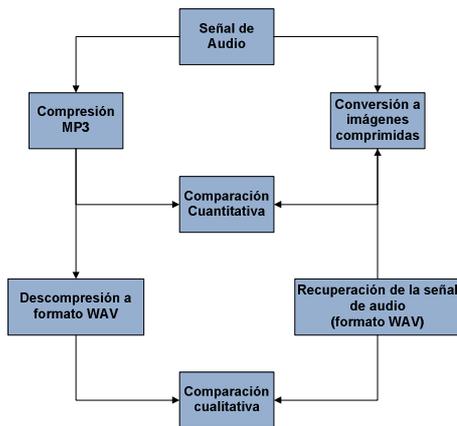


Fig 1. Procedimiento general para la comparación cuantitativa y cualitativa

## FORMULACION DEL ALGORITMO DE COMPRESIÓN

Esta novedosa técnica consiste en convertir tramas de audio en imágenes equivalentes comprimidas con el objetivo reducir el tamaño del archivo y que luego al enviarse por cualquier medio, en el receptor se pueda recuperar una versión muy cercana al audio original.

A la trama de voz se le aplicará la transformada de Fourier. En matlab hacemos uso del algoritmo de esta transformada usando la transformada rápida de Fourier FFT (Fast Fourier Transform). La transformada de Fourier nos entrega muestras complejas por lo cual es necesario separar la parte real e imaginaria y convertirlas por separado en una imagen equivalente. A éstas se les aplica un formato de compresión de imágenes como lo son jpg, png y tif. Ambas imágenes son las que se transmiten a través de algún canal de

datos. El receptor hará el proceso inverso para la recuperación del audio de la señal. [6]

## FORMULACION DEL ALGORITMO DE DESCOMPRESIÓN

Cuando las tramas transmitidas llegan al receptor, se aplica un proceso inverso al algoritmo anterior. Se sabe que estas tramas ahora son dos imágenes, las cuales representan la parte real e imaginaria del vector complejo de la señal de audio, las cuales el receptor primeramente las decodifica y las convierte en el vector complejo y luego obtiene su transformada inversa para recuperar la señal de audio. [6]

## PROCEDIMIENTO DE LA APLICACIÓN DE LOS ALGORITMOS EN MATLAB

En el transmisor se ha diseñado el codificador que utiliza el algoritmo de compresión, mientras que en el receptor se ha diseñado el decodificador que utiliza el algoritmo de descompresión. El archivo que se utiliza es un archivo de audio en formato WAV que se cargará en la interfaz del codificador y será convertido en imágenes ya sea en formato JPEG, TIF o PNG para luego ser procesado por el decodificador y así obtener nuevamente un archivo de audio.

## DESCRIPCION DEL CODIFICADOR

El objetivo del codificador es convertir la señal de audio que está en formato WAV a un archivo de imagen en cualquier formato de compresión, que puede ser JPEG, TIF, PNG.

## Aplicación de la transformada de Fourier

Se guarda los datos de la señal de audio en un vector al cual se le aplica el algoritmo de la transformada rápida de Fourier FFT [9]. De esta manera se obtienen N muestras complejas conjugadas. Para los propósitos de transmisión basta con procesar la mitad de esas N muestras. Si se toma solo una muestra de cada par conjugado, se podrá predecir el otro par cuando se esté recuperando la señal en el decodificador.

De estas  $N/2$  muestras se separan la parte real e imaginaria para tratarlas por separado en la obtención de las imágenes equivalentes [7 - 8].

### Adaptación a formato de imagen

Una imagen es representada como una matriz donde cada elemento es un pixel de la señal. Para una imagen en escala de grises (blanco y negro) solo se necesita una matriz donde cada pixel toma un valor que indica la intensidad de gris. Las imágenes que se obtendrán a partir de las  $N/2$  muestras complejas separadas en parte real y parte imaginaria serán en escala de grises.

El rango de niveles de gris que cada pixel puede tomar depende de la cantidad de bits que se utilice. Estos datos deben ser enteros sin signo, es decir cada pixel debe tomar un valor entero positivo o cero. El rango se definirá eligiendo los bytes a utilizar; por ejemplo si se utiliza un byte (ocho bits) el rango para los niveles de gris será desde 0 hasta 255, si se utilizan dos bytes (16 bits) el rango será desde 0 hasta 65535.

Para definir el tipo de dato entero sin signo usando un byte en Matlab se utiliza el comando `uint8`, para el tipo de dato entero sin signo usando dos bytes el comando es `uint16`. Si se aplica el comando `uint8` a un valor menor a cero o mayor a 255 Matlab asigna el valor mas cercano en el rango permitido; por ejemplo se asigna `uint8` al valor -20, `uint8(-20)` Matlab lo traducirá en 0 y si se asigna `uint8` al valor 300, `uint8(300)` Matlab lo traducirá a 255, de la misma manera funcionará para `uint16`.

Para convertir las  $N/2$  muestras de la parte real e imaginaria a imágenes equivalentes, es necesario primeramente adaptarlas a formato de imagen. Esto debido a que las muestras contienen valores negativos, los que se perderían usando el tipo de dato entero sin signo y si usamos 1 byte también se perderían los valores mayores a 255.

La técnica que se usa es la normalización de las muestras a valores dentro del rango de 0 a 255. Para lograrlo se deben obtener los valores máximos y mínimos de las muestras tanto para la parte real como para la parte imaginaria. Se exponen las fórmulas para la normalización en 1 y 2 siendo  $Xr$  el vector que contiene las muestras de la parte real. La parte imaginaria se deberá normalizar de la misma manera.

$$XRN = \frac{Xr - \min(Xr)}{\max(Xr) - \min(Xr)} \times 255 \quad (1)$$

$$XRI = \frac{Xi - \min(Xi)}{\max(Xi) - \min(Xi)} \times 255 \quad (2)$$

$XRN$ : Es el vector que representa la parte real del espectro de la señal de audio convertido a una imagen con valores desde 0 a 255.

$XRI$ : Es el vector que representa la parte imaginaria del espectro de la señal de audio convertido a una imagen con valores desde 0 a 255.

$Xr$ : Es la parte real del vector que representa el espectro de la señal de audio

$Xi$ : Es la parte imaginaria del vector que representa el espectro de la señal de audio

Para poder desnormalizar correctamente en el receptor, los valores máximos y mínimos son ubicados al final de las imágenes, los que se ubican como si fueran dos pixeles más. Para un archivo de audio relativamente grande se hace necesario utilizar el tipo de dato entero sin signo pero usando dos bytes (16 bits) ya que los valores que se encuentran en las muestras tanto como para la parte real como para la parte imaginaria son valores mucho mayores a 255. Sin embargo la normalización se la sigue realizando para el rango desde 0 a 255, lo cual aporta a una mayor compresión y permite guardar dentro de las imágenes valores grandes que corresponden a los máximos y mínimos.

### Aplicación de los formatos de compresión de imágenes

Una vez que tenemos las imágenes equivalentes de la parte real y la parte imaginaria se pueden aplicar diferentes tipos de formatos de compresión de imágenes. Se

ha elegido JPG porque es un tipo de formato que permite decidir la calidad de la compresión a usar. También con el objeto de comparar la compresión con otros tipos de formatos se ha elegido el formato TIF el cual se usa casi exclusivamente como formato de almacenamiento de imágenes sin pérdidas y sin ninguna compresión al igual que PNG, lo que indica que la imagen que se recupera es exacta a la original.

### Recuperación del archivo de audio en el receptor

Una vez que en el receptor se reciben las imágenes equivalentes, primeramente las descomprimimos y obtenemos sus píxeles. Estos píxeles son los que debemos desnormalizar valiéndonos de los valores máximos y mínimos incluidos en las imágenes. Para ello debemos cambiar el tipo de dato de entero sin signo a tipo double el cual proporciona las magnitudes más grandes y más pequeñas posibles para un número. Se muestra en 3 y 4 la fórmula para la desnormalización.

$$Xr = \frac{(XRN) \times (\max(Xr) - \min(Xr))}{255} + \min(Xr) \quad (3)$$

$$Xi = \frac{(XRI) \times (\max(Xi) - \min(Xi))}{255} + \min(Xi) \quad (4)$$

Una vez desnormalizada tanto la parte real como la parte imaginaria las unimos para obtener las muestras complejas ( $a + jb$ ) y completamos las  $N/2$  muestras restantes construyendo sus pares conjugados ( $a - jb$ ). Luego aplicamos la transformada inversa de Fourier utilizando el algoritmo de la transformada rápida de Fourier inversa (IFFT). La señal recuperada no será exactamente igual a la original ya que proviene de un proceso de compresión de imágenes que aporta pérdidas, sin embargo como se ha explicado el oído humano no reconoce muchas de ellas.

El proceso de convertir la señal a imágenes aporta ruido. Para solucionar este inconveniente, en el receptor hemos incorporado un filtro wavelet para reducción

de ruido. Este filtro ayuda muchísimo en eliminar ruido blanco, logrando obtener una señal recuperada de muy buena calidad.

### Reducción del ruido

En nuestro proceso de convertir una señal de audio a un formato de imagen equivalente, y luego al recuperarla usando el proceso contrario de imagen a un formato de audio, la señal resultante sufre alteraciones con respecto a la original. Estas alteraciones pueden ser adición de ruido, o adición de otras señales no deseadas. Para la reducción de esta clase de ruido, se hará uso de una de las aplicaciones de la transformada de wavelet, pues se usará un filtro de wavelet. En esta parte se elegirá el tipo de filtro wavelet (deb1, deb2, deb3, etc.), y el nivel, el tipo de umbral. Para realizar la opción del ruido se utilizó la función wden de Matlab. Para nuestra señal hemos usado un filtro deb3, con un umbral de tipo "minimaxi", opción de escala "sln", y el nivel de descomposición máximo. Con estos parámetros se pudo obtener una señal de audio recuperada muy parecida a la original. [10 - 11]

### DESARROLLO DE LA SIMULACION EN MATLAB

En esta sección explicaremos el desarrollo de la simulación, hasta obtener el producto final, que es la señal recuperada.

#### Ingreso de la señal original

En esta simulación se ingresa una señal de audio en formato WAV, esta señal puede ser grabada en el instante, y puede ser de una señal de voz, o un archivo de música cualquiera. En el desarrollo de la simulación hemos grabado la siguiente oración: "Procesamiento digital de señales", cuya gráfica se muestra en la figura 2.

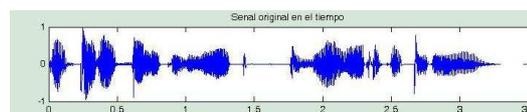


Fig. 2 Señal de la oración grabada

## Conversión de la señal de audio a un formato de imagen

Luego del ingreso de la señal de audio en formato wav, se procede a la conversión a una imagen, usando algún formato de compresión que puede ser JPEG, TIF, o PNG. Además de elegir el formato de compresión se puede elegir el tipo de dato que puede ser UINT 8, o UINT 16. Si el formato de compresión a elegir es JPEG, se puede elegir un porcentaje de compresión.

## Presentación de la señal en el dominio de la frecuencia

Se presenta en la figura 3 y 4 la señal de audio en el dominio de la frecuencia (parte real e imaginaria), esto se hace para hacer una comparación de la señal convertida en imagen

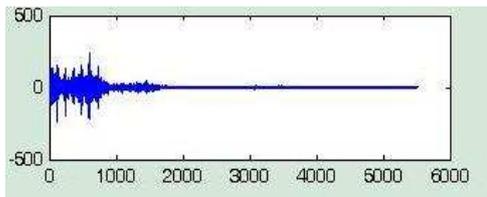


Fig 3. Parte real de la señal en el dominio de la frecuencia

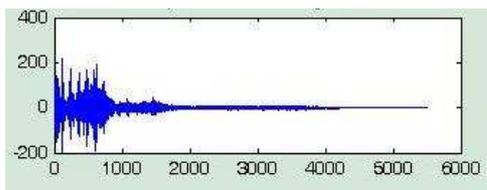


Fig 4. Parte imaginaria de la señal en el dominio de la frecuencia

## Presentación de las imágenes

Se presenta en la figura 5 y 6 las señales de audio convertidas en imágenes. Se tiene una imagen que representa a la parte real de la señal en el dominio de la frecuencia, y otra imagen que representa a la parte imaginaria de la señal en el dominio de la frecuencia [9].

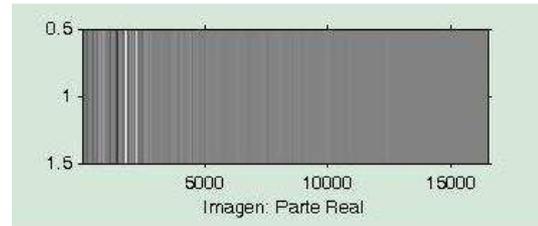


Fig. 5 Imagen que representa a la parte real de la señal en el dominio de la frecuencia

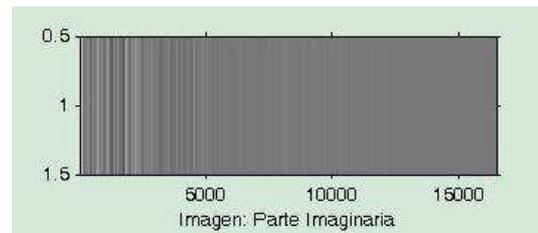


Fig. 6 Imagen que representa a la parte imaginaria de la señal en el dominio de la frecuencia

## Tamaño de los archivos

El tamaño del archivo que hemos elegido es de 87Kb y ha alcanzado niveles de compresión de hasta un 60%. El archivo de música puede ser un archivo de mucho mayor peso que el de 87KB.

## Recuperación del archivo de audio

Se seleccionan las imágenes que serán convertidas a un formato de audio, en este caso el formato WAV [5]. Se selecciona la imagen o las imágenes que representan a la parte real del espectro de la señal de audio y se selecciona la imagen o las imágenes que representan a la parte imaginaria del espectro de la señal de audio. A continuación se presenta en la figura 7 la señal recuperada después de todo el proceso.

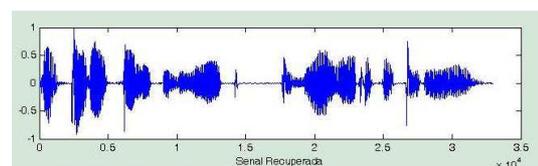


Fig. 7 Señal Recuperada

Una vez que se recupera la señal de audio a partir de las imágenes, es posible escuchar la señal recuperada, en el caso que la señal presente ruido blanco o señales no deseadas, se tiene la posibilidad de elegir un filtro de wavelet, para la reducción del ruido o la señal no deseada antes mencionada. La selección del filtro es opcional, en el caso de hacerlo se tiene que establecer ciertos parámetros.

### Parámetros del filtro de wavelet

Para disminuir considerablemente el ruido que se genera al recuperar la señal audio, se ha utilizado la función 'wden' de matlab. El formato de esta función es el siguiente:

XD = wden(X,TPTR,SORH,SCAL,N,'wname').

Donde XD es la versión con ruido disminuido de la señal de entrada X, en el campo de TPTR se coloca la regla para la selección del umbral; las opciones son: 'minimaxi', 'rigrsure', 'heursure' y 'sqrtwolog'. En SORH se coloca el tipo de la función de transferencia a utilizar para el proceso de thresholding; las opciones son 's' que significa soft-thresholding y 'h' que significa hard-thresholding. En SCAL se coloca el nombre de reescala para el umbral; las opciones son: 'one', para que sea sin reescala, 'sln', para reescala usando una sola estimación de nivel de ruido basada en el primer nivel de coeficientes y 'mln' para reescala usando una estimación de nivel de ruido dependiente del mismo. En N se elige el nivel de descomposición de la señal de entrada X para la eliminación del ruido. Y por último en 'wname' se coloca el nombre del filtro Wavelet ortogonal a usar. Las opciones son las familias Wavelet: 'Daubechies', 'Coiflets', 'Symlets', 'Discrete Meyer', 'Biorthogonal' y 'Reverse Biorthogonal' junto con sus respectivas variaciones. [10 - 11]

### Comparación Cuantitativa

Para comparar cuantitativamente la compresión en imágenes equivalentes con la compresión MP3 [1] se considera el tamaño

del archivo comprimido en estos dos formatos.

En este artículo se presenta el resultado de un archivo de audio WAV [5], en este caso una frase que ha sido generado desde el micrófono de una computadora.

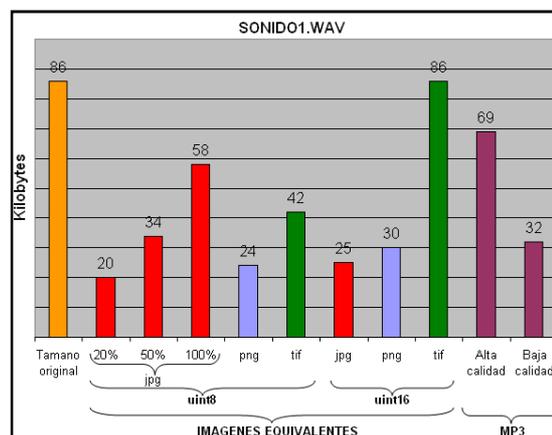


Fig. 8 Comparación de los tamaños usando el archivo SONIDO1.WAV

En la figura 8 se muestra que para el archivo de audio con tipo de dato uint8 y formato JPG se obtuvo con una calidad de 20 % de compresión, un archivo de 20 Kb. Con calidad de compresión de 50% se obtuvo un archivo de 34 Kb, y con una de 100% el archivo quedó con un tamaño de 58 Kb.

Con tipo de dato uint8 y formato PNG se redujo el archivo original a 24 Kb y con formato TIF fue de 42 Kb. Usando tipo de dato uint16 y formato jpg se redujo el tamaño del archivo original a 25 Kb. Con formato PNG en cambio se redujo a 35 Kb.

Usando tipo de dato uint16 y formato TIF no se obtuvo ninguna compresión. Con el formato MP3 usando alta solo se obtuvo una reducción del tamaño del archivo dejándolo en 69 Kb. Usando baja calidad para el formato MP3 se redujo a 35 Kb. Para ambos casos de la calidad del formato MP3, esta compresión es superada por la compresión en imágenes equivalentes.

## Comparación Cualitativa

Para comparar cualitativamente la compresión en imágenes equivalentes con la compresión MP3 se ha usado un archivo que contiene una frase grabada: “probando la calidad del audio recuperado” Con este archivo se ha realizado una encuesta que consta de cuatro partes.

En la primera parte se compara el audio original con el audio recuperado a partir de las imágenes comprimidas en los tres formatos que se ha utilizado, esto es JPG, PNG y TIF con tipo de dato uint8 y uint16 (Ver figura 9). Se hace escuchar al encuestado el audio original y luego el audio recuperado (sin mencionar cual es cada archivo) para que responda si encuentra diferencias o no en los archivos de audio que escuchó. En caso de que encuentre diferencias se les hace anotar en un campo de observaciones.

En la segunda parte se compara el audio original con el audio recuperado a partir de MP3. De igual manera se hace escuchar al encuestado el audio original y luego el audio recuperado (sin mencionar cual es cada archivo) para que responda si encuentra o no diferencias en estos archivos y que anote un comentario en el campo de observaciones (Ver figura 10)

En la tercera parte se compara el audio recuperado a partir de las imágenes equivalentes con el audio recuperado a partir de MP3. En esta parte primeramente se le hace escuchar al encuestado el audio original y luego los archivos de audio recuperados a partir de las imágenes y a partir de MP3 sin mencionar sus nombres. Luego el encuestado elije basándose en el audio original, cual de estos dos últimos archivos prefiere y se le hace marcar su elección en la hoja de encuesta. (Ver figura 11)

En la cuarta parte de la encuesta se anota la calificación de la máxima calidad en el audio recuperado a partir de las imágenes equivalentes en cualquiera de los formatos

que se ha utilizado. Para ello, se ha basado en que si el audio original tiene un valor de diez puntos y el encuestado en algún momento de la primera parte de la encuesta marca que no encuentra diferencias entre los archivos de audio que ha escuchado, entonces la máxima calidad de audio recuperado también sería de diez puntos. En el caso de que un encuestado haya marcado en la primera parte de la encuesta que si hay diferencias en todos los formatos, entonces se le hace escribir su calificación del uno al diez del audio recuperado a partir de las imágenes indicándole cuál es el audio original (indicándole que se base que este vale diez puntos) y cuál el audio recuperado.

## Resultados de la comparación cualitativa

En las figuras 9, 10, 11 se muestran los resultados de las encuestas realizadas, donde se usó el archivo FRASE.WAV

Formato	Opción	Elección de los encuestados	Total de encuestados	Porcentaje	
UINT 8	JPG	SI	13	26%	
	JPG	NO	37	74%	
	PNG	SI	12	50	24%
		NO	38	50	76%
	TIF	SI	12	50	24%
		NO	38	50	76%
UINT 16	JPG	SI	11	22%	
	JPG	NO	39	78%	
	PNG	SI	11	50	22%
		NO	39	50	78%
	TIF	SI	15	50	30%
		NO	35	50	70%

Fig. 9 Audio original vs audio recuperado a partir de los formatos de imagen

Opción	Elección de los encuestados	Total de encuestados	Porcentaje
SI	16	50	32%
NO	34	50	68%

Fig. 10 Audio original vs audio recuperado a partir de MP3

Método de recuperación de audio	Elección	Total de Encuestados	Porcentaje
A partir de la imágenes	13	50	26%
A partir de MP3	17	50	34%
Ambos	20	50	40%

Fig. 11 Audio recuperado a partir de las imágenes vs audio recuperado a partir de MP3

En la figura 9 se muestra que para un tipo de dato uint 8, un 74% de los encuestados respondieron que no encontraban diferencias entre el audio original y el audio recuperado para el formato JPG. Para el formato PNG un 76% no encontraron diferencias y de igual manera un 76% para el formato TIF (Tabla IX). Sin embargo, para el 26% de los encuestados que sí encontraron diferencias en el formato JPG, el 69% anotó en las observaciones comentarios favorables para el audio recuperado.

Para el 24% de los encuestados que sí encontraron diferencias en el formato PNG, el 58% anotó en las observaciones comentarios favorables para el audio recuperado. Y para el 24% de los encuestados que sí encontraron diferencias en el formato TIF, el 58% anotó en las observaciones comentarios favorables para el audio recuperado.

Para tipo de dato uint 16, un 78% de los encuestados respondieron que no encontraban diferencias entre el audio original y el audio recuperado para el formato JPG. Para el formato PNG un 78% no encontraron diferencias y de igual manera un 70% para el formato TIF. Sin embargo, para el 22% de los encuestados que sí encontraron diferencias en el formato JPG, el 64% anotó en las observaciones comentarios favorables para el audio recuperado.

Para el 22% de los encuestados que sí encontraron diferencias en el formato PNG, el 73% anotó en las observaciones comentarios favorables para el audio recuperado. Y para el 30% de los encuestados que sí encontraron diferencias en el formato TIF, el 60% anotó en las observaciones comentarios favorables para el audio recuperado.

En la figura 10 se muestra que para el caso de MP3, del 32% de los encuestados que dijeron que sí encontraron diferencias, un 56% anotó comentarios favorables para el audio recuperado. El 68% de los encuestados no encontraron diferencias.

Estos comentarios favorables refiriéndose al segundo archivo de audio que

escucharon, el cual se trataba del audio recuperado, fueron mencionados en cada formato de imágenes y en MP3, y fueron tales como:

Se escucha más clara y menos ruidosa  
Se escucha más alto  
Se escucha con más volumen  
Es más nítido  
Se escucha mejor  
Es más fuerte la voz  
Se escucha mejor la pronunciación  
La diferencia es mínima y otros muy parecidos

En la tercera parte de la encuesta, donde los encuestados elegían cuál audio recuperado preferían basándose en el audio original, se obtuvo que un 26% prefirieron el audio recuperado a partir de las imágenes, un 34% prefirieron el audio recuperado a partir de MP3 y un 40% eligieron ambos.

En la cuarta parte de la encuesta, donde se pedía una calificación a la máxima calidad del audio recuperado a partir de las imágenes, se indicó a los encuestados que se basen en la idea de que el audio original vale 10 puntos, es decir que si en algunos de los formatos de imágenes elegían que no encontraban diferencias entonces la máxima calidad del audio recuperado también era de 10 puntos. El resultado fue que todos los encuestados eligieron por lo menos una vez, para algún tipo de formato de imágenes que no encontraban diferencias entre el audio original y el audio recuperado, obteniendo de esta manera una calificación promedio de 10 puntos para la máxima calidad del audio recuperado (FRASE.WAV).

## **CONCLUSIONES Y RECOMENDACIONES**

Los formatos JPG y PNG son los que ofrecen mayor compresión para los archivos de audio. Con JPG se puede llegar a una gran compresión ya que es posible elegir hasta el 1% de calidad de compresión, sin embargo esto significaría obtener una calidad muy pobre en la señal recuperada.

El tipo de dato uint16 permite obtener un mayor porcentaje de compresión del audio en imágenes equivalentes en los formatos JPG y PNG; mientras que en el formato TIF el porcentaje de compresión es considerablemente menor.

El formato TIF ofrece una calidad muy buena al recuperar la señal, comparable con JPG y PNG, sin embargo debido a la poca compresión que se consigue, en una aplicación real podríamos descartar este formato. Como máximo con el formato TIF se pudo lograr una compresión al 42%.

El formato PNG ofrece una compresión comparable a JPG con calidad de compresión del 25% y utilizando tipo de dato uint8, pero con la diferencia que la calidad de la señal recuperada es muy superior en PNG.

La compresión de audio en imágenes equivalentes, ofrece una mayor compresión que en el formato MP3.

La técnica de recuperación de audio a partir de imágenes equivalentes permite obtener una señal de muy buena calidad, incluso comparable con MP3.

El tipo de dato uint16, permite obtener una mejor calidad de la señal recuperada que el tipo de dato uint8. Esto debido a que uint16 no admite pérdidas y aporta una muy buena compresión de datos.

La compresión de audio en imágenes equivalentes ofrece un gran ahorro en memoria de almacenamiento, comparable al formato MP3.

## REFERENCIAS

- [1] "Formato de compresión MP3"  
<http://es.wikipedia.org/wiki/MP3>
- [2] PULGARÍN JUAN D., ACOSTA CARLOS D., CASTELLANOS GERMÁN, "Reducción de perturbaciones mediante molificación discreta y umbralización por

Wavelets". Scientia et Technica Año XIII, No 35. Agosto. 2007.

[3] Llanos Diego R., Cardeñoso Valentín., "Filtrado de señales de voz a través de "Wavelet shrinkage". Julio. 1998

[4] "Compresión de imágenes"  
<http://alojamientos.us.es/gtocom/pid/pid6/pid61.htm>

[5] "Waveform Audio Format"  
[http://es.wikipedia.org/wiki/Waveform\\_Audio\\_Format](http://es.wikipedia.org/wiki/Waveform_Audio_Format)

[6] Kaschel Héctor, Watkins Francisco, San Juan Enrique, "Compresión de voz mediante técnicas digitales para el procesamiento de señales y aplicación de formatos de compresión de imágenes". Rev. Fac. Ing. - Univ. Tarapacá, vol. 13 N° 3, pp. 4-10. Octubre. 2005

[7] "Algoritmo de compression con pérdida".  
[http://es.wikipedia.org/wiki/Algoritmo\\_de\\_compresi%C3%B3n\\_con\\_p%C3%A9rdida](http://es.wikipedia.org/wiki/Algoritmo_de_compresi%C3%B3n_con_p%C3%A9rdida)

[8] PROAKIS J.G., MANOLAKIS D.G., "Digital Signal Processing: Principles, Algorithms, and Applications", MacMillan Publishing, 1992.

[9] MADISETI Vijay K., WILLIAMS Douglas B., "Digital signal processing", Chapman & HALL/CRCnetBASE, 1999.

[10] "Wavelet Toolbox".  
<http://www.mathworks.com/access/helpdesk/help/toolbox/wavelet/>

[11] WASEGHI Sated V. "Advanced Digital Signal Processing and Noise Reduction", John Wiley & Sons Ltda., Second Edition, 2000.