

Universidad de San Carlos de Guatemala.

FACULTAD DE INGENIERÍA.

ESCUELA DE MECÁNICA ELÉCTRICA.

LABORATORIO, COMUNICACIONES 4

ING. JOSE ANIBAL SILVA

SECCIÓN C

COMUNICACIONES 4: PRACTICA 5

27 DE MARZO DE 2023

INTEGRANTES:

LUIS ARTURO QUIÑONEZ BOLAÑOS 201801101

Este código en octave realiza la compresión de un archivo de audio usando la transformada discreta de coseno (DCT). A continuación se describe detalladamente cada una de las líneas del código:

I. CODIGO 1

Figura 1: Codigo 1

```
pkg load signal
# Leer archivo de audio
[y, fs] = audioread('audio.wav');
# Realizar DCT
dct_y=dct(y);
# Establecer el umbral para la compresión
umbral = 0.1;
```

- Esta línea carga el paquete de señales de MATLAB para tener acceso a funciones útiles para procesar señales.
- Esta línea lee un archivo de audio llamado 'audio.wav' y lo almacena en la variable y, junto con su frecuencia de sondeo en la variable fs.
- Esta línea realiza la transformada discreta de coseno (DCT) del archivo de audio yy lo almacena en la variable dcty.

II. CODIGO 2

Figura 2: codigo 2

```
# Comprimir DCT
dct_y_comprimido = dct_y.* (abs(dct_y) > umbral);
# Realizar la inversa de la DCT para obtener el archivo de audio comprimido
y_comprimido = idct(dct_y_comprimido);
# Graficar el archivo inicial y final
t = (0:length(y)-1)/fs;
t_comp = (0:length(y_comprimido)-1)/fs;
```

- Esta línea establece un umbral para la compresión del archivo de audio.
- Esta línea comprende el archivo de audio realizando una multiplicación punto a punto entre la DCT del archivo original y una máscara binaria que selecciona solo los coeficientes que superan el umbral. El resultado de la compresión se almacena en la variable dctycomprimido.
- Esta línea realiza la inversa de la DCT para obtener el archivo de audio comprimido en la variable ycomprimido.

III. CODIGO 3

Figura 3: codigo 3

```
subplot(2,1,1);
plot(t, y);
title('Archivo de audio inicial');
xlabel('Tiempo (s)');
ylabel('Amplitud');
subplot(2,1,2);
plot(t_comp, y_comprimido);
title('Archivo de audio comprimido');
xlabel('Tiempo (s)');
ylabel('Amplitud');
```

- Estas definen las líneas de tiempo para el archivo original y el archivo comprimido, respectivamente.
- Estas líneas grafican el archivo de audio original y el archivo comprimido en dos subgráficos. La primera subgrafica muestra el archivo original y la segunda subgrafica muestra el archivo comprimido. Ambas subgráficas tienen el tiempo en el eje xy la amplitud en el eje y.

IV. SCREENSHOT

Figura 4: codigo 4

```
while opcion ~= 8
    % Menú de opciones
    disp('Seleccione una opción:');
    disp('1. Grabar');
    disp('2. Reproducir');
    disp('3. Graficar');
    disp('4. Graficar densidad');
    disp('5. Procesamiento de señales ');
    disp('6. Procesamiento digital de señales');
    disp('7. Compresión de sonidos');
    disp('8. Salir');
    opcion = input('Ingrese su elección: ');
```

- Grabar: permite grabar una señal de audio y guardarla en un archivo de audio en formato WAV.
- Reproducir: permite reproducir la señal de audio grabada. Graficar: muestra un gráfico de la señal de audio grabada en el tiempo. Graficar densidad: muestra el espectro de frecuencia de la señal de audio grabada.
- Procesamiento de señales: aplica filtros digitales a la señal de audio grabada para reducir el ruido eléctrico (filtro RFI) y filtrar las frecuencias no deseadas (filtro RII).
- Para la opción 7, se diseñan los filtros digitales utilizando funciones de octave para calcular sus coeficientes y luego se aplican a la grabación de audio.

Se muestran gráficos de la señal de audio original y la señal filtrada con cada filtro.

Funcionamiento del código de compresión de audio.

V. SCREENSHOT

Figura 5: código 4

