

Copista - Sistema de OMR para a recuperação de acervo histórico musical

Avner Maximiliano de Paulo¹, Flávio Luiz Schiavoni¹,
Marcos Antônio de Matos Laia¹, Daniel Luiz Alves Madeira¹

¹Departamento de Computação
Universidade Federal de São João Del Rei
São João Del Rei – MG – Brasil

fls@ufsj.edu.br, avnerpaulo.mg@gmail.com

marcoslaia@gmail.com, dmadeira@ufsj.edu.br

Abstract. *Optical Music Recognition (OMR) is a Computer Science field applied to Music and has opening problems to recognize handwritten scores. This paper presents a project called “Copista” that is investigating techniques to develop a software to recognize handwritten scores.*

Resumo. *Recuperação Musical Ótica (OMR) é uma área da Computação aplicada à Música com problemas em aberto para reconhecimento de partituras manuscritas. O presente artigo apresenta um projeto chamado “Copista” que está investigando técnicas com o intuito de desenvolver uma ferramenta computacional para o reconhecimento de partituras manuscritas.*

1. Introdução

Os municípios de São João del-Rei, Tiradentes e Prados possuem, juntas, uma história musical com raízes no início do século XVIII, e são depositárias de um dos mais relevantes conjuntos documentais, ao lado de Ouro Preto e Mariana, da história musical de Minas Gerais e do Brasil. A riqueza de gêneros vai da música sacra, passando pela ópera e música de concerto, ao repertório das bandas de música. Cronologicamente, possuem cópias musicais da segunda metade do século XVIII aos dias atuais, fruto do labor de uma centena de nomes de compositores, profissional este, sempre presente e atuante na história daquelas sociedades.

Neste cenário, o Departamento de Música (DMUSI) da Universidade Federal de São João Del Rei (UFSJ) está lançando o Programa Memória Viva, que tem como principal objetivo atuar junto a arquivos musicais em São João del-Rei, Tiradentes, Prados e São Tiago, visando o desenvolvimento e implementação de metodologias para descrição de documentos musicais (partituras e partes), que melhor atendam à peculiaridade de cada instituição. Muitas destas partituras são documentos históricos de valor inestimável e a sua disponibilização ao público depende de publicação de cópias físicas ou digitais. As cópias digitais idênticas da música original, de grande valor para a área da Musicologia, trazem em si uma dificuldade maior de leitura devido ao seu estado de conservação, conforme ilustra a Figura 1.

A criação de uma nova partitura reescrita a partir da partitura original traz a possibilidade de este acervo ser mais explorado, estudado, executado e difundido. Uma vez que

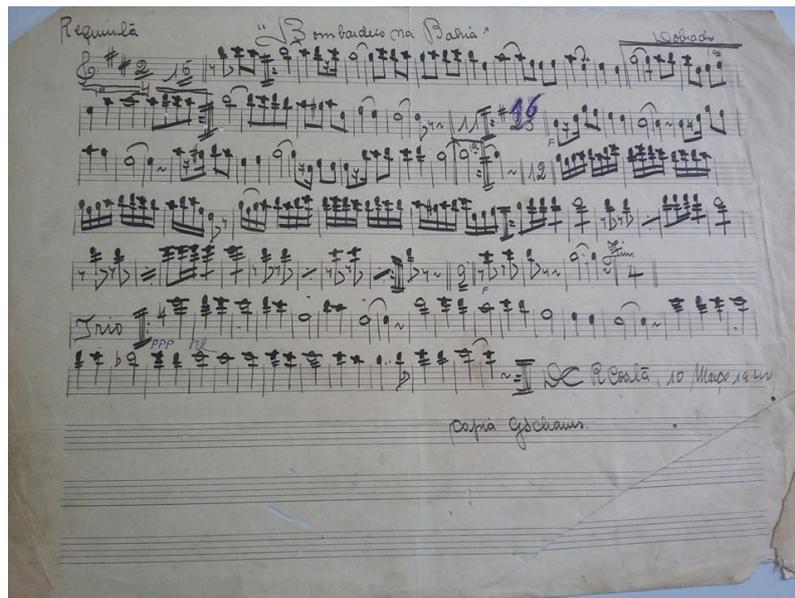


Figura 1. Exemplo de partitura presente no acervo

as peças musicais estejam catalogadas, pelo projeto Memória Viva, o próximo passo seria a conversão de tais partituras para um formato musical simbólico para a disponibilização do acervo ao público.

Como fruto de uma parceria entre o Departamento de Ciência da Computação (DCOMP) e o DMUSI, o presente projeto tem o intuito de prover uma ferramenta de amparo tecnológico para o Programa Memória Viva para a conversão destas partituras em formatos de música digital. Esta ferramenta é chamada **Copista**.

2. O Copista

O processo de conversão das partituras para um formato musical digital simbólico é também conhecido por OMR (Optical Music Recognition)[Rebello et al. 2012][Bainbridge and Bell 2001]. Apesar de existirem ferramentas que façam o reconhecimento de partituras a partir de uma imagem (OMR), a maior parte destas ferramentas a) não trabalha com reconhecimento de partituras manuscritas[Bainbridge and Bell 2001], b) possuem um alto custo de aquisição e c) não possuem código aberto de forma que a adaptação da mesma ao projeto em questão é impossível. Tais considerações foram levadas em conta para o início do desenvolvimento do projeto Copista.

O processo de OMR no Copista foi dividido em 3 partes distintas: o pré-processamento e restauro eletrônico das partituras originais, o reconhecimento dos elementos musicais por visão computacional e a representação digital dos elementos musicais reconhecidos, como ilustrado na Figura 2.

A intenção do projeto é que, a partir desta representação digital, será possível reescrever as partituras originais em programas de notação musical modernos de maneira a permitir a reedição destas partituras e a análise musical apoiada por computador das obras.

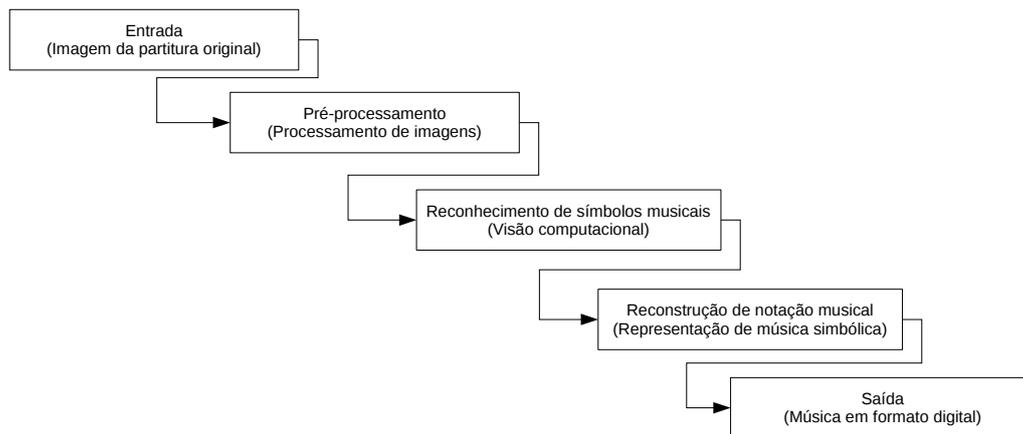


Figura 2. Estrutura básica de sistemas para OMR

2.1. Entrada

A entrada do Copista é a cópia digital das partituras que compõem o acervo regional. Nestes acervos, é comum que as partituras, muitas delas centenárias, tenham sido utilizadas em missas e procissões, e possuam dobras, marcas de cera de vela, amassados, rasgos e outros tipos de avarias, conforme pode ser visto na Figura 2.1:

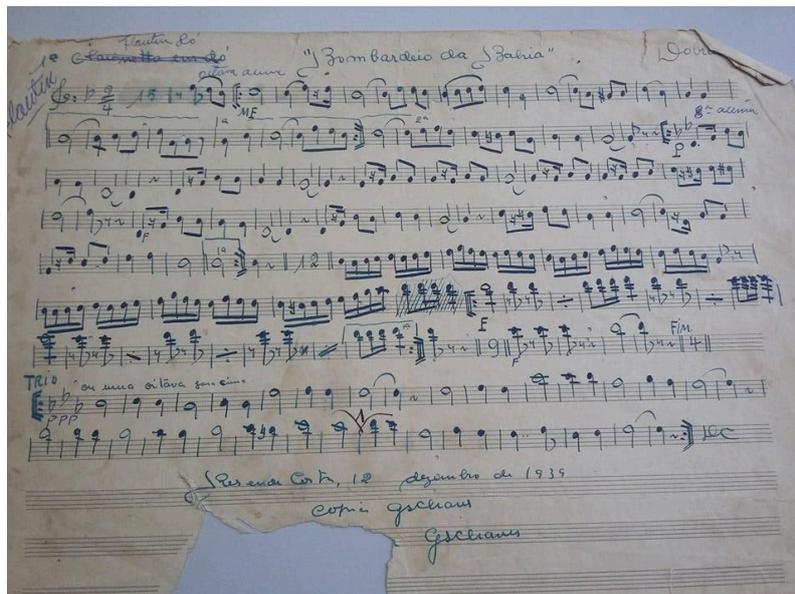


Figura 3. Partitura manuscrita e com avarias

Por esta razão, o Programa de Extensão deverá, primeiramente, tentar realizar o restauro físico das partituras dos acervos. Uma vez que este restauro for executado, a partitura deverá ser digitalizada para ser processado pelo Copista.

2.2. Pré-processamento

É comum que documentos muito antigos e manuscritos sofram de degradação ao longo do tempo. O manuseio constante pode fazer com que a tinta clareie, criando diferenças de tonalidades, ou pode criar diversas marcas. No exemplo das partituras, pingos de cera de

vela e marcas de suor de mãos ocorrem em vários casos. Além disso, o armazenamento inadequado provocou dobras, amassados, rasgos e até que traças fizessem buracos nas partituras. Todos esses artefatos não são relevantes à partitura e precisam ser retirados para, ao final do processo, obter a partitura o mais limpa possível, contendo somente os elementos necessários para o reconhecimento da mesma.

Não há uma técnica universal de pré-processamento, já que cada conjunto específico de documentos pode exigir tratamento diferenciado. Porém, dois passos podem ser destacados como básicos no processo de pré-processamento:

1. remoção de artefatos
2. limiarização

A primeira etapa envolve remoção de todo artefato não importante ao processo de reconhecimento da partitura. Estes artefatos, que se tornam ruído na imagem digitalizada, englobam as manchas, rasgos, furos e todas marcas que não são parte da partitura. O fundo do papel pode também ser considerado ruído devido ao fato de não ser importante para o reconhecimento da partitura em si. Para essa etapa, utiliza-se a filtragem da imagem original [Fujinaga 2004, Szwoch 2007]. Ao final deste processo, é necessária uma correção dos tons de cinza para ampliação do contraste, facilitando o próximo passo.

Com o ruído devidamente removido, a partitura deve ser convertida para preto-e-branco, através do processo de binarização (limiarização em dois tons de cinza). Este processo visa simplificar a representação da partitura, ao eliminar toda a variação de cores existente, e pode ser classificado em duas categorias: limiarização global ou local. Os métodos globais são classificados de acordo com um único limiar global [Otsu 1975, Pinto et al. 2011, Seixas et al. 2008], válido para toda a imagem. Ao utilizar um só limiar, os métodos tendem a ser mais simples e computacionalmente mais baratos. Porém ruídos que ocorram em somente parte da imagem irão influenciar na tomada de decisão do algoritmo, o que pode gerar resultados indesejáveis. Para contornar este problema, os métodos de limiarização local trabalham com subconjuntos da imagem de entrada, calculando o limiar ideal por região [Sauvola and Pietikäinen 2000, Bernsen 1986]. Nesses métodos há uma adaptatividade maior do método, ao permitir que a decisão sobre uma região dependa somente dela, independente das regiões vizinhas. Com isso, obtêm-se resultados melhores, mas também com custo computacional proporcionalmente maior.

Como nas partituras alvo do projeto temos artefatos como marcas de suor das mãos e pingos de vela, que não ocorrem em toda a área, os métodos locais tendem a ser mais indicados para o Copista. Nesta etapa então, o conjunto de técnicas de filtragem para remoção de diferentes ruídos e para binarização de imagens deverão ser avaliados. A avaliação dos resultados pode ser realizada através de partituras padrão, cujo conteúdo já é conhecido, em conjunto com a próxima etapa do Copista.

2.3. Reconhecimento de Símbolos Musicais

O Reconhecimento de Símbolos Musicais fará uso de técnicas de visão computacional determinadas em etapas específicas:

1. Segmentação de linhas e notações;
2. Localização das notações
3. Definição de descritores para cada notação

4. Reconhecimento de padrões das notações
5. Mapeamento das notas nas partituras
6. Identificação de propriedades específicas de notações musicais

A etapa de segmentação [Gonzalez et al. 2004] permite separar elementos como linhas e as demais notações para serem treinadas. As linhas podem ainda ser utilizadas para definir a localização de uma notação. Por exemplo, a altura da nota de acordo com seu posicionamento em relação às linhas, separar diferentes símbolos sobrepostos [Bainbridge and Bell 1997] e de diferentes tamanhos ou posições rotacionadas [Mundy et al. 1992].

Cada notação, por sua vez, pode ser descrita por um conjunto de *features* [Koendrik 1992] (características). Cada *feature* pode representar algo da imagem a ser reconhecida como bordas, curvaturas, bolhas, cumes e pontos de interesse. As *features* extraídas das imagens são utilizadas posteriormente em um processo de reconhecimento de padrões [Fukunaga 2013, Ripley 1996], após serem qualificadas e quantizadas para uma análise estatística por um filtro para diminuir incertezas entre as alturas de notas ou presença de artefatos nas imagens.

Nessa etapa pode ser utilizado um filtro de Kalman [Laia 2013] que permitirá a correção dos dados gerados pela extração de *features*. Ao combinar técnicas de visão computacional em OMR, haverá um ganho maior sobre como gerar esses dados e garantindo a integridade e fidelidade ao que está presente no documento.

Além disto, técnicas de visão computacional utilizadas para outras aplicações como reconhecimento de caracteres [Dori et al. 1996], reconhecimento de escrita à mão [Xu et al. 1992], realidade aumentada com marcadores em baixa resolução [Furht 2011] também serão utilizadas para completar esta etapa do processo.

2.4. Reconstrução de Notação Musical

No processo de OMR, a etapa de reconstrução da representação simbólica deve receber dados da visão computacional e mapeá-los para um alfabeto de símbolos musicais. Este mapeamento pode incluir uma validação de um determinado símbolo de modo a auxiliar a etapa de reconhecimento quanto à corretude de um determinado elemento gráfico com uma análise a partir do modelo notacional [Good 2001] ou baseada no contexto musical [Medina et al. 2003]. A validação pode ocorrer pela criação de um conjunto de regras léxicas, sintáticas e/ou semânticas que definirão o formato de representação simbólica.

Uma questão principal de definir uma representação musical simbólica é encontrar uma representação genérica suficiente, bastante flexível mas ao mesmo tempo restrita em relação à suas regras para permitir uma validação da estrutura musical como um todo [Selfridge-Field 1997].

A maior parte dos modelos existentes parte de uma estrutura musical hierárquica [Buxton et al. 1978] onde há uma visão geral da música, dividida em várias pautas (linhas), que são divididas em compassos e estes compassos em tempo e tempos em notas. Para este projeto, será adicionado ao modelo uma hierarquia ainda mais profunda onde serão incluídas informações sobre as pautas e a página da partitura. Uma possibilidade computacional de alcançar tal representação é utilizar um modelo orientado

a objetos [Travis Pope 1996], de maneira a definir a representação por um conjunto de objetos com atributos valorados.

Tais atributos valorados deverão armazenar a notação musical de um símbolo assim como registrar informações do símbolo dentro da imagem. Por esta razão, dividimos a representação simbólica musical para OMR em duas partes, uma que registra a informação musical e a outra que registra a informação da imagem.

Os dados valorados da imagem original em que foi encontrado um símbolo musical são necessários para permitir uma reavaliação de dados reconhecidos erroneamente. Isto permitiria requisitar a visão computacional que a validação de um determinado símbolo seja refeita para conferência, de maneira automática.

Outros dados da imagem original que poderão ser armazenados dizem respeito aos processamentos iniciais feitos na imagem. Armazenar informações como brilho, contraste, cor, rotação, translação, histograma e quais passos foram realizados para retirada dos artefatos torna-se necessário para que, se houver dificuldade na identificação ou caso o software identifique incorretamente um símbolo, confundindo com uma rasura ou furo na página e assim causando erro léxico, o pré-processamento possa ser reajustado através da alteração destes parâmetros, na tentativa de melhorar a qualidade na leitura da página.

Assim, temos uma cadeia de processamentos que poderão ser alteradas por um classificador baseando-se em acertos e erros de leitura. Ao digitalizar uma partitura de ponta-cabeça por exemplo, e identificar que uma transformação de rotação se faz necessário, o algoritmo irá armazenar esta operação como um atributo do elemento página e reprocessá-la hierarquicamente caso seja necessário reprocessar algum elemento.

Além disto, deverá ser possível converter os dados valorados armazenados pela ferramenta em algum formato de música simbólica existente.

2.5. Saída

Esta última etapa, a definição do formato de representação simbólica, é uma tarefa crítica no contexto do desenvolvimento desta ferramenta. Esta definição irá influenciar o desenvolvimento da ferramenta pois a validação dos símbolos reconhecidos dentro do modelo de representação pode auxiliar no algoritmo de aprendizado da etapa de visão computacional e com isto diminuir a necessidade de intervenção humana no processo de transcrição das partituras digitalizadas.

A definição da representação também é uma tarefa crítica pois a mesma irá prover um modelo para a geração da saída da ferramenta. A saída da ferramenta deverá ser o mais interoperável possível de maneira a permitir que haja possibilidade de edição e intervenção humana na correção de uma partitura gerada, caso isto seja necessário. A correção humana realizada em uma partitura com problemas de identificação pode servir como uma nova entrada no sistema pois permitiria uma nova etapa de aprendizagem para os algoritmos propostos.

Para a avaliação de adaptação será levado em conta a) o levantamento de todos os símbolos utilizados nestas partituras b) a representação computacional hierárquica deste conjunto de símbolos, c) as regras léxicas, sintáticas e semânticas do formato para permitir a correção de partituras em seu formato simbólico e d) a conversão deste conjunto de símbolos em formatos comumente utilizados em aplicações musicais.

Entre os formatos de música simbólica para a ferramenta estão:

- **ABC**[Oppenheim et al. 2010]: É uma linguagem para notação de músicas que utiliza caracteres no formato ASCII. É uma linguagem que pode ser lida e interpretada, mesmo sem a utilização de software, bastando a pessoa ter conhecimento musical para fazê-lo. Apesar de conseguir representar os símbolos presente em uma partitura, não é adequada para este projeto, pois é crucial que a linguagem para notação musical também represente a posição em que a partitura foi escaneada e ao invés de representar a nota que vai ser tocada, é necessário que seja armazenada a posição (X, Y) de cada símbolo na partitura digitalizada.
- **MusicXML**[Good 2001]: É uma linguagem para notação musical que utiliza formato XML. Diferente do formato ABC, MusicXML não é tão intuitiva. É necessário o estudo e prática da mesma, para que seja possível ser feita a leitura de uma musica escrita em MusicXML por uma pessoa. Consegue fazer uma ótima representação de uma partitura, com detalhes minuciosos. Porém também é inadequada para este projeto, pois apesar de fazer ótima representação de uma partitura, também não armazena a posição (X, Y) dos símbolos e nem a posição que a partitura foi digitalizada.
- **Lilypond**[Nienhuys and Nieuwenhuizen 2003]: Também em formato ASCII. Muito intuitiva para leitura, mas exige estudo para escrita a mão. Consegue representar uma partitura tão bem quanto o MusicXML, porém utilizando o nome das notas para tal representação, diferente do MusicXML. Por utilizar o nome das notas para representação e nenhum meio para definir sua posição (X, Y) na partitura, esse tipo de notação também é inadequado para o projeto.
- **Music21**[Ariza and Cuthbert 2011][Cuthbert and Ariza 2010]: Formato também baseado em ASCII. Esta notação oferece um modelo de objeto em linguagem Python. Esse kit de ferramenta foi desenvolvido para auxiliar na criação e manipulação de dados de música simbólica. Consegue armazenar e ordenar a posição e a duração dos elementos. As notas são criadas a partir de comandos. A nota em si, que é escrita como cifra, é passada como parâmetro. Contudo, esta linguagem de notação musical também é inadequada para o projeto, por não ter um meio para o armazenamento das características como, cor, brilho, contraste nem histograma da partitura.
- **GUIDO**[Hoos et al. 1998]: Também em ASCII. Representa utilizando a cifra das notas, seguido da duração da mesma, e utiliza um método de localização das notas por oitava, onde é colocado o respectivo número ao lado da nota. Também é inadequado ao projeto. Apesar de dispor de um método de localização, não é o método necessário para o projeto. Neste projeto é necessário que exista um meio de localizar a nota por meio de uma posição (X, Y) a partir da página. Além disso, este formato também não dispõe de um meio para definir a posição em que foi digitalizada a partitura.

Também foram pesquisados os formatos de música simbólica MIDI[Association et al. 1996] e NIFF (Notation Interchange File Format)[Grande 1997] mas tais formatos são binários e por isto considerados inadequados para este projeto. Quanto ao formato MuseData[Hewlett 1997], não foi encontrado referência suficiente sobre o mesmo.

3. Resultados obtidos

O processo de reconhecimento de partituras musicais é feito por meio de etapas que abrangem o pré-processamento da imagem (retirada de possíveis ruídos e artefatos), segmentação (separação de elementos nas imagens), detecção, classificação e reconhecimento dos elementos musicais.

Na primeira etapa, o pré-processamento irá preparar a imagem para o processo de visão de computacional. Para esta etapa, podem ser utilizados filtros que transformam a imagem colorida em escala de cinza, para posterior conversão em preto-e-branco, filtragem de ruídos (que podem ocorrer devido à manipulação da partitura ou do próprio processo de digitalização da mesma), detecção e retirada de artefatos, ou seja, defeitos ou presença de objetos que não pertencem originalmente às partituras. Nesta etapa, também é feita a correção da orientação das pautas como também a equalização da luz sobre a pauta. Concluindo esta etapa, é feita a binarização da imagem, onde há a possibilidade de presença apenas dos elementos da partitura.

Os testes preliminares do processamento da visão computacional utilizou partituras geradas digitalmente para uma série de testes de implementações de algoritmos. Por esta razão, nestes testes não foi necessário realizar o pré-processamento das imagens.



Figura 4. Partitura original digitalizada

Na segunda etapa, é feita a segmentação de elementos para serem detectados e posteriormente reconhecidos nas próximas etapas. A imagem binarizada (apresentada na Figura 4), oriunda da etapa de pré-processamento, recebe um processo de segmentação onde é retirada as linhas das partituras para que cada elemento se torne desconectado, ou seja, não apresente pixels que possam conectar um elemento ao outro.



Figura 5. Partitura após o processo de dilatação

Para este processo, utilizou-se técnicas de operações morfológicas, primeiro aplicando a dilatação com o objetivo de se eliminar as linhas. A dilatação é um processo onde a parte clara da imagem é expandida, comprimindo a parte escura. Pode-se ver o resultado dessa operação na Figura 5.



Figura 6. Partitura após o processo erosão

Com a dilatação, as notas se apresentaram mais delgadas, devido a perda de pixels escuros. Para retornar o estado anterior a esse processo, pode-se utilizar outra operação

morfológica chamada erosão. A erosão é um processo onde a parte clara sofre uma diminuição enquanto a parte escura aumenta, em um processo inverso à dilatação. O resultado pode ser visto na Figura 6. Após a retirada das linhas, chega-se na etapa de detecção dos elementos e etiquetagem. Cada grupo de pixels conectados recebe um único valor, que o diferencia dos demais.

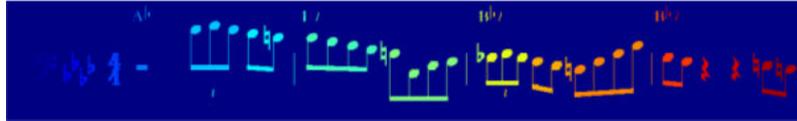


Figura 7. Partitura com os elementos etiquetados

Nesta etapa, os elementos podem ser selecionados um a um pelo valor ao qual é identificado. O fundo da imagem é apresentado com o menor valor (no caso da Figura 7, o valor é igual a 0) e o primeiro elemento com o valor consequente. Assim, se a imagem apresentar 200 elementos, o último elemento terá pixels no valor de 200.



Figura 8. Elementos encontrados na imagem da partitura: (a) primeiro, (b) décimo sexto, (c) décimo nono e (d) trigésimo quinto

Na etapa de separação dos elementos detectados, cada elemento é cortado da imagem e seus pixels normalizados para 1 (branco) para objeto e 0 (preto) para fundo, como se pode ver na Figura 8. O fundo é deixado em preto, pois para essa etapa são utilizados os momentos invariantes de Hu, onde é feita a descrição da imagem de cada elemento. Momentos de Hu se baseiam em momentos invariantes (invariação à escala, translação e rotação). O primeiro momento, por exemplo, fornece o centro do elemento. Logo, o elemento pode estar em escalas diferentes (apresenta-se em tamanhos maiores ou menores) ou posições diferentes, rotacionado ou espelhado na imagem. Na etapa de descrição ou identificação dos elementos, é criado um objeto. Esse objeto apresenta uma imagem recortada do elemento, seus momentos de Hu, posição do seu centro dentro da imagem e a área que ele ocupa. A partir dessa etapa, pode-se reconstruir a partitura novamente usando os dados disponíveis e futuramente usá-los para a etapa de classificação e reconhecimento, conforme ilustrado na Figura 9. Para averiguar a coerência dos dados, uma imagem contendo os elementos detectados foi reconstruída a partir dos dados da descrição.

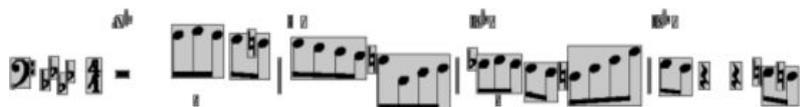


Figura 9. Imagem reconstruída com os dados da etapa de descrição

No futuro, após a etapa de classificação, os objetos não conterão mais imagens, apenas sua posição na imagem original, seu tamanho, seus momentos invariantes e um

código identificador quanto a sua classificação: se é uma clave de fá ou sol, notas conectadas ou não, etc.

4. Conclusão

Este projeto desencadeou a pesquisa conjunta de pesquisadores de diferentes áreas da Ciência da Computação como Visão computacional, Processamento de Imagens, Computação Musical, Inteligência Artificial, Compiladores, Redes complexas e Recuperação de Informação. A união destas áreas deverá auxiliar o desenvolvimento da ferramenta pretendida pelo projeto trazendo também ganhos pela interdisciplinaridade da pesquisa dentro da grande área da Ciência da Computação. Além de colaborar como pesquisa interdisciplinar na área da Ciência, o projeto deverá também auxiliar a área da música gerando uma ferramenta open-source para reconhecimento e reescrita de partituras.

Os primeiros passos deste projeto envolveram a pesquisa por técnicas e ferramentas computacionais a serem utilizadas em cada um dos passos do processamento do Copista. O levantamento destes algoritmos permitiu testes preliminares na área de Visão computacional com bons resultados iniciais. Os próximos passos do projeto deverão mesclar as técnicas levantadas e os códigos implementados nos passos individuais desta pesquisa em um primeiro protótipo funcional. Possivelmente, este primeiro protótipo deverá trabalhar ainda com partituras digitais e não-manuscritas para o treinamento do reconhecimento de uma rede neural que deverá ser utilizada para a tomada de decisão em relação à correção de um símbolo identificado.

Outro passo que deverá ser dado em breve será a integração da representação com a Visão Computacional e a verificação dos elementos identificados por meio de um compilador de música simbólica. Tal passo deverá auxiliar também no treinamento da ferramenta sendo mais um passo em busca de um melhor resultado para atender o objetivo proposto.

4.1. Agradecimentos

Agradecemos a Universidade Federal de São João Del Rei e o Programa de Iniciação Científica que está apoiando este projeto.

Referências

- Ariza, C. and Cuthbert, M. (2011). *The music21 stream: A new object model for representing, filtering, and transforming symbolic musical structures*. Ann Arbor, MI: MPublishing, University of Michigan Library.
- Association, M. M. et al. (1996). *The complete MIDI 1.0 detailed specification: incorporating all recommended practices*. MIDI Manufacturers Association.
- Bainbridge, D. and Bell, T. (1997). Dealing with superimposed objects in optical music recognition.
- Bainbridge, D. and Bell, T. (2001). The challenge of optical music recognition. *Computers and the Humanities*, 35(2):95–121.
- Bernsen, J. (1986). Dynamic thresholding of gray-level images. In *International Conference on Pattern Recognition*.

- Buxton, W., Reeves, W., Baecker, R., and Mezei, L. (1978). The use of hierarchy and instance in a data structure for computer music. *Computer Music Journal*, pages 10–20.
- Cuthbert, M. S. and Ariza, C. (2010). music21: A toolkit for computer-aided musicology and symbolic music data.
- Dori, D., Doerman, D., Shin, C., Haralick, R., Phillips, I., Buchman, M., and Ross, D. (1996). Handbook on optical character recognition and document image analysis, chapter the representation of document structure: a generic object-process analysis.
- Fujinaga, I. (2004). Staff detection and removal. *Visual perception of music notation: on-line and off-line recognition*, pages 1–39.
- Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. Academic press.
- Furht, B. (2011). *Handbook of augmented reality*. Springer Science & Business Media.
- Gonzalez, R. C., Woods, R. E., and Eddins, S. L. (2004). *Digital image processing using MATLAB*. Pearson Education India.
- Good, M. (2001). Musicxml for notation and analysis. *The virtual score: representation, retrieval, restoration*, 12:113–124.
- Grande, C. (1997). The notation interchange file format: A windows-compliant approach. In *Beyond MIDI*, pages 491–512. MIT Press.
- Hewlett, W. B. (1997). Beyond midi. chapter MuseData: Multipurpose Representation, pages 402–447. MIT Press, Cambridge, MA, USA.
- Hoos, H. H., Hamel, K. A., Renz, K., and Kilian, J. (1998). The guido notation format – a novel approach for adequately representing score-level music.
- Koendrik, J. J. (1992). Computational vision (book). *Ecological Psychology*, 4(2):121–128.
- Laia, M. A. d. M. (2013). *Filtragem de Kalman não linear com redes neurais embarcada em uma arquitetura reconfigurável para uso na tomografia de Raios-X para amostras da física de solos*. PhD thesis, Universidade de São Paulo.
- Medina, R. A., Smith, L. A., and Wagner, D. R. (2003). Content-based indexing of musical scores. In *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '03*, pages 18–26, Washington, DC, USA. IEEE Computer Society.
- Mundy, J. L., Zisserman, A., et al. (1992). *Geometric invariance in computer vision*, volume 92. MIT press Cambridge.
- Nienhuys, H.-W. and Nieuwenhuizen, J. (2003). Lilypond, a system for automated music engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, volume 1. Citeseer.
- Oppenheim, I., Walshaw, C., and Atchley, J. (2010). The abc standard 2.0.
- Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27.

- Pinto, T., Rebelo, A., Giraldi, G., and Cardoso, J. S. (2011). Music score binarization based on domain knowledge. In *pattern recognition and image analysis*, pages 700–708. Springer.
- Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A., Guedes, C., and Cardoso, J. (2012). Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190.
- Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge university press.
- Sauvola, J. and Pietikäinen, M. (2000). Adaptive document image binarization. *PATTERN RECOGNITION*, 33:225–236.
- Seixas, F. L., Martins, A., Stilben, A. R., Madeira, D., Assumpção, R., Mansur, S., Victor, S. M., Mendes, V. B., and Conci, A. (2008). Avaliação dos métodos para a segmentação automática dos tecidos do encéfalo em ressonância magnética. *Simpósio de Pesquisa Operacional e Logística da Marinha SPOLM*.
- Selfridge-Field, E. (1997). Beyond codes: issues in musical representation. In *Beyond MIDI*, pages 565–572. MIT Press.
- Szwoch, M. (2007). Guido: A musical score recognition system. In *icdar*, pages 809–813.
- Travis Pope, S. (1996). Object-oriented music representation. *Organised Sound*, 1(01):56–68.
- Xu, L., Krzyzak, A., and Suen, C. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3):418–435.