

Universidade de São Paulo
Instituto de Matemática e Estatística

Trabalho de Conclusão de Curso

**Técnicas Computacionais para a Recuperação
de Informação Musical em Repertórios
Indígenas Brasileiros**

Aluna: Bruna Bazaluk M V
Orientador: Prof. Marcelo Queiroz

Dezembro 2022

Agradecimentos

Gostaria de agradecer primeiramente ao PUB-USP (Programa Unificado de Bolsas da USP), que me financiou durante parte deste projeto, enquanto era minha Iniciação Científica.

Segundo, mas não menos importante, meu orientador, Prof. Marcelo Queiroz, que aceitou trabalhar comigo esse tema tão interessante que é a cultura dos povos originários brasileiros. Além disso, agradeço por nossas reuniões semanais, que me ensinaram tanto e me direcionaram não somente neste projeto mas também na continuidade da minha vida acadêmica.

Agradeço também aos meus amigos do LIAMF, Sub e Julissa, por ouvirem meus ensaios para as apresentações deste trabalho e sempre estarem prontos pra me ajudar com críticas construtivas (também pizzas e pique-niques para descansar de vez em quando).

Também gostaria de agradecer profundamente a Erikinha por estar do meu lado desde sempre; sem seu apoio não teria mudado de curso e não estaria aqui fazendo este trabalho.

Re, obrigada por tudo, principalmente por não me deixar desistir nem quando eu estava mais desanimada. Agradeço por estar sempre ao meu lado, ajudando-me com sugestões, críticas, elogios e me dando todo o apoio possível para que eu terminasse este trabalho. Mesmo estando em silêncio, o simples fato de estar ao meu lado enquanto eu trabalhava já me dava segurança, obrigada.

Finalmente, agradeço a minha família, Paulo, Carla, Pedro e Rita, por nunca duvidarem de mim, nem mesmo quando eu duvidava, e sempre apoiarem meus trabalhos com entusiasmo (mesmo quando não entendem do assunto). Amo muito vocês.

Resumo

Certos estilos musicais apresentam padrões rítmicos característicos que podem ser identificados por qualquer ouvinte. Na transcrição de tais padrões, frequentemente são realizadas aproximações das durações a fim de simplificar a representação simbólica, principalmente no caso da notação musical padrão. Por isso, é frequente encontrarmos pequenas discrepâncias entre a representação simbólica e as realizações musicais, e quando tais discrepâncias são observadas de forma sistemática, é possível investigar certas características do estilo musical, tais como a expressividade, frequentemente inacessíveis através da notação simbólica. A área de estudo que investiga estas questões é denominada microtiming (microtemporalidade) [8]. Neste trabalho, faremos uma análise de padrões rítmicos similares seguindo a metodologia de Fabien Gouyon em seu artigo “Microtiming in ‘Samba de Roda’ — Preliminary experiments with polyphonic audio” [8]. O objetivo deste trabalho é investigar duas peças do repertório Parixara, um estilo de música e dança que está presente nas culturas indígenas roraimenses Wapichana e Macuxi [6], e analisar as variações em “microtiming” dos padrões rítmicos encontrados na partitura. Nossas conclusões foram que a metodologia original tem sucesso bastante limitado neste repertório, que outros caminhos de investigação usando *clusterização* de onsets são plausíveis, e que a sonoridade longa e reiterada do chocalho influenciou os resultados e dificultou a análise.

Palavras-chave: música indígena, cultura indígena, macuxi, wapichana, povos originários, computação musical, microtemporalidade, MIR.

Conteúdo

1	Introdução	1
1.1	Motivações	1
1.2	Objetivos	1
2	Fundamentação Teórica	3
2.1	Contextualização Cultural	3
2.2	Processamento de Sinais Musicais	5
2.2.1	Detecção de Onsets	5
2.2.2	Funções de Novidade	5
2.2.3	Separação de Fontes	6
2.2.4	Filtro e Clusterização de Picos	6
3	Experimentos	8
3.1	Dados	8
3.2	Pré-processamento	8
3.3	Extração de Características	9
3.3.1	Funções de Detecção de Onsets	9
3.3.2	Segmentação de Compassos	10
3.3.3	Grupos de Padrões Rítmicos	11
3.3.4	Reamostragem	13
3.4	Sumarização das Funções de Novidade	14
3.5	Clusterização de Onsets	20
3.5.1	Seleção de Picos	20
3.5.2	Sumarização por Centroides	21
4	Conclusões	27
	Referências	28
	Apêndices	30
	Anexos	33

1 Introdução

No presente trabalho serão analisadas duas peças do estilo Parixara, um estilo de música e dança proveniente dos povos Macuxi e Wapichana [6] do norte de Roraima. O foco das análises é o estudo de variações de microtemporalidades para comparar registros em áudio e partituras existentes.

1.1 Motivações

O Brasil é um país que nasceu às custas de sua população nativa [13]. Alguns povos foram dizimados, muitos já nem existem mais e ainda assim muitos povos indígenas influenciam a rica cultura brasileira. A valorização das culturas indígenas brasileiras se mostra cada vez mais necessária, principalmente neste momento em que diversos conflitos são frequentes em suas terras [15].

Por outro lado, a maior parte das pesquisas da área de Computação Musical concentradas em técnicas de Music Information Retrieval (MIR) são direcionadas aos repertórios mais populares como música pop, rock, jazz, etc. Portanto, é de extrema importância que existam trabalhos focados em outros estilos para verificar que os diversos algoritmos focados em obter diferentes informações musicais a partir de arquivos de áudio não sejam viesados [20].

No caso deste trabalho, dentre os repertórios não populares, a escolha de um estilo de uma cultura nativa brasileira se deu pela vontade de difundir e homenagear as culturas indígenas do nosso país. Inicialmente, a escolha de um estilo específico foi difícil; o estilo musical precisaria ser bem estabelecido e já ter informações que o descrevessem e caracterizassem, bem como contar com gravações de áudio com o mínimo possível de ruído. Nesta jornada, foram encontrados alguns projetos interessantes de gravações de músicas de culturas indígenas, como o Projeto Música das Cachoeiras¹, que concentra gravações de músicas de comunidades indígenas desde o Alto do Rio Negro, no Amazonas, até o Monte Roraima, em Roraima. Através desse projeto, conhecemos o Parixara, estilo musical escolhido para ser analisado neste trabalho, e no site do projeto Pantan Pia' [7] encontramos gravações acompanhadas de suas transcrições musicais.

1.2 Objetivos

Diferentes estilos musicais têm padrões rítmicos diferentes que podem muitas vezes ser identificados por qualquer ouvinte. Na transcrição de tais padrões, frequentemente são realizadas aproximações (quantizações) das durações a fim de simplificar a representação simbólica, principalmente no caso na notação musical padrão (partitura), predominantemente baseada em subdivisões temporais binárias e ternárias. Por isso, é frequente encontrarmos pequenas discrepâncias entre as durações na representação simbólica e em realizações/performances musicais, e quando tais discrepâncias são observadas de forma sistemática, é

¹Site do projeto: <http://www.musicadascachoeiras.com.br/>. Perfil com o CD disponibilizado livremente: <https://soundcloud.com/musicadascachoeiras>.

possível investigar características do estilo musical, tais como a expressividade, frequentemente inacessíveis através da notação simbólica. A área de estudo que investiga estas questões no contexto rítmico-temporal é denominada *microtiming* (microtemporalidades) [8].

Neste trabalho, faremos uma análise por compassos de padrões similares seguindo o procedimento feito por Fabien Gouyon em seu artigo “*Microtiming in ‘Samba de Roda’ — Preliminary experiments with polyphonic audio*” [8]. O objetivo do nosso trabalho é reproduzir os experimentos do artigo citado acima no repertório Parixara e analisar as variações em “microtiming” de cada tipo de compasso, a fim de buscar estabelecer correspondências entre os registros em áudio e as partituras existentes.

sua cultura [19].

O Parixara é dançado em pares ou trios enfileirados³ com movimentos simples. Os dançarinos cantam, acompanhados de um chocalho comprido que faz marcações batendo no chão, músicas que falam sobre atividades cotidianas, a natureza ou os animais. No passado, as apresentações também contavam com outros instrumentos, principalmente de sopro [10], mas hoje em dia o instrumento principal é o próprio chocalho [6].



Figura 2: Tukúí Grupo de Dança Parixara Macuxi, da comunidade de Boqueirão em Alto Alegre (Roraima). Extraído de: <https://www.youtube.com/watch?v=ZqLgX6PwSU&t=1s>

A dança é um elemento essencial do Parixara, pois os passos dos dançarinos definem o andamento da música [19] que é acompanhada pelo chocalho.

Tradicionalmente, Parixaras não eram transcritos, mas sim passados oralmente atravessando gerações [19] (como a maior parte dos estilos musicais de povos indígenas brasileiros), porém, diversos projetos de transcrição surgiram ao longo dos anos com o objetivo de preservar a memória e estudar a música dos povos originários. Neste trabalho, as partituras utilizadas na análise foram transcritas no contexto do projeto Panton Pia' [7].

Uma das referências para este trabalho foi a dissertação de mestrado de Jucicleide Pereira Mendonça dos Santos [19], uma pesquisadora Wapichana que ajudou na transcrição de diversas partituras no projeto Panton Pia' [7].

³No link a seguir é possível apreciar a apresentação do Tukúí Grupo de Dança Parixara Macuxi, da comunidade de Boqueirão em Alto Alegre (Roraima): <https://www.youtube.com/watch?v=ZqLgX6PwSU&t=1s>. Este vídeo faz parte do projeto Música das Cachoeiras, que também realizou gravações de áudios de outros Parixaras, inclusive Wapichana.

2.2 Processamento de Sinais Musicais

Para desenvolver este trabalho, alguns conceitos de processamento de sinais musicais foram estudados e aplicados; tais conceitos serão definidos a seguir.

2.2.1 Detecção de Onsets

Onset é uma palavra utilizada para definir o momento em que um evento musical se inicia, geralmente associado a alguma variação abrupta no sinal de áudio, como por exemplo um ataque (início) de uma nota, uma batida percussiva, etc.

Desse modo, detectar os *onsets* de uma música significa detectar os momentos em que há ataques percussivos ou inícios de notas musicais específicas.

Para fazer a detecção, são computadas a partir do sinal de áudio algumas funções específicas, chamadas ODFs (Onset Detection Function) ou funções de novidade (*novelty functions*) [12], a partir das quais os *onsets* são estimados como os picos da função.

2.2.2 Funções de Novidade

O ataque de uma nota normalmente coincide com o aumento de energia do sinal de áudio [12], portanto, algumas funções de novidade são baseadas em medir a energia local do sinal, como a amplitude RMS (*Root Mean Square*), por exemplo.

Root Mean Square (RMS)

A amplitude RMS de um conjunto de n valores $\{x_1, x_2, \dots, x_n\}$ é

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}. \quad (1)$$

No caso da biblioteca em python utilizada neste trabalho [5], todas as funções foram calculadas no domínio da frequência. Portanto, utilizando a identidade de Parseval [3], podemos derivar a função aplicada pela biblioteca. Seja $X[m] = FFT\{x[n]\}$ e N o tamanho da amostra; então

$$\begin{aligned} \sum_{n=1}^N x^2[n] &= \frac{1}{N} \sum_{m=1}^N |X[m]|^2 \text{(Parseval)} \implies \\ RMS\{x[n]\} &= \sqrt{\frac{1}{N} \sum_{n=1}^N x^2[n]} = \sqrt{\frac{1}{N^2} \sum_{m=1}^N |X[m]|^2} \end{aligned}$$

e portanto

$$RMS\{x[n]\} = \frac{1}{N} \sqrt{\sum_{m=1}^N |X[m]|^2}. \quad (2)$$

High Frequency Content (HFC)

Outra função que, de certa forma, mede a energia do sinal é a *High Frequency Content*. Esta função caracteriza a quantidade de altas frequências no sinal de áudio.

Sua definição é a seguinte [11]: se $X[i] = FFT\{x[n]\}$ e N é o tamanho da amostra, então

$$HFC = \sum_{i=0}^N i |X(i)|^2. \quad (3)$$

Complex Spectral Difference

A função de novidade utilizada por Fabien Gouyon em seu artigo [8] é a *Complex Spectral Difference*. Esta função combina a abordagem baseada na quantidade de energia com informações referentes à fase do sinal; enquanto a primeira favorece *onsets* percussivos, a segunda enfatiza *onsets* “tonais” [2].

A CSD (*Complex Spectral Difference*) é definida da seguinte forma [12]: seja $X[n] = FFT\{x[n]\}$, N o tamanho da amostra e $\varphi(n)$ a fase, então

$$\hat{X}[n+1] = |X[n]| \exp(2\pi i(\varphi(n) + \varphi'(n)))$$

e

$$CSD = |\hat{X}[n+1] - X[n+1]|. \quad (4)$$

2.2.3 Separação de Fontes

Como nosso objetivo é comparar a rítmica da melodia vocal na partitura com o áudio da performance, nossa primeira ideia foi separar as fontes de áudio para que trabalhássemos somente com a melodia vocal, sem a percussão do chocalho. Para isso, utilizamos o *Spleeter* [9], uma ferramenta para separação de fontes construída a partir de modelos pré-treinados. Esses modelos são *U-nets*, que é uma Rede Neural Convolucional (CNN) do tipo *encoder/decoder* [9]. A ferramenta foi treinada principalmente utilizando o *Bean Dataset* [16], que é composto majoritariamente por canções do estilo pop e rock.

2.2.4 Filtro e Clusterização de Picos

Com o objetivo de diminuir ruídos nas funções de novidade e *clusterizar* somente os picos (que representariam *onsets*) mais claros, uma função de filtragem foi necessária⁴.

Clusterização é o agrupamento de pontos em um plano cartesiano, ou espaço multi-dimensional. No caso do presente trabalho, a intenção era considerar os picos, após a filtragem, como pontos e agrupá-los em uma quantidade específica de grupos (*clusters*) para então definir o centroide de cada *cluster*, ou seja, o ponto médio de cada grupo de pontos.

⁴Mais detalhes sobre a função na página 32 do presente trabalho.

O algoritmo utilizado para encontrar os centroides foi o *KMeans* implementado na biblioteca em Python *scikit-learn* [14]. Este algoritmo tem como *input* os pontos a serem agrupados e a quantidade de *clusters* que devem ser definidos, assim, ele agrupa os dados separando amostras em n grupos buscando minimizar a variância intra-*cluster*, tentando minimizar a inércia, definida como

$$\text{Inércia} = \sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2),$$

onde C é o conjunto de *clusters* e μ_j a média das amostras do *cluster* j , ou seja, o centroide [1].

3 Experimentos

Como nosso objetivo é, resumidamente, comparar a rítmica da performance com as notas da partitura, primeiramente precisamos encontrar e separar cada compasso dentro da música. Com os compassos divididos, aplicaremos as funções de novidade para fazer as análises.

3.1 Dados

Os experimentos foram realizados em 2 Parixaras diferentes⁵, divulgados pelo projeto Panton Pia', que “registra e analisa narrativas orais de indígenas de Roraima” [7]. Ambos são Parixaras Macuxi interpretados por Manaaka e Yauyo. A escolha das músicas se deve principalmente ao fato das peças possuírem uma transcrição em partitura.

Os áudios são arquivos *mp3*, amostrados à frequência de 44.1 KHz, em um só canal (mono). Os dois parixaras têm somente 2 vocais e um chocalho; além disso, o Parixara 1 tem uma poesia em Macuxi no início, que foi cortada manualmente antes de realizarmos as análises de áudio.

No Parixara 1, por ter uma fórmula de compasso 6/8, o chocalho aparece duas vezes em cada compasso, marcando o início e a metade do compasso; já no Parixara 2, que é 2/4, o chocalho sempre marca o início do compasso.

3.2 Pré-processamento

Inicialmente, pensamos em utilizar uma ferramenta de separação de fontes para trabalhar apenas com a parte vocal do áudio; no caso, utilizamos o *Spleeter* [9], considerado estado-da-arte nessa tarefa. Porém, ao aplicar a ferramenta, percebemos que ainda havia uma grande presença do chocalho, apenas levemente abafada⁶, que continuou gerando ruído nas análises.

Acreditamos que essa ferramenta, apesar de ser considerada estado-da-arte, não trouxe os resultados desejados, pois foi treinada em gêneros musicais como pop e rock, que diferem significativamente do repertório aqui investigado. Essa observação se relaciona com a afirmação de Xavier Serra, que diz que não é comum a inclusão e investigação de repertórios não-populares na área de computação musical [20].

Todas as análises, portanto, foram feitas nos áudios originais, contendo chocalho e melodias vocais.

⁵Ambas as partituras se encontram nos anexos deste trabalho e podem ser ouvidas aqui: Parixara 1, Parixara 2.

⁶É possível escutar o resultado da separação dos vocais pelo Spleeter aqui: Parixara 1 (vocal), Parixara 2 (vocal).

3.3 Extração de Características

3.3.1 Funções de Detecção de Onsets

Como parte do nosso objetivo é detectar os *onsets* que representam o início de cada nota, aplicamos três ODFs diferentes: a RMS, que mede a energia total do sinal; a HFC, que está relacionada com o centroide espectral e prioriza a energia contida nas componentes de alta frequência do sinal; e a *Complex Spectral Difference*, que combina tanto a energia quanto informações referentes à fase do sinal. As duas primeiras foram escolhidas por serem relativamente simples e frequentemente utilizadas no contexto de detecção de *onsets*, enquanto a última foi utilizada por Fabien Gouyon [8] no artigo que inspirou este trabalho.

Utilizando a biblioteca Essentia [5], aplicamos as ODFs primeiramente no sinal de áudio inteiro, e em um segundo momento, em cada compasso separadamente. Os parâmetros definidos para a aplicação das funções foram os mesmos que Fabien Gouyon [8] utilizou em seu trabalho, e são valores comuns para esse tipo de análise: o tamanho de cada *frame* é 1024 e o *hop size* é 512. Isso significa que a função será calculada nas primeiras 1024 amostras do sinal e depois novamente nas 1024 amostras a partir do índice 512, seguindo assim até o final do vetor de áudio.

Abaixo seguem os gráficos referentes às funções de novidade aplicadas aos arquivos de áudio do Parixara 1 e do Parixara 2; os picos mais altos representam os *onsets* relativos às batidas do chocalho.

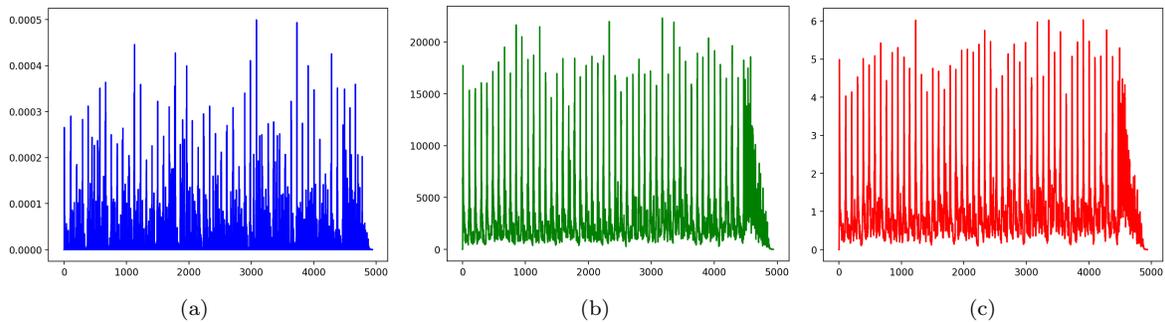


Figura 3: Gráficos das funções de novidade aplicadas ao sinal de áudio do Parixara 1. (a) RMS, (b) HFC, (c) Complex Spectral Difference.

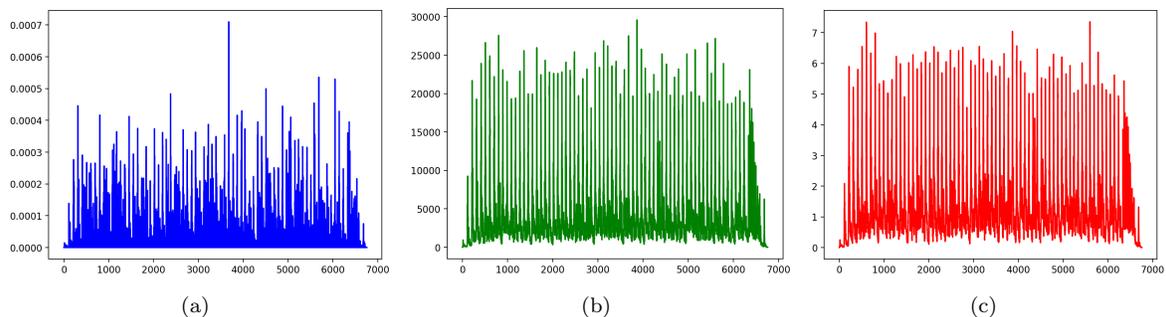


Figura 4: Gráficos das funções de novidade aplicadas ao sinal de áudio do Parixara 2. (a) RMS, (b) HFC, (c) Complex Spectral Difference.

3.3.2 Segmentação de Compassos

O próximo passo é segmentar o áudio por compassos, para encontrar o início de cada compasso. Inicialmente, utilizamos o algoritmo “*Rhythm – Onsets*” [22], que faz parte da coleção de plugins Vamp do software Sonic Visualiser [4] e marca somente os *onsets* percussivos. Após a aplicação do algoritmo, é feito um ajuste manual para garantir que os onsets marcados correspondem aos toques do chocalho. Abaixo segue como exemplo a visualização de um pedaço do sinal de áudio do Parixara 1 (em azul) com as marcações dos *onsets* encontrados pelo algoritmo e ajustados manualmente (em vermelho):

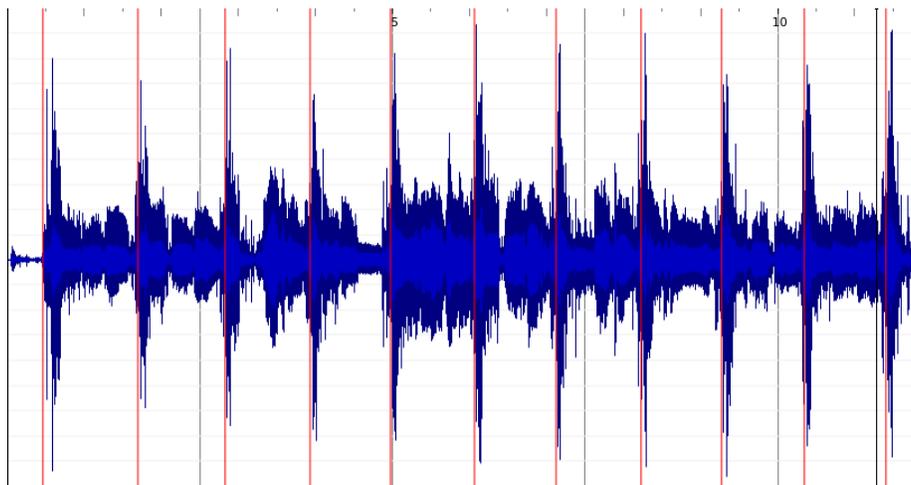


Figura 5: Uma parte do sinal de áudio do Parixara 1 visto no Sonic Visualiser. Sinal em azul e marcações dos *onsets* referentes ao chocalho em vermelho.

Porém, como o algoritmo marca o exato momento antes do chocalho bater, todo o som do chocalho estaria dentro do compasso a ser analisado, o que gerou ruídos demais na análise. Portanto, uma função simples (chamada `filtro_picos`⁷) foi escrita para definir o instante mais forte do ataque do chocalho (o máximo do pico da função de novidade ao invés do seu início), utilizando a ODF HFC (*High Frequency Content*), por ser a mais homogênea em relação ao tamanho dos picos.

Abaixo segue como exemplo a visualização de um pedaço da HFC do Parixara 1 (em azul) com as marcações dos *onsets* referentes ao chocalho encontrados pela função `filtro_picos` (em vermelho).

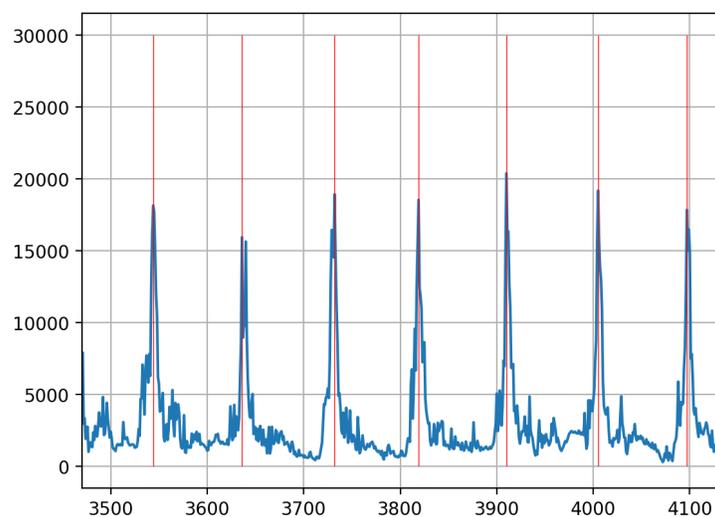


Figura 6: Uma parte da HFC do sinal de áudio do Parixara 1 em azul e as marcações dos *onsets* referentes ao chocalho em vermelho.

3.3.3 Grupos de Padrões Rítmicos

Com o objetivo de comparar os padrões rítmicos de cada peça, após dividir o sinal em compassos, estes foram agrupados. Os compassos do Parixara 1 foram divididos em 5 grupos principais⁸, e os do Parixara 2 em 3 grupos. Cada grupo é formado pelos compassos que apresentam o mesmo padrão rítmico.

Na Figura 7 abaixo podem ser vistos os grupos de compassos do Parixara 1.

⁷A função `filtro_picos` se encontra no final deste trabalho no apêndice 1, página 30.

⁸Alguns compassos aparecem apenas 3 vezes durante a música toda e portanto não foram considerados para análise.

Grupo 1

Grupo 2

Grupo 3

(a) (b) (c)

Grupo 4

Grupo 5

(d) (e)

Detailed description: This figure shows five groups of musical notation for Parixara 1. Each group consists of a treble staff and a bass staff. Grupo 1 (a) has a treble staff with a sequence of eighth notes and a bass staff with a sequence of eighth notes. Grupo 2 (b) has a treble staff with eighth notes and a bass staff with eighth notes and a dotted quarter note. Grupo 3 (c) has a treble staff with eighth notes and a bass staff with eighth notes and a dotted quarter note. Grupo 4 (d) has a treble staff with eighth notes and a bass staff with eighth notes and a dotted quarter note. Grupo 5 (e) has a treble staff with eighth notes and a bass staff with eighth notes and a dotted quarter note.

Figura 7: Padrões de compasso do Parixara 1.

Na Figura 8 abaixo podem ser vistos os grupos de compassos do Parixara 2.

Grupo 1

Grupo 2

(a) (b)

Grupo 3

(c)

Detailed description: This figure shows three groups of musical notation for Parixara 2. Each group consists of a treble staff and a bass staff. Grupo 1 (a) has a treble staff with a sequence of eighth notes and a bass staff with a sequence of eighth notes. Grupo 2 (b) has a treble staff with a sequence of eighth notes and a bass staff with a sequence of eighth notes. Grupo 3 (c) has a treble staff with a sequence of eighth notes and a bass staff with a sequence of eighth notes.

Figura 8: Padrões de compasso do Parixara 2.

3.3.4 Reamostragem

Ambas as canções têm um pulso quase constante, mas há pequenas diferenças de duração entre os compassos, o que nos impediria de comparar com acurácia os padrões rítmicos dentro de cada grupo. Para exemplificar, Na Figura 9 abaixo está o gráfico da *Complex Spectral Difference* de 2 compassos do mesmo grupo do Parixara 2. Podemos ver que o compasso representado em vermelho acaba antes do compasso em azul, assim não conseguiríamos fazer uma comparação direta a partir destas duas representações.

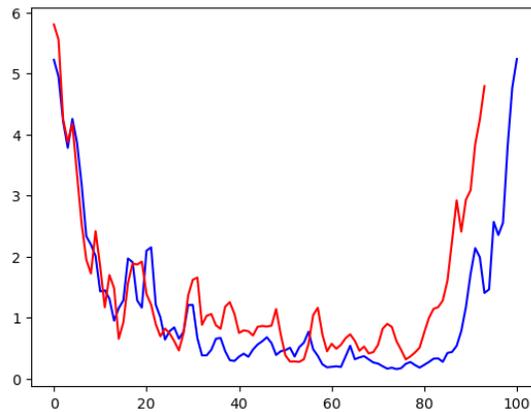


Figura 9: Gráfico da *Complex Spectral Difference* de 2 compassos do Parixara 2.

Para resolver esta dificuldade, adotamos a mesma abordagem de Gouyon [8] e fizemos uma reamostragem para que todas as representações associadas a compassos da mesma peça tivessem exatamente 100 amostras. Nossa função de reamostragem⁹ utiliza a biblioteca *scipy* [21] para fazer uma interpolação cúbica em cada trecho correspondente a um compasso, para todos os compassos e para as três funções de novidade. Na Figura 10 abaixo podem ser vistos os gráficos reamostrados da *Complex Spectral Difference* dos mesmos 2 compassos do Parixara 2. Nesta representação em que ambos terminam juntos faria sentido localizar os onsets medidos em tempos relativos ao início e fim do compasso.

⁹Esta função está no fim deste trabalho, no apêndice 2, página 31.

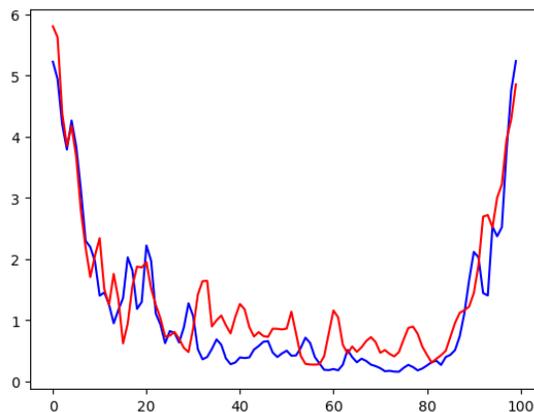


Figura 10: Gráfico da *Complex Spectral Difference* de 2 compassos reamostrados do Parixara 2.

3.4 Sumarização das Funções de Novidade

Seguindo a metodologia de Fabien Gouyon [8], em busca dos picos das funções de novidade que representassem as notas de cada grupo de compassos, utilizamos todos os gráficos reamostrados de cada compasso daquele grupo, separadamente para cada grupo e para cada uma das três ODFs, a *Complex Spectral Difference*, a HFC e a RMS.

Para cada grupo, foi gerada uma figura com os gráficos de todos os compassos pertencentes a ele, bem como com o padrão médio¹⁰, onde esperávamos encontrar os picos referentes aos ataques (*onsets*) das notas do compasso correspondente, a partir dos resultados obtidos por Fabien Gouyon em sua análise do estilo “Samba de roda” [8].

Abaixo seguem as imagens dos padrões médios referentes ao Parixara 1. Como no Parixara 1 o chocalho é batido para marcar o início e a metade do compasso, o grande pico na metade do gráfico e os picos no início e no fim sempre representam o chocalho.

¹⁰Os padrões médios são as médias simples das funções de novidade reamostradas de cada grupo.

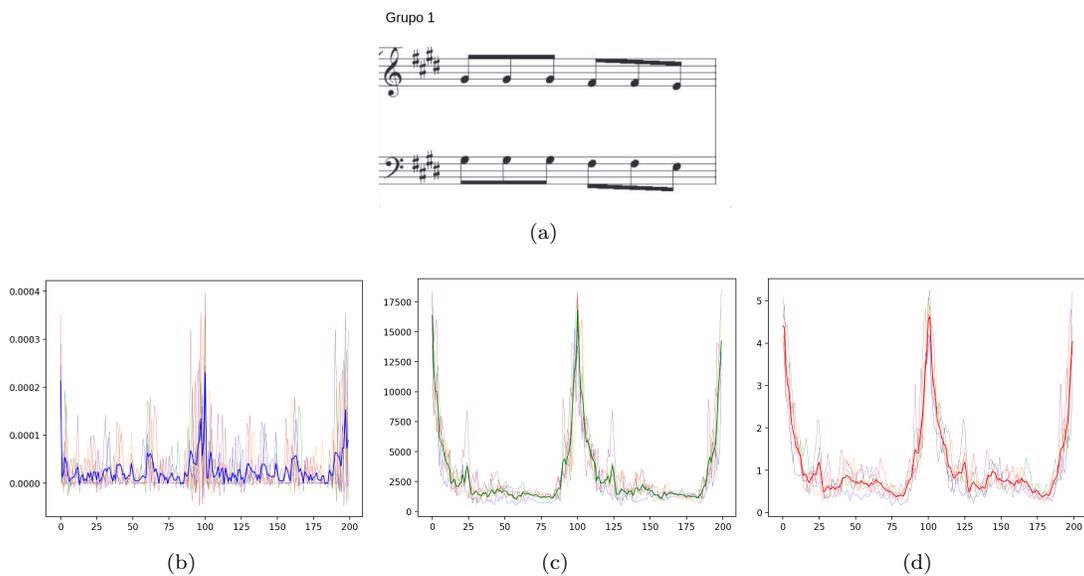


Figura 11: Padrões médios referentes ao Grupo 1 do Parixara 1. (a) Grupo de compasso número 1, (b) Função RMS média, (c) Função HFC média, (d) Função *Complex Spectral Difference* média.

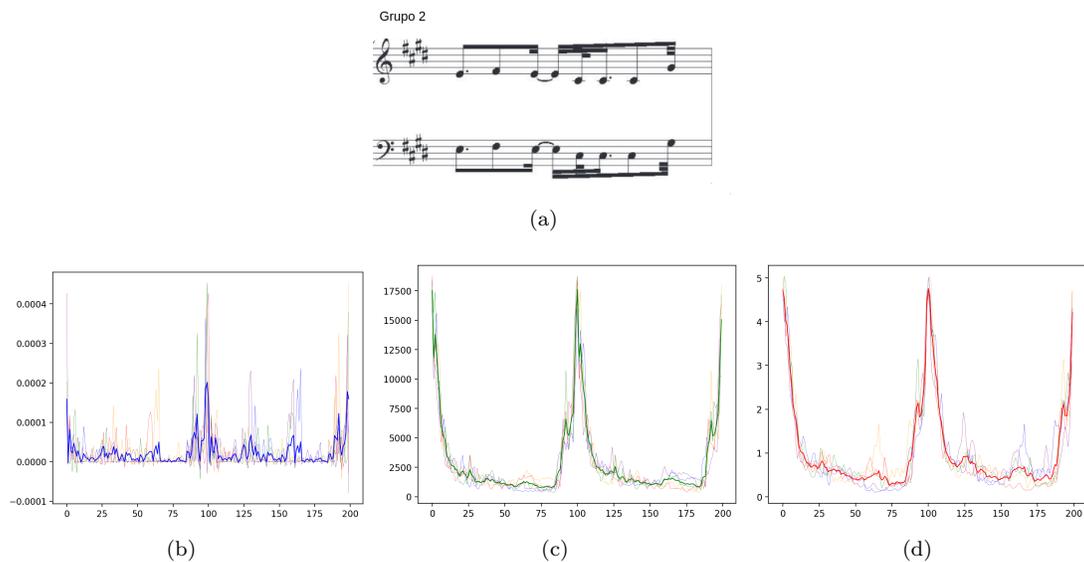


Figura 12: Padrões médios referentes ao Grupo 2 do Parixara 1. (a) Grupo de compasso número 2, (b) Função RMS média, (c) Função HFC média, (d) Função *Complex Spectral Difference* média.

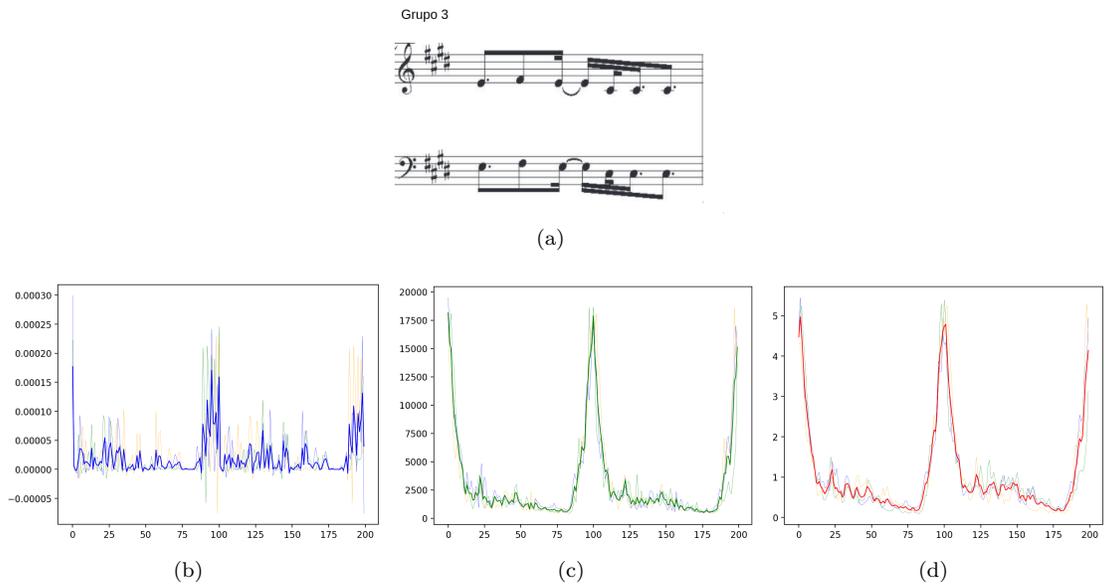


Figura 13: Padrões médios referentes ao Grupo 3 do Parixara 1. (a) Grupo de compasso número 3, (b) Função RMS média, (c) Função HFC média, (d) Função *Complex Spectral Difference* média.

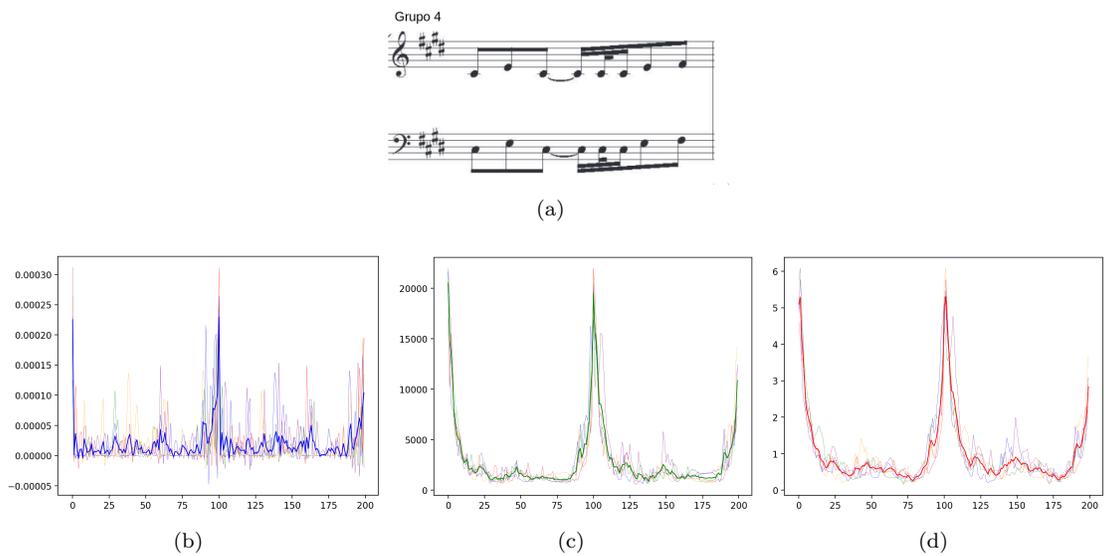


Figura 14: Padrões médios referentes ao Grupo 4 do Parixara 1. (a) Grupo de compasso número 4, (b) Função RMS média, (c) Função HFC média, (d) Função *Complex Spectral Difference* média.

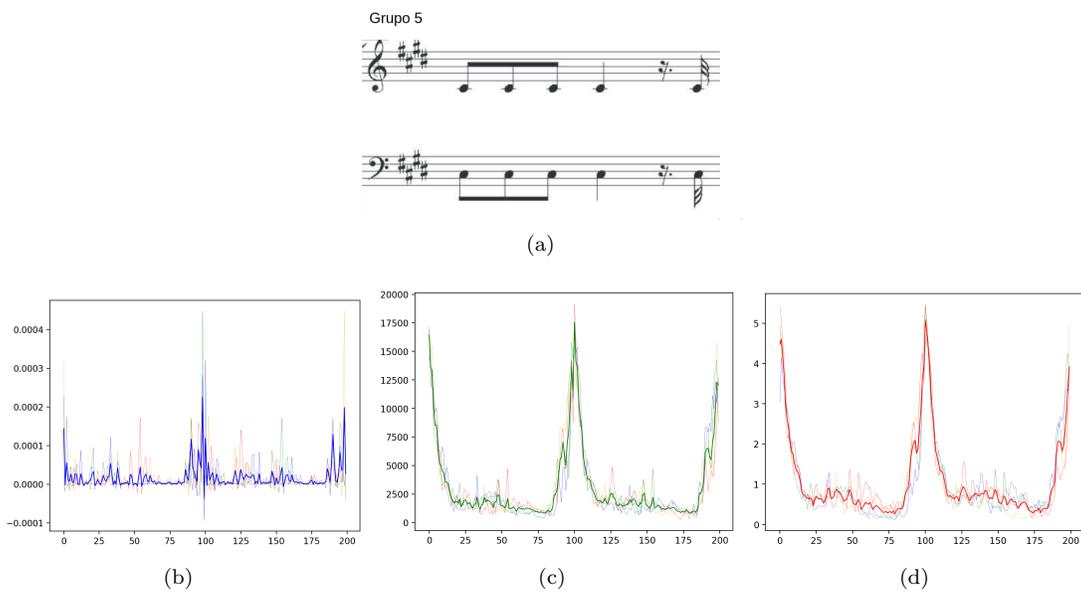


Figura 15: Padrões médios referentes ao Grupo 5 do Parixara 1. (a) Grupo de compasso número 5, (b) Função RMS média, (c) Função HFC média, (d) Função *Complex Spectral Difference* média.

Abaixo seguem as imagens dos padrões médios referentes ao Parixara 2. Como no Parixara 2 o chocalho é batido somente para marcar o início do compasso, os grandes picos no início e no fim representam este ataque percussivo.

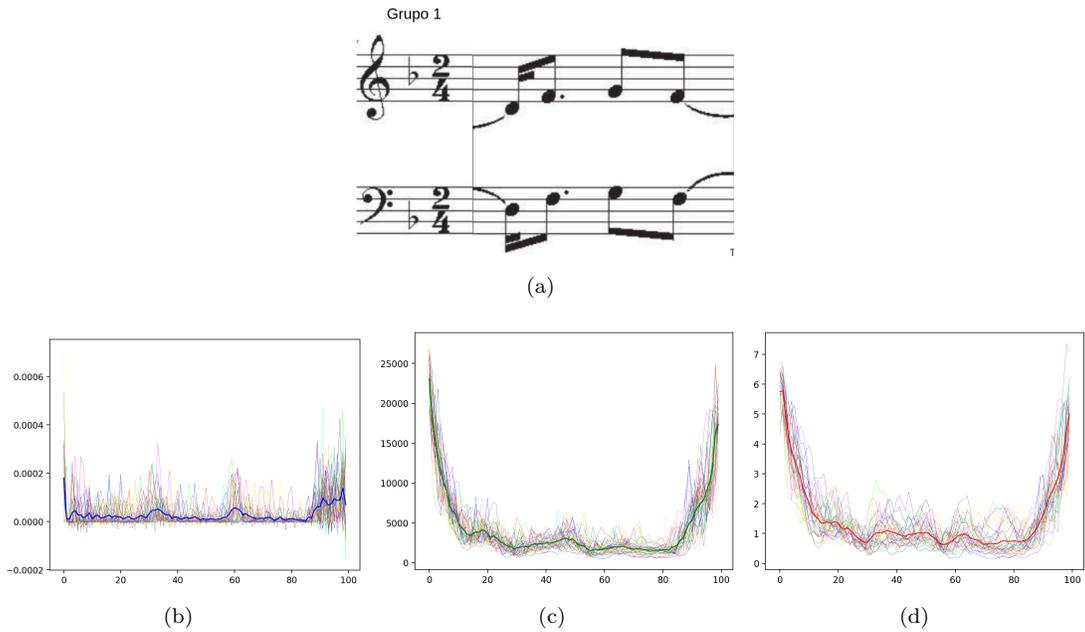


Figura 16: Padrões médios referentes ao Grupo 1 do Parixara 2. (a) Grupo de compasso número 1, (b) Função RMS média, (c) Função HFC média, (d) Função *Complex Spectral Difference* média.

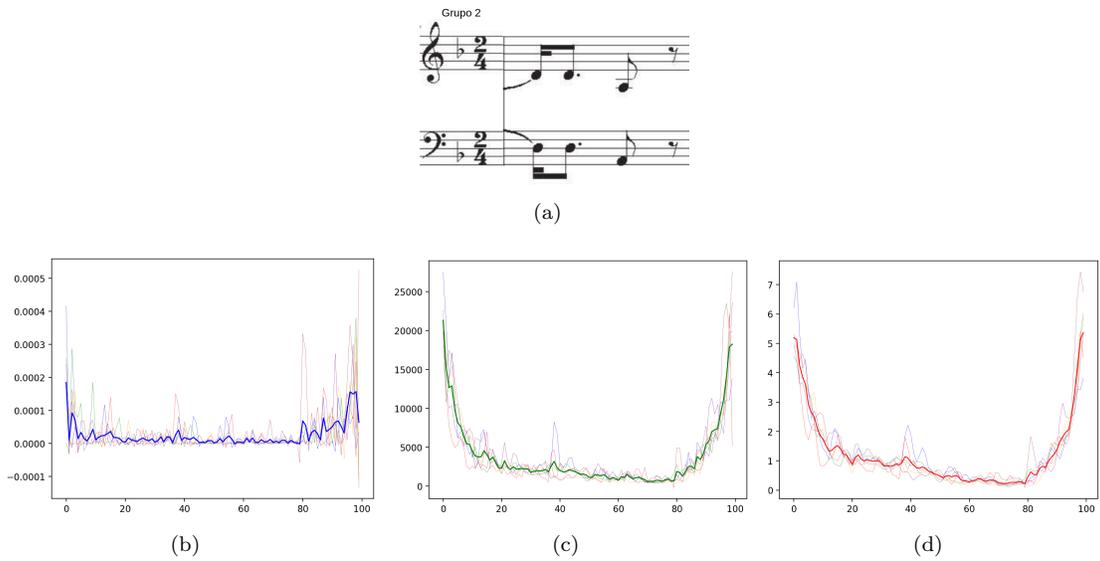


Figura 17: Padrões médios referentes ao Grupo 2 do Parixara 2. (a) Grupo de compasso número 2, (b) Função RMS média, (c) Função HFC média, (d) Função *Complex Spectral Difference* média.

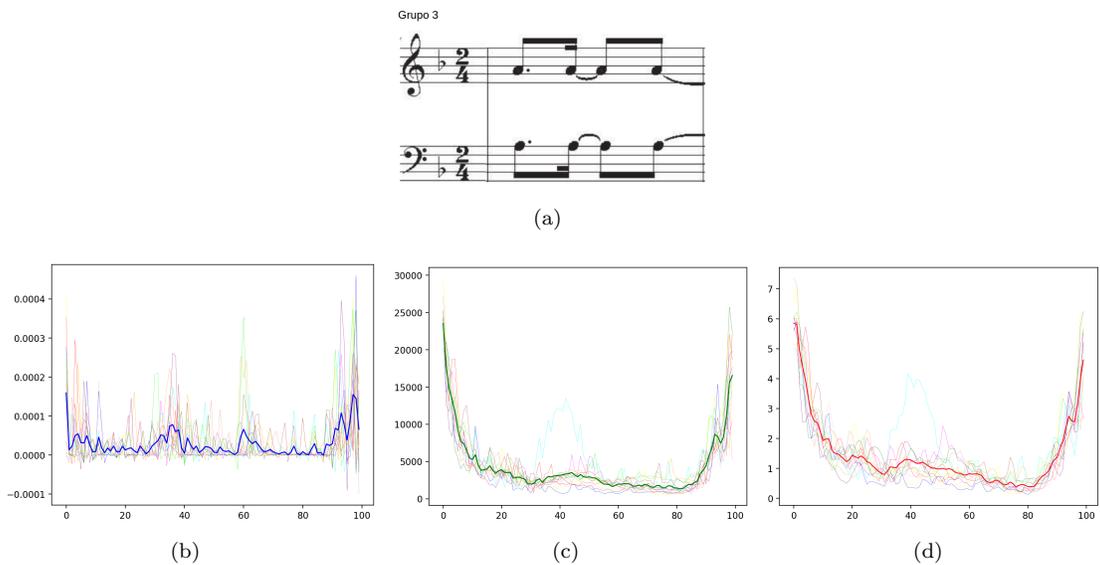


Figura 18: Padrões médios referentes ao Grupo 3 do Parixara 2. (a) Grupo de compasso número 3, (b) Função RMS média, (c) Função HFC média, (d) Função *Complex Spectral Difference* média.

O que esperávamos observar, a partir do exemplo de Gouyon [8], seriam os picos correspondentes à cada nota do compasso, por exemplo, no Grupo 3 do Parixara 2 (Figura 18) há 3 ataques em cada compasso, então esperávamos encontrar 3 picos muito claros e destacados nos padrões médios, o que não aconteceu na maioria dos casos.

A partir das observações acima, surgiu a ideia de trabalhar com outra técnica de agrupamento para encontrar a localização dos ataques das notas no sinal de áudio: a *Clusterização*.

3.5 Clusterização de Onsets

A *clusterização* dos *onsets* é uma estratégia consideravelmente diferente daquela proposta por Gouyon [8], visto que leva em consideração apenas os instantes dos onsets e não o perfil completo das funções de novidade. Por se tratar de um método exploratório e empírico, decidimos restringir sua aplicação aos casos mais simples, para posteriormente considerar casos mais gerais.

Uma restrição foi considerar apenas os picos da função RMS, pois dentre as ODFs, esta foi a que teve mais clareza na definição dos picos. Adicionalmente, devido à maior complexidade do Parixara 1, que tem um andamento mais rápido do que o Parixara 2 e uma batida do chocalho no meio de cada compasso, que poderia acabar se misturando com os picos de algumas das notas e interferir na *clusterização*, decidimos realizar o processo seguinte apenas no Parixara 2.

Além disso, a *clusterização* foi feita de duas formas diferentes, ambas utilizando o algoritmo *KMeans*. A primeira foi *clusterizar* cada instância de um grupo de compassos separadamente. Já a segunda foi *clusterizar* conjuntamente os picos de todos os compassos de um mesmo grupo. Nos dois casos, a quantidade de *clusters* foi definida a partir da quantidade de notas que aparecem na representação simbólica (partitura) do grupo de compassos correspondente.

3.5.1 Seleção de Picos

Como podemos observar no exemplo da Figura 19 (exemplo do compasso número 2, que faz parte do segundo grupo de padrões de compasso do Parixara 2), existem picos mais proeminentes, e outros picos comparativamente menores e provavelmente espúrios (visto que aquele compasso teria apenas 2 notas após o chocalho inicial). Isso torna necessária a definição de um critério para seleção de picos relevantes para a *clusterização*.

Este critério¹¹ leva em consideração a amplitude relativa de cada pico tanto em relação aos picos próximos como também em relação ao maior pico do compasso (associado ao chocalho). Além disso, devido ao som percussivo do chocalho não ser “seco”, mas possuir diversos *onsets* secundários ocupando uma duração relativamente grande do compasso, a borda (início e fim) de cada compasso é desconsiderada na análise.

¹¹Implementado na função `filtro_compasso`, que pode ser encontrada na seção de Apêndices, na página 32.

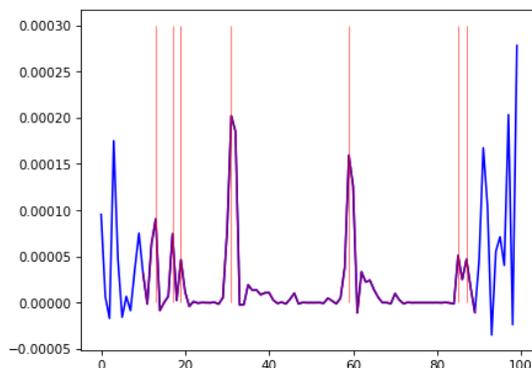


Figura 19: Gráfico da RMS do segundo compasso. A borda que não foi considerada para *clusterização* está em azul e a curva em roxo teve os picos filtrados (picos selecionados com linhas verticais).

Para definir o tamanho da borda a ser desconsiderada, consideramos a duração da menor nota do compasso. No caso do Parixara 2, a menor nota de todos os compassos é sempre uma semi-colcheia, que representa $\frac{1}{8}$ ou 12,5% da duração do compasso. Levando em consideração que flutuações rítmicas são comuns, definimos como borda de um compasso um intervalo de 10% da duração do compasso, uma porcentagem levemente menor do que aquela duração mínima, para evitar que a borda acabasse excluindo uma nota que não fosse o chocalho.

3.5.2 Sumarização por Centroides

Como dito anteriormente, o agrupamento foi realizado de duas formas diferentes. A primeira, *clusterizar* cada compasso separadamente, foi feita da seguinte maneira: para cada compasso de um mesmo grupo, os picos selecionados pela nossa função de filtragem foram *clusterizados* em um número de *clusters* igual à quantidade de notas do compasso. Após a *clusterização*, os centroides de cada *cluster* foram calculados utilizando a biblioteca *scikit-learn*. Utilizando o mesmo exemplo da figura anterior, o segundo compasso (que pertence ao grupo 1) do Parixara 2, podemos ver na Figura 20 a divisão de *clusters* e centroides.

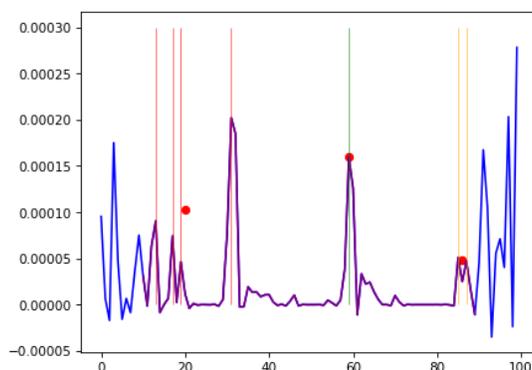


Figura 20: Gráfico da RMS do segundo compasso. A borda que não foi considerada para *clusterização* está em azul e a curva em roxo teve os picos filtrados (picos selecionados com linhas verticais, cada cor representa um *cluster*). Os pontos vermelhos são os centroides de cada *cluster*.

Para obter a localização representativa de cada nota dentro de um grupo de compassos, foi feita uma média dos centroides obtidos em cada compasso separadamente. Nas Figuras 21 e 22 abaixo, vemos os centroides de cada compasso do respectivo grupo representados como círculos, usando cores para identificar os *clusters* correspondentes, e quadrados para representar a média de cada *cluster*.

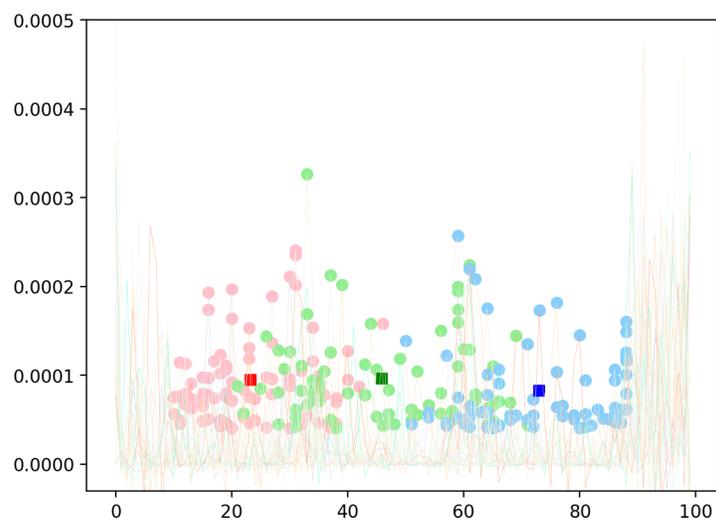


Figura 21: Gráfico da RMS de cada compasso do grupo 1. Os círculos são os centroides dos picos de cada compasso e os quadrados são as médias desses centroides.

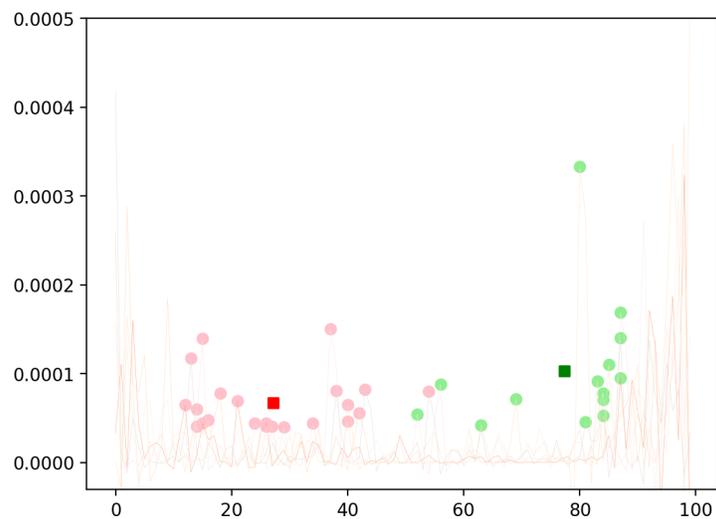


Figura 22: Gráfico da RMS de cada compasso do grupo 2. Os círculos são os centroides dos picos de cada compasso e os quadrados são as médias desses centroides.

A segunda forma de agrupamento é feita *clusterizando* conjuntamente os picos de todos os compassos de um mesmo grupo. Após filtrar os picos de cada compasso, com a mesma função citada anteriormente (`filtro_compasso`), os picos são *clusterizados* utilizando a biblioteca *scikit-learn*. Nas Figuras 23 e 24 abaixo vemos os picos *clusterizados* como círculos (sendo cada *cluster* de uma cor) e os centroides como quadrados, para cada um dos 2 grupos de compasso principais.

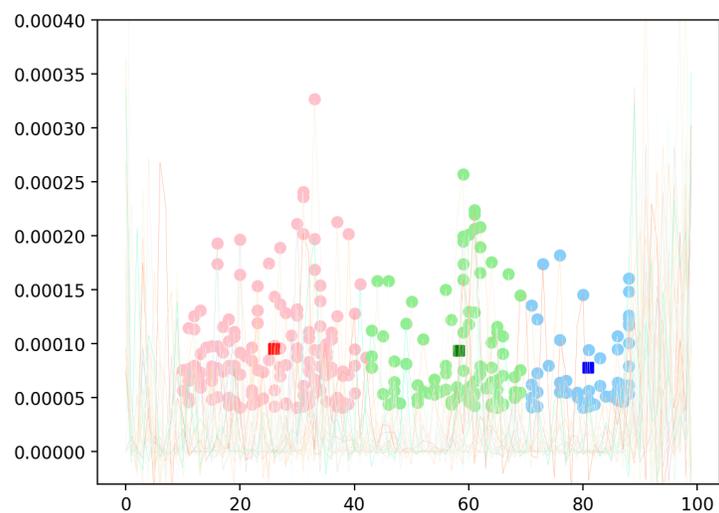


Figura 23: Gráfico da RMS de cada compasso do grupo 1. Os círculos são os picos *clusterizados* e os quadrados são os centroides de cada *cluster*.

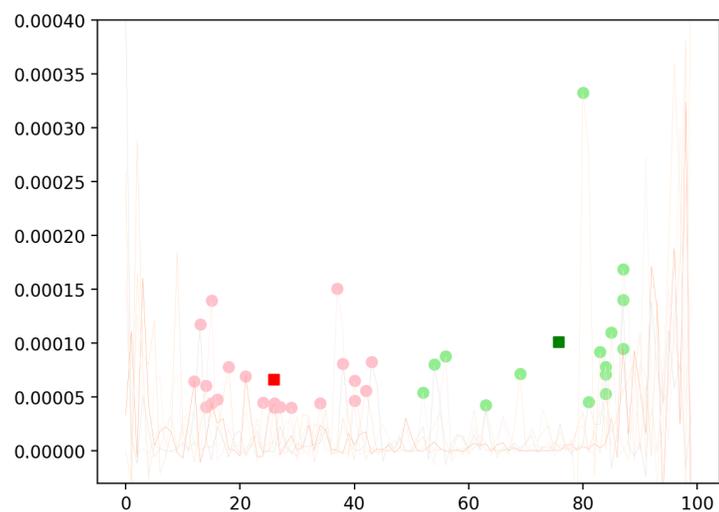


Figura 24: Gráfico da RMS de cada compasso do grupo 2. Os círculos são os picos *clusterizados* e os quadrados são os centroides de cada *cluster*.

Para compararmos visualmente a localização esperada das notas e os pontos encontrados na análise, segue abaixo uma imagem com os pontos encontrados utilizando a primeira forma de *clusterização* como um quadrado azul, a segunda forma como um quadrado vermelho e a localização das notas de acordo com a

partitura sendo uma linha vertical preta.

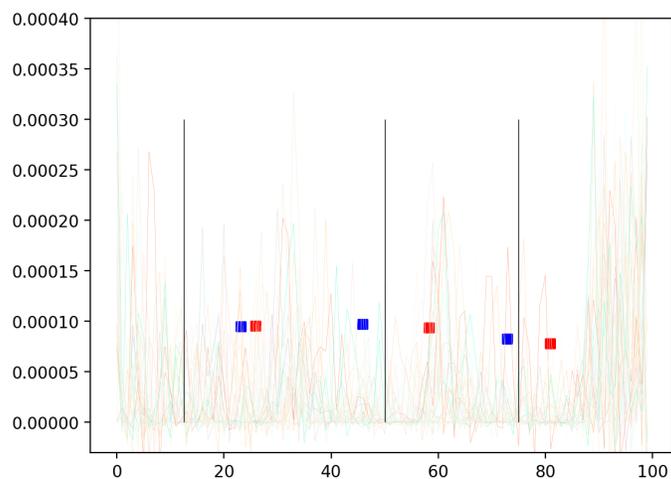


Figura 25: Gráfico da RMS de cada compasso do grupo 1. Os quadrados azuis são as médias dos picos *clusterizados* e os quadrados vermelhos são os centroides de cada *cluster* quando *clusterizados* todos os picos juntos. A linha vertical preta representa a localização das notas na partitura.

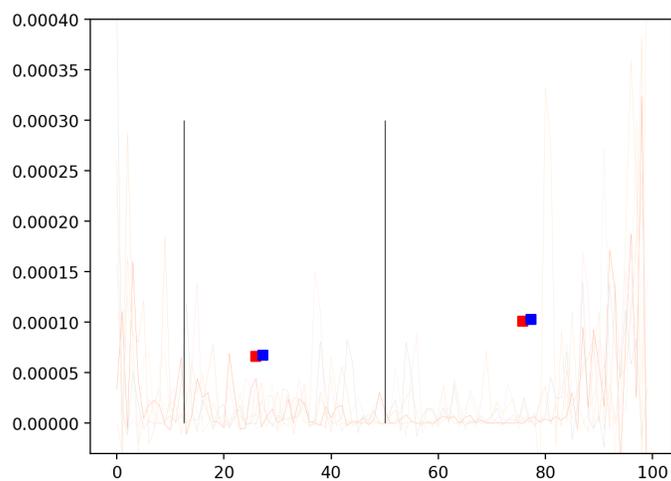


Figura 26: Gráfico da RMS de cada compasso do grupo 2. Os quadrados azuis são as médias dos picos *clusterizados* e os quadrados vermelhos são os centroides de cada *cluster* quando *clusterizados* todos os picos juntos. A linha vertical preta representa a localização das notas na partitura.

Como podemos observar nas imagens acima, os onsets detectados no sinal de áudio são bem diferentes dos instantes onde a partitura prescreve, geralmente ocorrendo depois das respectivas posições teóricas. Também foi observada uma maior divergência entre as duas técnicas de clusterização no grupo 1 do que no grupo 2.

Isso mostraria um desvio entre a performance e a escrita formal da canção, que poderia ser uma característica intencional desse estilo de música. Porém, também é possível observar, principalmente nas figuras de 21 a 24, que os pontos encontrados vieram de dados bem dispersos, o que sugere que mesmo os compassos que contêm a mesma rítmica seriam realizados de maneira diferente na performance, com um alto grau de flutuação dos onsets.

Tais observações também poderiam sugerir que a metodologia utilizada ainda não é a ideal para analisar este repertório, principalmente por conta da forte interferência do chocalho, que atravessa quase todo o sinal.

4 Conclusões

As análises por compassos de padrões similares que seguiram o procedimento feito por Fabien Gouyon em seu artigo “*Microtiming in ‘Samba de Roda’ — Preliminary experiments with polyphonic audio*” [8] não se mostraram eficientes para este repertório em específico, pois esperávamos observar claramente os picos correspondentes a cada nota de cada grupo de compassos, o que não aconteceu.

Então, a fim de avançar em direção ao objetivo do nosso trabalho, ou seja, estabelecer correspondências entre os registros em áudio e as partituras existentes, trabalhamos com outra técnica de agrupamento para encontrar a localização dos ataques das notas no sinal de áudio: a *Clusterização*. Tal técnica se mostrou mais eficaz do que a proposta de Gouyon, porém ainda apresentou alguns problemas, como o fato de os pontos definidos como ataques das notas terem vindo de dados bem dispersos.

O som reverberante do chocalho certamente influenciou os resultados e dificultou a análise, e o ideal seria filtrar este som para analisarmos apenas o material melódico. Entretanto, tal filtragem também se mostrou uma tarefa difícil, devido ao fato de que as técnicas estado-da-arte de separação de fontes, no caso deste trabalho a biblioteca Spleeter [9], são construídas em sua maioria a partir de algoritmos de aprendizado de máquina e seus dados de treinamento são, em sua maioria, compostos de repertórios como música pop, rock, jazz, etc. Desse modo, ao aplicar os modelos treinados em repertórios diferentes, o resultado não é tão bom quanto nos dados originais.

Uma ideia para a continuidade do trabalho é aprofundar as análises baseadas na *clusterização*, estudando variações das funções de novidade ou da função de seleção de picos mais apropriadas ao material investigado. Outro caminho possível para a pesquisa futura é investigar técnicas mais específicas para a separação do som do chocalho e do material melódico, para que as análises do ritmo da parte vocal possam ser feitas sem aquela interferência.

Referências

- [1] David Arthur e Sergei Vassilvitskii. «K-Means++: The Advantages of Careful Seeding». Em: vol. 8. Jan. de 2007, pp. 1027–1035. DOI: 10.1145/1283383.1283494.
- [2] Juan Bello et al. «On the Use of Phase and Energy for Musical Onset Detection in the Complex Domain». Em: *Signal Processing Letters, IEEE* 11 (jul. de 2004), pp. 553–556. DOI: 10.1109/LSP.2004.827951.
- [3] S. Broughton e Kurt Bryan. *Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing*. Out. de 2008. ISBN: 9780470294666. DOI: 10.1002/9781118032442.
- [4] C. Cannam, C. Landone e M. Sandler. «Sonic Visualiser: An Open Source Application for Viewing, Analysing, and Annotating Music Audio Files». Em: *Proceedings of the ACM Multimedia 2010 International Conference*. Firenze, Italy, out. de 2010, pp. 1467–1468.
- [5] Music Technology Group - Universitat Pompeu Fabra. *Essentia Library*. <https://github.com/MTG/essentia>. 2019.
- [6] Felipe Munhoz Martins Fernandes. «Do parixara ao forró, do forró ao “parixara”: uma trajetória musical». Tese de mestrado. São Carlos, SP: Universidade Federal de São Carlos, 2015.
- [7] Devair Antônio Fiorotti. *Panton Pia’*. URL: <https://pantonpia.com.br/>.
- [8] Fabien Gouyon. «Microtiming in “Samba de Roda” — Preliminary experiments with polyphonic audio». Em: (jan. de 2007).
- [9] Romain Hennequin et al. «Spleeter: a fast and efficient music source separation tool with pre-trained models». Em: *Journal of Open Source Software* 5 (jun. de 2020), p. 2154. DOI: 10.21105/joss.02154.
- [10] Theodor Koch-Grünberg. *Do Roraima ao Orinoco*. Vol. 1. Editora UNESP, 1916/2001.
- [11] Paul Masri. «Computer Modelling of Sound for Transformation and Synthesis of Musical Signals». Tese de doutoramento. Bristol, UK: University of Bristol, 1996.
- [12] Meinard Müller. *Fundamentals of Music Processing*. Jan. de 2015. ISBN: 978-3-319-21944-8. DOI: 10.1007/978-3-319-21945-5.
- [13] Helena Palmquist. «Questões sobre genocídio e etnocídio indígena: a persistência da destruição». Tese de mestrado. Belém, Pará: Universidade Federal do Pará, 2018.
- [14] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». Em: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [15] Fernanda Frizzo Bragato e Pedro Bigolin Neto. «Conflitos territoriais indígenas no Brasil: entre risco e prevenção». Em: *Revista Direito e Praxis* 8.1 (2016). DOI: 10.12957/dep.2017.21350.

- [16] Laure Prétet et al. «Singing Voice Separation: A Study on Training Data». Em: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mai. de 2019, pp. 506–510. DOI: 10.1109/ICASSP.2019.8683555.
- [17] Diocese de Roraima. *Índios de Roraima - Coleção histórico-antropológica No. 1*. Coronário, 1989.
- [18] Isabella Coutinho Santos Josué e Costa. *Povos Indígenas de Roraima*. 2022. URL: <https://povosindigenasrr.uerr.edu.br/>.
- [19] Jucicleide Pereira Mendonça dos Santos. «Do Parixara ao Areruaia». Tese de mestrado. Boa Vista, RR: Universidade Federal de Roraima, 2018.
- [20] Xavier Serra. «A Multicultural Approach in Music Information Research.» Em: jan. de 2011, pp. 151–156.
- [21] Pauli Virtanen et al. «SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python». Em: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [22] Chris Baume e Yves Raimond. *BBC Vamp plugin collection*. <https://github.com/bbc/bbc-vamp-plugins>. 2014.

Apêndices

Apêndice 1

O código a seguir está em Python e pode ser encontrado aqui. Seus *inputs* são: `odf`, o vetor da função de novidade, `largura`, a largura aproximada dos picos e `min_altura`, a altura mínima para que o pico seja considerado um pico que representa o chocalho.

A função `min_index(vetor)` também pode ser encontrada no mesmo arquivo, ela retorna o índice do menor número do `vetor`.

```
1 def filtro_picos(odf, largura, min_altura):
2
3     tamanho = len(odf)
4     onsets = []
5     index_min = min_index(odf)
6     aux = index_min
7     i = 0
8
9     while i < tamanho:
10        if odf[i] >= min_altura:
11            for j in range(largura):
12                if odf[i+j] >= odf[aux]:
13                    aux = i+j
14                    onsets.append(aux)
15                    aux = index_min
16                    i += largura
17            else:
18                i += 1
19    return onsets
```

Apêndice 2

O código a seguir está em Python e pode ser encontrado aqui. Seus *inputs* são: `odf`, o vetor da função de novidade e `onsets`, o vetor com os *onsets* que representam o chocalho.

A função `interp(xi, xp, yp)` também pode ser encontrada no mesmo arquivo, ela retorna a interpolação cúbica da função `(xp, yp)`.

```
1 def resample(odf, onsets):
2     n = len(onsets)
3     new_sample = list()
4
5     rate = 100
6
7     for i in range(n-1):
8
9         inicio = int(onsets[i])
10        fim = int(onsets[i+1])
11        step = (fim-inicio)/rate
12
13        yp = odf[inicio:fim]
14        xp = np.array(range(inicio,fim))
15
16        xi = np.arange(inicio,fim,step)
17
18        #itera nos 100 pontos novos de cada segmento
19        new_sample.append(interp(xi, xp, yp))
20
21    return new_sample
```

Apêndice 3

O código a seguir está em Python e pode ser encontrado aqui. Seus *inputs* são: `odf`, o vetor da função de novidade, `altura_min`, a menor altura de pico a ser considerada e `borda`, o tamanho da borda.

```
1 def filtro_compasso(odf, altura_min, borda):
2     i=0
3     tamanho = len(odf)
4     picos = []
5     while i < tamanho-1:
6         j = i
7         while j<tamanho and odf[j] > altura_min:
8             if odf[j-1] < odf[j] and odf[j+1] < odf[j]:
9                 picos.append([j+borda,odf[j]])
10                j += 1
11            if i != j:
12                i = j
13                if i>=tamanho:
14                    break
15            else:
16                i += 1
17
18     return picos
```

Anexos

PARIXARA 1

Autores: Povos indígenas do circun-Roraima

Intérpretes: Manaaka e Yauyo

Transcrição/Editoração: Jucicleide Pereira e
Mozart Mendсан

$\text{♩} = 60$

Voz (F)

Voz (M)

Chocalho

4

7

10

1.

2.

PARIXARA 2

Autores: Povos indígenas do circum-Roraima

Intérpretes: Manaaka e Yauyo

Transcrição/Editoração: Jucicleide Pereira e Mozart Mendsan

$\text{♩} = 100$

Voz (F)

Voz (M)

Chocalho

6

11

15

20