

DESCRIPTORES PARA ANÁLISIS AUTOMÁTICO DE MÚSICA

Federico Feldsberg¹

¹Universidad Nacional de Tres De Febrero, Buenos Aires, Argentina
fedefelds@hotmail.com

Resumen

Se hace un código que básicamente puede hacer muchas cosas. Se eligen 3 descriptores. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1. INTRODUCCIÓN

En este informe se describe el diseño y la implementación de un sistema capaz de analizar canciones y extraer información útil de las mismas. Para ello se desarrolló una serie de herramientas basadas en la librería *Librosa*. Dicha librería es de código abierto y está validada por...AGREGAR VALIDACION... Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend

at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Los objetivos de este trabajo son los siguientes:

- Implementar un sistema que pueda remover el silencio al principio y al final de una señal
- Implementar un sistema que pueda normalizar la amplitud de una señal
- Implementar un sistema que pueda visualizar la STFT de una señal
- Implementar un sistema que pueda estimar el tempo de una señal
- Implementar un sistema que pueda calcular 3 descriptores a elegir
- Procesar un disco de música con las herramientas desarrolladas
- Implementar un sistema que pueda normalizar los valores obtenidos

2. DESCRIPTORES ELEGIDOS

2.1. SILENCE RATE

Dicho descriptor es experimental. Consiuye una manera de medir que tan silenciosa es una canción. Sea una señal de N muestras. Se propone dividir la canción en intervalos silenciosos y no silenciosos. Se obtiene que de N muestras, M_s son silenciosas y $N - M_s$ no lo son.

El Silence Rate esta dado por la relacion entre la cantidad de muestras silenciosas y la cantidad total de muestras:

$$SR = \frac{N - M_s}{N} \quad (1)$$

Como $0 \leq N - M_s \leq N$, entonces $0 \leq SR \leq 1$. Una canción cuyo SR es 1 corresponde a una señal totalmente silenciosa. Similarmente, una canción cuyo SR es nulo corresponde a una señal totalmente no silenciosa.

En este trabajo, dicho analisis es implementado mediante el uso del metodo *librosa.effects.split*.

2.2. CHROMA ANALISYS

El Chroma analisys fue introducido por primera vez por Fujishima en [1]. Dicho analisis es una manera de representar las características espectrales de una señal sonora. En dicha representacion, el espectro de frecuencias es proyectado en 12 bins. Cada bin representa uno de los 12 distintos semitonos de una octava musical. En otras palabras, todas las octavas de una nota musical son mapeadas a uno de los 12 bins. Debido a esto es posible sintetizar, con cierta pérdida de informacion, una señal a partir de su Chroma Analisis, mediante Chroma Synthesis.

En [2] Ellis sostiene que el Chroma analisis puede dar informacion util acerca de la señal en cuestion que no es evidente en el espectro original de la señal. Por ejemplo, es capaz de señalar la similitud musical percibida en un tono de Shepard [3].

En este trabajo, dicho analisis es implementado mediante el uso de la funcion *librosa.feature.chroma_stft*.

2.3. SPECTRAL CONTRAST

Dada una señal musical, se presenta el problema de como identificar a que tipo de musica pertenece la señal en cuestion. Este descriptor resulta util a la hora de realizar esta tarea. Jiang et al. indica que es posible que este descriptor puede llegar a tener una mejor capacidad de discriminacion de tipos musicales que los MFCC [4].

Dicho descriptor considera las diferencias entre los picos espectrales y los valles espectrales para cada sub banda. En la mayoría de los casos en los que se analiza musica, los fuertes picos corresponden aproximadamente con las componentes armonicas mientras que las componentes no armonicas o ruidos corresponden con los valles. Por eso el descriptor en cuestion permite caracterizar la distribucion relativa de las componentes armonicas y no armonicas del espectro.

En este trabajo, dicho analisis es implementado mediante el uso de la funcion *librosa.feature.spectral_contrast*.

2.4. TEMPOGRAM

Este descriptor se basa en un tratamiento similar al presentado en la seccion 2.2. En el caso de dicha seccion, se recurre a mapear varias octavas o armonicos a una determinada cantidad de bins [5]. En el caso del tempograma se recurre a mapear varios tempos que difieren por un factor de 2 en un mismo bin. En terminos analogos, podemos pensar en los armonicos de una frecuencia dada, que no son mas que frecuencias relacionadas por un factor de 2.

En este trabajo, dicho analisis es implementado mediante el uso del metodo *librosa.feature.tempogram*.

3. RESULTADOS

En la siguiente sección se describe la implementacion de los objetivos propuestos en la sección 1. En las subsecciones 3.1, 3.2 y 3.3 se utiliza la canción *Brain Stew* de GreenDay como señal de prueba.

3.1. PREPARACIÓN DE LA SEÑAL TEMPORAL

La preparacion de la señal consistie en cargar el archivo, extraer el silencio al principio y al final y finalmente, normalizar la señal. Dichas tareas se implementan mediante un script en python disponible en

3.2. VISUALIZACIÓN DE LA STFT

Para la implementacion de la STFT, se considera el siguiente script, propuesto en [6]:

El resultado obtenido es la siguiente figura:

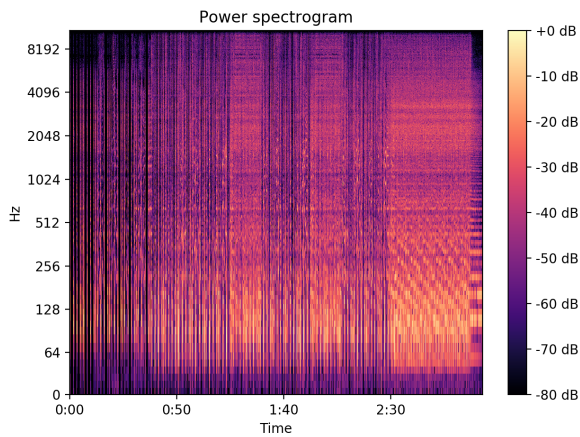


Figura 1: Visualización de la STFT

3.3. ESTIMACIÓN DEL TEMPO

Para la estimación del tempo se implementa el siguiente script:

4. ANÁLISIS DE RESULTADOS

5. CONCLUSIÓN

6. REFERENCIAS

- [1] Takuya Fujishima. *Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music*.
- [2] Dan Ellis. *Chroma Feature Analysis and Synthesis*. URL: <https://labrosa.ee.columbia.edu/matlab/chroma-ansyn/>.
- [3] Juan Pablo Bello. *Chroma and tonality*. URL: http://www.nyu.edu/classes/bello/MIR_files/tonality.pdf.
- [4] Dan-Ning Jiang y col. "Music type classification by spectral contrast feature". En: *Proceedings. IEEE International Conference on Multimedia and Expo. IEEE*. DOI: 10.1109/icme.2002.1035731. URL: <https://doi.org/10.1109/icme.2002.1035731>.
- [5] Frank Kurth, Thorsten Gehrmann y Meinard Müller. "The Cyclic Beat Spectrum: Tempo-Related Audio Features for Time-Scale Invariant Audio Identification." En: *ISMIR*. 2006, págs. 35-40.
- [6] "librosa.core.stft Documentation". En: (). URL: <http://librosa.github.io/librosa/generated/librosa.core.stft.html?highlight=stft#librosa.core.stft>.

A. RESULTADOS ADICIONALES

	Tema 1	Tema 2	Tema 3	Tema 4	Tema 5	Tema 6	Tema 7	Tema 8	Tema 9	Tema 10
Bin 1	0.504	0.530	0.457	0.570	0.417	0.324	0.551	0.459	0.406	0.351
Bin 2	0.507	0.449	0.436	0.470	0.514	0.463	0.505	0.489	0.392	0.489
Bin 3	0.559	0.484	0.562	0.423	0.683	0.609	0.507	0.576	0.464	0.645
Bin 4	0.523	0.428	0.712	0.463	0.486	0.472	0.558	0.483	0.470	0.494
Bin 5	0.618	0.431	0.496	0.543	0.434	0.451	0.650	0.501	0.487	0.514
Bin 6	0.704	0.458	0.365	0.428	0.489	0.402	0.525	0.514	0.512	0.490
Bin 7	0.521	0.602	0.381	0.430	0.478	0.343	0.475	0.512	0.495	0.384
Bin 8	0.458	0.579	0.447	0.556	0.531	0.371	0.502	0.555	0.462	0.418
Bin 9	0.494	0.469	0.399	0.564	0.550	0.471	0.602	0.512	0.396	0.532
Bin 10	0.570	0.476	0.428	0.627	0.626	0.638	0.765	0.579	0.550	0.723
Bin 11	0.466	0.583	0.543	0.478	0.455	0.457	0.580	0.518	0.704	0.509
Bin 12	0.427	0.726	0.464	0.500	0.376	0.308	0.511	0.446	0.570	0.328

Tabla 1: Valores del descriptor chroma obtenidos sin normalización

	Tema 1	Tema 2	Tema 3	Tema 4	Tema 5	Tema 6	Tema 7	Tema 8	Tema 9	Tema 10
Bin 1	0.659	0.693	0.597	0.745	0.545	0.424	0.720	0.600	0.531	0.459
Bin 2	0.663	0.587	0.570	0.614	0.672	0.605	0.660	0.639	0.512	0.639
Bin 3	0.731	0.633	0.735	0.553	0.893	0.796	0.663	0.753	0.607	0.843
Bin 4	0.684	0.559	0.931	0.605	0.635	0.617	0.729	0.631	0.614	0.646
Bin 5	0.808	0.563	0.648	0.710	0.567	0.590	0.850	0.655	0.637	0.672
Bin 6	0.920	0.599	0.477	0.559	0.639	0.525	0.686	0.672	0.669	0.641
Bin 7	0.681	0.787	0.498	0.562	0.625	0.448	0.621	0.669	0.647	0.502
Bin 8	0.599	0.757	0.584	0.727	0.694	0.485	0.656	0.725	0.604	0.546
Bin 9	0.646	0.613	0.522	0.737	0.719	0.616	0.787	0.669	0.518	0.695
Bin 10	0.745	0.622	0.559	0.820	0.818	0.834	1.000	0.757	0.719	0.945
Bin 11	0.609	0.762	0.710	0.625	0.595	0.597	0.758	0.677	0.920	0.665
Bin 12	0.558	0.949	0.607	0.654	0.492	0.403	0.668	0.583	0.745	0.429

Tabla 2: Valores del descriptor chroma obtenidos normalizados

B. IMPLEMENTACIONES

B.1. PREPARACIÓN DE LA SEÑAL TEMPORAL

```
import librosa
import numpy as np
import matplotlib.pyplot as plt
carpeta='/Users/Fede/Desktop/The Turn of a Friendly Card 1979 (GPF)/Canciones del trabajo/
filename='10'
formato='.mp3'
filename=carpeta+filename+formato
# cargar audio
y, sr = librosa.load(filename)
# extraer silencios al principio y final
y,index=librosa.effects.trim(y, top_db=60, ref=np.amax, frame_length=2048, hop_length=50)
# normalizar
valor_max=np.max(y)
y=y/valor_max
```

B.2. VISUALIZACIÓN DE LA STFT

```
import librosa
import numpy as np
import matplotlib.pyplot as plt
carpeta='/Users/Fede/Desktop/'
filename='burno mars'
formato='.mp3'
filename=carpeta+filename+formato
# cargar audio
y, sr = librosa.load(filename)
# extraer silencios al principio y final
y,index=librosa.effects.trim(y, top_db=60, ref=np.amax, frame_length=2048, hop_length=50)
# normalizar
valor_max=np.max(y)
y=y/valor_max
D = librosa.stft(y)
librosa.display.specshow(librosa.amplitude_to_db(D, ref=np.max)
, y_axis='log', x_axis='time')
plt.title('Power spectrogram')
plt.colorbar(format='%+2.0f dB')
plt.tight_layout()
plt.show()
```

B.3. DESCRIPTOR TEMPO

```
import librosa
import numpy as np

carpeta='/Users/Fede/Desktop/The Turn of a Friendly Card 1979 (GPF)/Canciones del trabajo/'
filename='10'
formato='.mp3'
filename=carpeta+filename+formato
# cargar audio
y, sr = librosa.load(filename)
# extraer silencios al principio y final
y,index=librosa.effects.trim(y, top_db=60, ref=np.amax, frame_length=2048, hop_length=50)
# normalizar
valor_max=np.max(y)
y=y/valor_max
# calcular tempo
hop_length = 512
oenv = librosa.onset.onset_strength(y=y, sr=sr, hop_length=hop_length)
tempogram = librosa.feature.tempogram(onset_envelope=oenv, sr=sr,
                                     hop_length=hop_length)
ac_global = librosa.autocorrelate(oenv, max_size=tempogram.shape[0])
ac_global = librosa.util.normalize(ac_global)
# Estimate the global tempo for display purposes
tempo = librosa.beat.tempo(onset_envelope=oenv, sr=sr, hop_length=hop_length) [0]
print(tempo)
```

B.4. SILENCE RATE

```
import librosa
import numpy as np
```

```

carpeta='/Users/Fede/Desktop/The Turn of a Friendly Card 1979 (GPF)/Canciones del trabajo/
filename='9'
formato='.mp3'
filename=carpeta+filename+formato
# cargar audio
y, sr = librosa.load(filename)
# # extraer silencios al principio y final
# y,index=librosa.effects.trim(y, top_db=60, ref=np.amax,
# frame_length=1024, hop_length=50)
# normalizar
valor_max=np.max(y)
y=y/valor_max
# calcular sr
intervals=librosa.effects.split(y, top_db=60, ref=np.amax,
frame_length=1024, hop_length=50)
M_ns=0
for i in range(0,intervals.shape[0]):
    M_ns = M_ns+intervals[i,1]-intervals[i,0]
N=len(y)
SR=((N-M_ns)/N)
SR=np.array([SR])
print(SR)

```

B.5. FACTOR DE CRESTA

```

import librosa
import numpy as np
import matplotlib.pyplot as plt
carpeta='/Users/Fede/Desktop/The Turn of a Friendly Card 1979 (GPF)/Canciones del trabajo/
filename='10'
formato='.mp3'
filename=carpeta+filename+formato
# cargar audio
y, sr = librosa.load(filename)
# extraer silencios al principio y final
y,index=librosa.effects.trim(y, top_db=60, ref=np.amax, frame_length=2048, hop_length=50)
# normalizar
valor_max=np.max(y)
y=y/valor_max
# calculo el valor rms
y_rms=y*y
y_rms=np.sum(y_rms)
y_rms=y_rms/len(y)
y_rms=np.sqrt(y_rms)
# calculo el valor maximo
y_max=np.max(y) # siempre vale 1
# calculo el factor de cresta
factor_de_cresta=(y_max)/(y_rms)
print(factor_de_cresta)

```

B.6. CHROMA

```

import librosa
import numpy as np
carpeta='/Users/Fede/Desktop/The Turn of a Friendly Card 1979 (GPF)/Canciones del trabajo/

```

```

filename='10'
formato='.mp3'
filename=carpeta+filename+formato
# cargar audio
y, sr = librosa.load(filename)
# extraer silencios al principio y final
y,index=librosa.effects.trim(y, top_db=60, ref=np.amax,
frame_length=1024, hop_length=50)
# normalizar
valor_max=np.max(y)
y=y/valor_max
# calculo el chromagram
chromagram=librosa.feature.chroma_cqt(y,sr)
chromagram=np.mean(chromagram,1)
np.savetxt('test.txt',chromagram,delimiter=' \\ ',fmt='%.3f', newline=' & ')

```