



Sociedade de Engenharia de Áudio

Artigo de Congresso

Apresentado no 9º Congresso de Engenharia de Áudio
15ª Convenção Nacional da AES Brasil
17 a 19 de Maio de 2011, São Paulo, SP

Este artigo foi reproduzido do original final entregue pelo autor, sem edições, correções ou considerações feitas pelo comitê técnico. A AES Brasil não se responsabiliza pelo conteúdo. Outros artigos podem ser adquiridos através da Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA, www.aes.org. Informações sobre a seção Brasileira podem ser obtidas em www.aesbrasil.org. Todos os direitos são reservados. Não é permitida a reprodução total ou parcial deste artigo sem autorização expressa da AES Brasil.

Projeto de banco de filtros digitais por janelas Kaiser para identificação de notas e acordes de violão usando imagens binarizadas por limiar de Otsu

M. M. Cordeiro Jr¹, H.M. de Oliveira¹

¹Universidade Federal de Pernambuco, Departamento de Eletrônica e Sistemas.
Recife, Pernambuco, 50.740-530, Brasil.
mauriciojunior19@hotmail.com, hmo@ufpe.br

RESUMO

Neste trabalho é proposto um novo procedimento para identificação de notas e acordes em registros de violão. O método apresentado utiliza técnicas de processamento de sinais de áudio e de imagem e pode ser dividido em cinco etapas, sendo elas: decomposição em componentes de energia utilizando banco de filtros digitais por janelas Kaiser; conversão matriz-imagem seguida de uma binarização por limiar de Otsu; ajuste a partir de técnicas de conectividade entre os pixels; sobreposição temporal e composição dos resultados. Testes de funcionamento usando registros de violão (gravados via sintetizador) foram realizados, obtendo taxas de acerto superiores a 99%.

0 INTRODUÇÃO

O estudo na área de análise e reconhecimento de padrões musicais vem se intensificando nas últimas décadas, devido ao avanço tecnológico em processamento de sinais. Dispositivos DSP (*Digital Signal Processor*) apresentam cada vez maior capacidade de execução de cálculos computacionais complexos, efetuando grande número de instruções em curtos intervalos de tempo. Considera-se natural, portanto, desenvolver *softwares* que emulem o comportamento do homem, capaz de interpretar estímulos sonoros. Estudos envolvendo processamento de sinais de áudio investigam a possibilidade de descrever sons gerados a partir de instrumentos musicais utilizando modelos matemáticos, desta forma, tornando possível analisá-los e identificá-los.

Na literatura existe uma vasta quantidade de pesquisas realizadas com este propósito, cada qual, utilizando técnicas distintas, algumas das quais são enunciadas: em [1], o autor propõe uma estratégia de classificação baseada em redes neuronais *feed-forward* de múltiplas camadas, treinadas com representações frequenciais com

transformadas com Q-constante (CQT); A técnica desenvolvida em [2] também é baseada no treinamento de uma rede que extrai um vetor característico contendo as energias da nota (fundamental e seus primeiros harmônicos), sendo a rede constituída por duas camadas – a camada de segmentação e a camada de reconhecimento; em [3], com objetivo similar, aborda-se o uso da transformada rápida de *Fourier* para localização dos picos de frequência do sinal; em [4] os autores deste artigo enunciaram outra técnica utilizada para reconhecimento de notas musicais isoladas no tempo, utilizando um diagrama de estados para segmentação das mesmas e a transformada rápida de *Fourier* para o reconhecimento. Em 2006, K. Lee [5] propôs um método de reconhecimento de acordes o qual denominou de EPCP (*Enhanced Pitch Class Profile*). Em [6,7,8], os autores propuseram outro método com base em cadeias de Markov ocultas. Em [9], Harte e Sandler usaram um *cromograma* de 36 bits para encontrar o valor de ajuste de entrada de áudio, usando a distribuição das posições dos picos. Em [10], Fujishima também descreveu um método para reconhecimento de acordes, baseado no mesmo princípio seguido por Harte e Sandler.

O reconhecimento de acordes pode ser usado como uma ferramenta na elaboração de aplicativos para transcrição automática de música [11].

1 VISÃO GLOBAL DO SISTEMA

Neste trabalho é proposta uma nova abordagem para a identificação de notas e acordes em registros de violão gerados a partir da gravação em um sintetizador digital, *software*, ou por meio do próprio instrumento. O método apresentado combina técnicas de processamento de sinais de áudio e de imagem e pode ser dividido em cinco etapas: na primeira, o arquivo de áudio é processado em um banco de filtros digitais por janelas *Kaiser* [12], compostos por 84 filtros passa-faixas cujas frequências centrais são as próprias frequências das notas musicais em escala temperada (a passagem pelo banco de filtros decompõe o sinal de áudio em componentes de energia e os armazenam em uma matriz); na segunda etapa, a matriz gerada é convertida em uma imagem em tons de cinza com 256 níveis, sendo então binarizada por limiar de *Otsu* [13]; na terceira, são realizados ajustes na imagem binária utilizando técnicas de conectividade entre os pixels [13], a fim de remover todas as oscilações presentes e manter apenas os intervalos nos quais as notas/acordes permanecem fixas; na quarta, analisa-se sua sobreposição temporal, com o objetivo de distinguir entre nota isolada, notas sobrepostas ou um acorde; finalmente, na última etapa, é feita uma análise na saída expressa na etapa anterior a fim de compor em tela os acordes ou notas reconhecidas pelo sistema. Vale mencionar que o sistema foi projetado para escala temperada.

1.1 Objetivo

Desenvolver um *software* em plataforma *Matlab*¹ capaz de ler um arquivo de áudio gravado no formato *wav* o qual contém notas e/ou acordes musicais (com ou sem sobreposição temporal) e identificá-las. A proposta é introduzir uma técnica que seja eficaz tanto no tempo quanto na frequência, sem utilização de nenhuma rede neuronal, e também sem trabalhar diretamente com a transformada de *Fourier*, pois esta limita a análise ao domínio frequencial, perdendo informações temporais.

A ideia consiste em projetar um banco de filtros digitais passa-faixas que opere em toda faixa de frequência a qual o sistema estará projetado para reconhecer. Cada filtro é um “filtro passa-faixa” com largura de banda de passagem proporcional à escala temperada e com frequência central na “nota alvo”.

O banco de filtros se baseia no mesmo princípio da transformada de *Wavelets* [14], que, ao contrário da transformada de *Fourier* [15], faz uma análise com resolução adequada no domínio tempo-frequência do sinal analisado.

2 METODOLOGIA

O sistema para reconhecimento de notas e acordes musicais consiste em uma técnica desenvolvida utilizando um banco de filtros digitais projetados usando janelas de *Kaiser* capaz de identificar uma nota musical pura (DÓ-RÉ-MI-FÁ-SOL-LÁ-SI), assim como seus acidentes

musicais (sustenidos ou bemóis), sendo um total de 12 notas em qualquer escala que se repetem à medida que se altera a escala. Subir uma escala implica duplicar a frequência das notas. Também esta mesma técnica permite a identificação de notas musicais compostas (acordes musicais), que de acordo com suas combinações, podem formar uma gama deles. A técnica é vantajosa, pois faz simultaneamente uma análise nos domínios tempo-frequência, sendo análoga ao princípio da transformada que utiliza banco de filtros *wavelets*.

Na escala temperada, tem-se a seguinte classificação, conforme ilustrado na Tabela 1:

Tabela 1: Frequências Sonoras: observe que ao passar de uma nota para outra, sua frequência fundamental é multiplicada por $^{12}\sqrt{2}$.

Notas		1	2	3	4	5	6	7
0	C	32,703196	65,40639	130,8128	261,6256	523,2511	1046,502	2093,005
1	C#	34,647829	69,29566	138,5913	277,1826	554,3653	1108,731	2217,461
2	D	36,708096	73,41619	146,8324	293,6648	587,3295	1174,659	2349,318
3	D#	38,890873	77,78175	155,5635	311,127	622,254	1244,508	2489,016
4	E	41,203445	82,40689	164,8138	329,6276	659,2551	1318,51	2637,02
5	F	43,653529	87,30706	174,6141	349,2282	698,4565	1396,913	2793,826
6	F#	46,249303	92,49861	184,9972	369,9944	739,9889	1479,978	2959,955
7	G	48,99943	97,99886	195,9977	391,9954	783,9909	1567,982	3135,964
8	G#	51,913088	103,8262	207,6524	415,3047	830,6094	1661,219	3322,438
9	A	55	110	220	440	880	1760	3520
10	A#	58,270471	116,5409	233,0819	466,1638	932,3275	1864,655	3729,31
11	B	61,735413	123,4708	246,9417	493,8833	987,7666	1975,533	3951,066

Para implementação deste sistema foram utilizados arquivos no formato *wav* do tipo monofônico e gravados a uma taxa de amostragem de 8 kHz, adequada para registros produzidos por violão.

Embora outros instrumentos possam ser usados, os testes de validação foram conduzidos exclusivamente usando gravações de violão. A identificação das notas/acordes se faz em escala temperada (incluindo # e β), através de um mapeamento de identificação tonal, conforme apresentado nas Tabelas 2,3 e 4.

Tabela 2: Função identificadora de notas musicais em função das suas respectivas cifras com código representativo de cada uma delas.

Cifras	C	D	E	F	G	A	B
Notas	Dó	Ré	Mi	Fá	Sol	Lá	Si
CodMusical	0	2	4	5	7	9	11
Cifras	C#	D#		F#	G#	A#	
Acidentes	Dó#	Ré#		Fá#	Sol#	Lá#	
CodMusical	1	3		6	8	10	

Tabela 3: Exemplos de acordes básicos formados de tônica, terça e quinta. Com qualquer escala maior ou menor, forma-se o campo harmônico semelhante ao indicado na tabela anterior.

		Escala											
		Dó	Ré	Mi	Fá	Sol	Lá	Si	Dó	Ré	Mi	Fá	
A C O R D E S	C												
	Dm												
	Em												
	F												
	G												
	Am												
Bdim													

¹ Registered trademarks of the MathWorks.™

Tabela 4: Exemplos de acordes dissonantes básicos formados por cinco notas.

Acordes Dissonantes	Escala												
	Dó	Ré	Mi	Fá	Sol	Lá	Si	Dó	Ré	Mi	Fá	Sol	Lá
C7M/9	■	■	■	■	■	■	■	■	■	■	■	■	■
Dm7/9	■	■	■	■	■	■	■	■	■	■	■	■	■
Em7	■	■	■	■	■	■	■	■	■	■	■	■	■
F7M/9	■	■	■	■	■	■	■	■	■	■	■	■	■
G7/9	■	■	■	■	■	■	■	■	■	■	■	■	■
Am7	■	■	■	■	■	■	■	■	■	■	■	■	■
Bdim7	■	■	■	■	■	■	■	■	■	■	■	■	■

Uma vasta quantidade de testes foi conduzida com arquivos gerados através do *software Curso de Violão ETM 6.3²*, simulando notas e acordes produzidas por um violão (com taxa de acerto acima de 99%). Implementou-se toda a abordagem usando a plataforma *Matlab*. Como mencionado, o processo de identificação pode ser dividido em cinco etapas, conforme diagrama de blocos mostrado na Figura 1.

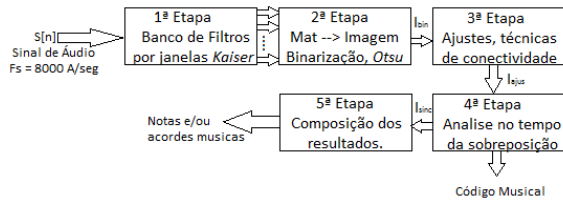


Figura 1: Diagrama de blocos do processo de reconhecimento.

2.1 Decomposição

A primeira etapa consiste em decompor o sinal digital amostrado com uma taxa de 8 kHz em 84 componentes a serem dispostas em uma matriz de tamanho $84 \times N$, em que N denota o número de amostras armazenados.

No primeiro passo desta etapa (1º bloco da Figura 1) é inicializado um banco de filtros digitais por janelas de *Kaiser*, tal como explanado a seguir:

2.1.1. Implementação do filtro digital usando janela de Kaiser

As magnitudes mais expressivas, no domínio frequencial, concentram-se em torno da condição $\omega = 0$. A atenuação (diferença entre as alturas dos lóbulos principal e secundário) é um importante para a escolha da janela. A solução adotada envolveria funções de ondas dificilmente computáveis; portanto, não é uma ideia atrativa para ser implementada. *Kaiser* [12] (1966, 1974) descobriu que uma janela quase ótima poderia ser formada usando a função de *Bessel* de ordem zero, uma função facilmente computável. A janela de *Kaiser* é definida através da relação:

$$\omega[n] = \begin{cases} I_0 \left\{ \beta \left[1 - \left(\frac{n-\alpha}{\alpha} \right)^2 \right]^{1/2} \right\}, & 0 \leq n \leq M \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

em que $\alpha = M/2$, e $I_0(\cdot)$ representa a função de *Bessel* de ordem zero do primeiro tipo [12].

A janela de *Kaiser* possui dois parâmetros: O comprimento ($M+1$) e o parâmetro de forma β .

Variando-se ($M+1$) e β , tanto o comprimento da janela como a forma podem ser ajustados. Assim, conhecendo as especificações do filtro, seja F_s (frequência de amostragem), A (Atenuação/Ganho), f_c (frequência de corte) e Δf (largura da região de transição) pode-se ajustar os parâmetros do filtro a partir das seguintes fórmulas:

$$\beta = \begin{cases} 0,1102(A - 8,7), & A > 50 \\ 0,5842(A - 21)^{0,4} + 0,07886(A - 21), & 21 \leq A \leq 50 \\ 0,0, & 21 < A. \end{cases} \quad (2)$$

$$M = \frac{A-8}{2,285 \Delta\omega}, A \geq 21, \quad (3)$$

$$\Delta\omega = \omega_p - \omega_s, \quad (4)$$

em que ω_p e ω_s são as frequências de corte (inferior e superior) do passa-faixa.

A proposta consiste em projetar um banco de filtros passa-faixas, de atenuação igual a -65 dB, utilizando janelas de *Kaiser*. Cada janela espectral deverá estar centrada numa frequência correspondente à frequência característica de uma nota musical (Dó, Dó#, Ré, ..., Si), repetindo-se até um total de sete oitavas musicais, conforme Tabela 1.

A largura espectral de cada janela não é fixa; esta deverá crescer à medida que a frequência aumenta (filtros com fatores de qualidade constantes), tendo largura correspondente a um semitom. A faixa da região de transição foi assumida constante, com largura de 20 Hz.

Como as frequências indicadas na Tabela 1 crescem em razão de uma P.G. (Progressão Geométrica) à medida que se avança um semitom, é possível definir:

$$f_N(i) := 32,703196 \cdot ({}^{13}\sqrt{2})^i \text{ Hz}. \quad (5)$$

Sendo a frequência $f_N(0) = 32,703196$ Hz correspondente ao primeiro Dó (Escala 1), tem-se:

- Frequência central de cada janela

$$f_c(i) := 32,703196 \cdot ({}^{13}\sqrt{2})^i \text{ Hz}. \quad (6)$$

- Largura espectral de cada janela

$$\delta f(i) = f_s(i) - f_i(i) = 31,7308795 \cdot [({}^{13}\sqrt{2})^{i+1} - ({}^{13}\sqrt{2})^i] \text{ Hz}. \quad (7)$$

- Frequências de corte (nota inferior)

$$f_{inf}(i) = 31,7308795 \cdot ({}^{13}\sqrt{2})^i \text{ Hz}. \quad (8)$$

- Frequências de corte (nota superior)

$$f_{sup}(i) = 31,7308795 \cdot ({}^{13}\sqrt{2})^{i+1} \text{ Hz}. \quad (9)$$

- Resposta ao impulso do filtro digital

$$h[i, n] = \frac{\text{sen}[2\pi f_c(i) \cdot (n-\alpha)]}{\pi(n-\alpha)} \cdot \omega[n]. \quad (10)$$

O código musical (#CM) e sua respectiva escala musical (#EM) podem ser vistos na Tabela 1 e calculados a partir do índice i das equações (11) e (12), usando a fórmula de recorrência:

$$\# CM = i \text{ mod } 12, \quad (11)$$

² Registrado por Airton de Igaratinga, MG, Brasil. TM

$$\# EM = \left\lfloor 1 + \frac{i}{12} \right\rfloor, \quad (12)$$

em que $\lfloor x \rfloor$ denota o maior inteiro menor do que x .

2.1.2. Construção de uma matriz utilizando filtros de Kaiser

O sinal digital de áudio pode ser representado como um vetor com N amostras de mesmo comprimento, e cada amostra está espaçada por um intervalo de 1/8000 segundos e denota-se o sinal de entrada por $S[n]$. Desta forma, pode-se construir uma matriz de tamanho $84 \times N$ efetuando-se a convolução discreta entre cada filtro $h[i,n]$ e o sinal $S[n]$. Assim, a matriz gerada assume a seguinte forma:

$$M[i, n] = h[i, n] * S[n]; 1 \leq i \leq 84, 0 \leq n \leq N - 1. \quad (13)$$

Sendo cada elemento da matriz denotado por $a_{i,n}$, em que i e n identificam à qual linha e coluna, respectivamente, pertence o elemento. As componentes de energia podem ser obtidas elevando-se cada elemento da matriz ao quadrado:

$$M_E[i, n] = (a_{i,n}^2); 1 \leq i \leq 84, 0 \leq n \leq N - 1. \quad (14)$$

2.2 Conversão Matriz - Imagem Binária

A segunda etapa do processo consiste em duas partes: primeiramente, converte-se a matriz $M_E[i,n]$ em uma *imagem digital* monocromática, com 256 níveis de tons de cinza; em seguida, faz-se uma binarização desta utilizando o limiar de *Otsu* [13], fazendo com que as componentes com alto nível de energia assumam nível lógico 1, enquanto que componentes com baixo nível de energia assumam o nível lógico 0.

Vale ressaltar que notas musicais puras devem apresentar uma grande amplitude na componente de frequência fundamental na qual elas são definidas. Acordes também devem apresentar grandes amplitudes nas respectivas frequências de suas notas. A binarização faz a separação entre as baixas e altas amplitudes.

2.2.1. Conversão da Matriz de Áudio para Imagem Monocromática

Para conversão da matriz $M_E[i,n]$ em uma imagem digital monocromática com 256 níveis de energia é utilizado um algoritmo de varredura que localiza o valor de maior energia contido nesta matriz e a partir dele, calcula o fator de conversão da seguinte forma:

$$F_{cov} = \frac{255}{\max(M_E[i,n])}. \quad (15)$$

Assim, a matriz que representa a imagem é convertida utilizando a expressão:

$$M_{Gray} = \left[F_{cov} \begin{pmatrix} [a_{1,1}^2] & \cdots & [a_{1,N}^2] \\ \vdots & \ddots & \vdots \\ [a_{84,1}^2] & \cdots & [a_{84,N}^2] \end{pmatrix} \right], \quad (16)$$

em que $\lfloor x \rfloor$ denota o inteiro mais próximo de x (arredondamento). Vale ressaltar que cada elemento desta matriz é quantizado por um número binário de 8 bits (1 byte/elemento), alocando na memória um espaço equivalente a $84 \times N$ bytes.

2.2.2. Binarização por limiar de Otsu

O processo de binarização ocorre aplicando-se a limiarização de Otsu [13] na matriz M_{Gray} .

O método de Otsu é um método de limiarização global para escolher o melhor *threshold*. Este se baseia no histograma normalizado (da imagem) como uma função de densidade de probabilidade discreta, na forma, $p_r(r_q) = m_q/m$, $q = 0, 1, 2, \dots, L - 1$; em que m é o número total de pixels na imagem, m_q é o número de pixels que tem intensidade r_q e L representa o número total de níveis de intensidade na imagem, neste caso, 256.

O método de Otsu constrói duas classes denotadas por C_0 e C_1 e escolhe para limiar um valor de k como sendo o nível de intensidade para o qual $C_0 = [0, 1, \dots, k-1]$ e $C_1 = [k, k+1, \dots, L-1]$, maximizando a variância entre classes σ_B^2 , que é definida como:

$$\sigma_B^2 := \Pi_0(\mu_0 - \mu_T)^2 + \Pi_1(\mu_1 - \mu_T)^2, \quad (17)$$

em que

$$\Pi_0 = \sum_{q=0}^{k-1} p_q(r_q), \quad (18)$$

$$\Pi_1 = \sum_{q=k}^{L-1} p_q(r_q), \quad (19)$$

$$\mu_0 = \sum_{q=0}^{k-1} \frac{qp_q(r_q)}{\Pi_0}, \quad (20)$$

$$\mu_1 = \sum_{q=k}^{L-1} \frac{qp_q(r_q)}{\Pi_1}, \quad (21)$$

$$\mu_T = \sum_{q=0}^{L-1} qp_q(r_q). \quad (22)$$

O limiar de Otsu é o valor calculado k , dessa maneira a matriz gerada assumirá a forma:

$$M_{Bin} = \begin{bmatrix} b_{1,1} & \cdots & b_{1,N} \\ \vdots & \ddots & \vdots \\ b_{84,1} & \cdots & b_{84,N} \end{bmatrix}, \quad (23)$$

em que

$$b_{i,n} = \begin{cases} 0, & \text{se } a_{i,n}^2 \leq k \\ 1, & \text{se } a_{i,n}^2 > k. \end{cases} \quad (24)$$

2.3 Conexão entre os pixels

É bastante natural esperar que cada linha da matriz binarizada apresente oscilações (alternâncias entre 0's e 1's) devido à natureza da energia instantânea do sinal analisado ser oscilatória. Uma vez que a energia instantânea do sinal é definida como sendo o módulo elevado ao quadrado do mesmo, logo, a frequência de oscilação da energia deverá ser estimada em média como sendo o dobro da frequência da nota à qual se deseja analisar.

Analisando “componente a componente” de acordo com a Tabela 1, a nota de menor frequência está em torno dos 32 Hz, ao passo que a maior por volta dos 4 kHz; isto significa que a cada um segundo podem aparecer de 64 a 8000 oscilações. Assim, sabendo que existem 8000 amostras em um segundo, pode ocorrer desde 1 nível lógico baixo para cada 1 nível lógico alto até 128 níveis lógico baixo para cada 128 níveis lógicos alto.

Na ausência de nota é assumido nível lógico baixo durante todo tempo em que esta ocorrer (cenário de baixo ruído).

A técnica de conectividade entre pixels [13] permite remover todas as oscilações presentes analisando linha a linha a matriz binária. Ela é baseada numa máscara de tamanho 1×150 ; assim, havendo uma conectividade de pixels (nível lógico alto ou baixo) de comprimento superior a 150, este é caracterizado como presença de silêncio. Caso contrário, caracteriza-se como presença de informação (nota musical) nas componentes da linha analisada.

Portanto, todo trecho do sinal que tiver oscilações que satisfaçam critério enunciado no parágrafo anterior, assumirá nível lógico alto (presença de informação), caso contrário, o nível lógico permanecerá inalterado (ausência de informação).

2.4 Sobreposição temporal

O processo de sobreposição é utilizado como critério de decisão para resolver se a sobreposição temporal entre duas ou mais notas será caracterizada como acorde ou simplesmente como notas sobrepostas. Caso o critério de decisão decida por acorde musical, este deverá ser sincronizado, caso contrário, todas as sobreposições deverão ser removidas.

Para melhor entendimento da técnica, supõe-se a ocorrência do acorde Dó maior (de acordo com a Tabela 3) a partir de t_1 e das notas isoladas Ré, Mi e Sol a partir de t_4 , conforme Figura 2.

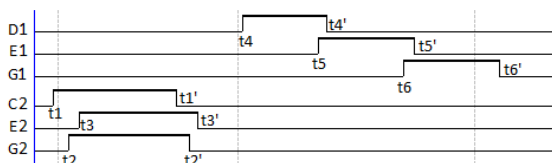


Figura 2: Exemplo ilustrativo de identificação de notas/acordes em trecho de um sinal qualquer.

Para o exemplo, o algoritmo desenvolvido visa analisar as sobreposições ocorridas entre C2, E2 e G2 e entre D1, E1 e G1 a fim de classificar estas como um acorde musical ou meramente como um conjunto de notas sobrepostas. A técnica utilizada se baseia em analisar a largura das notas sobrepostas. Isto é feito efetuando-se um AND lógico entre elas, como segue:

Na escala 1, tem-se:

$$\#1 = D1 \text{ and } E1 \text{ and } G1 = 0, \forall t$$

Na escala 2, tem-se:

$$\#2 = C2 \text{ and } E2 \text{ and } G2 = \begin{cases} 1, & \text{se } t_3 \leq t \leq t_1' \\ 0, & \text{caso contrário.} \end{cases}$$

Desta forma, as três notas presentes na escala 1 formam um conjunto de notas puras e sobrepostas; estas deverão ter suas sobreposições eliminadas, ao passo que as três notas presentes na escala 2 apresentam largura de sobreposição igual à diferença $t_1' - t_3$, que se for superior a um determinado limiar e ainda satisfizer o critério de formação dos acordes (Tabela 3), a mesma deverá ser classificada como um acorde musical e então sincronizada, conforme ilustra a Figura 3.

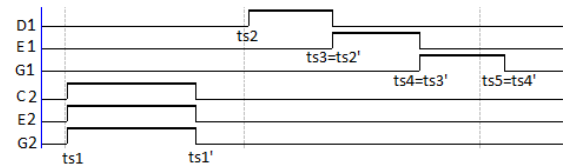


Figura 3: Exemplo de identificação de notas/acordes em trecho de um sinal qualquer sincronizado e ajustado.

De uma forma geral, para m sobreposições encontradas, assumindo que o método somente identifica acordes formados por 3, 5 ou 7 notas, as m sobreposições devem ser combinadas de acordo com o seguinte critério:

- Caso $m \geq 7$, realiza-se a função “and” em todas as combinações m tomada 7 a 7, 5 a 5 e 3 a 3, a fim de identificar algum acorde presente entre elas.
- Caso $5 \leq m < 7$, realiza-se a função “and” em todas as combinações m tomada 5 a 5 e 3 a 3 possíveis.
- Caso $3 \leq m < 5$, realiza-se a função “and” em todas as combinações m tomada 3 a 3 possíveis.
- Caso $m < 3$, não há acorde neste trecho de sobreposição, apenas a presença de notas puras.

2.5 Composição dos resultados

A 5ª e última etapa deste processo é a mais simples de todas. Ela é responsável pela composição dos resultados obtidos na saída da etapa anterior (cf. Figura 1), a fim de exibir ao usuário final um resultado mais amigável.

Pela 4ª etapa, sabe-se que a matriz final sincronizada apresenta níveis lógicos altos em instantes de tempo (colunas) onde a informação está presente e o número da linha da matriz informa qual nota está ali presente, inclusive sua escala musical, conforme identificado nas “equações 11 e 12”. A partir de então, pode-se formar os resultados a serem exibidos em tela.

No exemplo ilustrado na Figura 2, chega-se à Figura 3, e a partir dela podem-se inferir facilmente os resultados, assim sendo:

$$\text{Cifras} = \begin{cases} \text{Acorde Dó Maior,} & \text{quando } t_{s1} \leq t \leq t_{s1}'; \\ \text{Nota Ré}_1, & \text{quando } t_{s2} \leq t \leq t_{s2}'; \\ \text{Nota Mi}_1, & \text{quando } t_{s3} \leq t \leq t_{s3}'; \\ \text{Nota Sol}_1, & \text{quando } t_{s4} \leq t \leq t_{s4}'; \\ \text{Ausência de notas / acordes, caso contrário.} \end{cases}$$

3 CONCLUSÕES

Este trabalho introduz uma nova técnica para a detecção e reconhecimento de notas ou acordes musicais (com ou sem sobreposição temporal) produzidas por um violão e descreve a implementação de um método inovador em relação àquele publicado em [4] pelos mesmos autores. Vários métodos introduzidos podem ser adaptados para uso

em cenários como classificação de instrumentos e transcrição automática. [11, 16].

A técnica foi inteiramente implementada e simulada em *Matlab*, e foram realizados cinquenta testes, com duração entre 1” a 35”, resultando em taxa de acerto superior a 99% e precisão temporal em torno de 250 ms. O sistema conduz uma alternativa atrativa para aprendizado e serve como base para afinação de violão. A extensão para outros instrumentos parece ser relativamente simples. De fato, os testes incluíram acordes produzidos por outros instrumentos (e.g., piano, cello) sem degradar a taxa de acerto.

4 REFERÊNCIAS BIBLIOGRÁFICAS

[1] A.L. Szczupack e L.W.P. Biscainho, “Identificação de notas musicais em registros de violão solo” em anais do 7º Congresso de Engenharia e Áudio, AES Brasil 2009, pp. 26 – 33.

[2] A.S. Ogasawara, “Reconhecedor de notas musicais em sons polifônicos”, Departamento de Eletrônica e Computação, UFRJ, Abril de 2008.

[3] S.A.S. Ferreira, “Sistema especialista para reconhecimento de acordes musicais em tempo real para violão elétrico utilizando técnicas de DSP”, Dissertação de Mestrado, UFBA, Dezembro de 2006.

[4] M.M. Cordeiro Jr e H.M. de Oliveira, “Sistema de Reconhecimento Automatizado de Notas Musicais Isoladas” em anais XXXII CNMAC, 2009, Vol. 2, Mato Grosso, Brasil, pp. 1038 – 1039.

[5] K. Lee, “Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile”, in *proc. of the International Computer Music Conference*, 8p., 2006.

[6] K. Lee and M. Slaney, “Automatic chord recognition from audio using an HMM with supervised learning”, *Proc. ISMIR*, 5p., 2006.

[7] K. Lee and M. Slaney, “Automatic chord recognition from audio using a supervised HMM trained with audio-symbolic data”, in *proc. of the 1st ACM workshop on Audio and music computing multimedia*, pp.11–19, 2006.

[8] H-T. Cheng et al., “Automatic chord recognition for music classification and retrieval”, in *IEEE International Conference on Multimedia and Expo*, pp. 1505–1508, 2008 doi 10.1109/ICME.2008.4607732.

[9] C. Harte and M. Sandler, “Automatic chord identification using a quantised chromagram”, AES, in *Proceedings of the Audio Engineering Society*. Spain: Audio Engineering Society, May 2005.

[10] T. Fujishima, “Realtime chord recognition of musical sound: A system using Common Lisp Music”, In *Proc. of the Int. Computer Music Conf.*, Beijing: International Computer Music Association, 1999.

[11] T.F. Tavares, J.G.A. Barbedo and A. Lopes, “Towards the evaluation of automatic transcription of music” em anais do 6º Congresso de Engenharia e Áudio, AES Brasil 2008, pp. 96 – 99.

[12] A.V. Oppenheim, R.W. Schaffer, with J.R. Buck, “Filter Design Techniques” em *Discrete-Time Signal Processing*, 2ª Ed. Upper Saddle River: Prentice Hall, 1999, cap. 7, seção 7.2.3, pp. 474 – 477.

[13] R.P. Gonzales e R.C. Woods, *Processamento Digital de Imagens*, 3ª Ed. São Paulo: Prentice Hall, 2010.

[14] H.M. de Oliveira, “Wavelets: Uma Evolução na Representação de Sinais” em *Análise de Sinais para Engenheiros - uma Abordagem via Wavelets*, 1ª Ed. Rio de Janeiro, Brasil: Brasport, 2007.

[15] H. M. de Oliveira, “Espectro de Sinais” em *Análise de Fourier e Wavelets: Sinais Estacionários e não Estacionários*, 1ª Ed., Recife, Brasil: Editora Universitária da UFPE, 2007.

[16] J.G.A. Barbedo and A. Lopes, “Short-Term Classification of Musical Instruments: A Critical View” em anais do 6º Congresso de Engenharia e Áudio, AES Brasil 2008, pp. 100 – 107.