
PROYECTO 1: COMPRESIÓN DE SEÑALES DE AUDIO

Carnet 202200041 – Daniel Eduardo Velásquez Avila

Resumen

El centro de Investigaciones de la Facultad de Ingeniería está experimentando un nuevo método de compresión de señales de audio, este método se basa en los parámetros de frecuencia y amplitud de las ondas de sonido.

La frecuencia es la cantidad de ciclos que se realizaran en un segundo y esta se mide en Hert (Hz), mientras que la amplitud representa la altura de la onda y se puede interpretar como la intensidad del sonido, este parámetro se mide en decibelios (dB).

La propuesta que se tiene por parte del Centro de Investigaciones consiste en agrupar las señales de audio en base a los patrones de frecuencia, para eso se necesitara crear una matriz de patrones de frecuencia en la cual se indicara los tiempos y amplitudes que contiene la señal de audio.

En este método lo que se busca es reducir de manera significativa el tamaño de las señales de audio sin tener que afectar la calidad de la misma.

Palabras Clave

- Amplitud
- Frecuencia
- Matrices
- Patrones
- Señales de Audio

Abstract

The Research Center of the Faculty of Engineering is experimenting a new method of audio signal compression, this method is based on the parameters of frequency and amplitude of sound waves.

The frequency is the number of cycles that will be performed in one second and this is measured in Hert (Hz), while the amplitude represents the height of the wave and can be interpreted as the intensity of the sound, this parameter is measured in decibels (dB).

The proposal of the Research Center consists of grouping the audio signals based on frequency patterns, for that it will be necessary to create a matrix of frequency patterns in which the times and amplitudes contained in the audio signal will be indicated.

This method seeks to significantly reduce the size of the audio signals without affecting their quality.

KeyWords

- Amplitude
- Frecuency
- Arrays
- Patterns
- Audio Signals

Introducción

El Proyecto consta de realizar un software que reduzca señales de audio en base a los datos y parámetros que se nos proporcionan mediante un archivo XML el cual se leerá, extraerá y almacenaran los datos para luego ser manipulados.

Para lo anterior descrito es necesario el uso de listas enlazadas simples, estas se manejarán a través de nodos, para mostrar los datos en pantalla se hará uso de la librería graphviz con la cual se pueden realizar gráficos en los cuales se mostrarán los datos de la matriz original, así como de la matriz luego de hacer la comparación de los patrones.

Para la comparación se necesitará realizar una función la cual compare los patrones y al momento de encontrar coincidencia sume los datos.

Desarrollo del tema

Al iniciar el desarrollo del programa se creó una clase nodo la cual contiene lo necesario para poder crear y manipular los datos de las listas simples, los atributos que contiene la clase nodo son, id, dato, siguiente.

A partir de estas variables se puede hacer uso de las listas simples, esta clase contiene la función de 'agregar', esta función como su nombre lo indica nos sirve para poder agregar un nuevo elemento a la lista simple, entre las demás funciones que contiene la clase lista simple se encuentra una función para hacer una búsqueda de datos, esta función realiza una comparación entre los nodos que se tienen guardados y el dato que se está mandando para buscar, si encuentra una coincidencia se devuelve el dato y de lo contrario se regresa el valor 'None', a parte de estas funciones se encuentran

diferentes set y gets para la manipulación de los datos.

Al iniciar el programa se le mostrara al usuario un menú en el cual se tienen siete opciones, se realizara una serie de if para saber cuál fue la opción seleccionada esto a partir de un input que contenga el texto 'Seleccione una opción:' este input será guardado en una variable llamada opción, luego de escribir la opción correspondiente, se realizara una serie de if para verificar cual de todas las opciones fue la que se ingresó, si el dato ingresado no corresponde con ninguna de las opciones se mandara un mensaje al usuario en el cual se le indicara que la opción escrita es invalida.

Al seleccionar la primera opción que corresponde a 'cargar archivos' se le preguntara al usuario la ruta y el nombre del archivo que se va a cargar, al momento de seleccionar esta opción se mandara a llamar una función la cual preguntara la ubicación y nombre del archivo, el dato proporcionado se almacenara en una variable para posteriormente ser usada, luego de ingresar el dato se le permitirá al usuario volver al menú principal para poder seleccionar una opción diferente. al seleccionar la segunda opción que corresponde a el procesamiento de los datos se llamada a una función llamada 'getSenales' la cual esta almacenada en una clase diferente, al momento de realizar el llamado de la función se le proporcionara como parámetro el dato de la ubicación y el nombre del archivo que fueron guardados anteriormente.

Dentro de esta función se crearan 3 listas simples las cuales son necesarias para el almacenamiento de los datos, estas listas simples serán 'ListaS', 'amplitudes' y 'tiempos', la única lista que se creara al principio será la lista de señales, las demás listas serán creadas el momento de leer los datos ya que cada una de las

señales contendrán tiempos, amplitudes y datos diferentes

valor '1' dentro de una variable y que si el valor es igual a 0 guardar un valor 0 en la variable, al

Para el guardado de estos datos se hará uso de las diferentes características que tiene la librería de XML, se hará uso de '.findall' para encontrar todas las señales que contiene el archivo XML esto junto con el uso de un ciclo 'for' el cual ejecutara una serie de código a partir de la cantidad de 'senales que encuentre', al momento de encontrar un 'senal' guardara el nombre, tiempo máximo y amplitud máxima, para guardar los datos en variables se hará uso de '.get', luego de guardar estos datos se hará uso de un ciclo 'for' a partir del tiempo máximo encontrado anteriormente, dentro de este ciclo se realizara otro ciclo para almacenar el valor del dato, dentro de este último ciclo se hará uso de un 'if' para verificar que el tiempo coincide, si es así se guardara dentro de una lista simple la amplitud junto con el dato correspondiente, al finalizar ese ciclo se guardara los tiempos junto con la lista de las amplitudes, y al finalizar el siguiente ciclo se guardaran los datos del señal anteriormente mencionados, tiempo Máximo, amplitud máxima, justo con la lista de tiempos, la cual contiene las amplitudes y los datos, teniendo así una lista dentro de otra lista.

Luego de realizar el guardado de todos los datos se mandara a llamar a la lista que contiene las señales para poder así hacer la conversión a binario y posteriormente la comparación de las mismas, para acceder a los datos de la lista se requerirá de varios ciclos for y el uso de una función dentro de la clase 'ListaSimple' la cual se llama 'buscarId()', luego de realizar los ciclos for y estar dentro del ciclo en el cual podremos acceder a los datos individuales se realizara una condición en la cual si el valor es diferente de cero se guarde un

finalizar el ciclo se tiene una variable la cual contiene una cadena de texto con únicamente ceros o uno, correspondiente a el valor, ese nuevo dato se manda a guardar a una lista simple la cual contendrá todas las cadenas binarias junto con el tiempo y al senal al que corresponde.

Luego de guardar la cadena de texto se realizara una función que compare todos los elementos de la lista entre sí, esto se hará por medio de dos ciclos for y un if, en el cual compara los patrones y si encuentra una coincidencia almacena en una lista simple los tiempos en los que coincide junto con el patrón correspondiente, estos datos se usaran posteriormente para poder realizar las sumas y obtener el nombre correspondiente del grupo.

Al finalizar la comparación se realizara la suma de los datos, para esto se hace uso nuevamente de varios ciclos for para poder acceder a todos los datos, en este caso lee la señal original y compara los tiempos con los tiempos donde se encontró coincidencias en los patrones, esto con el fin de poder encontrar los datos almacenados en ese tiempo, y valida que se encuentren ambos ya que de lo contrario no se podría realizar la suma y terminaría mostrándose un error. Al validar que se encuentren ambos tiempos se usara una función la cual creara el nuevo nombre del grupo y sumara los datos de sus tiempos, luego de esto se regresaran esos datos para poder almacenarlos en una nueva lista junto con los tiempos donde no hubo coincidencias, así teniendo la lista con los datos reducidos.

Luego de ejecutar todas estas funciones y mostrar en pantalla los mensajes respecto a que parte del programa se estaba realizando, se regresara al menú principal.

Si se sigue el orden que está en el menú seguiría la opción 3. La cual corresponde a escribir el archivo de salida, en este apartado se le pregunta al usuario en que ruta quisiera que fuera guardado el archivo de salida en el formato XML, este dato será almacenado en una variable para poder mandar como parámetro a la función que genere el archivo XML

La cuarta opción del menú principal se encargará de mostrar todos los datos correspondientes del estudiante. La última opción del menú es la encargada de preguntar al usuario cual de las graficas se desea mostrar en pantalla, para realizar estas graficas se hará uso de la librería 'graphviz' junto con todas las características que trae para su uso, para generar las gráficas se hará uso de ciclos para acceder a los datos y cada uno de los datos serán agregados como nodos según la forma indicada en el enunciado del proyecto, cada uno de estos nodos estarán relacionados por medio de aristas y finalmente serán generados y guardados en el formato adecuado. La última opción que se encuentra en el menú sería la opción de salida, en la cual como su nombre lo indica se finalizara la ejecución del programa.

Conclusiones

Al momento de utilizar una librería nueva, es buena idea que se lea la documentación que esta tiene, ya que así se nos hará mas sencillo el saber sobre su funcionamiento y como enfocarlo hacia nuestro programa.

En este proyecto la lectura de la documentación fue de gran ayuda al

momento de generar las graficas al igual que con la manipulación de los archivos XML.

¿Cuáles son las ventajas y desventajas de utilizar listas enlazadas en comparación con otras estructuras de datos como los arrays?.

Referencias bibliográficas

Luis Joyanes Aguilar, (2008), Fundamentos de programación, algoritmos, estructuras de datos y objetos, McGraw-Hill

Apéndices:

