

TRABAJO PRÁCTICO SOBRE CODIFICACIÓN DE AUDIO

ESTÁNDARES DE COMUNICACIÓN DE AUDIO Y VÍDEO – 3er curso
GRADO DE INGENIERÍA EN SISTEMAS AUDIOVISUALES Y MULTIMEDIA
Curso 2019-2020

INTRODUCCIÓN Y OBJETIVOS

El objetivo de este trabajo práctico es aprender el funcionamiento de **FFmpeg** a la hora de codificar audio con diferentes códecs, y el efecto que tiene modificar parámetros como el **bitrate** o la **frecuencia de muestreo** tanto a nivel de **calidad** como a nivel de **tamaño de archivo**.

Para ello, vamos a hacer uso del siguiente material:

FFMPEG

Se trata de una colección de software libre que puede grabar, convertir (transcodificar) y hacer streaming de audio y vídeo. En Aula Virtual se os proporcionan diferentes binarios ya compilados tanto para Windows como para MacOS y Linux, y que son perfectamente utilizables desde una terminal.

Puede consultar su documentación a través de la siguiente URL: <https://www.ffmpeg.org/ffmpeg-codecs.html>

ALGORITMO PESQ EN MATLAB

Para sacar algún tipo de parámetro objetivo sobre la calidad de las diferentes codificaciones respecto a un audio de referencia se va a hacer uso de una serie de funciones en MATLAB que permiten calcular el Mean Opinion Score a partir del PESQ (Perceptual Evaluation of Speech Quality). Tanto el código a utilizar como los binarios necesarios están disponibles en Aula Virtual.

AUDIO DE EJEMPLO (SPEECH)

Para el inicio de la práctica se va a utilizar un pequeño *speech* de ejemplo, tanto limpio (*speech.wav*) como con ruido añadido (*speech_LQ.wav*).

GRABACIONES DE CADA GRUPO

Para la segunda parte de la práctica será necesario que un miembro de cada grupo grabe un determinado audio leyendo una línea correspondiente a alguna de las diapositivas del temario del bloque de audio (que no dure más de 10 segundos). Este audio se puede grabar tanto con ordenador como con un smartphone.

EJERCICIO 1

Para esta primera parte de la práctica se van a utilizar los dos archivos de *speech* proporcionados. Abra con un reproductor de audio cualquiera el archivo ***speech.wav***, y califique su calidad según el criterio del **Mean**

Opinion Score (MOS). Para ello, cada uno de los miembros del grupo deberá escuchar dicha grabación, dará su puntuación de calidad (u **ACR, Absolute Category Rating**), y el MOS resultante se calculará como el promedio de todas las puntuaciones.

A continuación, se repetirá el experimento con el archivo *speech_LQ.wav*. Dado que vamos a hacer una evaluación comparativa respecto al audio anterior, en este caso vamos a calcular el parámetro **DMOS (Degradation MOS)**. Para ello, cada miembro del grupo dará una puntuación del tipo **DCR (Degradation Category Rating)**, comparando la calidad del *speech_LQ* respecto del *speech* original. Finalmente, el DMOS resultante se calculará como el promedio de todas las puntuaciones.

Discuta brevemente los resultados obtenidos. En el caso de que haya habido muchas discrepancias entre las puntuaciones de cada miembro del grupo, justifique por qué ocurre esto y qué implicaciones tiene a la hora de diseñar un estudio sobre calidad de audio utilizando esta metodología.

EJERCICIO 2

A continuación, en vez de calcular el DMOS de forma cualitativa y subjetiva, se va a hacer uso de un **parámetro objetivo** con el que estimar dicho DMOS. Este parámetro es el **PESQ (Perceptual Evaluation of Speech Quality)**. Describe brevemente en qué se basa este parámetro para calcular la calidad de un determinado segmento de audio.

En esta práctica se va a hacer uso de una implementación en **MATLAB** de dicho parámetro de calidad, en el que es necesario introducir a la entrada un audio de referencia y el audio sobre el que queremos hacer la comparación. Como resultado, el algoritmo devuelve una estimación del DMOS.

Abra en MATLAB el script *test_pesq.m*, y obtenga la estimación del DMOS entre el *speech.wav* y el *speech_LQ.wav*. ¿Es muy diferente el resultado respecto a lo obtenido en el Ejercicio 1? ¿Por qué?

Finalmente, modifique el script para comparar el archivo *speech.wav* consigo mismo, y calcule de nuevo el DMOS. ¿A qué se debe el resultado obtenido?

EJERCICIO 3

En este ejercicio vamos a trabajar directamente con un archivo de audio grabado por cada grupo (utilizando la metodología que tenga cada uno a su alcance, sea un smartphone, un micrófono de PC, etc.). Denomine a este archivo de sonido grabado *recorded_audio*. Respecto al formato del audio grabado, aunque FFmpeg soporta gran variedad de formatos, es preferible utilizar alguno de los más conocidos, como MP4 (generalmente utiliza AAC), OGG o MP3.

En primer lugar, vamos a **comprobar algunas de las características técnicas del archivo grabado**. Para ello, ejecute el comando siguiente (recuerde añadir la extensión del archivo antes de ejecutar el comando):

```
ffprobe -i recorded_audio
```

Trate de identificar alguno de los parámetros más importantes, como la **duración del stream**, **bitrate**, **sampling rate**, **número de canales** y **códec utilizado**.

Finalmente, de forma análoga respecto al Ejercicio 1, **trate de estimar el MOS de esta grabación en base a los ACR de cada miembro del grupo** y discuta el resultado en base a la metodología de grabación y los parámetros obtenidos con el comando *ffprobe*.

EJERCICIO 4

A continuación, vamos a realizar un pequeño estudio sobre la **influencia del bitrate y el sampling rate** en el **tamaño y calidad del stream de audio**. Para ello, vamos a convertir el audio grabado por cada grupo en un archivo MP3 con diferentes bitrates y sampling rates.

Suponiendo que el archivo grabado por cada grupo es *recorded_audio.mp4* (modificar según sea vuestro archivo), el comando a ejecutar sería el siguiente:

```
ffmpeg -i recorded_audio.mp4 -codec:a libmp3lame -b:a 320k -ar 16000 output.mp3
```

donde *output.mp3* es el archivo de salida. Los argumentos introducidos a **ffmpeg** son los siguientes:

- **codec:a**. Este argumento permite elegir el codificador que vamos a utilizar para obtener el archivo de audio de salida. En este caso, libmp3lame (utilizado para MP3).
- **b:a**. Con este argumento seleccionamos el bitrate que queremos que tenga el archivo de salida. En este caso, 320k.
- **ar**. Este argumento nos permite seleccionar el sampling rate del archivo de salida. En este caso, 16000Hz.

Si nos fijamos, en los argumentos anteriores está presenta la letra 'a' de diferentes formas. Esto nos permite indicar (para que no haya ambigüedad) que vamos a trabajar con un stream de audio, ya que FFmpeg permite trabajar tanto con vídeo como con audio.

En primer lugar, debido a las limitaciones del algoritmo PESQ proporcionado, vamos a **convertir el archivo grabado a un archivo MP3 cuyo sampling rate sea de 16000Hz, con un bitrate de 320k**. Para ello, utilice el comando de ejemplo dado al principio de este ejercicio. Llame a este archivo *recorded_audio.mp3*.

Una vez hecho esto, se va a realizar un estudio sobre la **influencia del bitrate, dejando fijo el sampling rate en un valor concreto**. Modifique el comando dado como ejemplo para utilizar un **sampling rate de 16000Hz, variando el bitrate desde 320k hasta 64k. Estudie tanto el tamaño del archivo generado como su calidad**. Para estudiar la calidad, utilice la misma función de MATLAB que en el Ejercicio 2, utilizando como audio de referencia el archivo *recorded_audio.mp3*. Discuta los resultados obtenidos.

Finalmente, **repita el mismo ejercicio solo que, en este caso, se va a dejar fijo el bitrate en 128k**, mientras que **el sampling rate lo va a ir variando** según los siguientes valores: 16000, 11025 y 8000. De nuevo, discuta los resultados obtenidos.

En ambos casos, **compruebe con el comando ffprobe -i** (ver Ejercicio 3) que las **características de bitrate y sampling rate del archivo de salida se corresponden con los introducidos a ffmpeg**. ¿Existe algún caso en el que esto no sea así? ¿Por qué cree que ocurre esto?

Recomendación: Para facilitar la compresión de los resultados, tanto a nivel de tamaño de archivo como de calidad, se recomienda crear 4 gráficas (con MATLAB, Excel, etc.) que relacionen los siguientes parámetros:

- Bitrate vs Tamaño de archivo.
- Bitrate vs Calidad de audio (PESQ-DMOS).
- Sampling rate vs Tamaño de archivo.
- Sampling rate vs Calidad de audio (PESQ-DMOS).

EJERCICIO 5

Después de haber realizado el estudio sobre la influencia del bitrate y sampling rate en el tamaño del archivo y en la calidad resultante, finalmente, se va a realizar un **estudio sobre el uso de diferentes códecs a igualdad de sampling rate y bitrate**. Para ello, visite la página <https://www.ffmpeg.org/ffmpeg-codecs.html#Audio-Encoders> y escoja 3 encoders diferentes.

Compare los tamaños de archivo y calidad obtenidos respecto a MP3 y respecto al archivo original. ¿Qué conclusión puede sacar de este estudio? ¿Son todos los códecs igual de eficientes?

ENTREGA DE RESULTADOS

Se deberá realizar una **memoria donde se discutan cada uno de los ejercicios dispuestos** (formato libre, con el nombre de cada uno de los miembros del grupo), además de cada uno de los **archivos de audio generados** a lo largo de la práctica. Este trabajo práctico se deberá realizar en los **mismos grupos con los que se han ido llevando a cabo las diferentes prácticas de la asignatura**.

El formato de entrega será un **archivo .zip** que contenga la **memoria en PDF** y los **diferentes archivos de audio generados**. El **nombre del archivo .zip** entregado deberá seguir el siguiente **patrón**:

ECAV_Audio_GXX_1920.zip

donde **XX** es el **número del grupo**. Se habilitará en Aula Virtual una **entrega por grupos**, por lo que bastará con que un miembro de cada grupo suba la entrega a Aula Virtual.

La **fecha límite** para realizar la entrega de este trabajo práctico será el día **1 de mayo de 2020, a las 23.59**.