

Análise Semantica de Sinais Audio Audio Fingerprinting

»» Relatório Intercalar

Aluno_ Miguel Henriques
Orientador_ Prof. Artur Pimenta Alves

Maio2003



INDICE

INTRODUÇÃO	3
EXTRACÇÃO DE CARACTERÍSTICAS	5
BASE DE DADOS (pesquisa e matching)	11
TESTES E RESULTADOS	14
DISCUSSÃO	17
IMPLEMENTAÇÃO	18
RESULTADOS OBTIDOS	20
REFERÊNCIAS	24

INTRODUÇÃO

O objectivo deste projecto é o de desenvolver um protótipo de um sistema de áudio fingerprinting. Ou seja, pretende-se conceber e implementar um sistema capaz de extrair determinadas características de sinais áudio que permitam gerar informação (fingerprints) que as identifiquem exclusivamente. Esta informação funcionará então como uma espécie de assinatura da musica e permitirá a sua correcta identificação independentemente do seu formato e mesmo da sua origem.

Existem alguns estudos sobre este tipo de sistemas, estando inclusivamente já disponível software que os implementa [1] [2] [3].

O que se verifica é que todos esses sistemas são semelhantes do ponto de vista conceptual, ou seja partem das mesmas ideias base e têm, todos eles, uma estrutura muito similar. Assim, todos evidenciam a existência de 3 passos essenciais (ver Figura 1): a extracção de características; a modelização de fingerprints e a exploração da base de dados (construção, pesquisa e matching)

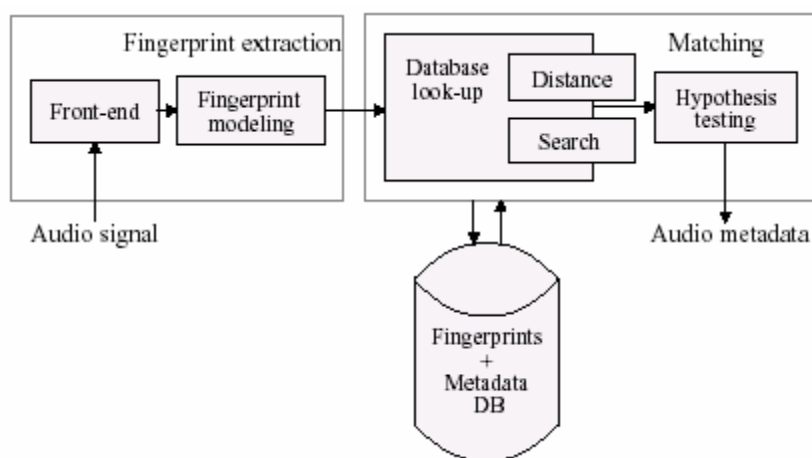


Figura 1 - Content-based Audio Identification Framework (retirado de [4])

No entanto são apresentadas as mais variadas soluções para cada uma destas tarefas base. O que se pretende neste relatório é exactamente apresentar um levantamento dessas soluções de forma a tirar conclusões que permitam definir qual, ou quais, os sistemas a implementar.

Independentemente dos métodos utilizados para implementar as diferentes tarefas de um sistema de áudio fingerprinting, é exigível que um sistema deste tipo seja eficiente e robusto,

sendo estas duas características essenciais para que o sistema tenha aplicação prática:

Robustez

- O sistema e mais concretamente o correcto reconhecimento de uma musica, tem que ser insensível a factores como o ruído de fundo e consequente degradação do sinal, o formato do ficheiro áudio derivado de diferentes tipos de codificação, a equalização, os deslocamentos temporais. Ou seja, pretende-se que o sistema seja capaz de superar degradações severas no sinal que se pretende identificar, pretende-se que o sistema seja *versátil*.
- A *granularidade* é outra característica essencial do sistema pois torna-o mais robusto e completo. A *granularidade* não é mais do que o tamanho mínimo do extracto de uma musica que permite a sua correcta identificação. É por isso de desejar que o sistema não apresente uma *granularidade* muito elevada.
- Muito importante é também que a *confiança* no sistema seja elevada, ou seja que a sua capacidade de identificação nos leve a ter um elevado grau de confiança no sistema. Para isso é muito importante que o sistema tenha um grande poder de discriminação sobre enormes quantidades de fingerprints, de forma a termos elevadas taxas de acerto e casos de identificação incorrectas reduzidos ao mínimo.

Eficiência

Uma questão essencial para a eficiência de um sistema é a sua *velocidade*. É essencial que os tempos de processamento de sinal e extracção de características, de construção do fingerprint e de pesquisa e matching na base de dados, sejam o mais pequeno possível mesmo para grande quantidade de dados. Há diversos factores com influência directa na velocidade do sistema e consequentemente na sua eficiência:

- *Tamanho do fingerprint* - o modelo de fingerprint utilizado não deve ser muito extenso nem muito pesado, principalmente para permitir pesquisas e identificações rápidas.

- *Complexidade* - o sistema não deve apresentar uma elevada complexidade computacional, não só para não diminuir excessivamente a sua velocidade como para não aumentar os seus custos. Deve portanto ser um sistema *compacto*.

EXTRACÇÃO DE CARACTERÍSTICAS

	Sistema (consultar referências)	Características (features) extraídas
1	Philips [5]	Diferença de energias de bandas consecutivas
2	AudioID [9] [10] [11] [12]	SFM; SCF
3	AudioDNA [6] [7] [8]	MFCC; Energia; (e respectivas derivadas)
4	TRM [1]	Average zero crossing rate; BPM (beats per minute); average spectrum; diferença de energias; Haar wavelet transform; Potência do espectro.

1 Philips: Diferença de energias de bandas consecutivas

É sugerido neste sistema usar essencialmente o domínio das frequências pois muitas das características mais importantes para a caracterização de sinais áudio (principalmente as relacionadas com as tonalidades) encontram-se neste domínio. Como se pode observar na Figura 2 é feita uma divisão de bandas, no espaço logarítmico, e é, para cada uma delas, calculada a energia. Realce-se que apenas se consideram as bandas mais significativas para a percepção humana, entre os 300Hz e os 3000Hz. O que se extraí efectivamente é então uma diferença de energias entre bandas consecutivas de frames consecutivos.

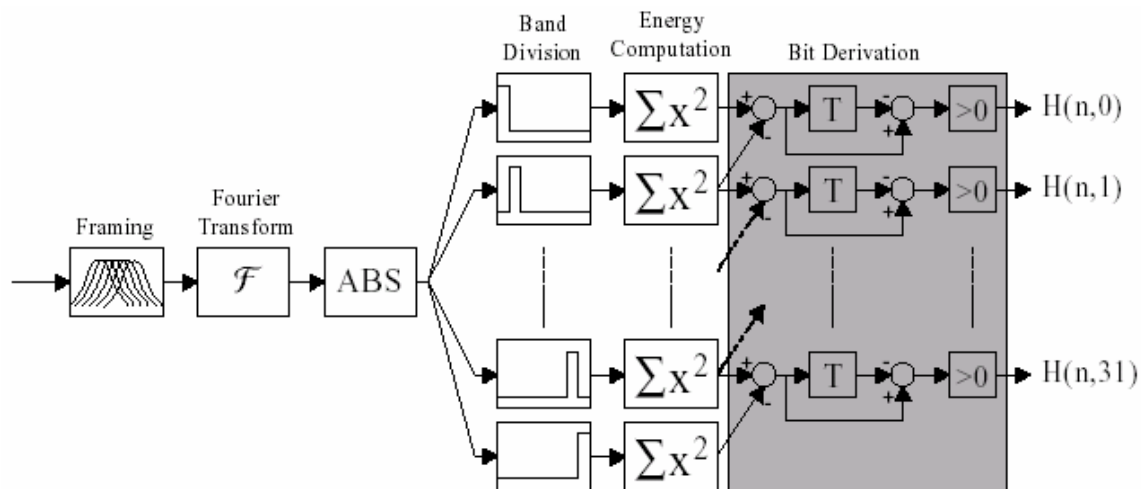


Figura 2 - Overview fingerprint extraction scheme (retirado de [5])

2 AudioID: SFM; SCF

Também neste caso a análise será efectuada no domínio das frequências, para o efeito é efectuada uma DFT (Discrete Fourier Transform). Depois é extraído um conjunto de características psicoacústicas (intimamente relacionadas com a capacidade perceptual do ouvido humano) com o qual se construirá um vector de características. É inicialmente sugerido que desse conjunto façam parte as seguintes características: loudness, SFM, sharpness e spectral tilt, mas após análise dos resultados experimentais chega-se à conclusão que é suficiente a extracção do SFM (Spectral Flatness Measure) e do SCF (Spectral Crest Factor). O SCF é uma característica derivada do SFM e por isso intimamente relacionada com este. Uma grande vantagem desta opção é permitir a integração com a norma MPEG-7 que já prevê a extracção destas duas características (especialmente o SFM). Uma fase importante do processo é então a construção do vector a partir das características extraídas, a qual envolve uma série de operações de normalização e operações de dimensionalização por forma a tornar o sistema mais robusto e eficiente.

3 AudioDNA: MFCC; Energia; (e respectivas derivadas)

O vector de características que neste caso servirá como base para a construção do fingerprint é constituído pelos coeficientes mel cepstrais e também pela energia do sinal (mais especificamente de cada frame do sinal). De forma a incluir alguma informação sobre a evolução temporal, de frame para frame, são também incluídos neste vector a

primeira e segunda derivadas de cada uma das 2 características referidas.

4 TRM: Average zero crossing rate; BPM (beats per minute); average spectrum; diferença de energias; Haar wavelet transform; Potência do espectro.

Neste sistema um extracto do sinal áudio (cerca de 26 segundos) é representado por um vector que inclui uma série de características (ver tabela), algumas delas de mais alto nível, com um maior significado musical.

NOTA: Como se pode observar todas as soluções apresentadas se baseiam numa divisão do sinal a analisar em janelas (frames) e a extracção de características é portanto efectuada para cada uma delas.

CONSTRUÇÃO DO FINGERPRINT

	Sistema (consultar referências)	Modelo de fingerprint utilizado
1	Philips [5]	Sequência binária (blocos de 32x256 bits)
2	AudioID [9] [10] [11] [12]	Codebook de características
3	AudioDNA [6] [7] [8]	Sucessão de estados (estados esses que podem por exemplo ser descritos por uma letra)
4	TRM [1]	Cadeia de 32 símbolos hexadecimais

1 Philips: Sequência binária (blocos de 32x256 bits)

É proposto pelos autores utilizar uma regra de decisão (ver Figura 3 onde n representa o frame m a banda e EB a energia) para codificar as características extraídas - energias - em sequências binárias. No fundo os bits destas sequências representam o sinal da diferença entre as derivadas no tempo e nas frequências da energia. Cada frame foi dividido em 33 bandas (compreendidas entre os 300Hz e os 3000Hz) e, utilizando a regra referida, dá origem a 32 bits que constituirão o que se chamará um sub-fingerprint. Para identificar adequadamente uma musica são necessários 256 sub-fingerprints que equivalem a aproximadamente 3 segundos de musica. Esta estrutura de 8152 (32x256) bits é chamada de fingerprint block e é portanto a unidade básica de todo o processo de reconhecimento e identificação.

$$H(n,m) = \begin{cases} 1 & \text{if } EB(n,m) - EB(n,m+1) - (EB(n-1,m) - EB(n-1,m+1)) > 0 \\ 0 & \text{if } EB(n,m) - EB(n,m+1) - (EB(n-1,m) - EB(n-1,m+1)) \leq 0 \end{cases}$$

Figura 3 - Regra de decisão aplicada em [5]

2 AudioID: Codebook de características

Esta solução é toda baseada nos vectores de características criados e na construção de um codebook de classificação. A exploração das redundâncias globais existentes numa musica permite assumir a existência de um elevado grau de semelhança entre os vectores de características de diferentes frames de uma mesma musica. No fundo é esta ideia que dá origem à geração do codebook referido, que não é portanto mais do que uma aproximação aos vectores de características, com a grande vantagem de ser

significativamente mais pequeno. Esta diminuição de dimensão é resultado do uso do algoritmo de K-means à informação contida nos vectores de características.

A Figura 4 procura exemplificar a aplicação desse algoritmo na construção do codebook representativo de uma musica, cada ponto representa um frame.

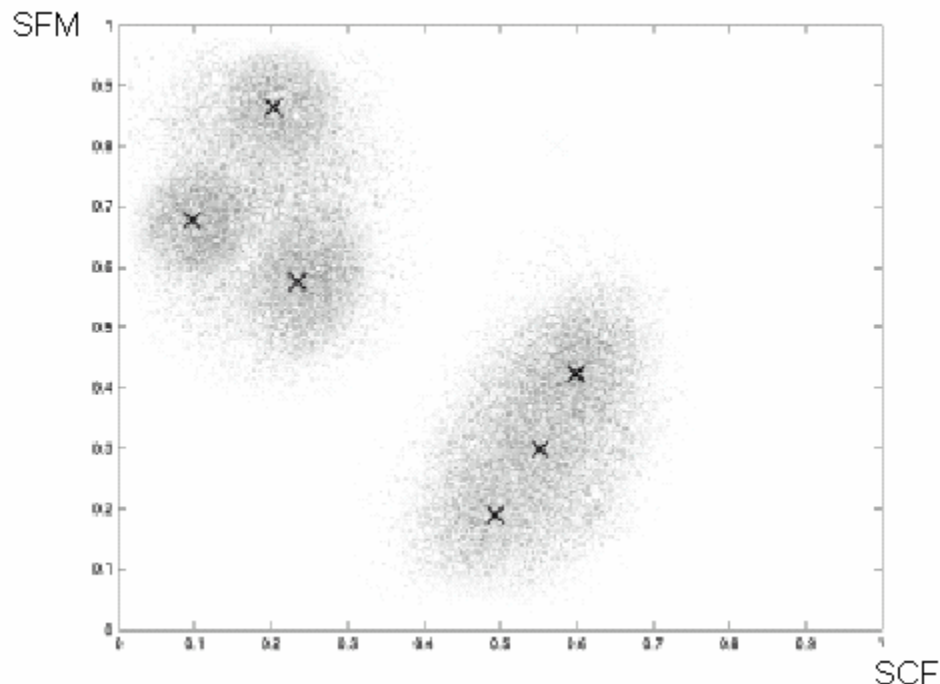


Figura 4 - Exemplo de construção do codebook usando 6 centroides (adaptado de [9])

Concluindo, o fingerprint da musica não é mais que um vector que representa os pontos correspondentes aos centros do algoritmo aplicado (K-means), é no fundo uma espécie de esqueleto da musica no plano das duas características escolhidas (SFM e SCF).

Para além disso e de forma a melhorar o desempenho na fase de reconhecimento é proposto q se introduzam dependências temporais entre os frames. Assim os valores correspondentes a cada frame não são instantâneos mas dependem dos valores dos frames anteriores. Para o efeito é usada uma memória e são calculadas médias.

3 AudioDNA: Sucessão de estados (estados esses que podem por exemplo ser descritos por uma letra)

Este sistema propõe uma abordagem bastante diferente dos restantes pois assenta na ideia de que um sinal áudio não é mais do que uma sequência de *eventos acústicos*, daí o nome de audioDNA. No entanto também esta solução parte da assunção da existência das redundâncias em cima referidas, já que é com base nesse facto que é criado um alfabeto de eventos acústicos. Este alfabeto é constituído por elementos que procuram representar os estados escondidos do sistema, ou seja são todos os estados que segundo diferentes combinações dão origem a todas as musicas.

O passo essencial na construção dos fingerprints é portanto a definição deste alfabeto e a codificação de cada musica segundo esse alfabeto, passo esse que se baseia na técnica estocástica de reconhecimento de padrões HMM (hidden markov models).

Todo este processo exige uma extensa fase de treino que se torna assim na base de todo o sistema.

Concluindo, a ideia apresentada nesta solução é a de usar as características extraídas (MFCC e energias) e usar os HMM para, calculando probabilidades associadas a cada estado, codificar a musica como uma sequência de eventos, de estados, de "AudioGenes". Esta solução concentra portanto a maior parte do processamento e da complexidade nesta fase de construção do fingerprint.

NOTA: Um passo intermédio importante, pois permite reduzir o numero de parâmetros do sistema e conseqüentemente aumentar a eficiência do sistema, é a descorrelação das características extraídas.

4 TRM: Cadeia de 32 símbolos hexadecimais

Neste caso o que é proposto é uma simples codificação das características extraídas, bem como, de alguma informação temporal de forma a gerar uma sequência de 32 (8+4+4+4+2+2+2+2+2+2+2) caracteres. Estes são finalmente convertidos em símbolos ASCII.

NOTA: A informação disponível sobre este sistema é reduzida e bastante confusa e por isso não permite compreender na sua totalidade o seu funcionamento.

BASE DE DADOS (pesquisa e matching)

	Sistema (consultar referências)	
1	Philips [5]	Look-up Table
2	AudioID [9] [10] [11] [12]	Nearest neighbour algorithm
3	AudioDNA [6] [7] [8]	Pesquisa circular e algoritmo de Viterbi.
4	TRM [1]	(informação não disponível)

1 Philips: Look-up Table

Esta solução propõe uma forma de evitar a solução "à força bruta", isto porque é impraticável implementar um sistema de dimensão apreciável em que cada operação de reconhecimento exija que a pesquisa e conseqüente matching se estenda a toda a base de dados. É sugerido então que se crie uma tabela adicional à base de dados que contenha todos os sub-fingerprints possíveis (neste caso 2^{32} entradas) e o seu respectivo posicionamento na base de dados. Ou seja cada subfingerprint está associado a apontadores para todas as suas posições na BD. A Figura 5 ilustra como funciona esta solução com um pequeno exemplo:

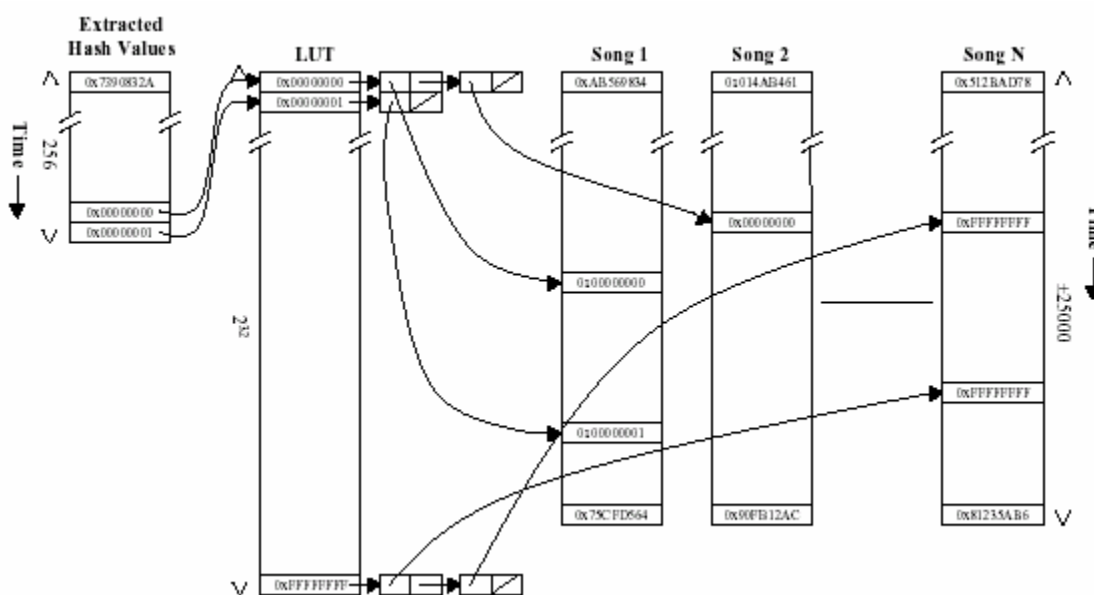


Figura 5 - Overview database setup (retirado de [5])

O procedimento é então o seguinte: começa-se com o primeiro sub-fingerprint do fingerprint block que se pretende identificar e, usando a tabela (LUT), descobrem-se todas as posições da base de dados onde o podemos encontrar; depois para cada uma dessas posições procura-se fazer um matching de todo o bloco; no caso do erro estar abaixo de um determinado limite, então temos uma identificação positiva, em caso contrário repete-se o procedimento para o sub-fingerprint original seguinte;

A solução apresentada parte do importante pressuposto de que cada fingerprint block contém pelo menos um sub-fingerprint sem erros, presunção essa que é suportada por testes realizados ao sistema.

Concluindo, esta solução propõe um sistema de pesquisa e matching na base de dados de fingerprints muito eficiente, com tempos de pesquisa muito satisfatórios.

2 AudioID : Nearest neighbour algorithm

Esta solução propõe um sistema de reconhecimento e identificação baseado num simples algoritmo de vizinho mais próximo. Assim, para cada musica a identificar é extraído um codebook (como já foi visto na secção anterior) e depois aplica-se um algoritmo de vizinho mais próximo de forma a determinar qual o codebook da base de dados mais próximo. No fundo é uma pesquisa "à força bruta" mas, como os codebooks armazenados são de dimensão bastante reduzida, acaba por não ser muito pesada.

3 AudioDNA: Pesquisa circular e algoritmo de Viterbi.

Neste sistema propõe-se que a base de dados seja uma lista circular das sequências HMM criadas para cada musica tal como se pode observar na Figura 6. É esta estrutura que em combinação com a aplicação do algoritmo de Viterbi permite a pesquisa e o reconhecimento das musicas.

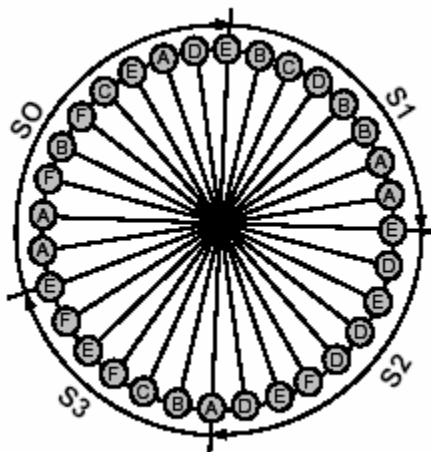


Figura 6 - HMM model for a signature database with four songs: S0-S1-S2-S3 (retirado de [6])

O algoritmo de matching é constituído por dois passos essenciais :

- É feito um primeiro matching (exacto) de extractos da sequência original.
- Nas posições encontradas no primeiro passo procura-se encontrar semelhanças em sequências mais longas

Estes passos (ilustrados na Figura 7) permitem que se encontre na base de dados sequências de AudioDNA similares à pretendida.

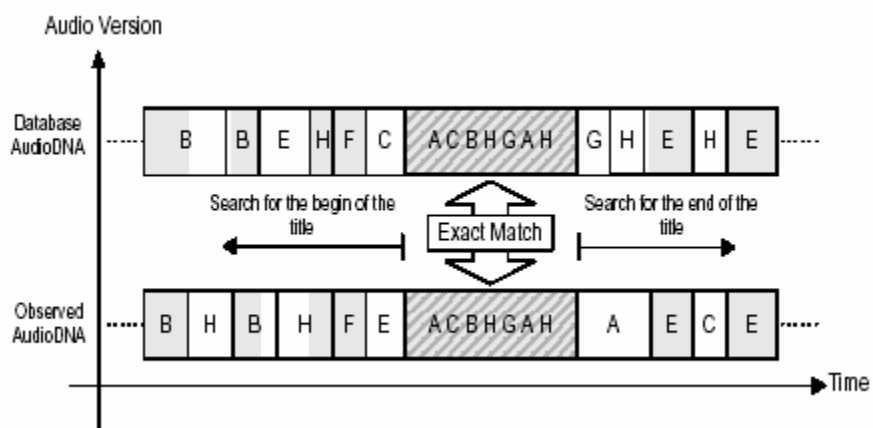


Figura 7 - Exact matching process (retirado de [7])

4 (informação não disponível)

TESTES E RESULTADOS

	Sistema (consultar referências)	Dimensão da Base de Dados	Tempos de pesquisa	Desempenho
1	Philips [5]	10000 músicas	1.5 ms	Ver Figura 8 e Figura 9
2	AudioID [9] [10] [11] [12]	10000 músicas	0.3 s	
3	AudioDNA [6] [7] [8]	a) 3852 músicas * b) 2000 músicas **	b) 2 minutos	a) Ver Figura 11

* Testes realizados no contexto de uma emissão rádio [6]

** Testes realizados no contexto da Internet. O tempo de pesquisa equivale ao matching de toda a base de dados, ou seja das 2000 músicas (12 horas de música) [7].

1 Philips

Processing	Orff	Sinead	Texas	ACDC
MP3@128Kbps	0.078	0.085	0.081	0.084
MP3@32Kbps	0.174	0.106	0.096	0.133
Real@20Kbps	0.161	0.138	0.159	0.210
GSM	0.160	0.144	0.168	0.181
GSM C/I = 4dB	0.286	0.247	0.316	0.324
All-pass filtering	0.019	0.015	0.018	0.027
Amp. Compr.	0.052	0.070	0.113	0.073
Equalization	0.048	0.045	0.066	0.062
Echo Addition	0.157	0.148	0.139	0.145
Band Pass Filter	0.028	0.025	0.024	0.038
Time Scale +4%	0.202	0.183	0.200	0.206
Time Scale -4%	0.207	0.174	0.190	0.203
Linear Speed +1%	0.172	0.102	0.132	0.238
Linear Speed -1%	0.243	0.142	0.260	0.196
Linear Speed +4%	0.438	0.467	0.355	0.472
Linear Speed -4%	0.464	0.438	0.470	0.431
Noise Addition	0.009	0.011	0.011	0.036
Resampling	0.000	0.000	0.000	0.000
D/A A/D	0.088	0.061	0.111	0.076

Figura 8 - BER for different kinds of signal degradation

Na Figura 8 encontram-se as percentagens de erro (BER) dos diferentes fingerprints extraídos para diferentes codificações e distorções do sinal audio original.

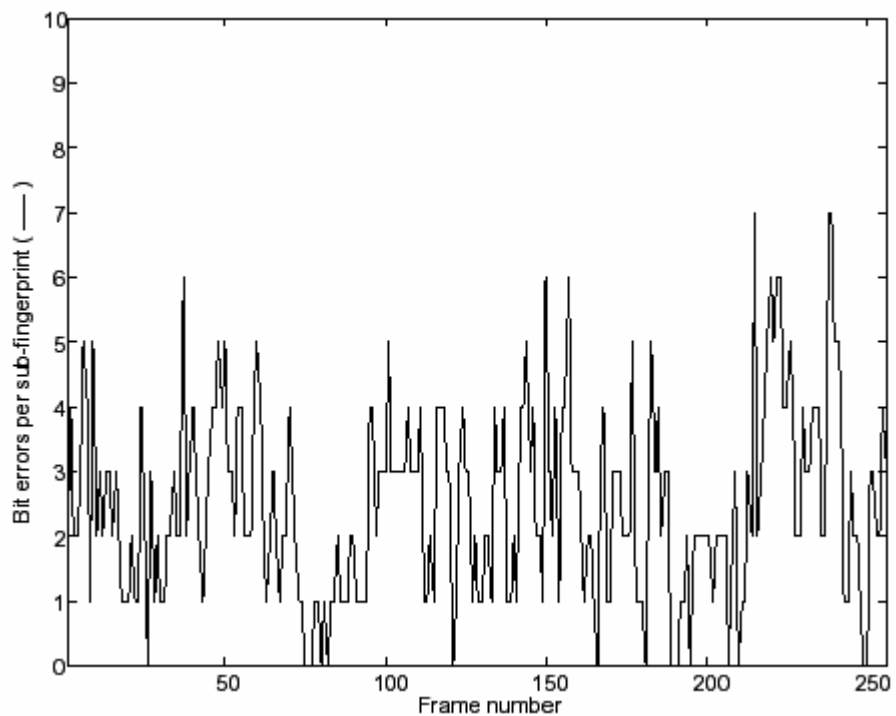


Figura 9 - Bit errors per sub-fingerprint (example)

A Figura 9 procura mostrar a validade da condição já referida de haver pelo menos um sub-fingerprint sem erros num sub-fingerprint block. Neste caso temos 17 sub-fingerprints totalmente correctos.

2 AudioID

Feature	SFM	SCF
No distortion	100.0% / 100.0%	100.0% / 100.0%
Cropping	100.0% / 100.0%	100.0% / 100.0%
MPEG-1/2 Layer 3 @ 96 kbit/s	99.6% / 99.8%	100.0% / 100.0%
MPEG-1/2 Layer 3 @ 96 kbit/s & Cropping	97.9% / 99.9%	99.7% / 100.0%
Loudspeaker / Microphone Chain & Cropping	98.0% / 99.0%	98.8% / 99.5%

Figura 10 - Recognition Performance of SFM and SCF features (15000 items setup)

3 AudioDNA

Audio Source	Identification with False Positives	Identification with no False Positives
Original	100%	100%
Radio capture	100%	100%
MP3 128 kbps	100%	100%
MP3 32 kbps	99.83%	100%
MP3 24 kbps	99.04%	100%

Figura 11 - Desempenho do sistema (retirado de [6])

DISCUSSÃO

Após esta breve apresentação dos diferentes métodos e sistemas existentes associados à identificação de música utilizando audiofingerprint, o passo seguinte é escolher a solução a implementar.

A primeira decisão tomada foi a de antes de, avançar para a construção da base de dados e desenvolvimento dos algoritmos de pesquisa e matching, explorar a parte de extração de características e construção do fingerprint pois é uma fase decisiva para o funcionamento de todo o sistema.

Resolveu-se então começar por procurar implementar o sistema da Philips. Esta escolha teve essencialmente a ver com o facto de ser a solução que está melhor documentada.

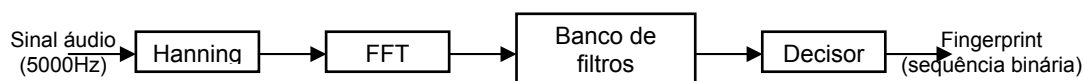
Posteriormente procurar-se-á implementar a solução designada em cima por audioID por forma a comparar os resultados obtidos numa e noutra solução e poder tirar conclusões relativamente ao percurso a seguir daí para a frente. Isto porque esta é também uma solução muito interessante e sobretudo tem uma vertente muito apelativa que é a integração com a norma MPEG-7 e seus descritores.

No entanto tanto uma como outra solução permitem que, dependendo dos resultados obtidos, possam ser pensadas variantes para essas soluções originais, ou mesmo que se pense na implementação de soluções combinadas.

Quanto às outras duas soluções, apesar de serem na realidade as que têm uma maior exploração prática e comercial, concluiu-se serem soluções menos interessantes de explorar, nomeadamente no caso do TRM, devido à falta de informação.

IMPLEMENTAÇÃO

1 Philips



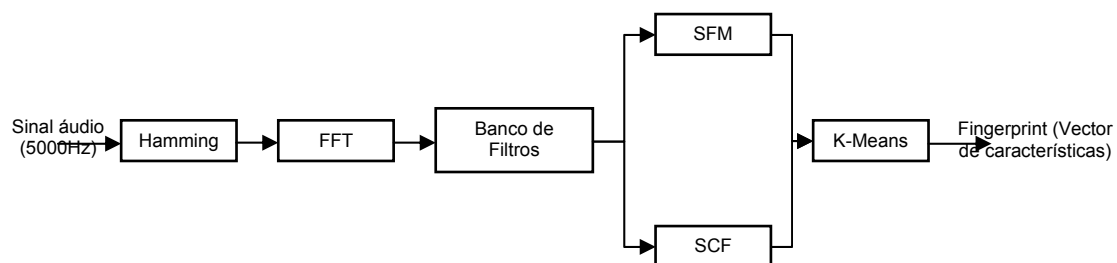
A figura acima apresenta o diagrama de blocos do sistema implementado e totalmente baseado na solução apresentada em [5]. Convém referir e esclarecer alguns aspectos mais importantes relacionados com algumas opções de implementação:

- Em primeiro lugar, é referido em [5] que o sinal áudio original (44100Hz) sofre uma sub-amostragem para os 5000Hz. Nesta fase optou-se por não incluir para já este bloco de sub-amostragem no sistema e fazê-lo externamente, tendo por isso os nossos sinais de entrada a frequência de 5000Hz.

- Um aspecto muito importante prende-se com o tamanho da janela de Hanning, com o seu "hop-size" e com a FFT realizada. Optou-se por utilizar os parâmetros sugeridos pois é para estes que temos informação sobre os resultados obtidos. Assim, utilizou-se uma janela de Hanning de 1850 amostras com um "hop" de 58 amostras, ou seja um factor de "overlap" de $\frac{31}{32}$. Como é utilizada uma DFT de 2048 pontos é necessário efectuar um zero padding de $2048-1850=198$ amostras.

- Finalmente, quanto ao banco de filtros foi também implementado fielmente de acordo com o sistema apresentado em [5]. O espectro foi limitado à banda compreendida entre os 300Hz e os 2000Hz, banda essa que foi por sua vez dividida em 33 bandas logarítmicas não sobrepostas. Resolveu-se ainda, apesar de não ser feita nenhuma referência a este aspecto na solução original, efectuar uma normalização dos diferentes filtros, que ficaram portanto, todos com a mesma área.

2 AudioID



Esta solução encontra-se ainda em fase de implementação. Já se obtiveram os primeiros vetores de características, faltando no entanto a aplicação do algoritmo de clustering que permitirá construir os codebooks de dimensão mais reduzida. No entanto, e tal como aconteceu com a primeira solução referida, procurar-se-á, na medida do possível, seguir a bibliografia disponível, pelo menos nesta fase inicial de implementação.

RESULTADOS OBTIDOS

Apresentam-se de seguida os primeiros resultados obtidos para o primeiro sistema de extracção de fingerprints implementado:

A Tempo de extracção do fingerprint de uma música completa.

B Numero total de bits errados no fingerprint completo para diferentes tipos de codificação.

C Número total de sub-fingerprints sem erros e com um só erro.

TESTES ADICIONAIS:

D BER sem normalização do banco de filtros.

E Codificação MP3 de alta qualidade e com uma taxa de compressão variável.

F Identificação falsa.

A Tempos (Pentium III - 450Mhz)

	Muse	Maxwell	dEUS	Carl Orff	Gorillaz
Tamanho (s)	121	220	116	153	238
Tempo de extracção (s)	41.790	76.169	40.658	53.460	86.123
	2.9 x	2.9 x	2.9 x	2.5 x	2.8 x

B BER

	Muse	Maxwell	dEUS	Carl Orff	Gorillaz
MP3@32Kbps	0.219452	0.173775	0.178405	0.188116	0.200731
MP3@128Kbps	0.180772	0.225156	0.158777	0.177698	0.171095
Real@20Kbps	0.177777	0.072665	0.130095	0.115292	0.165221
GSM	0.197000	0.130383	0.136294	0.131090	0.168837

Como se pode ver e comparando os resultados apresentados na tabela com os expostos na Figura 8, obtiveram-se BER (bit error rate) muito semelhantes com os do sistema original. A excepção é o caso da codificação MP3 a 128Kbps. Os

resultados obtidos foram, neste caso, bastante piores do que os esperados.

Uma possível explicação poderá ter que ver com o codificador utilizado, pois como se pode verificar, tanto para a codificação GSM, como para a codificação da Real os resultados obtidos são até mais favoráveis do que os apresentados em [5].

C Sub-fingerprints

Este é um teste importante tendo, em conta a já referida condição essencial em que se assume que num fingerprint block haverá pelo menos um sub-fingerprint sem erros.

	Muse	Maxwell	dEUS	Carl Orff	Gorilla z
Total de Subfingerprints	10326	18796	9906	13091	20301
MP3@32Kbps					
0 erros	0.136 % (14)	0.729 % (137)	0.717 % (71)	0.420 % (55)	0.404 % (82)
1 erro	0.591 % (61)	3.394 % (638)	2.958 % (293)	1.520 % (199)	1.207 % (290)
MP3@128Kbps					
0 erros	0.271 % (28)	0.271 % (51)	1.363 % (135)	0.435 % (57)	1.015 % (206)
1 erro	1.937 % (200)	1.202 % (226)	4.654 % (461)	1.925 % (252)	3.670 % (745)
Real@20Kbps					
0 erros	0.407 % (42)	12.481 % (2346)	2.675 % (265)	3.002 % (393)	0.818 % (166)
1 erro	2.160 % (223)	23.851 % (4483)	9.257 % (917)	10.381 % (1359)	3.453 % (701)
GSM					
0 erros	0.165 % (17)	1.277 % (240)	2.948 % (292)	2.590 % (339)	1.177 % (239)
1 erro	1.714 % (177)	4.086 % (768)	9.237 % (915)	8.387 % (1098)	4.029 % (818)

Como se pode verificar através dos resultados obtidos, na maior parte dos casos testados a validade da condição é mais que provável.

D Sem normalização

Por forma a verificar a validade da normalização efectuada ao banco de filtros, realizaram-se alguns testes sem essa normalização:

	Muse	dEUS
MP3@128Kbps	0.169581	0.159297

Como se pode observar, não há grandes alterações no erro obtido.

E Codificação MP3 de alta qualidade

Em consequência dos resultados obtidos para a codificação MP3, resolveu-se efectuar testes para uma codificação MP3 de alta qualidade com taxa de compressão de bits variável alternativamente à taxa constante:

	Muse	dEUS
MP3@128Kbps	0.319055	0.300847

Verifica-se neste caso que as taxas de erro aumentam significativamente. Este facto dever-se-á provavelmente à utilização de uma taxa de compressão variável.

F Falsa Identificação

O objectivo neste último teste foi o de procurar demonstrar que os resultados apresentados são bastante razoáveis e não inviabilizam o reconhecimento e consequente identificação correcta de uma musica. Calculou-se então o número de bits errados de uma musica relativamente a outra:

	dEUS	Orff
Muse (MP3@128Kbps)	0.499946	---
Gorillaz (GSM)	---	0.500294

Como era de esperar foram obtidas percentagens de erro muito próximas dos 50%, o que corresponde a uma situação de aleatoriedade de bits.



REFERÊNCIAS

- [1] (2002) Musicbrainz trm. musicbrainz-1.1.0.tar.gz. [Online]. Available: <ftp://ftp.musicbrainz.org/pub/musicbrainz/>
- [2] (2002) Etantrum. [Online]. Available: <http://www.freshmeat.net/projects/songprint>
- [3] (2001) Clango (TM) [Online] Available: <http://www.clango.com/download.html>
- [4] E. Batlle, P. Cano, T. Kalker and J. Haitsma, "A Review of Algorithms for Audio Fingerprinting"
- [5] J. Haitsma and T. Kalker, "A Highly Robust Audio Fingerprint System"
- [6] E. Batlle, J. Masip, and E. Guaus, "Automatic song identification in noisy broadcast audio," in *Proc. of the SIP*, Aug. 2002.
- [7] E. Batlle, H. Mayer, and H. Neuschmied, "Identification of audio titles on the internet" in *International Conference on Web Delivering of Music* (2001).
- [8] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust sound modeling for song detection in broadcast audio," in *Proc. AES 112th Int. Conv.*, Munich, Germany, May 2002.
- [9] E. Allamanche, J. Herre, O. Helmuth, B. Froba, T. Kasten, and M. Cremer, "Content-based identification of audio material using mpeg-7 low level description," in *Proc. of the Int. Symp. of Music Information Retrieval*, Indiana, USA, Oct. 2002.
- [10] J. Herre, E. Allamanche, and O. Helmuth, "Robust Matching of Audio Signals Using Spectral Flatness Features," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (2001).
- [11] J. Herre, E. Allamanche, T. Kastner, M. Cremer, W. Hirsch and O. Helmuth, "Advanced Audio Identification Using MPEG-7 Content Description"
- [12] J. Herre, E. Allamanche, B. Froba, M. Cremer and O. Helmuth, "AudioID: Towards Content-Based Identification of Audio Material"