

Leitura de partituras em imagens digitais

Erick Seiji Furukawa

Hélio Pedrini

Relatório Técnico - IC-PFG-20-26

Projeto Final de Graduação

2021 - Janeiro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Leitura de partituras a partir de imagens

Erick Seiji Furukawa* Hélio Pedrini †

Resumo

Este trabalho tem como objetivo realizar o estudo de sistemas OMR (*reconhecimento musical óptico* ou *optical musical recognition* em inglês) e o desenvolvimento de um leitor de partituras em imagens digitais. Estes sistemas buscam realizar a conversão de uma imagem contendo uma partitura, em um formato de arquivo musical que pode ser facilmente lido por um computador. Neste projeto, experimentos foram realizados com alguns métodos frequentemente utilizados em sistemas OMR e também foi desenvolvido um sistema capaz de realizar a conversão de imagens contendo partituras monofônicas simples para um arquivo MIDI, utilizando alguns dos métodos existentes na literatura. O projeto está disponível em um repositório público.

1 Introdução

A escrita é um conjunto de símbolos padronizados que é utilizado para transmitir a linguagem falada graficamente. De forma similar, a notação musical é um sistema que foi desenvolvido para representar uma peça musical de forma gráfica, para que possa ser utilizada por um artista para performar uma música de forma semelhante à ideia do compositor.

As formas mais antigas de notação musical podem ser encontradas em forma de escrita cuneiforme de aproximadamente 2000 AC. Apesar dos sistemas antigos de notação musical serem capazes de transmitir algum grau de significado musical, muitos não possuíam formas de indicar a altura absoluta das notas musicais, o que impossibilitava pessoas que nunca tivessem ouvido a música de interpretá-las. Outro problema comum era a falta de indicação da duração das notas. O sistema de notação vem se evoluindo desde suas origens, para proporcionar uma forma eficiente de se transmitir música de forma gráfica. Com isso, centenas de símbolos e técnicas diferentes foram incorporadas, tornando o sistema de notação musical moderno bastante complexo.

Mesmo com a existência de ferramentas computacionais capazes de gerar partituras em excelente qualidade, muitos compositores ainda preferem escrever suas peças de maneira manuscrita. A utilização de mouses, teclados ou mesmo pianos digitais que podem ser conectados ao computador para gerar partituras é um processo que pode ser considerado lento ou difícil por muitas pessoas. Além disso, ainda existem diversos documentos antigos que possuem partituras manuscritas que não foram convertidas para uma forma digital. A

*Instituto de Computação, Universidade Estadual de Campinas, 13083-852 Campinas, SP.

†Instituto de Computação, Universidade Estadual de Campinas, 13083-852 Campinas, SP.

conversão destes manuscritos é uma tarefa que exige muito esforço e tempo, que poderia ser parcialmente automatizada com o uso de técnicas da área de OMR.

OMR é um campo de pesquisa que possui inúmeras aplicações como preservar documentos antigos, auxiliar músicos a praticar e compor, auxiliar a área de musicologia (estudo científico da música) e converter documentos musicais para diferentes formatos. É uma área de pesquisa muitas vezes confundida com OCR (*reconhecimento óptico de caracteres* ou *optical character recognition* em inglês). Em uma partitura, símbolos possuem um significado diferente dependendo da sua posição horizontal, vertical e sua posição relativa a outros na partitura. Um símbolo no começo da página pode interagir com diversos símbolos até o final da página, alterando o significado musical de cada um deles. Além disso, uma partitura pode conter uma densidade muito alta de símbolos que podem até mesmo estar sobrepostos, que torna a tarefa de detecção e atribuição de significado musical aos símbolos bastante complexa.

Tendo em vista as aplicações e os desafios encontrados em OMR, este projeto visa realizar o estudo e a aplicação de diferentes técnicas utilizadas nesta área de pesquisa, buscando desenvolver um método capaz de converter uma partitura, presente em uma imagem, em um arquivo de som como MIDI ou mp3. Serão abordadas técnicas como binarização de imagens, detecção de linhas, operações morfológicas e detecção de objetos.

A Seção 2 dedica-se à apresentação da notação musical moderna. Sua leitura não é necessária para aqueles que já conhecem o básico. A Seção 3 apresenta o funcionamento geral de sistemas OMR e os métodos mais utilizados. A Seção 4 descreve os experimentos realizados com alguns métodos apresentados na Seção 3 e a metodologia que foi utilizada para desenvolver o projeto. A Seção 5 traz algumas considerações finais.

2 Notação Musical

Nesta seção, alguns conceitos básicos relacionados à notação musical são apresentados.

2.1 Pauta

A música é escrita em um conjunto de cinco linhas horizontais paralelas chamado de pauta, que é utilizada para definir a altura (frequência) das notas musicais. Quanto mais acima estiver, mais aguda uma nota será. Uma nota musical pode estar posicionada ou em uma linha ou em um espaço da pauta e, caso seja necessário representar alturas que não estão contidas neste intervalo, podem ser utilizadas linhas e espaços suplementares acima ou abaixo da pauta como pode ser visto na Figura 1. Notas musicais podem estar invertidas para economizar espaço.

2.2 Notas

No sistema de notação musical moderno, há sete notas musicais: dó, ré, mi, fá, sol, lá, e si. Cada uma dessas notas pode ser representada por uma letra do alfabeto, como mostra a Tabela 1. Apesar de existirem apenas sete notas diferentes, isso não significa que só existam sete alturas diferentes. Conforme as notas vão subindo na pauta, os nomes das

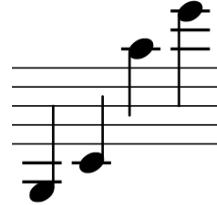


Figura 1: Linhas suplementares acima e abaixo da pauta.

notas começam a se repetir (Figura 2) e as notas passam a pertencer a uma nova oitava. Uma nota estar uma oitava acima de outra significa que sua frequência é o dobro da outra. Similarmente, uma nota estar uma oitava abaixo, significa que possui a metade da frequência. Como existem várias notas com o mesmo nome, quando é necessário referenciar uma nota específica, é indicada com um número na frente da nota, qual a oitava que esta nota pertence como, por exemplo, sol 4 ou G4, para indicar a nota sol na quarta oitava.

Nota	Representação
Dó	C
Ré	D
Mi	E
Fá	F
Sol	G
Lá	A
Si	B

Tabela 1: Notas musicais

(a)

(b)

Figura 2: (a) Clave de sol e suas notas; (b) Clave de fá e suas notas.

Na Figura 3, podem ser observadas as sete notas musicais e suas posições em um piano.

As teclas pretas presentes entre as brancas correspondem a notas intermediárias, que serão apresentadas na Seção 2.7. A distância entre uma tecla do piano e a tecla seguinte mais próxima (incluindo teclas pretas) corresponde ao intervalo musical chamado de semitom. Dois semitons formam um tom.

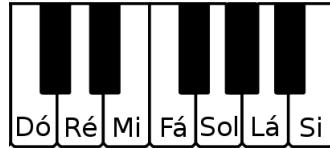


Figura 3: Notas em um piano.

2.3 Claves

Há quatro tipos de clave no sistema de notação moderno: clave de sol, clave de fá, clave de dó e a clave neutra. Claves são símbolos musicais utilizados no começo de uma pauta para definir uma nota de referência, para que seja possível determinar as posição das outras notas a partir desta. O efeito de uma clave dura por toda a pauta, ou até que uma nova clave apareça. Na Figura 4, podem ser observadas a clave de sol, a clave de fá e as suas notas de referência. Na Figura 2, a primeira nota dó da clave de sol e a última nota dó da clave de fá indicadas são a mesma nota (possuem a mesma frequência). Esta nota é frequentemente chamada de dó central por estar localizada na região central do piano. Esta nota também é chamada de dó 4.

Enquanto as claves de sol, fá e dó são utilizadas estabelecer as notas de uma pauta, instrumentos que não possuem alturas definidas, como alguns instrumentos de percussão, utilizam a clave neutra. Nesta situação, as notas podem ser posicionadas arbitrariamente. Nem todos os instrumentos de percussão utilizam a clave neutra. O tímpano, por exemplo, possui alturas definidas e é escrito na clave de fá.

Alguns instrumentos como o piano, cravo, e órgão, possuem uma tessitura (extensão de notas) muito grandes e, por isso, suas partituras geralmente possuem duas pautas para serem tocadas ao mesmo tempo, com claves de sol e fá. Para representar diferentes tessituras, as claves podem ser posicionadas em algumas posições diferentes na pauta, alterando a posição da nota de referência e, portanto, a posição de todas as outras notas da pauta.

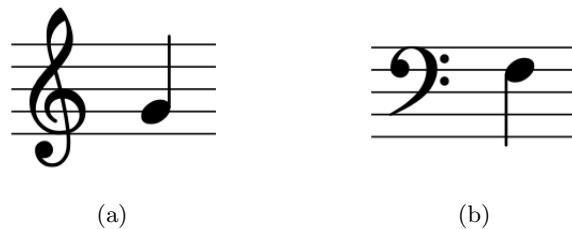


Figura 4: (a) Clave de sol e a nota de referência sol 4; (b) Clave de fá e a nota de referência fá 3.

2.4 Andamento

O andamento da música (velocidade) é muitas vezes indicada pelo compositor no início da pauta, com a utilização de termos musicais italianos. Na Tabela 2 encontram-se algumas marcações de andamento, o seu significado e o número de batidas por minuto (BPM). O número de BPM é apenas uma orientação geral. O real andamento da música deve ser decidido pelos músicos que a interpretam e pode ser fluido, variando durante a música conforme a necessidade.

Termo	Significado	BPM
Largo	Muito devagar	40-60
Adagio	Devagar	66-76
Andante	Ritmo de caminhada	76-108
Moderato	Moderadamente	108-120
Allegro	Rápido	120-168
Vivace	Rápido e vivo	140-176
Presto	Muito rápido	168-200

Tabela 2: Algumas marcações de andamento [2].

2.5 Duração de notas e pausas

Na Tabela 3, encontram-se os símbolos utilizados para indicar a duração das notas. Cada nota possui um símbolo para indicar que a nota será tocada (coluna de nota) e um símbolo para indicar uma pausa (coluna de pausa). As notas musicais são representadas basicamente por três componentes:

- Cabeça da nota: parte elíptica da nota, que não é preenchida apenas nas notas semi-breve e mímina.
- Haste: linha vertical conectada na cabeça da nota.
- Bandeirola: linha(s) conectada(s) na haste, no extremo oposto da cabeça da nota.

Na situação em que se encontram várias notas que possuem bandeirolas em sequência, as bandeirolas são conectadas para facilitar a leitura. As bandeirolas são substituídas por uma linha conectando as hastes, como pode ser visto na Figura 5. Se as notas possuírem um número diferente de bandeirolas, as que sobram ficam desconectadas.

O ponto de aumento é um símbolo colocado à direita de uma nota ou pausa, para indicar que a nota deve ter sua duração aumentada em 50%. Pontos de aumento adicionais podem ser utilizados e, nessas situações, o novo ponto de aumento aumenta a duração da nota em 50% do aumento do ponto anterior. Isto é, se forem utilizados dois pontos de aumento, uma nota passa a ter uma duração adicional de 50% (primeiro ponto) mais 25% (segundo ponto), totalizando uma duração de 175% da nota original.

Número	Nota	Pausa	Nome	Duração
1			Semibreve	1
2			Mínima	1/2
4			Semínima	1/4
8			Colcheia	1/8
16			Semicolcheia	1/16
32			Fusa	1/32
64			Semifusa	1/64

Tabela 3: Notas e pausas.



Figura 5: Semicolcheia, colcheia e duas fusas conectadas.

Uma outra forma de adicionar duração a uma nota musical é utilizando a ligadura. A ligadura é um símbolo que conecta duas notas musicais com uma mesma altura, fazendo com que a nota seja tocada apenas uma vez, com a duração das duas notas somadas. Na Figura 6, podem ser observadas uma semínima pontuada e uma semínima conectada com uma colcheia por uma ligadura. Ambas as notas possuem a mesma duração.

2.6 Compasso e fórmula de compasso

Compassos são divisões da música em intervalos de tempos iguais utilizados para organizar a partitura e facilitar sua leitura. A soma da duração de todas as notas e pausas dentro de um compasso deve ser exatamente igual à duração do compasso, que é definida pela fórmula de compasso. Na Figura 7, podem ser observados dois compassos, onde cada um possui uma duração de quatro semínimas.

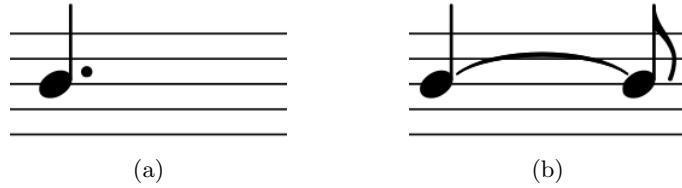


Figura 6: (a) Semínima pontuada; (b) Semínima e colcheia conectadas por uma ligadura.

A fórmula de compasso encontra-se no começo de uma partitura e é composta por dois números. O número superior indica quantas unidades de tempo cada compasso possui, enquanto o número inferior indica qual o símbolo será utilizado como uma unidade de tempo. O número inferior refere-se à coluna “número” da Tabela 3. Para representar uma fórmula de compasso, utilizam-se os números superior e inferior separados por uma barra, como por exemplo, “3/4” (lido como: três por quatro). Uma música com uma fórmula de compasso 3/4 significa que cada compasso possui uma duração de três (número superior) semínimas (número inferior).

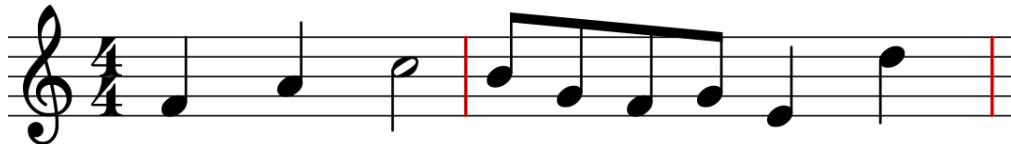


Figura 7: Pauta com dois compassos, demarcados pelas linhas verticais destacadas em vermelho.

2.7 Acidentes

Acidentes são símbolos utilizados na esquerda de notas musicais para aumentar ou diminuir sua altura. Quando um acidente é colocado na pauta, seu efeito persiste até o fim do compasso atual, afetando todas as notas que se encontram na mesma linha ou espaço que o acidente. Na Tabela 4, podem ser observados alguns tipos de acidentes que existem no sistema de notação musical e seus efeitos nas notas musicais. O sustenido, bemol, e bequadro são acidentes bastante comuns, enquanto os outros são raramente utilizados. Para representar um acidente em uma nota de forma escrita, podem ser utilizados os símbolos '#' e 'b' após uma nota para representar o sustenido e bemol respectivamente. Por exemplo, a nota sol sustenido pode ser indicada como Sol# ou G#.

Adicionar um sustenido em uma nota significa aumentar a altura da nota em um semitom. No caso de um piano, isto significa mover uma tecla à direita. Similarmente, o bemol diminui a altura de uma nota em um semitom, ou move uma tecla do piano à esquerda. Na Figura 3, a primeira tecla preta pode ser chamada tanto como dó sustenido por estar um semitom acima da nota dó, ou de ré bemol por estar um semitom abaixo da nota ré. O nome da nota vai depender do contexto musical.

No geral, tenta-se utilizar o mínimo de acidentes possíveis para representar as notas. No

caso de mi sustenido, a nota pode ser representada por fá, e dó bemol pode ser representado por si. Existem algumas situações em que essas notas seriam representadas do primeiro modo, mas se encontram fora do escopo deste projeto e, por isso, não serão citadas.

Na Figura 8 (a), pode ser observado um sustenido sendo aplicado em duas notas musicais por causa do seu efeito persistente. Na Figura 8 (b), as duas primeiras notas possuem a mesma altura apesar de serem representadas de maneira diferente. A última nota não é afetada pelo sustenido pois está em um outro compasso.

Nome	Acidente	Efeito
Sustenido	#	Aumenta a altura em um semitom
Bemol	b	Abaixa a altura em um semitom
Bequadro	—	Cancela outros acidentes
Dobrado sustenido	x	Aumenta a altura em um tom
Dobrado bemol	bb	Abaixa a altura em um tom

Tabela 4: Diferentes tipos de acidentes.

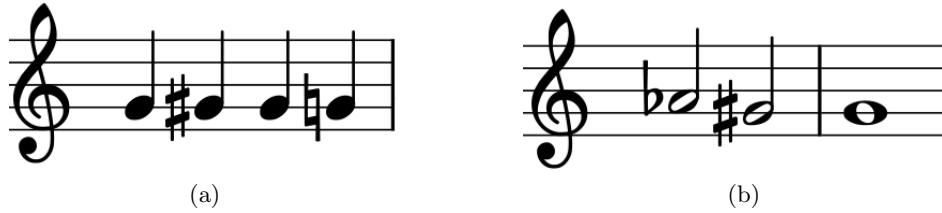


Figura 8: (a) Notas G, G#, G# e G; (b) Notas Ab, G# e G.

2.8 Armadura de clave

A armadura de clave é um conjunto de acidentes colocado ao lado da clave que, diferentemente dos acidentes que ocorrem durante um compasso, afetam as notas em todas as oitavas, e persistem por toda a pauta. Armaduras de claves não estão limitadas a aparecer ao lado de uma clave e podem ser utilizadas no começo de compassos para cancelar uma clave anterior e aplicar novos acidentes.

Na armadura de clave no primeiro compasso da Figura 9 estão presentes sustenidos nas notas fá, dó e sol. Isto significa que todos essas notas são aumentadas em um semitom, independente de sua oitava. A primeira nota exibida da imagem é um fá, entretanto, por causa da armadura de clave, torna-se um fá sustenido. A segunda armadura de clave anula o efeito da primeira e coloca bemóis nas notas si e mi. A terceira anula os efeitos da segunda e não aplica nenhum acidente.

Acidentes não podem ser colocados em uma armadura de clave de qualquer forma. O método para inserir acidentes em uma armadura de clave está fora do escopo deste projeto,

e não será citado.



Figura 9: Pauta com armaduras de clave.

2.9 Articulação

Articulação refere-se ao modo de tocar as notas, que pode ser alterada com a utilização de símbolos musicais. Diferentes instrumentos possuem diferentes formas para criar as articulações. Instrumentos de cordas podem variar a velocidade do arco, o ponto de contato, e a pressão sobre as cordas. Pianos possuem pedais que alteram a forma que as notas soam.

Na Figura 10, podem ser vistos alguns dos símbolos de articulação: legato, staccato e acento. O símbolo de legato é o mesmo da ligadura, mas é utilizado para indicar que as notas devem ser tocadas de forma “ligada”, sem silêncio entre elas. O símbolo de staccato é o mesmo que o ponto de aumento, mas é utilizado na parte superior ou inferior da nota, indicando que a nota deve ser executada de maneira curta, com um período de silêncio entre as notas. O símbolo de acento indica que a nota deve ser executada com maior intensidade, destacando-se das outras.



Figura 10: Legato conectando as duas primeiras notas, staccato na segunda nota e acento na terceira.

3 Arquitetura OMR

Um sistema OMR recebe como entrada uma imagem digital contendo a partitura que será processada. As imagens geralmente são obtidas por meio de uma fotografia, captura de tela ou digitalização utilizando um *scanner*. Essas imagens passam por um fluxo que consiste tipicamente em quatro etapas:

- Pré-processamento: técnicas como binarização, remoção de ruído, desfoque, e remoção de distorção são utilizadas para transformar a imagem original para auxiliar as etapas seguintes da pipeline. Também nesta etapa são calculados comprimentos de referência como altura e espessura das linhas da pauta.

- Detecção de símbolos musicais: recebe como entrada a imagem após a realização do pré-processamento. Tipicamente, possui como subetapas a detecção das linhas de pauta, remoção das linhas, detecção e classificação de símbolos musicais.
- Reconstrução da notação musical: a partir de regras sintáticas e gráficas, são atribuídos valores musicais aos símbolos detectados na etapa anterior. Esta etapa também pode resolver problemas com símbolos ambíguos.
- Construção da representação final: é gerado um arquivo contendo as informações musicais obtidas na etapa anterior. Os formatos de arquivos mais comuns incluem MIDI e MusicXML.

3.1 Pré-processamento

O pré-processamento é uma etapa muito importante para todos os sistemas OMR, pois ajusta a imagem para que se torne mais fácil de ser analisada e também extrai informações importantes como posição, espessura, e distância das linhas da pauta. A técnica mais comumente aplicada é a binarização de imagem. Outras técnicas incluem remoção de ruído [6, 8], desfoque [6], remoção de distorção [5, 6, 8] e operações morfológicas [5, 6]. Para extrair informações referentes as linhas da pauta geralmente, utiliza-se a codificação RLE (*run length encoding*) [7, 11, 14, 15].

3.1.1 Binarização

Binarização, segmentação binária ou segmentação por limiarização é uma técnica utilizada para transformar uma imagem colorida ou preto e branco, em uma imagem binária, reduzindo a quantidade de informação que deve ser processada. Isso faz com que os algoritmos das próximas etapas não necessitem ser tão complexos e também diminui significativamente o tempo total de processamento. Apesar das vantagens citadas, a binarização também pode introduzir alguns problemas como segmentação dos símbolos musicais e acentuar a presença de ruído.

Os métodos de binarização calculam um limiar utilizado para decidir se os pixels devem se tornar pretos ou brancos. Pixels com intensidade acima do limiar tornam-se brancos, enquanto aqueles com intensidade menor tornam-se pretos. Os métodos são ditos globais quando apenas um limiar é calculado para toda a imagem e são chamados de adaptativos ou locais quando calculam vários limiares para serem utilizados em diferentes regiões da imagem.

Um dos métodos de binarização global mais utilizado em sistemas OMR é o método de Otsu [3]. Este método calcula um limiar para separar os pixels em duas classes: primeiro plano e segundo plano. O limiar é calculado de modo que minimize a variação de intensidade intra-classe dos pixels. É um método bastante conhecido, rápido e eficiente. Um problema comum de métodos de binarização global ocorre quando a imagem possui uma iluminação não uniforme, fazendo com que regiões escuras tornem-se totalmente pretas.

Métodos de binarização adaptativos determinam um limiar para cada região ou pixel individual a partir da intensidade dos pixels vizinhos. A vantagem desses métodos é que

regiões escuras da imagem não se tornam completamente pretas como no caso dos métodos globais. Um dos métodos adaptativos mais utilizado é o método de Niblack, que utiliza a média e desvio padrão dos pixels vizinhos para determinar o limiar do pixel. Uma variável importante que deve ser considerada ao utilizar métodos adaptativos é o tamanho da janela, que deve ser adaptada para produzir resultados mais adequados.

Na Figura 11, podem ser vistos os resultados do método global de Otsu e dos métodos adaptativos de Niblack e Sauvola com uma janela de 25 pixels. Pode ser observado uma grande quantidade de ruído no método de Niblack pelo fato da janela ser muito pequena para a imagem. O método de Sauvola é uma adaptação do método de Niblack e produz um resultado com menos ruído.

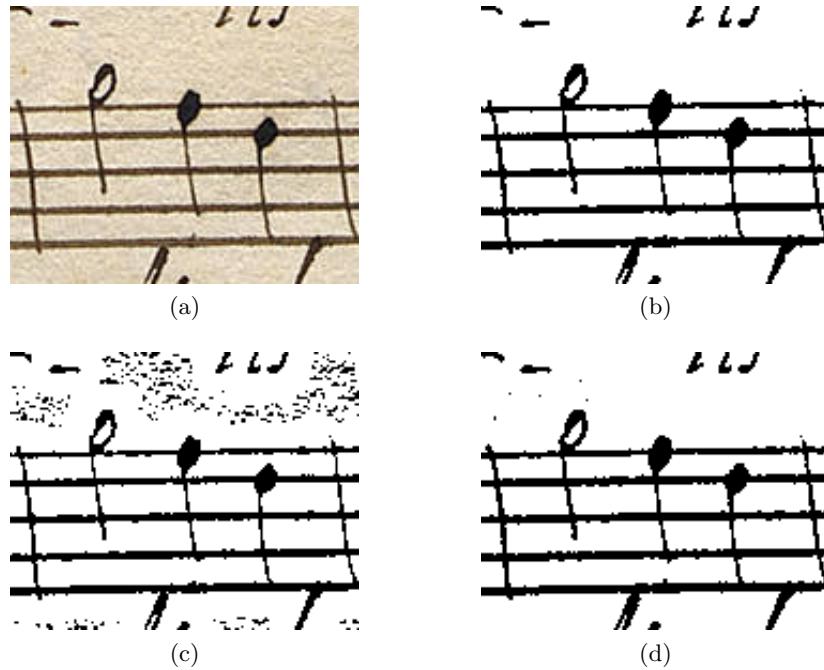


Figura 11: Binarização de imagens. (a) Original; (b) Otsu; (c) Niblack; (d) Sauvola.

3.1.2 Comprimentos de referência

A espessura das linhas de pauta e a distância entre as linhas são chamadas de comprimentos de referência e são medidas utilizadas por diversos sistemas OMR.

Um dos métodos mais comuns para determinar os comprimentos de referência é utilizando RLE (*run-length encoding*), que é uma forma simples de compressão sem perda de dados. RLE codifica uma sequência de valores repetidos em um único valor seguido do número de caracteres em sequência. Como imagens binárias possuem apenas dois valores diferentes de pixels, a codificação RLE pode ser ainda mais compacta, contendo apenas o comprimento das corridas. A sequência de pixels [111001100001011] pode ser codificada em [3,2,2,4,1,1,2] (assumindo primeiro valor como 1) ou [0,3,2,2,4,1,1,2] (assumindo primeiro

valor como 0).

Para determinar os comprimentos de referência utilizando RLE, a imagem é codificada verticalmente e da codificação é extraído o valor de corrida de pixels pretos mais comum, e corrida de pixels brancos mais comum, que correspondem aos valores de espessura das linhas da pauta e distância entre as linhas da pauta, respectivamente. O problema desta abordagem é que o resultado pode ser incorreto em situações em que a imagem possui muito ruído, ou em que as linhas da pauta não estejam muito bem definidas.

Uma outra técnica para se obter os comprimentos de referência é utilizando projeções horizontais. Para se obter a projeção horizontal de uma imagem binária, basta realizar a contagem do número de pixels presentes em cada linha da imagem. Os comprimentos de referência podem ser obtidos fazendo ao análise os picos locais observados na projeção. A Figura 12 ilustra uma partitura e sua projeção horizontal.



Figura 12: (a) Trecho da quinta sinfonia de Beethoven; (b) Projeção horizontal da imagem.

3.2 Detecção de símbolos musicais

A detecção de símbolos musicais é uma etapa bastante complexa e sujeita a erros dada a natureza complexa das partituras. Fatores como alta densidade de símbolos, símbolos sobrepostos, diferenças entre partituras manuscritas e digitais, bem como diferentes formas de representar símbolos por cada compositor, tornam o processo de detecção bastante complicado de se obter uma alta precisão. Nesta etapa, alguns métodos são aplicados, tais como projeções [10, 9, 11], *stable paths* [13, 14, 15], *line track height* (LTH) [14, 15] para detecção e remoção de linhas de pauta e redes neurais [17, 12, 18] para a detecção e classificação de símbolos.

3.2.1 Detecção e remoção das linhas de pauta

Em muitos sistemas OMR, a etapa de detecção e remoção das linhas de pauta é importante pois facilita a identificação dos símbolos musicais nas etapas seguintes. Uma das dificuldades encontradas é que as linhas da pauta podem estar distorcidas por causa de dobrões no papel,

má digitalização ou má fotografia. No caso de partituras manuscritas, a tinta pode estar esvaecida dificultando a detecção da pauta. Outra dificuldade encontrada nesta etapa é garantir que os símbolos não sejam segmentados acidentalmente durante o processo.

Apesar de ser uma etapa essencial para muitos sistemas OMR, esta etapa pode ser opcional na situação em que redes neurais são utilizadas para realizar a detecção dos símbolos, desde que tenham sido treinadas com imagens contendo as linhas de pauta. Pacha e Eidenberger [12] utilizaram CNNs para classificar símbolos musicais com e sem linhas de pauta, obtendo resultados similares nas duas situações.

A forma mais simples para realizar a detecção das linhas de pauta é utilizando projeções horizontais, de onde podem ser extraídas as posições das linhas ao encontrar a posição os picos locais da projeção. Este método requer que a pauta esteja perfeitamente horizontal, caso contrário, pode acabar produzindo resultados inadequados. Uma pauta pode ser alinhada horizontalmente aplicando-se pequenas rotações incrementais na imagem e selecionando aquela que possui o maior pico em sua projeção horizontal. Uma alternativa às projeções horizontais é o método de Hough, que é uma técnica matemática que capaz de detectar linhas e outras formas geométricas em imagens digitais.

No método proposto por Cardoso et al. [13], as linhas da pauta são detectadas pelo algoritmo de *stable paths*. Nesse algoritmo, cada pixel da imagem é representado por um nó que é conectado aos oito nós dos pixels de sua vizinhança-8 e o algoritmo busca achar os caminhos entre o lado esquerdo até o lado direito da imagem que possuem o menor peso. Para arestas que conectam algum pixel preto, um peso baixo é atribuído, enquanto pesos altos são atribuídos às outras. Além disso, arestas que seguem longas corridas de pixels pretos tem seu peso reduzido pois provavelmente pertencem a linhas de pauta. Esse método também utiliza os comprimentos de referência para melhorar a escolha dos pesos das arestas.

Após a detecção, a remoção das linhas de pauta pode ser realizada com técnicas como LTH, RLE e esqueletização. LTH percorre as linhas de pauta removendo as corridas verticais de pixels que são menores que um limiar, que geralmente é calculado a partir da espessura da linha de pauta. Na et al. [15] utilizaram um método em duas etapas, em que a primeira utiliza o algoritmo LTH com um limiar de 2 vezes a espessura local da linha de pauta. Na segunda etapa, as linhas de pauta são percorridas novamente e, realizando uma análise de componentes conexos, os componentes com um número de pixels menor que um limiar são removidos. A segunda etapa deste método é bastante eficiente na remoção de ruído gerado pela primeira etapa.

Na Figura 13, podemos observar uma partitura com e sem linhas de pauta. As linhas suplementares não são removidas nos métodos mais comuns encontradas na literatura. Os símbolos musicais tornam-se mais fáceis de serem identificados, permitindo algoritmos menos complexos nas etapas futuras.

3.2.2 Detecção de símbolos primitivos

Há diversas técnicas para a detecção de símbolos musicais propostos na literatura, mas com o aumento da capacidade de processamento dos computadores e a popularização de técnicas de aprendizado profundo, grande parte dos sistemas OMR modernos utilizam redes neurais

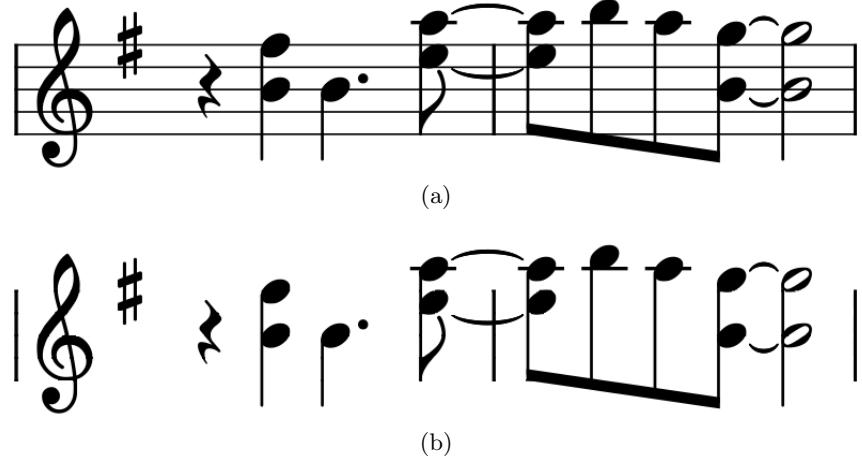


Figura 13: (a) Imagem com linhas de pauta; (b) Remoção de linhas de pauta ideal.

convolucionais (CNNs) para esta etapa de detecção. O objetivo desta etapa é determinar a posição e tipo dos símbolos musicais encontrados em uma imagem. Na maioria dos casos, os símbolos musicais são detectados em sua forma mais primitiva, detectando-se bandeirolas, hastes e cabeças de notas ao invés de notas inteiras.

A decomposição hierárquica é um método encontrado em vários sistemas OMR. Bellini et al. [10] utilizaram a projeção horizontal para detectar e separar as pautas de uma imagem. Cada pauta é analisada separadamente por uma janela deslizante, que reparte as pautas em regiões que contém grupos de notas. Cada região tem sua projeção vertical analisada para realizar a separação em símbolos individuais. Projeções horizontais são novamente utilizadas em cada símbolo para detectar o número de bandeirolas.

Rossant e Isabelle [16] desenvolveram um método para detectar símbolos musicais utilizando *template matching*. Segmentos verticais são detectados na partitura ao analisar corridas de pixels pretos verticais e são utilizados para detectar a posição e comprimento das hastes das notas. Para encontrar a cabeça das notas, as regiões em volta da haste são examinadas para ver se contém regiões que combinam com os *templates* de cabeças de notas. As bandeirolas são detectadas ao aplicar transformadas de Hough modificadas próximas às extremidades das hastes. Após a detecção dos símbolos primitivos, um modelo *fuzzy* é utilizado junto com regras sintáticas para determinar o tipo de nota mais provável.

A utilização de redes neurais para detecção de símbolos musicais está sendo bastante frequente em sistemas OMR mais modernos. Redes neurais podem ser treinadas para detectar os símbolos mesmo na presença de linhas de pauta. Pacha e Calvo-Zaragoza [17] utilizaram CNNs para realizar a detecção de notas musicais escritas em notação mensural, que é um sistema de notação utilizado na Europa do século 13 até o século 16. Este sistema é escrito de forma monofônica, ou seja, apenas uma nota é tocada por vez. Além disso, o número de símbolos diferentes é muito menor do que o sistema de notação moderno, que torna esta etapa consideravelmente mais simples. O modelo atingiu um w-mAP (*weighted mean average precision*) de 76% e possuí uma precisão baixa (10-40%) no caso de classes com poucas amostras e símbolos pequenos. Pacha et al. [18] realizaram experimentos com

diferentes métodos de aprendizado de máquina profundo, tais como Faster R-CNN, R-FCN e *single shot detectors* para realizar a detecção de símbolos primitivos, que obtiveram um medida w-mAP de até 94%. Os experimentos foram realizados em imagens com e sem linhas de pauta, atingindo valores de precisão praticamente iguais, o que sugere que sistemas OMR não necessitam mais da etapa de remoção de linhas de pauta.

Em partituras para músicas cantadas, a letra da música também pode estar presente debaixo da pauta como pode ser vista na Figura 15. Dinh et al. [11] utilizaram métodos como RLE e *stable paths* para obter informações sobre as linhas de pauta. Regiões entre as linhas de pauta potencialmente contêm a letra da música e são analisadas utilizando projeções horizontais, RLE e um conjunto de regras gráficas utilizadas para filtrar as regiões de interesse. Métodos da área de OCR também podem ser utilizados para realizar a detecção e classificação dos caracteres presentes na letra da música.

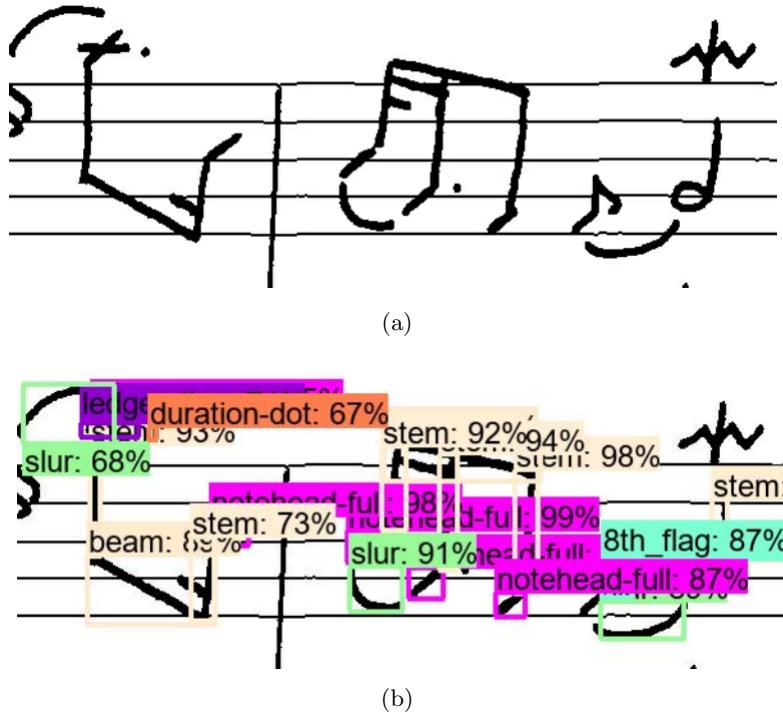


Figura 14: Detecção de símbolos primitivos. Imagem extraída de [18].



Figura 15: Trecho da ópera “A Flauta Mágica” de Mozart.

3.3 Reconstrução da notação musical

A reconstrução da notação musical tem como objetivo atribuir um significado musical para todos os símbolos detectados. Esta é uma tarefa consideravelmente difícil dada a quantidade de símbolos que interagem entre si, mesmo estando espacialmente distantes. Esta etapa pode corrigir alguns erros de classificação de símbolos com mesma representação gráfica. Normalmente, são adotadas regras gramaticais, espaciais e musicais para se construir um modelo que será utilizado para montar a representação final.

Durante a etapa de reconstrução, regras musicais devem sempre estar sendo avaliadas no momento de se atribuir um significado musical aos símbolos. Fórmulas de compasso provém informações importantes sobre a duração dos compassos, que pode ser utilizada para detectar a falta ou excesso de símbolos musicais. Símbolos iguais como ligadura e legato; ponto de aumento e staccato; podem ser distinguidos através da análise de símbolos próximos e da aplicação de regras. É importante ressaltar que há situações em que compositores não seguem as regras musicais à risca, omitindo símbolos em favor de uma melhor legibilidade da partitura ou até mesmo colocando mais ou menos notas que o exigido em um compasso.

Para se determinar o espaço ou linha que uma nota se encontra na pauta, a posição vertical das notas pode ser comparada com as posições das linhas de pauta obtidas nas etapas anteriores. Uma forma alternativa foi apresentada por Pacha e Calvo-Zaragoza [17], que utiliza CNNs. Ao detectar uma nota musical, a porção da imagem que contém a nota musical é extraída e fornecida como entrada para uma CNN que faz uma predição de qual espaço ou linha a nota está localizada.

Brisset [19] utilizou regras gramaticais que são separadas em nível gráfico e nível sintático para descrever a relação entre símbolos primitivos. No nível gráfico estão regras referentes à posição espacial dos símbolos na partitura, descrevendo onde devem estar posicionados para formarem elementos de maior nível. Exemplos de regras do nível gráfico incluem: notas são formadas por cabeças de notas, e opcionalmente hastes e bandeirolas; notas podem ser afetadas por acidentes localizados perto horizontalmente da cabeça de nota; cabeças de nota podem estar na parte de cima ou de baixo da haste. No nível sintático encontram-se regras relacionadas a notação musical: compassos contém notas; o início da pauta possui um cabeçalho composto por clave, armadura de clave e fórmula de compasso; uma partitura é composta pelo cabeçalho e por diversos compassos.

Um método que utiliza aprendizado de máquina foi utilizado por Pacha et al. [20] para determinar se símbolos primitivos relacionam-se ou não. Cada símbolo detectado é modelado como um vértice em um grafo, que pode ter arestas conectadas a outros vértices caso os símbolos se relacionem. Neste modelo, são definidas arestas sintáticas, que conectam dois símbolos que se relacionam sintaticamente como no caso de uma cabeça de nota e uma haste, bem como arestas de precedência, que conectam símbolos que se relacionam temporalmente, como duas notas em sequência. O modelo é treinado para determinar se a cada par de símbolos existe uma aresta, e qual o tipo da aresta. Para evitar testar todos os pares possíveis de vértices, são utilizadas algumas regras para filtrar vértices que podem se relacionar. Fatores como distância e tipo dos vértices são considerados para isso. O modelo atingiu boa eficácia, com uma precisão de até 93% ao determinar a relação entre símbolos detectados por um sistema de detecção de símbolos real.

3.4 Construção da representação final

A última etapa de sistemas OMR consiste na construção da representação final da música. Esta é uma etapa relativamente simples, que busca produzir como saída um arquivo musical como MIDI ou MusicXML. Arquivos de áudio tais como MP3, MP4, FLAC e WAV também podem ser possíveis formas de saída de sistemas OMR, mas não são muito comuns na literatura, dado que essas extensões não permitem a extração da informação musical de forma fácil como arquivos designados especialmente para isso, tais como MIDI e MusicXML.

MIDI (*Musical Instrument Digital Interface*) é um formato de arquivo que não contém áudio, mas sim, dados que representam as notas de uma música, de forma similar a partituras. Informações musicais como tom, dinâmicas, duração e andamento são armazenadas no arquivo de forma seja facilmente lida por outros pacotes, que podem utilizá-los para gerar partituras digitais ou arquivos de áudio com a música sendo tocada por diversos instrumentos diferentes.

MusicXML é um formato de arquivo baseado em XML que foi criado para representar partituras de uma forma que seja fácil de ser lida e escrita por um computador. É um formato amplamente utilizado por diversos softwares para a criação e edição de partituras digitais. Assim como MIDI, MusicXML não contém áudio.

4 Metodologia e Resultados

Esta seção descreve os experimentos realizados com métodos das diferentes etapas de um sistema OMR visando desenvolver um software capaz de realizar a leitura de partituras. O código associado a este projeto está disponível em um repositório do [github](#) [1]. Para a etapa de detecção de símbolos musicais, será utilizado um modelo de rede neural treinado para a detecção de símbolos em partituras digitais em [21], disponível em [22].

4.1 Binarização

Para esta etapa, foram testados métodos locais e globais de binarização em imagens de diferentes dimensões, contendo partituras tanto manuscritas como digitais, obtidas pela digitalização, fotografia e captura de tela. Dado que métodos de binarização são difíceis de ser avaliados numericamente, foram avaliados empiricamente através da análise visual das imagens produzidas. Os métodos de binarização utilizados estão disponíveis na biblioteca `scikit-image` para a linguagem Python.

O método global de Otsu, e os métodos adaptativos de Niblack e Sauvola foram utilizados para a realização dos experimentos. O método global de Otsu produziu bons resultados no geral independente do tamanho da imagem, mas teve problemas em imagens com iluminação não uniforme, que é um problema que ocorre em todos métodos globais. Os métodos adaptativos por outro lado, foram capazes de filtrar as regiões sombreadas quando utilizando um tamanho de janela adequado para o tamanho da imagem. A principal desvantagem dos métodos adaptativos é que se o tamanho da janela for muito pequeno, uma grande quantidade de ruído pode ser inserida na imagem, e caso a janela seja muito grande, os métodos não produzem bons resultados em imagens regiões com iluminação não

uniforme. Na Figura 16, pode ser vista a diferença ao utilizar diferentes tamanhos de janela. O método de Sauvola e o método de Otsu produziram resultados similares quando em condições de iluminação uniforme.

Imagens obtidas por meio de *scanners*, capturas de tela e câmeras fotográficas possuem uma grande variabilidade nas suas dimensões, fazendo com que métodos adaptativos não se comportem de forma ótima em todas as imagens quando se utiliza um tamanho de janela fixo. Entre escolher um tamanho de janela grande e um tamanho de janela pequeno, é desejável que se escolha um tamanho grande pois os métodos adaptativos podem produzir imagens com muito ruido e com características indesejáveis em situações em que o tamanho de janela é muito pequeno, como pode ser visto na Figura 17.

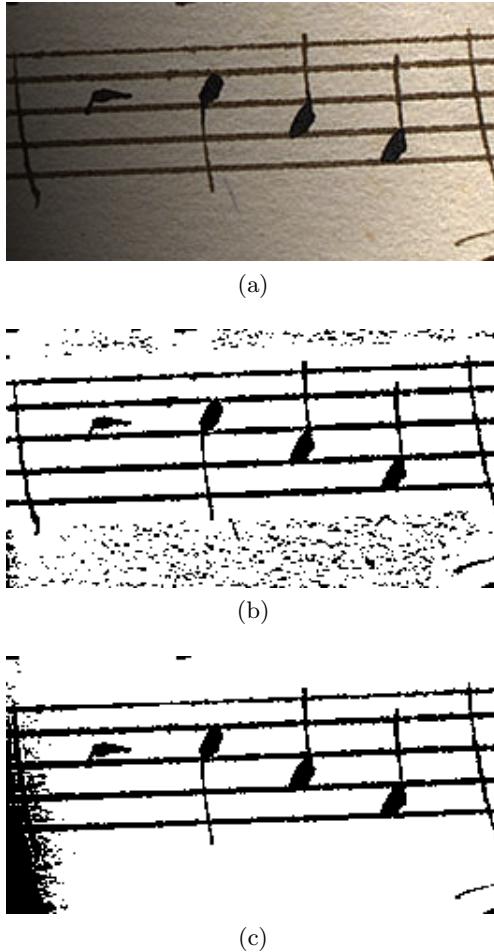


Figura 16: Método de Niblack. (a) Original; (b) Janela pequena; (c) Janela grande.

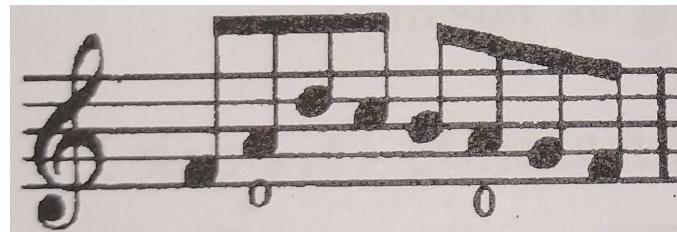
A tinta utilizada para imprimir uma partitura pode, por vezes, refletir o *flash* de uma câmera fotográfica fazendo com que pontos claros apareçam na imagem, que acabam se tornando branco após a aplicação da binarização (Figura 18). Foram realizados experimentos aplicando operadores morfológicos de fechamento e filtros de mediana antes da etapa



Figura 17: Método de Sauvola aplicado em uma imagem utilizando um tamanho de janela pequeno. O centro das cabeças de notas tornaram-se brancos. O tamanho de janela utilizado era adequado para imagens menores, mas não para esta.

de binarização na tentativa de remover os pontos brancos causados pelo *flash*. Ambos os métodos foram capazes de remover os pixels brancos, mas com a desvantagem de introduzirem problemas como a junção de símbolos musicais. Na Figura 19, pode-se observar que a aplicação do filtro da mediana resultou na junção de bandeirolas, desconectou a haste de uma das notas, fechou o espaço do número quatro e removeu um pouco de ruído.

Um resultado interessante, porém de pouca utilidade prática, pode ser visto na Figura 20, onde a aplicação do filtro de mediana removeu as linhas de pauta de uma partitura digital. As condições para que isso ocorra são muito específicas e praticamente qualquer outro método de remoção de linhas de pauta consegue remover as linhas de forma mais eficiente.



(a)



(b)

Figura 18: *Flash* criando pontos brancos na imagem. (a) Foto tirada com *flash*; (b) Método de Otsu.

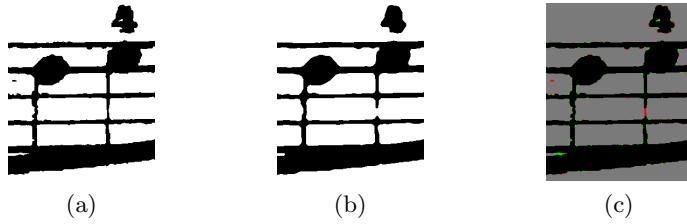


Figura 19: Comparação do método de Otsu aplicado em uma imagem não filtrada e filtrada por um filtro de mediana. (a) Sem aplicação do filtro; (b) Com aplicação do filtro; (c) Diferença entre as imagens. Os pixels que são adicionados ou removidos ao utilizar o filtro de mediana são mostrados em verde e vermelho, respectivamente. Os pixels pretos que não mudaram com a aplicação do filtro são mostrados em preto. Os pixels brancos que não mudaram são mostrados em cinza. A escolha da cor cinza é apenas para facilitar a visualização da cor dos pixels verde e vermelho.

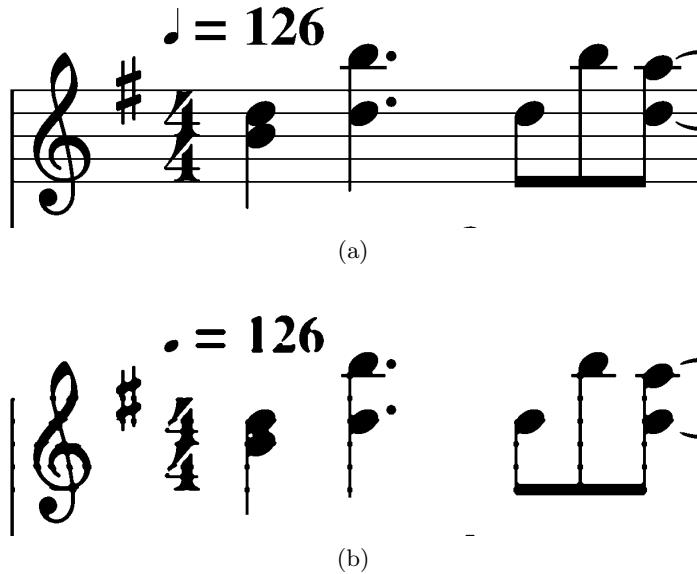


Figura 20: Filtro de mediana removendo linhas da pauta. (a) Imagem original (b) Imagem após aplicação do filtro.

4.2 Comprimentos de referência

Antes da aplicação de métodos para determinar os comprimentos de referência, como o RLE, é necessário que a pauta esteja o mais horizontal possível para extrair os valores corretamente. Para este propósito, foram geradas imagens com rotação em intervalos de 0.25 graus, de -10 até 10 graus, que tiveram suas projeções horizontais analisadas de forma a encontrar a rotação que maximiza o maior pico. As rotações foram aplicadas às imagens antes da execução do método de binarização e todas produziram bons resultados. Na Figura 21, uma imagem com uma partitura não horizontal é alinhada horizontalmente.

Após alinhar a imagem, os comprimentos de referência são extraídos por meio da contagem dos comprimentos de corridas verticais obtidos pelo método RLE, como discutido na Seção 3.1.2. O método foi capaz de determinar os comprimentos de referência corretamente em todas as imagens, mas é possível que obtenha valores incorretos em imagens com ruído ou em baixa qualidade.

4.3 Detecção de símbolos musicais

Para realizar a detecção dos símbolos musicais, foi utilizado o modelo CNN agnóstico disponível em [22], treinado para detectar símbolos em partituras digitais com a base de dados PrIMuS [23]. Um modelo CNN semântico também encontra-se disponível no mesmo local. A diferença entre o modelo agnóstico e semântico é que o modelo semântico detecta os símbolos e já atribui um significado musical a eles, enquanto o modelo agnóstico apenas indica qual símbolo foi detectado e qual sua posição.

O modelo escolhido foi treinado para detectar símbolos em imagens de partituras digitais, mas é capaz de detectar símbolos em fotos binárias com uma menor precisão, desde que a partitura, imagem e a imagem binária sejam adequadas. Notas musicais são detectadas de forma completa ao invés de serem divididas em símbolos menores como bandeirolas, cabeça de notas e hastes. O modelo foi desenvolvido para detectar notas em partituras monofônicas e, portanto, não é capaz de detectar símbolos em imagens que possuem mais de uma nota tocada por vez. Como saída, é produzida uma lista de caracteres que contém o tipo do símbolo e em qual linha ou espaço o símbolo está localizado como por exemplo “note.quarter-L3”, que indica que uma semínima foi detectada na linha três da pauta. É importante notar que o modelo não detecta retângulos envolventes (*bounding boxes*).

A base de dados PrIMuS consiste de imagens binárias de partituras digitais de alta qualidade e dimensão. A detecção dos símbolos utilizando imagens da base é praticamente perfeita, atingindo até 99% de precisão. Em imagens geradas pelo software Musescore [24], imagens obtidas através de capturas de tela e fotografias de partituras impressas, o modelo frequentemente classificou erroneamente o número de bandeirolas de notas e não conseguiu detectar alguns símbolos como acidentes. A Figura 22 ilustra essa situação. No caso de fotos de partituras impressas, a presença de ruído e pequenas imperfeições na impressão e/ou fotografia fazem com que a imagem binária não fique perfeita assim como as partituras da base PrIMuS. No caso de capturas de tela e imagens geradas pelo software Musescore, a diferença na resolução das imagens, os artefatos introduzidos pelo método de binarização e a diferença da fonte utilizada para gerar as imagens das partituras podem ser as principais causas dos erros.

4.4 Reconstrução da notação musical e representação final

A reconstrução da notação musical foi realizada aplicando-se regras espaciais e musicais para determinar a altura e duração das notas. Como o modelo utilizado para a etapa de detecção de símbolos não produz retângulos envolventes, alguns símbolos são impossíveis de terem seu significado extraído corretamente. Um acidente localizado no começo de uma pauta na maioria das vezes pertence à armadura de clave, mas também é possível que este

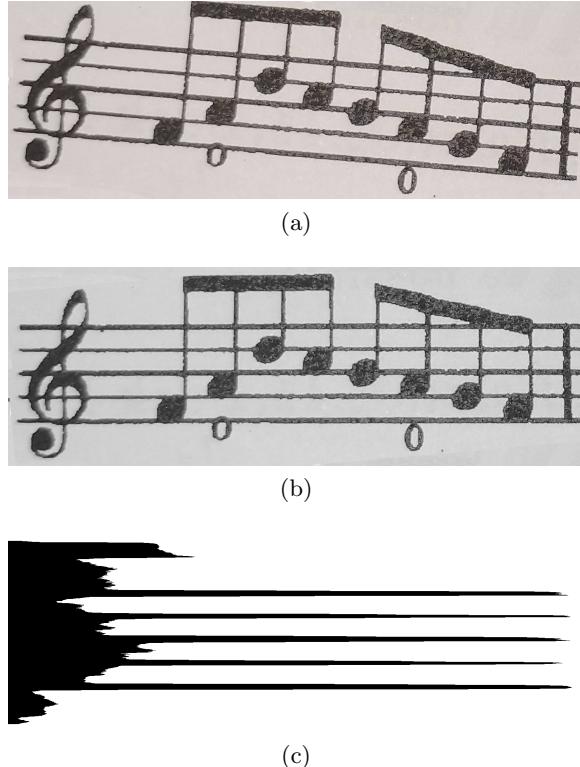


Figura 21: Alinhamento da pauta. (a) Imagem original; (b) Imagem em escala cinza após a rotação; (c) Projeção horizontal da imagem após rotação.

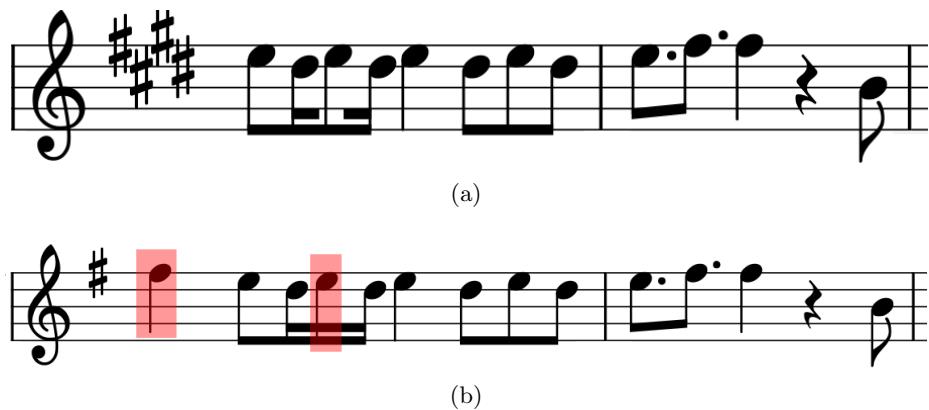


Figura 22: Detecção de símbolos musicais. (a) Imagem utilizada; (b) Símbolos detectados pelo modelo; Símbolos classificados erroneamente estão marcados em vermelho. Alguns acidentes da armadura de clave também não foram detectados.

acidente esteja atrelado à primeira nota da pauta. Em ambos os casos, a saída do modelo agnóstico é a mesma e não é possível distingui-las. Caso fosse utilizado o modelo semântico,

seria possível diferenciar essas duas situações. Esta distinção também seria possível na situação em que o modelo indicasse os retângulos envolventes para cada símbolo, permitindo que a distância entre os símbolos fosse analisada e, por meio de algumas regras espaciais, determinar se o acidente pertence à nota ou à armadura de clave.

Para prevenir a situação em que o modelo não seja capaz de detectar símbolos essenciais como clave e fórmula de compasso, foram definidos como padrão a clave de sol e a fórmula de compasso 4/4 por serem os mais utilizados. O andamento da música também teve um valor padrão definido, pois o modelo utilizado não foi projetado para fazer a detecção de texto ou outras indicações de andamento.

A representação final da partitura foi feita em um arquivo MIDI, que tem como extensão “.mid”, utilizando a biblioteca MIDIUtil. Construir o arquivo a partir das informações obtidas pelas outras etapas é uma tarefa bem simples e direta. A ordem, duração e altura das notas já foram determinadas e são facilmente escritas no arquivo de saída. O arquivo final foi testado em pacotes como Musescore e Synthesia [25], que conseguiram ler e reproduzir a música contida no arquivo corretamente.

Imagens da base PrIMuS foram detectadas e convertidas perfeitamente na maioria dos casos. Algumas partituras não seguiam as regras musicais à risca e continham compassos com maior ou menor duração que o necessário. Esse fator, em conjunto com limitações da biblioteca MIDIUtil e do formato de arquivo MIDI, impossibilitaram que essas imagens fossem convertidas perfeitamente. Fotografias de partituras e imagens obtidas por meio de capturas de tela, após a conversão, frequentemente possuíam notas com número de bandeirolas incorreto. Este é um problema que decorre da etapa de detecção dos símbolos musicais e pouco pode ser feito nesta etapa para realizar a correção dos símbolos.

5 Conclusões

Neste projeto, estudos foram realizados com diferentes técnicas computacionais aplicadas na área de OMR e um sistema OMR foi desenvolvido para realizar a conversão de imagens contendo uma partitura para um arquivo MIDI. O sistema ainda possui limitações, sendo capaz de converter apenas imagens que contém partituras monofônicas.

Muitos dos problemas encontrados durante o desenvolvimento do projeto estão na etapa de detecção de símbolos musicais. A detecção de símbolos completos, como é realizada neste projeto, é bastante limitada, pois requer que notas musicais estejam isoladas, impossibilitando a detecção de múltiplas notas tocadas ao mesmo tempo. Para a construção de um sistema OMR mais completo e robusto, a etapa de detecção poderia ser alterada para que fossem detectados símbolos primitivos ao invés de notas completas. Essa mudança tornaria a etapa de reconstrução da notação musical mais complexa, entretanto, como vantagem, a conversão de músicas polifônicas se tornaria possível.

Os métodos para detecção de linhas de pauta discutidos neste relatório acabaram não sendo utilizados de forma completa, dado que o modelo utilizado para a detecção dos símbolos já indicava a linha ou o espaço em que se encontravam. Caso fosse utilizado um modelo que indicasse retângulos envolventes ao invés da linha ou espaço do símbolo, os métodos de detecção de linhas de pauta teriam sido utilizados para determinar a posição

dos símbolos na pauta. A vantagem da utilização dos retângulos envolventes encontra-se na possibilidade de analisar as posições dos símbolos mais precisamente, que poderia resolver os problemas de símbolos ambíguos encontrados na última etapa.

Neste projeto, o formato MIDI foi escolhido para construir a representação final, entretanto, dadas algumas limitações do formato, nem todas as informações contidas na partitura puderam ser representadas no arquivo final. O formato MusicXML foi criado com o intuito de representar partituras e provavelmente seria uma melhor escolha para a representação final, dado que permite a adição de detalhes que não podem ser representados em um arquivo MIDI.

Referências

- [1] <https://github.com/erickfurukawa/OMR-PFG>, 2021.
- [2] <https://www.musicca.com/pt/termos-musicais>, 2021.
- [3] Otsu, N. (1979). A Threshold Selection Method from Gray Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9, 62-66.
- [4] Stathis, P., Kavallieratou, E., Papamarkos, N. (2008). An Evaluation Technique for Binarization Algorithms. *Journal of Universal Computer Science*, 14, 3011-3030.
- [5] Calvo-Zaragoza, J., Barbancho, I., Tardon, L., Barbancho, A. (2014). Avoiding Staff Removal Stage in Optical Music Recognition: Application to Scores Written in White Mensural Notation. *Formal Pattern Analysis & Applications*, 18, 1-6.
- [6] Goecke, R. (2003). Building a System for Writer Identification on Handwritten Music Scores. *International Conference on Signal Processing, Pattern Recognition, and Applications*, 1-6.
- [7] Na, I., Kim, S.H., Nquyen, T. (2015). A Robust Staff Line Height and Staff Line Space Estimation for the Preprocessing of Music Score Recognition. *Journal of Internet Computing and Services*, 16, 29-37.
- [8] Fujinaga, I. (2004). Staff Detection and Removal. *Visual Perception of Music Notation: On-Line and Off-Line Recognition*. 1-39.
- [9] Fujinaga, I. (1988). Optical Music Recognition using Projections, Tese de Doutorado. McGill University.
- [10] Bellini, P., Bruno, I., Nesi, P. (2001). Optical Music Sheet Segmentation. *First International Conference on WEB Delivering of Music*, 183-190.
- [11] Dinh, C.M., Yang, H.J., Lee, G.S., and Kim, S.H. (2016) Fast Lyric Area Extraction from Images of Printed Korean Music Scores, *IEICE Transactions on Information and Systems*, vol. 99, no. 6, 1576–1584.

- [12] Pacha, A., Eidenberger, H. (2017). Towards Self-Learning Optical Music Recognition. 16th IEEE International Conference on Machine Learning and Applications, 795-800.
- [13] Cardoso, J., Capela, A., Rebelo, A., Guedes, C., Costa, J. (2009). Staff Detection with Stable Paths. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31, 1134-1139.
- [14] Ashley, A.G., Sujatha, C.N. (2017). Optical Music Recognition: Staffline Detection and Removal. International Journal of Application or Innovation in Engineering & Management, vol. 6, no. 5, 48-58.
- [15] Na, I., Kim, S.H., Nquyen, T. (2015). A Robust Staff Line Height and Staff Line Space Estimation for the Preprocessing of Music Score Recognition. Journal of Internet Computing and Services, 16, 29-37.
- [16] Rossant, F., Isabelle, B. (2007). Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection. EURASIP Journal on Advances in Signal Processing, 1-25.
- [17] Pacha, A., Calvo-Zaragoza, J. (2018). Optical Music Recognition in Mensural Notation with Region-Based Convolutional Neural Networks, 240-247.
- [18] Pacha, A., Choi, K.-Y., Eidenberger, H., Ricquebourg, Y., Couasnon, B., Zanibbi, R. (2018). Handwritten Music Object Detection: Open Issues and Baseline Results. 13th IAPR International Workshop on Document Analysis Systems, 163-168.
- [19] Brisset, C. (1995). Using Logic Programming Languages For Optical Music Recognition. Third International Conference on The Practical Application of Prolog, 1-20.
- [20] Pacha, A., Calvo-Zaragoza, J., Hajic, J. (2019). Learning Notation Graph Construction for Full-Pipeline Optical Music Recognition. 20th International Society for Music Information Retrieval Conference, 75-82.
- [21] Calvo-Zaragoza, J., Rizo, D. (2018). End-to-End Neural Optical Music Recognition of Monophonic Scores. Applied Sciences, 8, 606.
- [22] <https://github.com/OMR-Research/tf-end-to-end>, 2021.
- [23] <https://grfia.dlsi.ua.es/primus/>, 2021.
- [24] <https://musescore.com/>, 2021.
- [25] <https://synthesiagame.com/>, 2021.