

Producción y análisis de música

Autores: Alejandro Magnorsky, Andrés Mata Suárez, Mariano Merchante

Instituto Tecnológico de Buenos Aires

Resumen

Palabras clave

Archivo de sonido; Partitura; Transformada de Fourier; Frecuencia.

I. INTRODUCCIÓN

II. DESARROLLO

A. Construcción de una partitura a partir de un archivo de sonido

El objetivo es la creación de un programa que, dado un archivo de sonido (.wav), genere una partitura. El código se divide en dos archivos o funciones, que son las que se detallan en la figuras 1 y 2.

Para comenzar, la función de la figura 1 recibe la ruta del archivo de sonido que se desea procesar. Utilizando la función de Octave, `wavread`, se obtiene el vector que representa la onda del sonido, x , junto a la frecuencia de muestro de dicha señal, f_s . Considerando intervalos de tiempo de $30ms$, se divide a x en vectores de $\frac{30}{1000}f_s$ elementos, ya que esa es la cantidad de muestreos del sonido que se realizan a lo largo de $30ms$. Para cada uno de dichos vectores, se calcula la Transformada Discreta de Fourier usando la función que provee Octave, `fft`, la cual usa el algoritmo de la Transformada Rápida de Fourier. La ventaja de esto es que tiene $O(n \log n)$ en lugar de $O(n^2)$.

Cada vector X que devuelve la función `fft`, pasa a ser la entrada para otra función llamada `fftshift`. El efecto de `fftshift` es mover los valores de X para que queden centrados en torno a la frecuencia 0. Finalmente, considerando los valores en que la frecuencia es positiva, es decir, la segunda mitad del vector X , se averigua en qué valor de la frecuencia X se maximiza. Ese valor es la frecuencia fundamental del intervalo considerado, que se guarda en el vector de frecuencias que devuelve la función en cuestión.

Utilizando el vector de frecuencias generado a partir de la función de la figura 1, la función de la figura 2 genera un vector de caracteres que representan los tripletes de cada una de las frecuencias.

III. RESULTADOS

IV. CONCLUSIONES

REFERENCIAS

Mathews, John H., Fink, Kurtis D., “Numerical Methods Using MATLAB”, Prentice Hall, 1999

Listing 1: Implementación de la generación de la secuencia de frecuencias de un archivo de sonido.

```

function frequencies = getFrequencies(wavFile)

% x es la onda del sonido
% fs es la frecuencia de muestreo (cantidad de muestreos por segundo).
% Es decir, un segundo equivale a fs muestras de x
% bits es la cantidad de bits de cada muestra
[x, fs, bits] = wavread(wavFile);

interval = floor(fs*30/1000);
quant = floor(length(x)/interval);

for k=1:quant
    lower = (k-1)*interval+1;
    upper = k*interval;

    X = fft(x(lower:upper));
    X = fftshift(X);
    X = X(floor(length(X)/2)+1:length(X));
    f = 0:fs/interval:fs/2-1;

    [number, pos] = max(X);
    frequencies(k) = f(pos);
endfor
plot(frequencies)
endfunction

```

Listing 2: Implementación de la escritura de una partitura dada una secuencia de frecuencias.

```

function partiture = writePartiture(frequencies)
    for i=1:length(frequencies)
        partiture(i,:) = frequencyToNote(frequencies(i));
    endfor
endfunction

```