

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

*Optical Music Recognition in the
grey-scale domain*

Vitor Hugo Couto Vidal

Mestrado Integrado em Engenharia Eletrotécnica e de computadores

Orientador: Jaime S. Cardoso (Ph.D.)

Responsável pelo acompanhamento do trabalho na instituição: Ana Maria Rebelo (MSc)

Fevereiro de 2012

Conteúdo

1	Introdução	1
1.1	Contextualização	1
1.2	O problema e a sua caracterização	1
1.2.1	<i>Optical Music Recognition</i> - OMR	2
1.3	Tecnologias a utilizar	4
1.3.1	<i>Python</i>	4
1.3.2	<i>Matlab</i>	4
1.3.3	<i>VisualStudio</i>	4
1.4	Metologia	5
1.4.1	Grafos	5
1.4.2	Programação dinâmica	5
1.5	Objectivos	6
1.6	Benefícios esperados com a solução encontrada	6
2	Estado da Arte	9
2.1	Caracterização do domínio de intervenção	9
2.2	Pré-processamento das imagens	9
2.3	Comprimentos de referência	10
2.4	Binarização	11
2.5	Detecção de Pautas	11
2.5.1	Método de Otsu	13
3	Plano de trabalho	15

Lista de Figuras

1.1	Arquitetura do sistema <i>in</i> [CCRG08]	2
1.2	Procedimento para a digitalização de um documento <i>in</i> [KB07]	3
1.3	Exemplo da linearização de um grafo <i>in</i> [DPV06]	5
2.1	Representação dos comprimentos de referência	10
2.2	Resultado de um erro de binarização (<i>figura 2b in</i> [CR10])	13
3.1	Calendarização do projeto	16

Abreviaturas e Símbolos

IMSLP	<i>International Music Score Library Project</i>
OMR	<i>Optical Music Recognition</i>
MusicXML	<i>Music Extensible Markup Language</i>
MIDI	<i>Musical Instrument Digital Interface</i>
RLE	<i>Run-Length Encoding</i>
DAG	<i>Directed acyclic graph</i>

Capítulo 1

Introdução

1.1 Contextualização

No momento em que o ser humano se apercebeu que era possível registrar os seus feitos e passa-los de geração em geração, começou a ser “escrita” a História. A partir desse momento, o Homem foi evoluindo e desenvolvendo formas mais rápidas, eficientes e acessíveis de armazenar e partilhar conhecimento. Desde o aparecimento do alfabeto Fenício e do desenvolvimento dos hieróglifos Egípcios entre 3500 A.C. e 2900 A.C. até à criação da World Wide Web em 1994, todas as tecnologias foram criadas tendo em conta a existência de alguma retro-compatibilidade, de forma a poder existir uma transição suave entre tecnologias e a poder ser usada toda a informação existente previamente.

Nos últimos anos, tem existido uma crescente preocupação e um grande esforço para encontrar soluções para a preservação de documentos musicais manuscritos originais. Um dos casos paradigmáticos, são as partituras musicais. Atualmente, a digitalização e a cópia são a opção mais usada pois revelam uma forma rápida e eficiente de distribuição e preservação destes documentos.

1.2 O problema e a sua caracterização

Existem atualmente algumas bibliotecas, como por exemplo a *International Music Score Library Project* (IMSLP)¹, que possuem documentos que se encontram já indexados às bases de dados não só pelos parâmetros de pesquisas usuais, mas também pela sua própria melodia. A existência de documentos nestes formatos, deve-se muitas vezes ao trabalho de pessoas que transcrevem os documentos manualmente, ou então, mais recentemente, a métodos computacionais que através de técnicas de processamento de imagem compilam essa informação de forma automática (cf. capítulo 1.2.1 sobre sistemas reconhecimento ótico de música (Optical Music Recognition - OMR)).

¹<http://imslp.org/>

O método a ser desenvolvido ao longo desta dissertação, encontra-se inserido no projeto “Automatic recognition of handwritten music scores”, que visa criar um sistema OMR automático que facilite a criação de futuros repositórios de obras musicais. Serão desenvolvidos métodos que permitam a deteção de símbolos musicais manuscritos apoiados em inteligência artificial. Numa fase seguinte, prevê-se o desenvolvimento de uma base de dados, de um servidor web e de um website. Os documentos originais digitalizados, a sua componente digital guardada como MusicXML e os meta-dados inseridos pelo utilizador serão armazenados no repositório. O servidor web, será responsável pelo processamento dos módulos de pesquisa e do módulo de reconhecimento ótico das partituras. Estará também incumbido de fazer o interface entre o repositório e o website. Finalmente, o website será responsável pelo interface com o utilizador e pela gestão de todo o sistema (figura 1.1 in [CCRG08]).

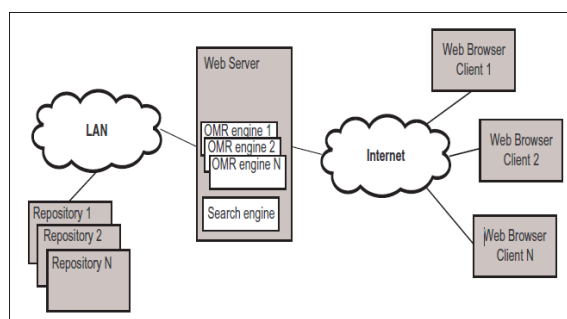


Figura 1.1: Arquitetura do sistema in [CCRG08]

1.2.1 Optical Music Recognition - OMR

Grande parte dos autores da literatura relacionada com a área do reconhecimento ótico de música, concorda que a melhor abordagem para a implementação deste método, passa pela fragmentação do problema em partes menos complexas [RFP⁺12]. Como exemplo desta fragmentação modular, vai ser utilizado o modelo encontrado em [CR10]. Neste artigo a leitura da partitura é feita baseada em 4 passos principais:

1. Pré-processamento;
2. Deteção dos símbolos musicais;
3. Reconstrução da informação musical num formato digital;
4. criação de uma notação musical modelo para, recorrendo ao formato digital, representar a partitura simbolicamente.

Existem vários métodos para solucionar cada um dos módulos acima descritos. No pré-processamento da imagem podem ser aplicadas várias técnicas desde, aprimoramentos, binarizações, remoção de ruídos, compensar rotações do documento, entre outros (cf. capítulo 2.2).

Neste módulo são também recolhidas informações quanto ao espaçamento entre linhas da mesma pauta e à sua espessura, de forma a criar referências para futuras comparações (cf. capítulo 2.3).

O módulo de pré-processamento da imagem é subdividido em três sub-módulos: a deteção e remoção de pautas (cf. capítulo 2.5) de forma a obter apenas os símbolos musicais; segmentação dos símbolos e, por fim, reconhecimento dos símbolos.

O segundo e o terceiro módulo aparecem por vezes combinados. Nesta etapa, para além da análise gráfica, é também feita uma análise de contexto baseada em regras de sintaxe, o que permite tratar indecisões que surjam na deteção de símbolos.

Por fim, no último módulo, é analisada a informação obtida nos passos anteriores, e posteriormente compilada num ficheiro final tipo MIDI ou MusicXML.

Apesar deste ser um dos moldes mais frequentes para a arquitetura de sistemas OMR, foram propostos outros modelos de forma a melhorar os resultados obtidos. Uma das formas encontradas de melhorar resultados é através da comparação de resultados obtidos [BS06, KB07]. Os métodos utilizados em [KB07] pode ser descrito da seguinte forma (figura 1.2): uma partitura é digitalizada e convertida para MusicXML, usando diferentes programas OMR; os resultados são normalizados e guardados na etapa correspondente; posteriormente, os resultados de cada etapa são comparados e são aplicadas regras que combinam todas as versões de forma a minimizar os erros aplicados. O núcleo deste procedimento são as regras para combinar as versões. Isto requer a criação manual de um mapa com as vantagens e desvantagens de todas as etapas de cada programa, tal como encontrado em [BS06].

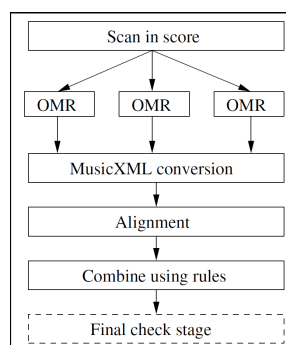


Figura 1.2: Procedimento para a digitalização de um documento in [KB07]

Outros trabalhos, combinam por exemplo a informação obtida através dos métodos OMR, com informação áudio, de forma a resolver ambiguidades [KMF⁺07, DFK⁺08].

1.3 Tecnologias a utilizar

Neste projeto vão ser utilizados principalmente três softwares: *Python*, *Matlab* e *VisualStudio*. A escolha desta combinação de aplicações visa usufruir do que cada uma tem de melhor. Todo este trabalho poderia ser, por exemplo, elaborado recorrendo a apenas um destes programas, mas a sua complexidade iria ser desnecessariamente maior. Por estes motivos, o *Python* (cf. 1.3.1)

foi utilizado para a interface com o utilizador, devido a facilidade que este proporciona quando se tratar de visualização e manipulação gráfica. O *Matlab* (cf. 1.3.2), para além do seu interface *user-friendly*, devido a sua enorme biblioteca de funções orientadas às operações sobre matrizes, tornam-no uma ferramenta essencial para quando é necessário manipular matematicamente uma imagem. Quanto ao *VisualStudio* (cf. 1.3.3) é uma das melhores aplicações para se desenvolverem projetos em C/C++ para Windows. Esta linguagem é muito versátil e devido ao seu baixo nível, permite a elaboração de programas de forma bastante eficiente, controlando o seu funcionamento com precisão.

1.3.1 Python

*Python*² é uma linguagem de programação dinâmica. Esta linguagem é usada em vários domínios de aplicação e é caracterizada pela sua versatilidade e simplicidade, uma vez que possui várias bibliotecas que podem ser integradas noutras linguagens de programação como por exemplo a biblioteca *Jython* para *Java* ou *IronPython* para .NET. No âmbito desta dissertação será usada como interface para criação de uma base de dados que servirá numa fase futura do trabalho como ponto de referência para verificar a eficiência do programa desenvolvido.

1.3.2 Matlab

*Matlab*³ é uma linguagem de alto nível com ambiente interativo, que permite desenvolver tarefas computacionalmente complexas, de uma forma mais rápida do que as linguagens de programação tradicionais como C e C++.

1.3.3 VisualStudio

*VisualStudio*⁴ é uma plataforma gratuita com várias aplicações de desenvolvimento para criar aplicações web, de cliente (Windows) e para o *Windows Phone 7*. Neste caso esta plataforma irá funcionar como compilador de C/C++, linguagem que vai ser utilizada para desenvolver a aplicação central deste projeto.

1.4 Metodologia

1.4.1 Grafos

Um grafo⁵ é um par $G = (V, E)$ de conjuntos tais que $E \subseteq [V]^2$ ⁽⁶⁾, assim, os elementos de E são subconjuntos de dois elementos de V . Os elementos de V são os vértices ou nós do grafo

²<http://python.org/>

³<http://www.mathworks.com/products/matlab/description1.html>

⁴<http://www.microsoft.com/visualstudio/en-us/products/2010-editions/expresswhatIsExpress>

⁵A palavra “grafo” é um neologismo derivado da palavra graph em inglês. Ela foi usada pela primeira vez no sentido abordado ao longo deste projeto, pelo matemático inglês James Joseph Sylvester

⁶ V^2 é o conjunto de todos os pares não ordenados de elementos de V .

G , enquanto que os elementos de E são as suas arestas [Die05]. Um grafo com um conjunto de vértices em V , é denotado como *grafo em V* . A ordem de um grafo G é representada por $|G|$ e é dada pelo número de vértices que este possui, o número de arestas é representado por $\|G\|$. Se v pertencer a e , então e é uma aresta de v . Se $x \in X$ e $y \in Y$, então xy é uma aresta entre os nós X e Y e pertence a $E(X, Y)$ que é o conjunto de todas as arestas que ligam X a Y . Todas as arestas que pertencem a um vértice v , são referenciadas como $E(v)$. Dois vértices x, y de um grafo $G = (V, E)$ são adjacentes se $xy \in E$. Quanto às arestas, estas são adjacentes se tiverem pelo menos um nó em comum. Um grafo, dependendo da aplicação, pode ter atribuído às suas arestas um peso e/ou uma direção, neste último caso, o grafo é denominado dígrafo. Na teoria dos grafos, existem alguns algoritmos (e.g. programação dinâmica) que permitem solucionar alguns problemas mais genéricos como, por exemplo, os problemas de caminho mínimo que visam encontrar o percurso com a rota mais curta entre dois nós.

1.4.2 Programação dinâmica

A programação dinâmica esta associada a metodologias de optimização, à resolução de problemas que requerem decisões sequenciais interrelacionadas que podem ou não ser afectadas por custos que afectem decisões futuras.

Uma das características de um DAG⁷ [DPV06] é a capacidade dos seus nós poderem ser linearizados, ou seja, podem ser reorganizados de forma a ficarem dispostos numa linha e com as arestas orientadas da esquerda para a direita (figura 1.3).

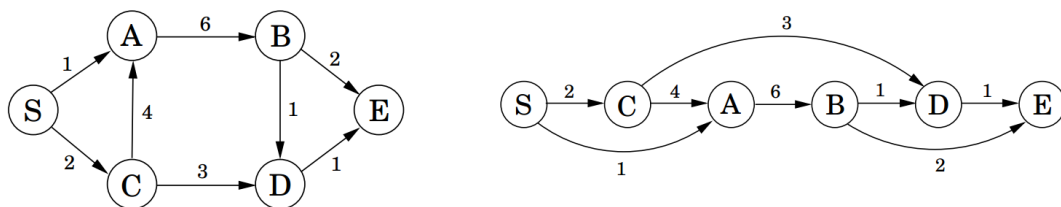


Figura 1.3: Exemplo da linearização de um grafo in [DPV06]

Sejam consideradas as distâncias ao nó S , mais especificamente do nó D , de forma a demonstrar o porque desta reorganização no cálculo do caminho mínimo. A única forma de se alcançar o nó D , é através dos nós B e C . Logo, para encontrar o caminho mínimo para D , é necessário comparar estas duas rotas:

$$dist(D) = \min\{dist(B) + 1, dist(C) + 3\}.$$

Todos os nós podem ser referenciados de forma semelhante. Se forem processados todos os valores $dist$ dos nós da figura 1.3 da esquerda para a direita, podemos assegurar que sempre que é alcançado um novo nó, por exemplo o nó v , já possuímos toda a informação necessária para

⁷do inglês *Directed acyclic graph* (dígrafo sem ciclo)

proceder ao cálculo da sua distância ao nó inicial. Conclui-se portanto que todos os valores $dist(\cdot)$ podem ser calculados de uma só vez.

inicializar todos os valores $dist(\cdot)$ como ∞

$$dist(s) = 0$$

para cada $v \in V \setminus \{s\}$, na forma linear:

$$dist(v) = \min_{(u,v) \in E} \{dist(u) + l(u, v)\}$$

O algoritmo demonstrado, aborda o problema através de pequenas etapas mais simples de resolver. À medida que progride no seu algoritmo, também a complexidade dos problemas vai aumentando. O mais fácil de resolver, é o cálculo do valor de $dist(s)$ que facilmente se conclui ser zero. De seguida são calculados os valores dos nós adjacentes. Esta operação é repetida até ser encontrado o valor do nó mais distante.

1.5 Objectivos

- Estudar e comparar algoritmos do estado da arte;
- Desenvolver uma nova base de dados constituída por imagens em níveis de cinzento;
- Investigar e desenvolver novos métodos para deteção de pautas, trabalhando diretamente com imagens em cinzento sem binarização;
- Avaliar os resultados obtidos;
- Possível integração do método desenvolvido no sistema OMR protótipo.

1.6 Benefícios esperados com a solução encontrada

Esta dissertação tem como intuito desenvolver um algoritmo de deteção de pautas de partitura em imagens com níveis de cinzento. Desta forma, é esperado que, ao deixar de se fazer a binarização, passo frequente em vários processos OMR (cf. capítulo 2.5), este processo fique mais robusto a variações de luminosidade ao longo da partitura, a imperfeições no papel, a deteriorações devido à idade dos documentos musicais em análise, entre outros fatores. Para além disto, irá também ser criada uma base de dados referente a imagens em escala de cinzentos que poderá vir a ser utilizada mais tarde noutros trabalhos.

Capítulo 2

Estado da Arte

2.1 Caracterização do domínio de intervenção

A detecção da pauta de uma partitura, é o primeiro passo de grande parte dos processos OMR. A sua grande relevância advém da facilidade com que posteriormente se pode proceder à extração dos símbolos musicais. A obtenção das pautas é normalmente feita recorrendo à binarização das imagens através da definição de *thresholds* que podem ser locais ou globais. Estas técnicas podem porém, remover alguma informação relevante da pauta e introduzir ruído nas etapas posteriores do processo. Surge portanto, a necessidade de se criar um método que permita eliminar a perda de informação resultante da binarização, uma forma de obter as características desejadas diretamente da imagem em tons de cinzento.

2.2 Pré-processamento das imagens

No que diz respeito à qualidade de digitalização, não existe nenhum estudo que analise o impacto deste passo no desempenho global dos algoritmos, daí não existir nenhum padrão na escolha da resolução feita pelos autores. Se por um lado alguns utilizaram resoluções de 300dpi (como encontrado por exemplo em [CCR⁺09, RC10]), outros como por exemplo [Pug06] e [FR02] utilizam, correspondentemente, 400dpi e 600dpi.

Uma vez que normalmente os grandes problemas encontrados estão maioritariamente associados a ruído, a variações de luminância indesejadas e até mesmo à má qualidade do documento que está a ser digitalizado, o aumento do número de pixéis por polegada não vai contribuir para um melhor desempenho das fases posteriores. Existem no entanto formas de trabalhar as imagens mais eficientes como por exemplo:

- Tratamento da qualidade da imagem inicial [Gö03]: por exemplo a equalização do histograma, permite obter melhores resultados durante a binarização;

- Binarização [Gö03, Fuj04, FLS06, CCR⁺09] (cf. capítulo 2.4);
- Remoção de ruído [Fuj04, Gö03]: são aplicados filtros passa-baixo de forma a eliminar pequenos pontos isolados que não constam no documento original¹, que podem dificultar futuramente a detecção de informação relevante;
- *blurring* [Gö03]: é aplicado um filtro Gaussiano de forma a suavizar a imagem;
- compensação da rotação de um documento (*deskwing*) [Gö03, Fuj04, FLS06]: por vezes os documentos não vêm perfeitamente alinhados. Quando este efeito é compensado e os documentos ficam alinhados, a detecção das pautas, das barras de compasso e das próprias hastes dos símbolos musicais fica mais fácil.
- operações morfológicas [Gö03]: estas operações têm o objetivo de reduzir ainda mais o ruído, eliminando todos os conjuntos de 4*4 pixéis. Para além disto, as zonas que ficam a esquerda e a direita das pautas são apagadas ou marcadas como uma zona sem informação útil.

2.3 Comprimentos de referência

Os algoritmos OMR, recorrem normalmente a duas medidas que servem de referência ao longo de todo o processo (figura 2.1), a espessura das linhas de cada pauta (*Staffline Height*) e o seu espaçamento (*Staffspace Height*).

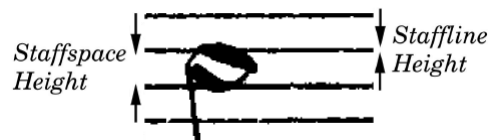


Figura 2.1: Representação dos comprimentos de referência

Por vezes é usado o algoritmo de codificação RLE - *Run-Length Encoding* em imagens já binarizadas [Fuj04, RB07, DDPF08, CCR⁺09], como forma de representação dos pixéis obtidos ao longo de uma coluna. Este algoritmo, para além de compactar a informação, permite obter inerentemente, a contagem de pixéis de cada nível pois deteta as repetições de cada valor e substitui-os pelo número de vezes que estes aparecem, e.g. a sequência 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 é codificada como 1,2,2,1,3,2,4. O primeiro número da sequência obtida é sempre referente a quantidade de uns pelo que, se a sequência a ser codificada começar com um zero, o primeiro número do código será 0. A partir deste resultado podemos obter os pixéis brancos nas posições pares e os pretos nas posições ímpares. Para saber o valor estimado da *Staffline Height* é analisado qual o valor mais frequente nas posições ímpares e para obter o valor do *Staffspace Height* é analisado qual o valor mais frequente nas posições pares. Este método é imune a rotações das partituras(*deskwing*). [CR10]

¹este efeito é conhecido como "salt and pepper"

2.4 Binarização

Na maior parte dos sistemas OMR, a sua primeira etapa é o processo de binarização. Este processo consiste em analisar a imagem inicial de forma a destacar os símbolos musicais e as pautas, do ruído e do fundo da imagem a analisar. Vários algoritmos foram desenvolvidos no passado de forma a tornar este processo automático e eficiente, mas o seu resultado é sempre dependente do caso específico que estão a analisar. O processo de binarização raramente é alvo de uma grande análise por parte dos investigadores, que preferem normalmente usar processos de binarização generalistas como por exemplo o método de Otsu (cf. capítulo 2.5.1) que, segundo [TJ95, TT95], é um dos melhores e mais rápidos. É evidente, que o desenvolvimento de métodos específicos de binarização de partituras, permite obter resultados mais fiáveis e, conseqüentemente, melhorar as restantes etapas do processo OMR [RFP⁺12]. Noutros artigos como [NB96, NCS⁺99] é usada a técnica de [RTWC78]. Os métodos de binarização podem calcular o *threshold* globalmente ou localmente. Usar o *threshold* calculado globalmente, permite extrair objetos de uma forma bastante rápida e eficiente em imagens com fundo uniforme, mas quando o fundo da imagem tem muito ruído ou sofre algumas variações no seu plano de fundo é mais eficiente utilizar o cálculo de *threshold* local, apesar de ter uma carga computacional mais *pesada*.

2.5 Detecção de Pautas

Uma vez que a deteção e identificação dos símbolos musicais estão condicionadas pela eficiência da deteção e remoção da pauta, o desempenho e a qualidade dos resultados finais de todo o projeto, depende em grande parte do sucesso desta etapa.

Normalmente, a deteção das pautas aparece relacionada com a sua remoção, salvo algumas exceções [MO07, Szw05, Pug06].

Um dos métodos para deteção consiste em procurar horizontalmente, secções onde o número de pixels pretos seja máximo, identificando uma linha sempre que encontra um máximo local [KI90, Fuj04, RCF⁺93, MN96, BBN01, TSM06]. Este método tem porém a grande desvantagem de só conseguir detetar linhas que sejam perfeitamente horizontais, o que pode não acontecer devido a pequenas rotações da partitura. Para retificar esta falha, podem ser feitas várias pesquisas com várias rotações e considerar a rotação que maximiza o número de pixels pretos encontrados por linha.

Outra forma de encontrar é recorrendo a linhas de pesquisa verticais [Car89]. Desta forma, as linhas são percorridas de forma a ir detetando potenciais linhas de pautas, esta deteção é feita analisando secções que satisfaçam padrões pré-definidos como *aspect ratio*, curvatura e continuidade. Existem também alguns métodos que melhoram os métodos anteriores recorrendo a projeções horizontais e verticais dos símbolos e das pautas [Bai97, BBN01, RB05].

Como referido no capítulo 2.2, trabalhos como por exemplo o de Fujinaga [Fuj04], utilizam técnicas de processamento de imagem como RLE, conexão de componentes e projeções. O primeiro passo deste método é determinar a *staffline height* e o *staffspace height* (cf. capítulo 2.3),

recorrendo ao RLE. De seguida, a posição das pautas tem de ser determinada antes de esta ser removida. Para tal, estas são isoladas dos símbolos seguindo dois passos. Primeiro todas as linhas verticais pretas que tenham mais do dobro da *staffline heigh* são removidas; Depois, é analisada a conectividade entre componentes e são removidas todas as componentes que tenham uma largura menor do que o *staffspace height*.

Existem outras técnicas para detetar pautas, como por exemplo agrupar colunas tendo em conta o seu espaçamento, largura e posição vertical relativa à imagem [RP96], classificar de segmentos horizontais baseados em regras [Mah82] ou fazer o rastreamento de linhas [Pre70, RT88]. No artigo

Apesar da variedade de métodos disponíveis, não existem nenhum sem alguma limitação. Uma das situações mais complicadas de se resolver é a existência de linhas com descontinuidades e que não sejam perfeitamente planas. Um dos poucos trabalhos que realizou algum esforço no sentido de resolver as descontinuidades foi o [LCL93]. Este algoritmo efetua uma pesquisa pixel a pixel ao longo de toda a imagem, classifica-os como mancha ou como traço e em seguida tenta construir linhas que permitam ligar esses traços.

Um ponto comum em todos os métodos analisados, é o facto de as partituras necessitarem de ser binarizadas antes de serem trabalhadas. Isto levanta o inconveniente da definição de um *threshold* que tome um valor que seja funcional para todo o documento, de forma a separar os objetos em análise do fundo da partitura. Em documentos musicais deteriorados pode não existir um valor para o qual esta condição se verifique, sendo que, após a binarização, a partitura pode ficar com zonas onde os dados vão ser perdidos (figura 2.2). Ao longo desta dissertação, será elaborado um algoritmo que permite a deteção de pautas sem recurso a binarização, trabalhando com as imagens em níveis de cinzento.



Figura 2.2: Resultado de um erro de binarização (figura 2b in [CR10])

Um dos poucos artigos que aborda esta estratégia é o [CR10]. Neste artigo, é usado um método de análise numa imagem com níveis de cinzento para obter a *staffline heigh* e o *staffspace height*. Para este método, em vez de se calcular um histograma para uma partitura binarizada uma vez com um valor de *threshold*, a partitura original é binarizada várias vezes recorrendo ao método de Otsu (cf. capítulo 2.5.1) utilizando vários valores de *threshold*. Desta forma é possível trabalhar com todos os valores obtidos e analisar a sua tendência, conseguindo um valor esperado para

estas medidas de referência mais próximo do real e mais robusto em relação, por exemplo, aos gradientes de luminosidade e à qualidade do papel.

2.5.1 Método de Otsu

Este método de binarização generalista [Ots79], é muitas vezes utilizado em vários trabalhos na área dos OMR [Gö03, Pug06, CCR⁺09, RC10], e é composto por cinco passos fundamentais:

- O primeiro passo recorre a um filtro *Wiener* passa-baixo de forma a reduzir o ruído;
- No segundo passo é feita uma primeira aproximação rude das regiões do primeiro plano;
- No terceiro passo, é avaliado o segundo plano da imagem, interpolando as intensidades da vizinhança deste plano com o resultado obtido no segundo passo;
- No quarto passo, procedemos à última binarização, combinando a informação calculada no passo anterior e a imagem original. O *threshold* é adaptado ao valor de *gray-scale* da imagem de segundo plano, preservando a informação mesmo em zonas mais escuras;
- Por fim, é feito um pós-processamento que elimina ruídos e melhora a qualidade das regiões seleccionadas.

Capítulo 3

Plano de trabalho

- Desenvolvimento da base de dados;
- Aquisição do conhecimento relacionado com o processamento de imagem;
- Aquisição do conhecimento necessário para desenvolver o trabalho de investigação, incluindo conceitos de grafos e programação dinâmica;
- Rever o estado da arte na deteção de pautas;
- Comparação das técnicas de processamento das partituras;
- Propor e implementar um novo algoritmo de deteção de pautas para OMR em *gray-scale*;
- Escrita da dissertação e de um artigo científico;
- Preparação da defesa da dissertação.

A calendarização destas etapas pode ser consultada na figura [3.1](#).

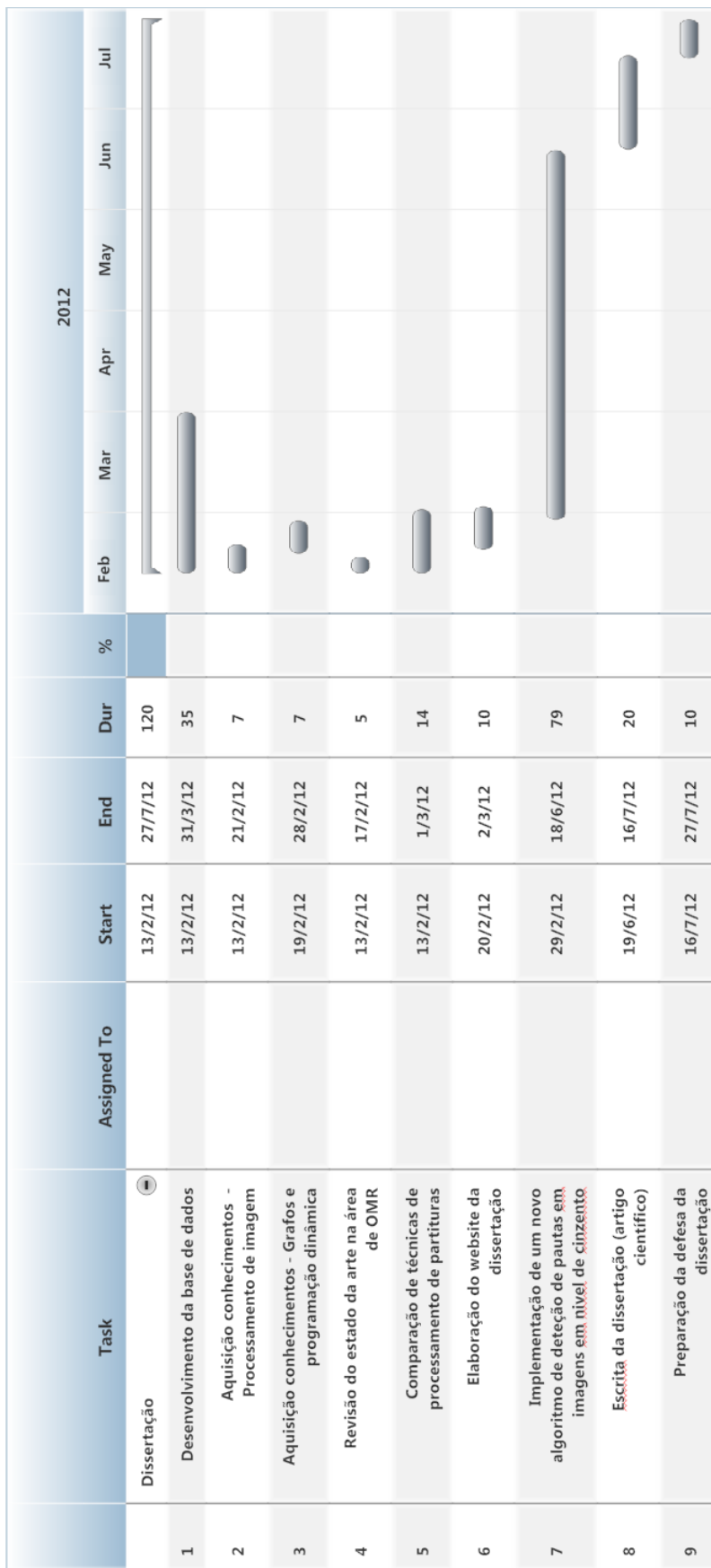


Figura 3.1: Calendarização do projeto

Bibliografia

- [Bai97] D. Bainbridge. *Extensible Optical Music Recognition*. PhD thesis, Department of Computer Science, University of Canterbury, Christchurch, NZ, 1997.
- [BBN01] P. Bellini, I. Bruno e P. Nesi. Optical music sheet segmentation. *Web Delivering of Music, 2001. Proceedings. First International Conference on*, pages 183–190, Nov. 2001.
- [BS06] Donald Byrd e Megan Schindele. Prospects for improving omr with multiple recognizers. In *ISMIR [icm06]*, pages 41–46.
- [Car89] N. P. Carter. *Automatic Recognition of Printed Music in the Context of Electronic Publishing*. PhD thesis, Departments of Physics and Music, 1989.
- [CCR⁺09] Jaime S. Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes e Joaquim F. Pinto da Costa. Staff detection with stable paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1134–1139, 2009.
- [CCRG08] Artur Capela, Jaime S. Cardoso, Ana Rebelo e Carlos Guedes. Integrated recognition system for music scores. In *Proceedings of the International Computer Music Conference*, 2008.
- [con02] *ISMIR 2002, 3rd International Conference on Music Information Retrieval, Paris, France, October 13-17, 2002, Proceedings*, 2002.
- [CR10] J.S. Cardoso e A. Rebelo. Robust staffline thickness and distance estimation in binary and gray-level music scores. In *20th International Conference on Pattern Recognition (ICPR)*, pages 1856–1859, August 2010.
- [DDPF08] C. Dalitz, M. Droettboom, B. Pranzas e I. Fujinaga. A comparative study of staff removal algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):753–766, may 2008.
- [DFK⁺08] David Damm, Christian Fremerey, Frank Kurth, Meinard Müller e Michael Clausen. Multimodal presentation and browsing of music. In *Proceedings of the 10th international conference on Multimodal interfaces, ICMI '08*, pages 205–208, New York, NY, USA, 2008. ACM.
- [Die05] Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, August 2005.
- [DPV06] S. Dasgupta, C.H. Papadimitriou e U.V. Vazirani. *Algorithms*, pages 169–179. McGraw-Hill Higher Education, 2006.

- [FLS06] Alicia Fornés, Josep Lladós e Gemma Sánchez. Primitive segmentation in old handwritten music scores. In Wenyin Liu e Josep Lladós, editors, *Graphics Recognition. Ten Years Review and Future Perspectives*, volume 3926 of *Lecture Notes in Computer Science*, pages 279–290. Springer Berlin / Heidelberg, 2006. 10.1007/1176797825.
- [FR02] Ichiro Fujinaga e Jenn Riley. Digital image capture of musical scores. In *ISMIR* [con02].
- [Fuj04] Ichiro Fujinaga. Staff detection and removal. In Susan George, editor, *Visual Perception of Music Notation: On-Line and Off-Line Recognition*, pages 1–39. Idea Group Inc., 2004.
- [Gö03] Roland Göcke. Building a system for writer identification on handwritten music scores, 2003.
- [icm06] *ISMIR 2006, 7th International Conference on Music Information Retrieval, Victoria, Canada, 8-12 October 2006, Proceedings*, 2006.
- [KB07] Ian Knopke e Donald Byrd. Towards musicdiff: A foundation for improved optical music recognition using multiple recognizers. In *ISMIR'07*, pages 123–126, 2007.
- [KI90] H. Kato e S. Inokuchi. A recognition system for printed piano music using musical knowledge and constraints. In *Proceedings of the International Association for Pattern Recognition Workshop on Syntactic and Structural Pattern Recognition*, pages 231–248, 1990.
- [KMF⁺07] Frank Kurth, Meinard Müller, Christian Fremerey, Yoon ha Chang e Michael Clausen. Automated synchronization of scanned sheet music with audio recordings. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 261–266, Vienna, Austria, September 2007.
- [LCL93] I. Leplumey, J. Camillerapp e G. Lorette. A robust detector for music staves. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 902–905, 1993.
- [Mah82] J. V. Mahoney. Automatic analysis of music score images. B.Sc thesis. Technical report, Department of Computer Science and Engineering, MIT, 1982.
- [MN96] H. Miyao e Y. Nakano. Note symbol extraction for printed piano scores using neural networks. *IEICE TRANSACTIONS on Information and Systems*, E79-D:548–554, 1996.
- [MO07] Hidetoshi Miyao e Masayuki Okamoto. Stave extraction for printed music scores using DP matching. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 8:208–215, 2007.
- [NB96] K. C. Ng e R. D. Boyle. Recognition and reconstruction of primitives in music scores. *Image and Vision Computing*, 14(1):39–46, 1996.
- [NCS⁺99] K. C. Ng, D. Cooper, E. Stefani, R. D. Boyle e N. Bailey. Embracing the Composer: Optical Recognition of Handwritten manuscripts. In *Proceedings of the International Computer Music Conference, Beijing*, 1999.
- [Ots79] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.
- [Pre70] D. Prerau. *Computer pattern recognition of standard engraved music notation*. PhD thesis, Massachusetts Institute of Technology, 1970.

- [Pug06] Laurent Pugin. Optical music recognition of early typographic prints using hidden markov models. In *ISMIR*, pages 53–56, 2006.
- [RB05] F. Rossant e I. Bloch. Optical music recognition based on a fuzzy modeling of symbol classes and music writing rules. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 2:II-538–41, Sept. 2005.
- [RB07] Florence Rossant e Isabelle Bloch. Robust and adaptive omr system including fuzzy modeling, fusion of musical rules, and possible error detection. *EURASIP J. Appl. Signal Process.*, 2007:160–160, January 2007.
- [RC10] Ana Rebelo e Jaime S. Cardoso. Content aware music score pre-processing. In *Proceedings of 16th Portuguese Conference on Pattern Recognition (RECPAD)*., October 2010.
- [RCF⁺93] R. Randriamahefa, J.P. Cocquerez, C. Fluhr, F. Pepin e S. Philipp. Printed music recognition. *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 898–901, Oct 1993.
- [RFP⁺12] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Carlos Guedes e Jaime S. Cardoso. Optical music recognition - state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 2012. Accepted.
- [RP96] K.T. Reed e J.R. Parker. Automatic computer recognition of printed music. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 3:803–807 vol.3, Aug 1996.
- [RT88] J. W. Roach e J. E. Tatem. Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment. *Pattern Recogn.*, 21(1):33–44, 1988.
- [RTWC78] S Ridler T W Calvard. Picture thresholding using an iterative selection method. *Ieee Transactions On Systems Man And Cybernetics*, 8(Aug):630–632, 1978.
- [Szw05] Mariusz Szwach. A robust detector for distorted music staves. In *Computer Analysis of Images and Patterns*, pages 701–708. Springer-Verlag, Heidelberg, 2005.
- [TJ95] Øivind Due Trier e Anil K. Jain. Goal-directed evaluation of binarization methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(12):1191–1201, 1995.
- [TSM06] Fubito Toyama, Kenji Shoji e Juichi Miyamichi. *Symbol Recognition of Printed Piano Scores with Touching Symbols*. IEEE Computer Society, Washington, DC, USA, 2006.
- [TT95] Due Trier e Torfinn Taxt. Evaluation of binarization methods for document images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:312–315, March 1995.