

Trabalho 8 – Busca de Músicas

Diversos softwares têm sido projetados para realizar a busca de músicas a partir de um trecho de áudio. Por exemplo, atualmente há ferramentas para celulares ou Web em que o usuário grava um trecho de uma música, pois deseja descobrir qual é aquela música. Uma das abordagens mais comuns é por meio do cálculo do **Espectrograma** conforme segue abaixo:

1) Considere que as músicas serão entregues a seu programa no seguinte formato:

Um arquivo binário em que cada 2 bytes (isto é, um *short int*) contém o valor da amplitude do áudio. A amplitude é a energia com que o som foi gravado. Por exemplo, um áudio gravado muito perto do microfone tem alta energia.

2) Abra o arquivo de música e recorte trechos de tamanho WINDOW

2.1) O que é um trecho?

Resp: É um conjunto de WINDOW elementos short, em que cada short tem 2 bytes

Por exemplo, se o áudio total contém 1000 valores (short ints), então considere que WINDOW = 100. Neste caso você teria 10 trechos, cada um desses trechos com 100 shorts.

2.2) Aplique a **Transformada Rápida de Fourier 1-D** sobre cada um dos trechos

http://pt.wikipedia.org/wiki/Transformada_de_Fourier#Transformada_discreta_de_Fourier

http://pt.wikipedia.org/wiki/Transformada_r%C3%A1pida_de_Fourier

Você poderá utilizar código existente para essa função, adaptado para o Trabalho 8, não é preciso fazer sua própria implementação. A Transformada Rápida é necessária para permitir trabalhar com arquivos maiores.

2.3) Para cada um dos trechos, selecione os K coeficientes (em que K será dado como entrada para seu programa) de maior magnitude (ou seja, de maior amplitude) e faça o seguinte:

2.3.1) Sabe-se, na música ocidental, que as notas são divididas em:

C, C#, D, D#, E, F, F#, G, G#, A, A#, B

em que:

C → Dó

D → Ré

E → Mi

F → Fá

G → Sol

A → Lá

B → Si

→ Sustenido (indica adição de frequência em relação ao tom anterior, por exemplo, C tem certa frequência e C# tem uma frequência adicional)

2.3.2) Na música ocidental, utilizamos esses 12 tons (ou, mais formalmente, intervalos).

Por exemplo, a frequência 440 representa uma nota Lá. O dobro dela também representa uma nota Lá, i.e., 880. O dobro de 880 também representa outra nota Lá. Neste caso, considere as frequências abaixo:

A → 440 hz (hertz é uma medida de frequência)

Como encontrar a frequência da nota A# ou B ou outra qualquer ???

Para isso utilize a seguinte equação:

$$\text{frequencia_da_nota} = \text{nota_base} * (2^{(1/12)})^{\text{intervalo}}$$

Por exemplo, para encontrar a frequência da nota que vem após Lá (A), ou seja, A#, temos como base a nota A:

$$\text{nota_base} = A = 440 \text{ hz}$$

Utilizamos **intervalo = 1**, pois queremos recuperar a frequência do próximo intervalo após A, i.e., um intervalo à frente, assim temos:

$$A\# = 440 * (2^{(1/12)})^1 = 466.16$$

E assim sucessivamente. Agora observe outro caso, como descobrimos o intervalo para a frequência 65.41? Basta substituir na fórmula:

$$65.41 = 440 * (2^{(1/12)})^{\text{intervalo}}$$

OBS: Como o resultado dessa fórmula é um intervalo na forma de um número real, faça arredondamento no intervalo obtido utilizando a função round()

2.3.4) Para cada trecho (janela), selecione os K coeficientes de maior magnitude. Cada um desses coeficientes terá uma frequência associada. Pegue cada uma dessas frequências e encontre seu respectivo intervalo SEMPRE considerando como frequência base 440hz, i.e., a nota Lá (A). Para exemplificar, considere alguns possíveis intervalos para a frequência base 440 hz, cuja nota será referenciada como A₄ ou nota Lá na escala número 4 (conforme a escala aumenta, os sons ficam mais agudos):

Nota	Frequência	Intervalo
A ₄	440.00	0
A [#] ₄ /B ^b ₄	466.16	1
B ₄	493.88	2
C ₅	523.25	3
C [#] ₅ /D ^b ₅	554.37	4
D ₅	587.33	5

$D^{\#}_5/E^b_5$	622.25	6
E_5	659.26	7
F_5	698.46	8
$F^{\#}_5/G^b_5$	739.99	9
G_5	783.99	10
$G^{\#}_5/A^b_5$	830.61	11
A_5	880.00	12
$A^{\#}_5/B^b_5$	932.33	13
B_5	987.77	14
C_6	1046.50	15
$C^{\#}_6/D^b_6$	1108.73	16
D_6	1174.66	17
$D^{\#}_6/E^b_6$	1244.51	18
E_6	1318.51	19
F_6	1396.91	20
$F^{\#}_6/G^b_6$	1479.98	21
G_6	1567.98	22
$G^{\#}_6/A^b_6$	1661.22	23
A_6	1760.00	24

Por exemplo, considere o primeiro trecho da música e $K = 3$. Suponha, neste caso que os coeficientes com maior magnitude estão nas frequências 440, 880, 1567. Aplicando a equação anteriormente dada, você chegará aos seguintes intervalos:

$$\begin{aligned} 440 &\rightarrow 0 \\ 880 &\rightarrow 12 \\ 1567 &\rightarrow 22 \end{aligned}$$

Agora iremos descobrir quais são as notas para os intervalos do primeiro trecho analisado (0, 12, 22). Para isso podemos usar o resto da divisão por 12 na forma:

$$\begin{aligned} 440 &\rightarrow 0 \% 12 = 0 \\ 880 &\rightarrow 12 \% 12 = 0 \\ 1567 &\rightarrow 22 \% 12 = 10 \end{aligned}$$

Temos assim quais foram as notas dominantes naquele intervalo. Agora iremos criar uma matriz

que terá N linhas e 12 colunas, em que N representa o número total de trechos que iremos analisar da música. Para cada trecho, ou linha da matriz, iremos contabilizar quais notas ocorreram (representadas pelas 12 colunas).

Neste exemplo, iremos obter para o primeiro trecho a lista de valores abaixo:

0	1	2	3	4	5	6	7	8	9	10	11
2	0	0	0	0	0	0	0	0	0	1	0

Em que a primeira linha indica o número do intervalo (ou nota) e a segunda linha, de fato, indica **quantas vezes cada um desses intervalos ocorreu no primeiro trecho da música**.

Observe que após processar, por exemplo, 10 trechos da música, teremos algo como:

	Intervalos ou Notas Músicas Tocadas para cada Trecho da Música											
Trecho	0	1	2	3	4	5	6	7	8	9	10	11
1	2	0	0	0	0	0	0	0	0	0	1	0
2	0	3	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	1	0	1	0	0
4	2	0	0	0	0	0	0	0	0	0	1	0
5	0	3	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	1	0	1	0	0
7	2	0	0	0	0	0	0	0	0	0	1	0
8	0	3	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	1	0	1	0	0
10	2	0	0	0	0	0	0	0	0	0	1	0

Observação: Veja que os intervalos podem ser negativos em relação à frequência base 440 Hz. Por exemplo, considere que ocorreu a frequência 415, cujo intervalo = -1. Neste caso a nota correta é G#, ou seja, você deveria preencher a contagem de ocorrência do intervalo 11 e não do intervalo 1.

2.4) Seu programa irá receber uma **lista inicial** com **diversas músicas** que deverá processar e obter matrizes como a anterior. Cada música dessa lista irá ser utilizada para produzir uma matriz que irá representar as características da música, montando um tipo de base de músicas.

Em seguida, iremos apresentar uma **música de consulta** para seu programa. Você deverá, também, produzir uma matriz como a anterior para essa nova música e, em seguida, você deverá calcular qual das músicas da lista inicial que você extraiu anteriormente é a mais parecida com a música de consulta. **O programa deverá imprimir apenas o ID (número) da música mais parecida, com base na comparação entre as matrizes.**

Observe que a música de consulta pode ter mais ou menos trechos em relação às músicas da lista. Sendo assim, o número de trechos pode variar de música para música. Não daremos nenhuma técnica para você realizar tal comparação e encontrar a música mais próxima, a ideia é você explorar possibilidades de acordo com seus conhecimentos.

Dica: Converta algumas de suas músicas MP3 ou WAV para o formato RAW (ou formato cru). Esse formato armazena 1 short (ou 2 bytes) para cada amplitude de áudio. Exemplo:

```
sox musica.wav -b 16 -c 1 -r 16k -t raw musica.raw
```

Neste caso convertemos musica.wav para musica.raw. Esse arquivo musica.raw contém apenas bytes em que cada 16 bits (opção -b 16), i.e., 2 bytes (ou 1 short) contém valores de amplitude para o áudio. Os seguintes parâmetros significam:

- c 1 → Indica que o áudio será gravado como mono (1 canal apenas) em musica.raw
- r 16k → Indica que 16000 amplitudes serão gravadas por segundo, o que totalizam $16000 * 2 \text{ bytes} = 32000 \text{ bytes}$ por segundo de áudio
- t raw → apenas indica que iremos produzir um arquivo de áudio cru como resultado

Esse programa sox é disponível para Linux.

Se desejar tocar o arquivo musica.raw você pode fazer:

```
aplay -r 16k -t raw -c 1 -f S16_LE musica.raw
```

Esse comando aplay está disponível para Linux. Suas opções são as mesmas do comando **sox**. Exceto a opção -f S16_LE que é similar à -b 16 para o sox.

Exemplos de Entrada e Saída

Entrada para seu programa:

```
<TAMANHO_LISTA>  
<nome_arquivo_musica1.raw>  
<nome_arquivo_musica2.raw>  
...  
<nome_arquivo_musica_TAMANHO_LISTA.raw>  
<nome_arquivo_musica_consulta.raw>
```

Saída:

```
<Número da música na lista considerada mais próxima>\n
```

Exemplo:

3
audio.raw
teste.raw
abc.raw
consulta.raw

Digamos que a música mais parecida seja a primeira da lista, i.e., audio.raw. Então seu programa será gerar como saída o inteiro 1, o qual indica o índice da música mais parecida (**sempre iniciando em 1**).

Saída:

1\n