PDIH

.

Práctica 5. Experimentación con el sistema de salida de sonido

Autor: Miguel Molinero Martin

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Curso 2023 - 2024

DESARROLLO DE LA PRÁCTICA 5: Experimentación con el sistema de salida de sonido

Los objetivos concretos de esta práctica son: • Identificar y representar gráficamente la forma de onda de señales de sonido. • Conocer la estructura de un fichero típico de sonido (ficheros WAV). • Entender y operar con los parámetros principales de una señal de sonido. Para realizar esta práctica se puede programar en lenguaje R para manejar el sonido, utilizando el entorno de programación RStudio, o bien otro lenguaje, como Python, C++ o Java.

Lo primero que debemos hacer es crear los archivos .wav que vamos a usar con el siguiente comando:

```
espeak "Miguel" -w nombre.wav
espeak "Molinero Martin" -w apellido.wav
```

A continuación el script en R:

```r
library(tuneR)
library(seewave)

# Lo primero, leemos los archivos wav que hemos creado antes
nombre <- readWave("nombre.wav")
apellidos <- readWave("apellidos.wav")

# Dibujamos las ondas
par(mfrow=c(2,1))
plot(nombre, main="Forma de onda del Nombre")
plot(apellidos, main="Forma de onda del Apellidos")

# Información de cabeceras
str(nombre)
str(apellidos)

# Unión de los archivos en uno
nombreyapellidos <- pastew(nombre, apellidos, output = "Wave")

# Dibujamos la forma de onda de los archivos combinados
plot(nombreyapellidos, main="Forma de onda de los archivos combinados")

# Filtro de frecuencia para eliminar frecuencias entre 10000 y 20000 Hz
frecuencias_muestreo <- nombreyapellidos@samp.rate
from_normalizado <- 10000 / (frecuencias_muestreo / 2)
to_normalizado <- 20000 / (frecuencias_muestreo / 2)
sonidofiltrado <- bwfilter(nombreyapellidos, from = from_normalizado, to = to_normalizado, bandpass = FALSE, f = frecuencias_muestreo, ou

# Redondear los datos de los canales a enteros
sonidofiltrado@left <- round(sonidofiltrado@left)
if (!is.null(sonidofiltrado@right)) {
  sonidofiltrado@right <- round(sonidofiltrado@right)
}

# Almacenar el resultado como mezcla.wav
writeWave(sonidofiltrado, "mezcla.wav")

# Crear un nuevo sonido con eco y darle la vuelta al sonido
refran <- readWave("refran.wav")
refran_con_eco <- echo(refran, f = refran@samp.rate, amp = c(0.8, 0.4, 0.2), delay = c(0.2, 0.4, 0.6), output = "Wave")

# Darle la vuelta al sonido
refran_al_reves_matrix <- revw(refran_con_eco)
refran_al_reves <- Wave(refran_al_reves_matrix, samp.rate = refran@samp.rate, bit = refran@bit)

# Redondear los datos de los canales a enteros
refran_al_reves@left <- round(refran_al_reves@left)
if (!is.null(refran_al_reves@right)) {
```

```
# Almacenar el resultado
writeWave(refran_al_reves, "alreves.wav")
```

comienza cargando las librerías necesarias para el procesamiento de audio: tuneR y seewave, previamente instaladas desde la terminal.

Luego, lee dos archivos de audio previamente creados, uno correspondiente al nombre y otro a los apellidos. A continuación, representamos las formas de onda de ambos archivos en una figura dividida en dos paneles. Posteriormente, se examinan las cabeceras de los archivos para obtener información sobre su estructura y propiedades. Después, los dos archivos de audio se combinan en uno solo utilizando la función pastew, y se grafica la forma de onda del archivo resultante. Se establece un rango de frecuencias entre 10000 y 20000 Hz que se desea filtrar del archivo combinado, y se aplica un filtro de frecuencia utilizando la función bwfilter.

Los datos de los canales se redondean a enteros para facilitar su manipulación. El archivo filtrado se guarda como "mezcla.wav". Luego, se carga un nuevo archivo de audio que contiene un refrán y se le aplica un efecto de eco utilizando la función echo. Posteriormente, se invierte el sonido resultante utilizando revw y se redondean los datos de los canales a enteros. Finalmente, el sonido invertido se guarda como "alreves.wav".

Los resultados:

Environment | History | Connections | Tutorial

Import Dataset | 393 MiB | List

R | Global Environment

Data	
apellidos	Formal class Wave
nombre	Formal class Wave
nombreyapellidos	Formal class Wave
refran	Formal class Wave
refran_al_reves	Large Wave (94771 elements, 759.7 kB)
refran_al_reves_matrix	Large matrix (94771 elements, 758.4 kB)
refran_con_eco	Large Wave (94771 elements, 759.7 kB)
sonidofiltrado	Formal class Wave
Values	
frecuencias_muestreo	22050L
from_normalizado	0.90702947845805
to_normalizado	1.8140589569161

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish