Práctica A1
Formatos y edición de audio

1 Introducción

1.1 Formatos de audio

Cualquier fichero de audio viene definido principalmente por tres parámetros que deberemos manejar a la hora de trabajar:

1. **Frecuencia de muestreo**: Indica cuántas muestras por segundo se han tomado a la hora de convertir el sonido de analógico a digital. Se mide en Hertzios (Hz). La máxima frecuencia que podremos reproducir en un determinado archivo será igual a la mitad de la frecuencia de muestreo (esto se conoce como Teorema de Nyquist). Por tanto, cuanto mayor sea la frecuencia de muestreo, también podremos reproducir frecuencias más elevadas, que de otro modo desaparecerían. Por otra parte, si aumentamos la frecuencia de muestreo también aumenta el número de muestras que debemos almacenar por cada segundo de audio, lo que se traduce en un mayor tamaño del archivo de audio. Se trata, por tanto y como siempre, de alcanzar una solución de compromiso entre el ancho de banda del audio que vamos a poder reproducir y el tamaño del archivo. Los CDs de audio, por ejemplo, tienen una frecuencia de muestreo de 44.1kHz, es decir, 44100 muestras por segundo, lo que implica que pueden reproducir sonidos con frecuencias de hasta 22050Hz. Últimamente es fácil ver formatos de audio que "presumen" de utilizar frecuencias de muestreo mucho más altas (96kHz e incluso 192kHz). Si tenemos en cuenta que el oído humano no es capaz de percibir sonidos más allá de los 20kHz, no parece que utilizar estas frecuencias de muestreo tenga ningún sentido. Sí pueden tener una justificación, al menos hasta cierto punto, si pensamos que su uso hace que los filtros de los equipos de audio sean más fáciles de diseñar, lo que puede redundar en una mejor calidad.

2. **Número de bits por muestra**: Cada una de las muestras de audio se cuantifica utilizando un determinado número de bits. Cuanto mayor sea este número, más *fina* será la cuantificación, pareciéndose más la señal digital a la original analógica, y obteniéndose por tanto un menor ruido de cuantificación. Igual que antes, y como es lógico, utilizar un número de bits por muestra elevado implica que el tamaño del archivo aumentará, ya que serán necesarios más bits para almacenar cada una de las muestras de audio. Utilizar más de 16 bits por muestra no aporta grandes ventajas en cuanto a la calidad del audio, ya que con 16bits se consigue un ruido de cuantificación que va a quedar con casi total seguridad por debajo del umbral de audición de la mayor parte de las personas. No obstante, en aplicaciones profesionales, y sobre todo en el mundo de la mezcla y producción de música, sí que tiene sentido utilizar 20 ó 24 bits por muestra, para garantizar que al combinar las señales de cada uno de los instrumentos el sistema no sature.

3. **Número de canales**: Finalmente también hay que tener en cuenta el número de canales que tiene el archivo de audio, y que pueden ser 1 (ficheros mono), 2 (estéreo), o incluso más como en el caso de ficheros surround.

Con todo lo anterior, y a partir de esos tres parámetros, es muy sencillo calcular cuál lo que se conoce como régimen binario, que no es más que el número de bits que ocupa un archivo de audio por cada segundo de duración:

$$R_b[bits/s] = f_s[Hz] \cdot N_b[bits] \cdot C \tag{1}$$

donde fs es la frecuencia de muestreo medida en Hz, Nb el número de bits por muestra y C el número de canales. De forma inmediata, si sabemos lo que dura el audio, podremos calcular el tamaño del archivo simplemente multiplicando el régimen binario por la duración en segundos (T):

$$bits = R_b[bits/s] \cdot T[s] \tag{2}$$

1.2 Codificadores de audio

Cuando hablamos de un codificador de audio nos referimos a cualquier sistema capaz de reducir el tamaño del archivo de audio, de forma que éste se pueda luego recuperar, aunque no necesariamente con la misma calidad. Existen por tanto dos grandes tipos de codificadores de audio:

- **Codificadores sin pérdidas**: Son aquellos que permiten recuperar el archivo original de forma exacta, sin ninguna pérdida de calidad. Es el equivalente de un archivo .zip o .rar pero para ficheros de audio. Quizás el formato de codificación de audio sin pérdidas más conocido es el .flac (https://xiph.org/flac/).
- **Codificadores con pérdidas**: Este tipo de codificadores hacen uso de herramientas psicoacústicas para reducir aún más el tamaño del archivo codificado a costa de reducir su calidad. No podremos por tanto recuperar nunca el archivo original a partir del codificado, sino tan sólo una versión que perceptualmente será prácticamente idéntica, si todo se ha hecho bien. El ejemplo más conocido de este tipo de codificadores es el formato .mp3, así como los archivos .aac, .mpa ó .mp4.

En esta parte de la práctica nos vamos a centrar en los codificadores con pérdidas (en concreto utilizaremos el mp3), y también veremos de forma muy breve un ejemplo de codificación sin pérdidas con archivos .flac.

Cuando queremos convertir un archivo de audio a alguno de estos formatos, normalmente podremos seleccionar algunas de las siguientes opciones en el programa:

- **Bitrate**: Define el número de bits por segundo que se van a utilizar para codificar la señal de audio. Como vimos en la práctica 1, el fichero original, en formato wav, está codificado utilizando una frecuencia de muestreo de 48000Hz y 16 bits por muestra, y son dos canales, por lo que su tasa binaria es: $2 \cdot 48000 \cdot 16 \approx 1536 kbps$. En el menú podemos seleccionar tasas binarias que van desde 320kbps (unas 5 veces inferior al original) hasta 8 kbps.
- **Modo**: variable, intermedio o constante. El modo constante obliga a que el bitrate sea siempre del valor exacto fijado, independientemente de si el archivo tiene partes que son más "sencillas" de codificar que otras. El intermedio, permite que varíe cuando sea necesario, pero intentando que en media se mantenga en el valor fijado, y el variable permite variaciones mucho más amplias.
- **Modo de canales**: estéreo o estéreo unido. El modo estéreo codifica cada canal por separado (izquierdo y derecho), mientras que el estéreo unido utiliza los posibles parecidos entre ambos canales para reducir aún más la información a codificar. A cambio, es proclive a introducir más distorsión en la señal.

Jugando con estos parámetros podemos configurar el codificador para reducir más o menos el tamaño del archivo de audio a costa, claro está, de reducir también más o menos la calidad del mismo.

2 Explorando las propiedades de los ficheros de audio

1. Abre el fichero *else.wav*, que puedes encontrar en Aula Virtual, en Audacity.
2. Si te fijas, debajo de su nombre aparece la frecuencia de muestreo del fichero (48000Hz).
3. El archivo utiliza 16 bits por muestra (no os fiéis de los 32 bits que aparecen en la ventana de la pista de audio. Esos son los que utiliza el programa de forma interna). A partir de todo esto calcula el tamaño teórico del archivo en MBytes. Comprueba si coincide con el tamaño real del archivo según el explorador de archivos del ordenador.
4. Prueba ahora a bajar la frecuencia de muestreo hasta 8000 Hz. Para ello selecciona el archivo completo y vete a Pistas - Remuestrear.
5. Vuelve a dejar la frecuencia de muestreo en 48000 Hz, y ahora haz click sobre el nombre de la pista (a la izquierda de la representación de su forma de onda). En el menú que aparece, cambia el muestreo a 22050Hz. Ahora prueba a subirlo a 96000.

6. Para reducir el número de bits por muestra vamos a ir al menú Herramientas - Indicador Nyquist. Nyquist es un lenguaje de programación para composición y síntesis de sonido. Si os interesa, podéis encontrar más información sobre él en http://www.audacityteam.org/help/nyquist/. Nosotros sólo vamos a utilizar el comando *quantize*, que sirve precisamente para modificar el número de bits por muestra de un fichero de audio. La sintaxis es:
`(quantize *track* N)`
Donde `N` es el número de niveles con los que cuantificamos. Es decir, si `N`=7, quiere decir que vamos a tener 7 niveles para los valores negativos, 7 para los positivos y además el cero, lo que se corresponde con un cuantificador de 4 bits ($2^4 = 16$).
Probad a cuantificar la señal con 8, 6, 4 y 2 bits y observad como el ruido de cuantificación se hace cada vez más audible.

2.1 Cuestiones

1. ¿Cuál es el tamaño teórico del archivo de audio calculado en el paso 3 utilizando la fórmula dada en la Introducción?
2. ¿Qué sucede al cambiar el valor de la frecuencia de muestreo en el paso 4?
3. ¿Aprecias alguna diferencia al hacerlo tal y como se explica en el paso 5? Si es así, ¿a qué crees que se debe?
4. ¿Qué comandos has utilizado para cuantificar la señal en el paso 6?

3 Codificadores de audio

1. Trabajaremos todo el tiempo con el archivo else.wav, igual que antes.
2. Vamos a jugar con los distintos parámetros de configuración de un codificador mp3. Para ello, en *Archivo*, vete a *Exportar*. En donde pone *Formato*, selecciona mp3 y ahora vamos a la zona de *Opciones de formato*. Ahí podremos modificar todos los parámetros de los que hemos hablado antes.
3. Prueba a grabar el fichero con un bitrate de 64kbps (lo suficientemente bajo para que se note bien la distorsión), y en modo constante.
4. Ahora graba el fichero con el mismo bitrate y en los modos medio y variable (en este último caso selecciona la velocidad que va en el rango de 45 a 85kbps).
5. Otra opción es utilizar un codificador sin pérdidas, como el FLAC. Pruébalo y observa tanto la calidad como el tamaño del archivo codificado. Mantén la profundidad de bit a 16bits y el nivel como aparezca por defecto.
6. Por último coge el fichero que habías grabado a un bitrate constante de 64kbps. Ábrelo en el Audacity y expórtalo pero esta vez en formato WAV. Observa el tamaño del fichero generado y escúchalo para ver su calidad.

3.1 Cuestiones

1. ¿Qué tamaño tienen los archivos que has grabado en los pasos 3, 4 y 5 según el explorador del ordenador? ¿Cuál es el bitrate real en cada caso?
2. ¿Y para el caso del paso 6? ¿Qué conclusión sacas?

4 Análisis de la frecuencia

1. Abrid el fichero ParaElisa.wav
2. Visualiza el espectrograma, selecciona la primera nota y utiliza el analizador de frecuencia para poder ver de qué nota se trata (Spoiler, es un Mi)
3. Intenta hacer lo mismo con las primeras ocho notas. ¿Puedes identificarlas?

5 Cambio de ritmo y tono

Utilizaremos el mismo archivo que en el caso anterior.

1. Probad a acortar la duración del fichero de voz a la mitad. Alargadla ahora hasta que dure 3 veces más. Probad ahora a aumentar su duración hasta 10 veces su duración original. En este ejemplo hemos llevado muy al límite a esta herramienta, pero en la realidad es muy útil por ejemplo en aplicaciones de publicidad, donde muchas veces se necesita acortar o alargar la duración de una canción a la del vídeo en un anuncio.
2. Partid de nuevo del archivo original. Sobre la señal sin modificar, id al menú Efecto - Cambiar tono. El fichero originalmente debería aparecer como en Mi. Podéis probar a cambiarlo a otro tono (Sol, por ejemplo)

¿Habéis observado algún problema al modificar la duración del fichero? ¿Y al cambiar el tono?

6 Reducción del ruido de fondo

Es posible eliminar, o al menos disimular, ruidos de fondo debidos a equipos eléctricos, aires acondicionados, viento, o similares utilizando la herramienta automática de Audacity.

Para probarlo, abrid el fichero RuidoDeFondo.wav. Es una grabación de un arpa con bastante ruido de fondo.

Lo primero que hay que hacer es enseñarle al programa cómo es el ruido que queremos eliminar. Para eso seleccionad algo así como el primer segundo o los dos primeros segundos del fichero (donde todavía no ha empezado la música). Con eso seleccionado id al menú Efecto - Reducción de ruido - Obtener perfil de ruido.

Ahora deseleccionad el trozo de antes y seleccionad el archivo completo. Id al menú Efecto -> Reducción de ruido -> Reducción de ruido. Seleccionad un valor para la reducción de ruido de 20dB, dejando los otros dos parámetros con sus valores por defecto. Antes de aplicar definitivamente el efecto, podéis probar qué tal funciona dándole al botón de Vista Previa que hay en la parte de abajo a la izquierda. Así se pueden ir ajustando los tres parámetros hasta que el resultado sea el deseado. Pulsad en Aceptar y guardad el archivo obtenido como SinRuidoDeFondo.mp3 (Archivo -> Exportar audio, mp3 128kbps) y adjuntadlo a la práctica.

7 Vocoder

Es muy sencillo simular un sonido de voz robotizada con el efecto de vocoder. Podéis probar a hacerlo con un trozo de voz grabado por vosotros o utilizar el archivo vozanecoica44k.wav.

Además necesitaréis un archivo con música que será la que sirva de base para la voz. En principio es recomendable que sea un archivo de música en 8 bits, ya que da mejores resultados, pero vale cualquiera.

Cargad los dos archivos en Audacity asegurándoos de lo siguiente:

- Que los dos tengan la misma duración y las mismas características (misma frecuencia de muestreo)
- Que ambos ficheros sean mono (si son estéreo, quitadle uno de los dos canales)
- Que el fichero de voz esté por encima del de música.

Cread un sólo archivo estéreo a partir de esos dos archivos y ahora aplicad el efecto Vocoder. Exportad el resultado final como vocoder.mp3

8 Auto-duck

Aunque tiene un nombre muy llamativo, este es uno de los efectos más utilizados en el mundo de la radiodifusión o el podcast. Lo que hace es atenuar, de forma automática, una pista de sonido en función de lo que ocurre en otra.

Cargad los ficheros Trooper.wav y vozanecoica_largo.wav. En este caso el fichero de música debe quedar por encima del de voz, ya que es éste el que dirige el funcionamiento del efecto.

Si reproducís esto, veréis que la voz apenas es audible por culpa de la música. Para solucionarlo, seleccionad toda la música y utilizad el efecto Auto Duck del menú de efectos. Podéis ajustar la cantidad de atenuación (Duck) que se le va a aplicar a la música cuando esté la voz a vuestro gusto.

Grabad el resultado como autoduck.mp3.

9 Materiales a entregar

1. Respuestas a las preguntas de los apartados 2, 3, 4 y 5..
2. Archivo SinRuidoDeFondo.mp3
3. Archivo Vocoder.mp3
4. Archivo AutoDuck.mp3