

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Uma abordagem para auxílio na transcrição de textos
históricos em português por agrupamento de palavras

Lucas Schmidt Cavalcante



São Carlos - SP

Uma abordagem para auxílio na transcrição de textos históricos em português por agrupamento de palavras

Lucas Schmidt Cavalcante

Orientador: *Gustavo Enrique de Almeida Prado Alves Batista*

Monografia final de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP – para obtenção do título de Bacharel em Ciências de Computação.

Área de Concentração: Inteligência Computacional

USP - São Carlos

Novembro de 2012

Agradecimentos

Ao professor João do E. S. Batista Neto que foi de fundamental importância nas etapas de processamento de imagem.

Resumo

Algumas instituições possuem grandes acervos de documentos históricos de escrita cursiva digitados. A existência de uma imagem digitalizada auxilia a perpetuar a memória do documento, e tal imagem pode ser utilizada em futuras pesquisas. Entretanto, textos disponibilizados na forma de imagens tornam difícil a capacidade de pesquisar e correlacionar informações nelas contidas. É preciso que haja um modo de fazer busca, e para isto é preciso transcrever esse documento. Neste trabalho é apresentada uma abordagem que utiliza diversas técnicas de processamento de imagem para detectar e extrair o contorno de palavras em documentos históricos, para posterior comparação e agrupamento por técnicas de inteligência artificial que envolvem séries temporais. O resultado final é um dendograma para cada documento histórico com o agrupamento das palavras.

Sumário

Lista de Figuras	p. III
1 Introdução	p. 5
1.1 Contextualização e Motivação	p. 5
1.2 Objetivos	p. 6
1.3 Organização do Trabalho	p. 6
2 Revisão Bibliográfica	p. 7
2.1 Processamento de Imagem	p. 7
2.1.1 Considerações Iniciais	p. 7
2.1.2 Morfologia Matemática aplicada ao Processamento de Imagem . . .	p. 8
2.1.3 Transformada de Hough	p. 15
2.2 Inteligência Artificial	p. 17
2.2.1 Considerações Iniciais	p. 17
2.2.2 <i>Dynamic Time Warping</i> (DTW)	p. 17
2.2.3 Uso do Contorno como Série Temporal	p. 20
2.3 Agrupamento de Dados	p. 21
3 Desenvolvimento do Trabalho	p. 22
3.1 Consideração Iniciais	p. 22
3.2 Processamento de Imagem	p. 23
3.2.1 Esqueletização e Tratamento de Ruído	p. 23
3.2.2 Detecção de Linhas	p. 27

3.2.3	Agrupamento de <i>tokens</i>	p. 29
3.3	Inteligência Computacional	p. 32
3.3.1	Geração de Séries Temporais	p. 32
3.3.2	<i>Dynamic Time Warping</i>	p. 33
3.4	Resultados	p. 34
4	Conclusão	p. 39
4.1	Considerações Finais	p. 39
4.2	Considerações sobre o Curso de Graduação	p. 39
4.3	Trabalhos Futuros	p. 40
Referências Bibliográficas		p. 41

Lista de Figuras

2.1	Ilustração da vizinhança 8-conectada. Em preto são os pixels pixels que pertencem a vizinhança 8-conectada do pixel p	p. 7
2.2	Ilustração de elementos estruturantes.	p. 9
2.3	Ilustração da aplicação da operação de erosão sobre uma imagem.	p. 10
2.4	Ilustração da aplicação da operação de dilatação sobre uma imagem.	p. 10
2.5	Ilustração da MAT.	p. 11
2.6	Ilustração do processo de <i>prunning</i> da letra ‘e’.	p. 14
2.7	Ilustração da transformação do plano xy para o plano ab	p. 15
2.8	Ilustração do processo de detecção da reta que contém mais pontos	p. 16
2.9	Ilustração da matriz pd	p. 18
2.10	Ilustração da DTW sem restrição.	p. 19
2.11	Ilustração das restrições de Sakoe e Chiba e de Itakura.	p. 20
3.1	Ilustração dos processos de processamento de imagem em um documento com ruído.	p. 24
3.2	Ilustração dos processos de processamento de imagem em um documento com ruído.	p. 25
3.3	Ilustração dos processos de processamento de imagem em um documento com ruído.	p. 26
3.4	Documento com linhas ao fundo.	p. 27
3.5	Em vermelho as linhas detectadas de 3.4 pela transformada de Hough.	p. 28
3.6	Os <i>bounding boxes</i> indicam a localização das palavras.	p. 29
3.7	Os <i>bounding boxes</i> indicam a localização das palavras.	p. 30
3.8	Os <i>bounding boxes</i> indicam a localização das palavras.	p. 31

3.9	Extração da série temporal de uma palavra.	p. 33
3.10	Todas as etapas para a obtenção do agrupamento das palavras.	p. 34
3.11	Palavras escolhidas para agrupar da primeira página do esboço do hino nacional.	p. 35
3.12	Palavras escolhidas para agrupar da segunda página do esboço do hino nacional.	p. 36
3.13	Agrupamento das palavras da Figura 3.11 e 3.12.	p. 37
3.14	Ilustração do contorno extraído atualmente e do contorno desejado.	p. 38

1 *Introdução*

1.1 Contextualização e Motivação

Algumas instituições possuem grandes acervos de documentos históricos de escrita cursiva digitizados. A existência de uma imagem digitalizada auxilia a perpetuar a memória do documento, e tal imagem pode ser utilizada em futuras pesquisas. Entretanto, textos disponibilizados em forma de imagem tornam difícil a capacidade de pesquisar e correlacionar informações nele contido. É preciso que haja um modo de fazer busca, e para isto é preciso a transcrição deste documento. Dada a quantidade de documentos, a automação deste processo é desejada [Rath and Manmatha, 2003].

Uma possível solução para este problema seria a aplicação do *Optical Character Recognition* (OCR). No entanto, é aceito que OCR é inadequado para este caso, pois ele necessita de uma segmentação precisa da palavra [Tomai et al., 2002]. Além disso, fatores como a similaridade na forma de caracteres distintos, as sobreposições e a interligação de caracteres vizinhos presentes na escrita cursiva complicam ainda mais a situação [Arica and Yarman-Vural, 2002]. O consenso é de que o reconhecimento de escrita cursiva é um problema em aberto [Niels and Vuurpijl, 2005]. Neste contexto, opta-se por atacar um problema semelhante, que é a identificação da ocorrência de todas as instâncias de uma mesma palavra. Com isto, caso o método seja capaz de agrupar palavras idênticas, basta transcrever uma palavra de cada grupo para obter a transcrição de todo o documento, reduzindo assim o trabalho de transcrição a uma fração do total.

Além da escrita cursiva ser um grande desafio, os documentos apresentam seus próprios desafios, pois são documentos que se degradaram ao longo de inúmeros anos por mau uso e armazenamento impróprio, que se traduziram em diversos artefatos como manchas, ranhuras e buracos. Além disso, o próprio processo de digitalização pode introduzir mais artefatos, como ruído por iluminação não uniforme e o efeito *show through* (quando a folha de trás fica visível, ver Figura 3.2(b)) [Fernández et al., 2011].

1.2 Objetivos

Este trabalho visa investigar técnicas de processamento de imagem e de inteligência artificial que levem ao resultado desejado de agrupar palavras idênticas para auxilio na transcrição. Na área de processamento de imagem neste trabalho foram abordadas técnicas de redução de ruído e de um pré-processamento para tornar a escrita invariante em relação a espessura do traçado. Na área de inteligência artificial foram abordadas maneiras de extrair séries temporais das palavras para posterior comparação pela medida de distância *Dynamic Time Warping*.

1.3 Organização do Trabalho

Na Capítulo 2 é feita a revisão bibliográfica, que está dividida em duas grandes áreas: Processamento de Imagem e Inteligência Computacional. Em Processamento de Imagem, na Seção 2.1, é discutida a morfologia matemática aplicada ao processamento de imagem para tornar o traçado invariante à espessura e sua utilização indireta no tratamento de ruído, e também a transformada de Hough [Hough, 1962] para detecção de linhas; na Seção 2.2, sobre Inteligência Artificial, a proposta de extração de uma série temporal do contorno da palavra é justificada, e a medida de distância entre séries temporais, a *Dynamic Time Warping*, é explicada; por fim, na Seção 2.3, o algoritmo de agrupamento de dados *average linkage* é detalhado.

Em Desenvolvimento do Trabalho, no Capítulo 3, que segue a mesma estrutura do Capítulo 2, são discutidos os detalhes de implementação das técnicas propostas. Além disso, os resultados dos algoritmos sobre as imagens são discutidos e analisados.

Por fim, na Conclusão, no Capítulo 4, são feitas as considerações finais sobre o trabalho desenvolvido e uma breve descrição sobre trabalhos futuros.

2 *Revisão Bibliográfica*

2.1 Processamento de Imagem

2.1.1 Considerações Iniciais

No projeto de graduação anterior [Cavalcante, 2012], dentre as atividades, foi abordado o problema de segmentar uma imagem, ou seja, gerar uma imagem binária na qual a cor branca indica as regiões de interesse (*foreground*) e o resto da imagem tem a cor preta (*background*). Dentro os métodos avaliados, optou-se pelo *variable thresholding*, que utiliza a média e a variância da intensidade dos pixels em uma vizinhança para determinar se o pixel que está sendo avaliado pertence ou não a região de interesse, gerando assim a imagem binária. A partir da imagem binária é possível extrair as componentes conexas dos pixels segmentados. Uma componente conexa de pixels tem a mesma definição da componente conexa da teoria dos grafos, bastando que se tome cada pixel segmentado (pixel de *foreground*) um vértice e definir uma aresta entre pixels segmentados caso haja adjacência na vizinhança 8-conectada (ver a Figura 2.1, no qual em preto são os pixels que pertencem a vizinhança 8-conectada do pixel p). Uma componente conexa na parte de processamento de imagem recebe o nome de *token*, e uma palavra é formada pelo agrupamento de um ou mais *tokens*.

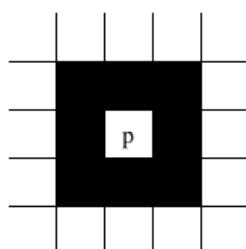


Figura 2.1: Ilustração da vizinhança 8-conectada. Em preto são os pixels pixels que pertencem a vizinhança 8-conectada do pixel p .

Neste projeto de graduação, foram definidas duas metas para a parte de processamento de imagem: aplicar uma técnica para redução de ruído e tornar a escrita invariante a espessura.

É comum em processamento de imagem, após a aplicação de uma técnica de segmentação, haver pixels de *foreground* que na verdade pertencem ao *background*, por isso é necessário uma técnica para eliminação ou redução de ruído. Uma das abordagens mais simples consiste na utilização da operação de erosão da morfologia matemática (ver Seção 2.1.2). Uma outra abordagem simples consiste na remoção de artefatos por regras pré-definidas por alguém com conhecimento do domínio, como por exemplo, é preciso que haja uma quantidade mínima de pixels em um *token* para que ele seja um candidato a uma palavra.

Como este trabalho extrai características do contorno da palavra, é importante reduzir o efeito de fatores que afetem o contorno e que distanciem palavras idênticas. Uma suposição deste trabalho é que a espessura do contorno é um desses fatores. Para neutralizar o efeito da espessura do traçado, decidiu-se implementar uma técnica de esqueletização, ou seja, uma técnica que deixa o traçado em 1 pixel de espessura, tornando assim, a escrita invariante a espessura. Esta técnica de esqueletização é apresentada na Seção 2.1.2, e na sequência, é apresentada uma técnica de pós-processamento para a retirada de artefatos espúrios criados durante a técnica de esqueletização.

Por fim, após a aplicação destas técnicas, é aplicada a técnica de componentes conexas para determinação dos *tokens*, e então, a união dos mesmos.

2.1.2 Morfologia Matemática aplicada ao Processamento de Imagem

A morfologia matemática, que tem como base a teoria dos conjuntos, é uma ferramenta utilizada na área de processamento de imagem para extrair informações como contornos e formas de regiões, esqueletos, componentes conexas, e etc. Em processamento de imagem, os elementos que compõe os conjuntos são os pixels de *foreground* da imagem binária. Neste contexto, cada elemento é uma tupla de 2 atributos, que são a sua coordenada (x, y) na imagem; e um conjunto é um grupo de um ou mais elementos. A partir desta definição básica, podemos definir dois conceitos básicos sobre um conjunto. O primeiro deles é a reflexão, que é definida como:

$$\hat{B} = \{w | w = -b, b \in B\}$$

no qual b é um elemento do conjunto B que tem as suas coordenadas invertidas $(-x, -y)$, e isto quando aplicado a todos os elementos do conjunto resulta na reflexão do conjunto, denotado

por \hat{B} . O outro conceito básico é a translação por $z = (z_x, z_y)$ unidades, que é denotada por $(B)_z$ e é definida como:

$$(B)_z = \{c | c = b + z, b \in B\}$$

Translação e reflexão são usados em conjunto com elementos estruturantes (*structuring elements*) para definir operações para extrair informações sobre imagens. Elementos estruturantes (SE) são pequenos conjuntos ou sub-imagens de forma pré-definida como ilustrado na Figura 2.2. Em SEs a convenção adotada é de que quadrados escuros denotam pixels de *foreground*, sem coloração denotam pixels de *background* e caso haja um *X* denota uma condição de *don't care* (tanto faz se é *background* ou *foreground*). Além disso, é preciso definir a origem do SE, que nesse trabalho será sempre no centro, então sua localização será omitida, mas em geral é ilustrado por um ponto.

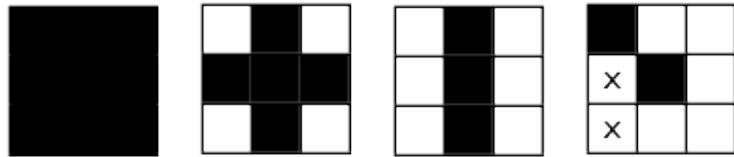


Figura 2.2: Ilustração de elementos estruturantes. Elementos que pertencem ao *foreground* são denotados pelos quadradinhos escurecidos; quadradinhos em branco denotam elementos que pertencem ao *background*; e quadradinhos com *X* denotam uma condição de *don't care*.

A partir destas definições podemos definir duas operações básicas: erosão e dilatação.

Erosão

Seja a imagem em estudo a presente na Figura 2.3(a), e seja o SE como descrito pela Figura 2.3(b). Então a operação de erosão consiste em, para cada pixel da imagem em estudo, posicionar o SE sobre ele (origem do SE alinhado sobre o pixel em análise). Caso todos os elementos de *foreground* do SE estejam sobrepostos sobre pixels de *foreground* da imagem, então torne o pixel em análise na imagem resultante como *foreground*, caso contrário o torne *background*. O resultado desta operação é ilustrado pela Figura 2.3(c).

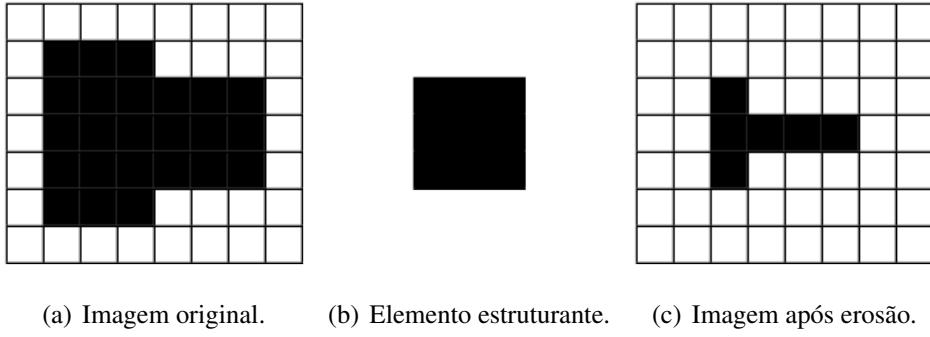


Figura 2.3: Ilustração da aplicação da operação de erosão sobre uma imagem.

De forma formal, seja a imagem em estudo o conjunto A , e seja o SE o conjunto B , então a operação de erosão pode ser definida por:

$$A \ominus B = \{z | (B)_z \subseteq A\}$$

Dilatação

A dilatação consiste em, para cada pixel da imagem em estudo, posicionar a reflexão do SE sobre o pixel em análise, então, caso algum dos pixels de *foreground* do SE esteja sobreposto sobre um dos pixels de *foreground* da imagem em estudo, torne-se o pixel em análise na imagem resultante *foreground*, caso contrário o torne *background*. Uma ilustração desse processo pode ser visto na Figura 2.4.

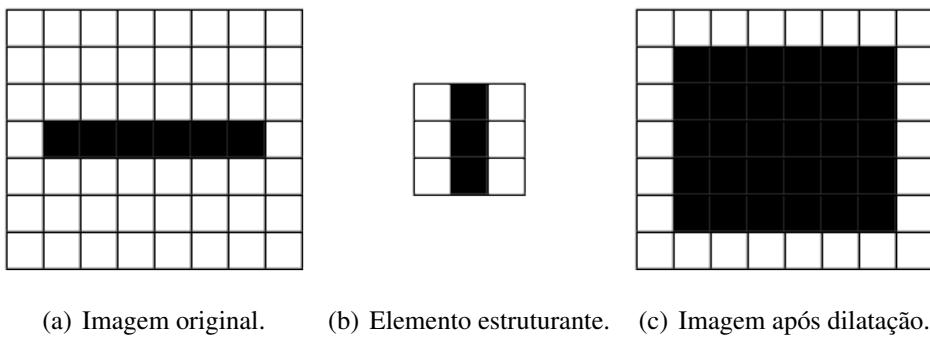


Figura 2.4: Ilustração da aplicação da operação de dilatação sobre uma imagem.

A formalização deste processo é dado por (mantendo a notação de que A é a imagem origi-

nal e B é o SE):

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

Skeletonizing

Uma forma para tornar a escrita invariante à espessura do traçado é modificar o traçado para que ele sempre tenha a espessura de um único pixel, e uma maneira de obter esse resultado é por extrair o esqueleto da imagem. Apesar da morfologia oferecer um método simples para obter o esqueleto de uma imagem [Serra, 1983], tal método não garante que o esqueleto final seja conexo, ou seja, pixels de *foreground* que antes pertenciam a mesma componente conexa continuem a pertencer a mesma componente. Para garantir que o esqueleto seja conexo é preciso lançar mão de uma formulação heurística [Gonzalez and Woods, 2007]. A razão para esqueletização (*skeletonizing*) estar dentro da seção de morfologia é que um pós-processamento aplicado a ele usa morfologia, e mesmo que ele não use morfologia diretamente, o seu entendimento é mais natural nesta sequência.

O esqueleto de uma região pode ser definido pela *medial axis transformation* (MAT) proposta por [Blum, 1967]. A MAT pode ser compreendida de forma intuitiva com a seguinte analogia: seja uma floresta delimitada por uma borda, então incendeie toda a sua borda, e vamos assumir que o fogo se propaga com velocidade constante para todos os lados. Então o esqueleto é dado por todos aqueles pontos em que o fogo alcançou tal ponto pela primeira vez por mais de uma frente ao mesmo tempo. A Figura 2.5 ilustra duas regiões com seus respectivos esqueletos (traçado pontilhado).

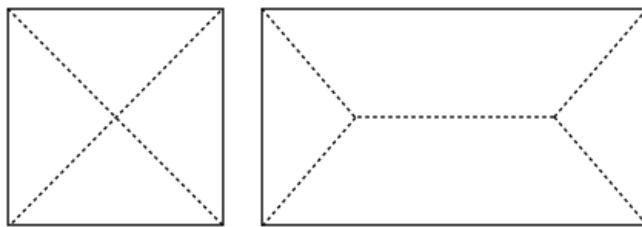


Figura 2.5: Ilustração do esqueleto (traçado pontilhado) de duas regiões.

Apesar da MAT ser de fácil compreensão, a sua implementação, como definida na analogia, é cara computacionalmente, então diversos algoritmos iterativos que tentam se manter fiel a MAT, mas ao mesmo tempo obter uma computação eficiente, foram propostos. O algoritmo

iterativo de 2 passos que será explicado foi retirado de [Gonzalez and Woods, 2007]. Este algoritmo assume que a cor branca tem valor 1 e a cor preta tem valor 0 na imagem binária, além disso definiremos a vizinhança como a vizinhança 8-conectada (arranjada da forma como ilustrada na Tabela 2.1, no qual o pixel p_1 é o pixel em análise), e um pixel de borda é aquele que tenha valor 1 e pelo menos um pixel de valor 0 na vizinhança.

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Tabela 2.1: Vizinhança 8-conectada do pixel p_1 .

O método consiste de sucessivas iterações de 2 passos. No passo 1, um pixel de borda é marcado para remoção caso respeite as seguintes restrições:

1. O número de pixels vizinhos diferentes de zero seja maior ou igual do que 2 e menor ou igual a 6;
2. O número de transições 0-1 na ordem $p_2, p_3, \dots, p_8, p_9, p_2$ seja igual a 1;
3. $p_2 \times p_4 \times p_6 = 0$;
4. $p_4 \times p_6 \times p_8 = 0$.

Ao fim do passo 1, todos os pixels marcados para remoção são removidos para então dar início ao passo 2. O passo 2 é executado sobre todos os pixels de borda, e, bem como no passo 1, um pixel de borda é marcado para remoção se respeitar as seguintes restrições:

1. O número de pixels vizinhos diferentes de zero seja maior ou igual do que 2 e menor ou igual a 6;
2. O número de transições 0-1 na ordem $p_2, p_3, \dots, p_8, p_9, p_2$ seja igual a 1;
3. $p_2 \times p_4 \times p_8 = 0$;
4. $p_2 \times p_6 \times p_8 = 0$.

Ao final do passo 2, todos os pixels marcados para remoção são removidos e o processo se repete até que nenhum pixel seja marcado para remoção.

Prunning

Após a execução da esqueletização, alguns artefatos indesejados são gerados, os quais podem ser retirados por uma técnica de *prunning*. Uma ilustração desses artefatos pode ser visto ao comparar a Figura 2.6(a) com a Figura 2.6(f).

O modo proposto para remoção destes artefatos em [Gonzalez and Woods, 2007] consiste em, primeiramente, determinar todos os *end-points* do conjunto (para a imagem da Figura 2.6(c) os *end-points* são os pontos ilustrados na Figura 2.6(d)). Para isto são usados os elementos estruturantes presentes em 2.6(b), cada qual aplicado uma vez e rotacionado em 90°, sendo assim 8 elementos estruturantes, denotados aqui por $\{B\}$. A detecção e remoção dos *end-points* é formalizada por:

$$X_1 = A \otimes \{B\} = A - (A \circledast \{B\}) = A - ((A \ominus \{B\}) \cap (A^c \ominus \{\bar{B}\}))$$

no qual \circledast é a operação *hit-or-miss*, que neste caso em específico se transforma na erosão de A por B com a intersecção do complemento de A (denotado por A^c) erodido pelo inverso do elemento estruturante B (denotado por \bar{B}). A operação *hit-or-miss* fornece os atuais *end-points* da imagem, os quais são subtraídos da imagem original por aplicar: $A - (A \circledast \{B\})$. Este processo é aplicado por N vezes, no qual N é o tamanho máximo que os artefatos a serem removidos podem ter. Ao final, X_1 resulta na imagem original menos a sucessiva subtração dos *end-points*, que é ilustrada na Figura 2.6(c) para $N = 2$.

Após obter X_1 , é preciso obter os *end-points* dele para recuperar os pixels perdidos que não fazem parte dos artefatos. Isto é feito por aplicar a operação *hit-or-miss* em X_1 :

$$X_2 = X_1 \circledast \{B\}$$

Para o exemplo da Figura 2.6, X_2 é a Figura 2.6(d). O processo de recuperação dos pixels é feito através de N dilatações de X_2 por um SE de 3x3 preenchido por 1s, com o cuidado de delimitar cada dilatação pela imagem original A para evitar novos elementos:

$$X_3 = (X_2 \oplus H) \cap A$$

A Figura 2.6(e) ilustra o resultado armazenado em X_3 . Por fim, basta incorporar estes pixels recuperados ao resultado armazendo em X_1 , para isto basta fazer uma união dos conjuntos, ou seja, $X_4 = X_1 \cup X_3$. O resultado final, X_4 , pode ser visto na Figura 2.6(f).

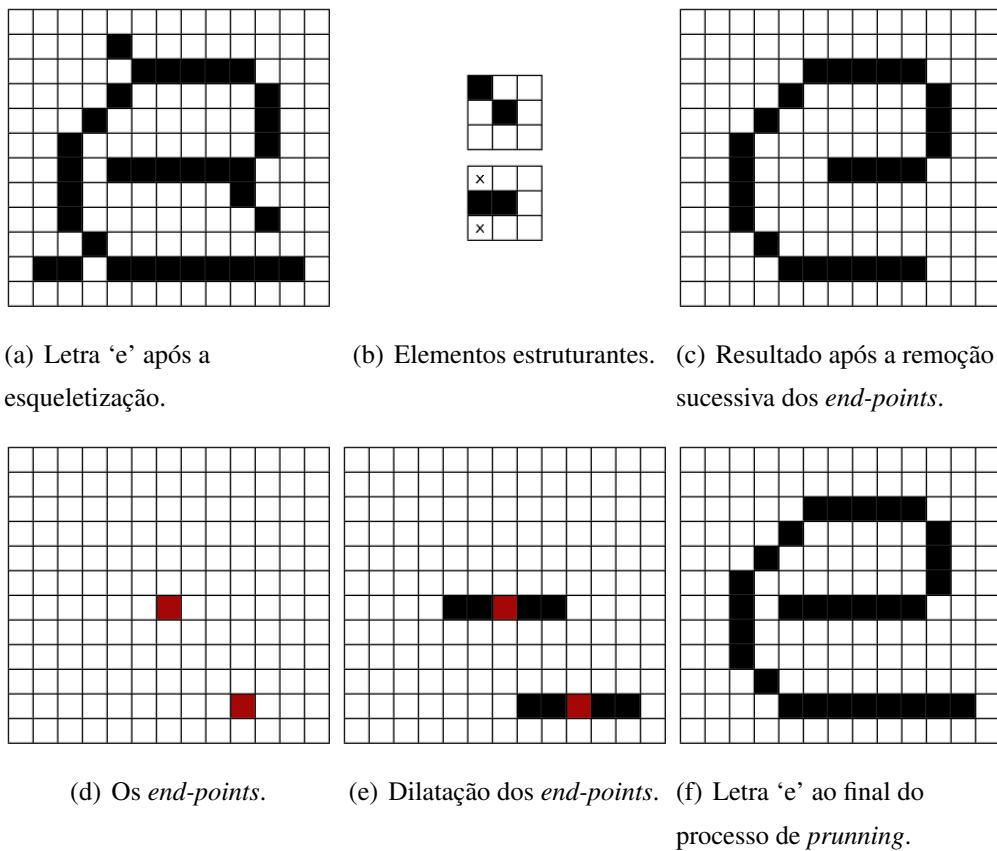
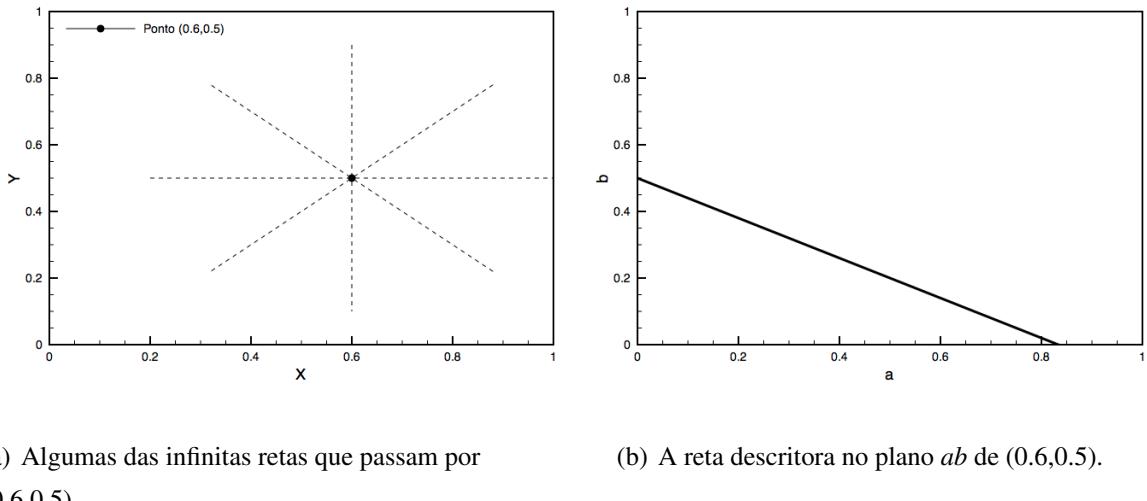


Figura 2.6: Ilustração do processo de *prunning* da letra ‘e’.

2.1.3 Transformada de Hough

Alguns documentos possuem a folha tracejada (ver Figura 3.4), e pelo método de segmentação empregado, essas linhas são segmentadas junto com o traçado da escrita. Como a determinação dos *tokes* que formam as palavras é feita pelas componentes conexas dos pixels segmentados, e dado que essas linhas costumam estar próximas ao traçado, é preciso detectá-las para em um passo futuro removê-las para não correr o risco de uma linha unir várias palavras.

O modo *naïve* para detectar linhas consiste em comparar cada pixel de *foreground* a todos os outros, e então a partir da reta formada por esses dois pixels ver quais outros pixels estão nessa reta (afinal, uma reta que tenha poucos pixels dificilmente se traduz em uma reta de interesse na imagem), tornando este algoritmo $O(N^3)$ em relação a quantidade de pixels de *foreground*. Para resolver este problema de forma mais eficiente, a primeira observação a ser feita é que, no plano xy , por cada ponto $p_{(x,y)}$ passam uma quantidade infinita de retas, dados que para a equação de reta $y = ax + b$ existem infinitas combinações de a e b que geram um determinado y para um dado x (ver Figura 2.7(a)). É possível re-arranjar a equação de reta para colocar em evidência o b , ou seja, $b = y - ax$, e assim podemos definir o plano ab (também chamado de *parameter space*), que tem a vantagem de ser capaz de descrever todas as infinitas retas que passam por um ponto $p_{(x,y)}$ por uma única reta, como ilustrado na Figura 2.7(b).



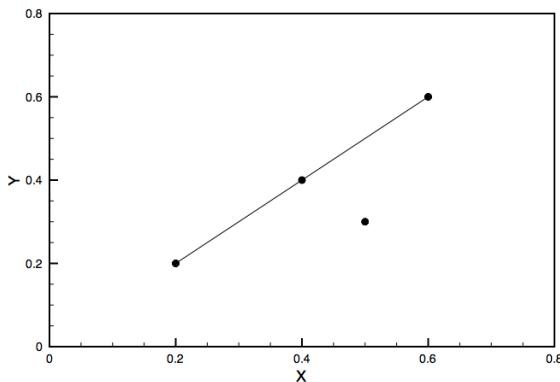
(a) Algumas das infinitas retas que passam por $(0.6,0.5)$.

(b) A reta descritora no plano ab de $(0.6,0.5)$.

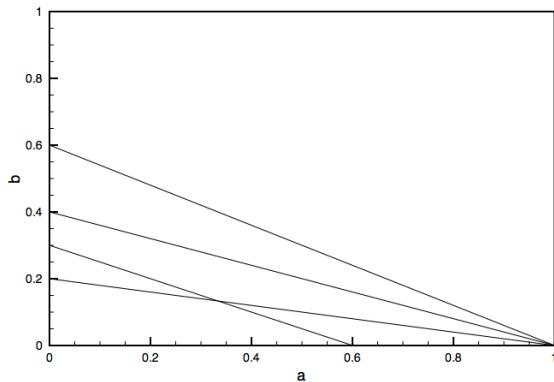
Figura 2.7: Ilustração da transformação do plano xy para o plano ab .

Uma vez que se tenha esse mecanismo para extrair uma reta descritora de cada pixel segmentado da imagem, o próximo passo é discretizar o plano ab . Na discretização do plano ab cada célula se torna um contador de pixels que pertencem à aquela reta. Então a cada vez que

se adiciona uma reta descritora ao plano ab , todas as células que pertencem a reta tem o seu contador incrementado. Um exemplo deste processo pode ser visto na Figura 2.8(a), na qual há 4 pontos e pode-se notar que há uma reta que passa por 3 pontos. Após a transformação para o plano ab é possível observar na Figura 2.8(b) que há um ponto onde 3 retas descritoras se intersectam, definindo assim a reta que contém os 3 pontos.



(a) Exemplo para detecção da reta que contém os 3 pontos.



(b) A intersecção das retas descritoras indica quantos pontos pertencem a reta definida pelo ponto da intersecção.

Figura 2.8: Ilustração do processo de detecção da reta que contém mais pontos

Do modo como foi proposta a implementação, há uma falha, que ocorre quando a reta se aproxima da vertical, e assim a tende ao infinito. Para contornar este problema, basta usar o plano $\rho\theta$, sendo assim, a equação da reta é dada por: $\rho = x\cos\theta + y\sin\theta$. Este método é conhecido como a transformada de Hough e foi proposto em 1962 [Hough, 1962].

2.2 Inteligência Artificial

2.2.1 Considerações Iniciais

Uma série temporal é um conjunto de amostras ordenadas pelo tempo no qual foram obtidas a partir de algum sinal. Uma das maneiras de se medir a semelhança entre duas séries temporais é pelo algoritmo *Dynamic Time Warping* (DTW), discutido na Seção 2.2.2. Inicialmente pode parecer estranho fazer uma comparação de algo visual como uma palavra por meio de uma série temporal, mas dado que documentos históricos possuem inúmeros artefatos, que a segmentação é difícil e que o traçado da escrita costuma variar ao longo do documento, uma abordagem que se distancie da imagem pode gerar bons resultados. Além disso, a medida de distância DTW possui certa robustez contra pequenas distorções, logo, ela pode atenuar estes efeitos [Rath and Manmatha, 2003].

A seguir, na Seção 2.2.2, é explicado o algoritmo DTW, para então, na Seção 2.3, ser feita uma breve revisão do artigo base deste trabalho, que além de propor e justificar o uso de séries temporais para este problema, também explica como é extraída a série temporal de uma imagem de uma palavra.

2.2.2 Dynamic Time Warping (DTW)

Dadas duas séries temporais compostas por uma sequência ordenada de pontos:

$$S = \{s_1, s_2, s_3, \dots, s_n\}$$

$$T = \{t_1, t_2, t_3, \dots, t_m\}$$

desejamos encontrar um caminho W (*warping path*), no qual cada $w_{i,j} \in W$ é um par (i, j) que mapeia um ponto $t_j \in T$ a um ponto $s_i \in S$, de tal forma que a soma cumulativa dos custos de $w_{i,j}$ seja mínimo, no qual o custo de $w_{i,j}$ é a distância entre os pontos s_i e t_j dado por $d(s_i, t_j)$. Além de ser mínimo, este caminho deve respeitar as seguintes restrições:

1. O primeiro elemento deve ser $w_{1,1}$ e o último $w_{n,m}$. Isto garante que uma série temporal será totalmente mapeada na outra.
2. Seja $w_{a,b}$ precedido por $w_{a',b'}$, então $a - a' \leq 1$ e $b - b' \leq 1$. Esta restrição impõe continuidade no caminho.

3. Seja $w_{a,b}$ precedido por $w_{a',b'}$, então $a - a' \geq 0$ e $b - b' \geq 0$. Esta restrição impõe que o caminho seja pelo menos não decrescente.

Este problema pode ser resolvido com programação dinâmica. Seja uma matriz pd de dimensão $n \times m$ ($pd[1,1]$ é o canto superior esquerdo e $pd[n,m]$ é o canto inferior direito), no qual o estado $pd[i,j]$ desta matriz guarda o custo do *warping path* de S_i (os s_i primeiros elementos de S) e T_j (os t_j primeiros elementos de T). Pela primeira restrição temos o caso base $pd[1,1] = d(s_1, t_1)$, e por como definimos o estado, a resposta está em $pd[n,m]$; pela segunda e terceira restrição, só é possível construir o caminho a partir dos seguintes estados adjacentes: $pd[i-1,j]$, $pd[i,j-1]$ e $pd[i-1,j-1]$ [Keogh and Ratanamahatana, 2005]. Com isto, a recorrência *forward* é dada por:

$$pd[i,j] = d(s_i, t_j) + \min\{pd[i-1,j], pd[i,j-1], pd[i-1,j-1]\} \quad (2.1)$$

Para uma ilustração da matriz pd ver Figura 2.9.

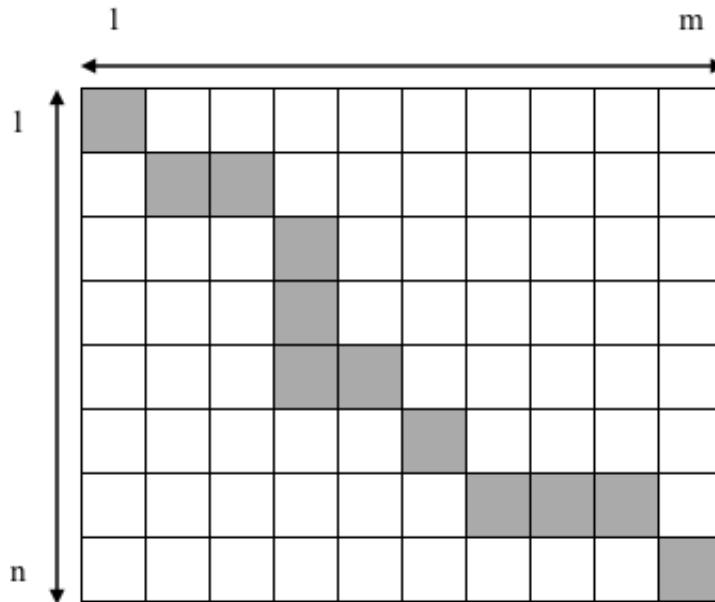


Figura 2.9: Ilustração da matriz pd . Note que o caminho respeita a primeira restrição por começar no canto superior esquerdo e terminar no canto inferior direito; respeita a segunda restrição por haver continuidade no caminho; e respeita a terceira restrição por nunca voltar atrás no que já foi calculado.

Faixa de Sakoe-Chiba e Paralelograma de Itakura

Da recorrência definida na Equação 2.1, uma leitura é: faça o mapeamento de s_i com t_j e escolha o mínimo de resolver o problema de S_{i-1} com T_j , de S_i com T_{j-1} ou de S_{i-1} com T_{j-1} . Quando o mínimo é S_{i-1} com T_j tanto s_{i-1} quanto s_i são mapeados para t_j , e de forma análoga o mesmo acontece quando o mínimo é S_i com T_{j-1} . Isto é bom pois permite alinhar as duas séries, ou seja, alinhar as fases delas, e também ajuda, no caso deste trabalho, a admitir e não penalizar o fato de que nem sempre o autor vai escrever uma mesma palavra do mesmo tamanho. No entanto, grandes sequências em que o mínimo é S_i com T_{j-1} ou S_{i-1} com T_j pode fazer com que duas séries diferentes tenham uma distância pequena, como ilustra a Figura 2.10.

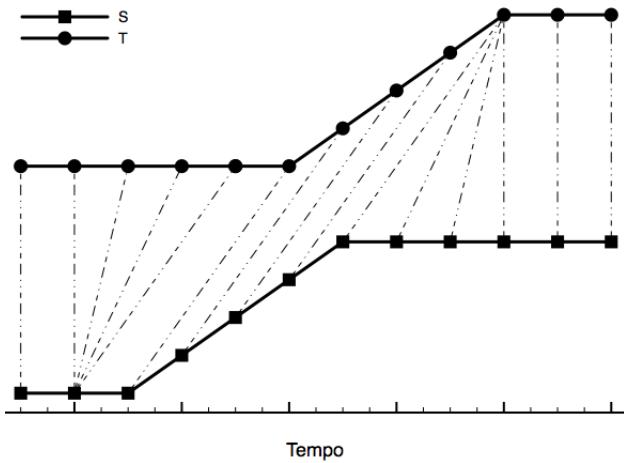


Figura 2.10: Ilustração do problema de não limitar a DTW para que muitos elementos sejam mapeados em poucos. A linha pontilhada indica que o elemento s_i foi mapeado em t_j . Para fácil visualização a série T foi transladada para cima, mas a distância entre as duas séries é 0, pois os elementos mapeados estão na mesma “altura”.

A solução para este problema está em observar o que significa graficamente no caminho (ver Figura 2.9) o mínimo ser S_i com T_{j-1} ou S_{i-1} com T_j . Quando estes estados são escolhidos, o caminho anda para baixo ou para o lado, e não para a diagonal. A solução consiste em impor uma restrição de por onde o caminho pode passar. Duas das mais famosas restrições são: faixa de Sakoe e Chiba [Sakoe and Chiba, 1978], que fixa uma porcentagem a partir da diagonal (ver Figura 2.11(a)); e o paralelograma de Itakura (ver Figura 2.11(b)) [Itakura, 1975].

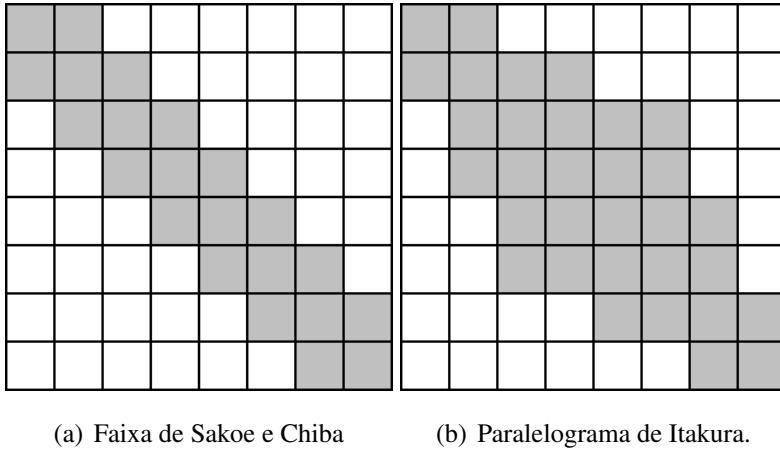


Figura 2.11: Em cinza ilustração dos caminhos possíveis após a aplicação das restrições de Sakoe e Chiba e de Itakura.

2.2.3 Uso do Contorno como Série Temporal

O artigo base deste trabalho é o “*Word Image Matching Using Dynamic Time Warping*” [Rath and Manmatha, 2003], no qual se identifica que algoritmos para reconhecer escrita cursiva ainda tem uma *performance* ruim em documentos históricos, e que pelo momento é mais interessante atacar o problema de determinar todas as ocorrências de uma mesma palavra, assim um humano necessita transcrever somente uma ocorrência de cada grupo.

No artigo, uma vez que se tenha o *bounding box*, que determina a localização de cada palavra no documento, são extraídas 4 características de cada coluna (f_1, f_2, f_3 e f_4), e esse vetor (de tamanho igual a quantidade de colunas da palavra) de 4 características é a série temporal usada para determinar a semelhança entre as palavras. A série temporal neste caso é justificada por uma fraca relação entre a ordem em que o traço foi feito com a disposição da coluna (na primeira coluna é onde a palavra começa a ser escrita e ela termina na última coluna). As 4 características por coluna são: soma dos valores de intensidade dos pixels; o contorno inferior e superior da palavra (a altura do pixel mais próximo do limitante inferior e superior da *bounding box*); e a quantidade de transições entre *foreground* e *background*. Por fim, para executar a DTW é preciso definir a função de distância entre os pontos (amostras) das séries em análise, que neste caso é a soma do quadrado da distância euclidiana das características (ver Equação 2.2).

$$d(s_i, t_j) = \sum_{k=1}^4 (s_i(f_k) - t_j(f_k))^2 \quad (2.2)$$

2.3 Agrupamento de Dados

Uma vez que se tenha uma matriz de distâncias entre as palavras é possível fazer o agrupamento das mesmas. Neste trabalho, o método utilizado para fazer o agrupamento das palavras foi o *average linkage* (ou *Unweighted Pair Group Method using Arithmetic averages*), que tem como resultado final uma hierarquia que pode ser visualizada por um dendograma.

Neste método, inicialmente, cada palavra forma um grupo próprio, então, de modo iterativo os dois grupos mais próximos são unidos até que haja somente um grupo. A definição de proximidade entre dois grupos A e B é dada pela equação 2.3:

$$\text{proximidade} = \frac{1}{|A| \times |B|} \sum_{a \in A} \sum_{b \in B} d(a, b) \quad (2.3)$$

no qual $d(a, b)$ é a distância dada pela matriz de distâncias entre as palavras a e b .

3 *Desenvolvimento do Trabalho*

3.1 Consideração Iniciais

Nesta seção, será mostrado como as técnicas e métodos definidos na Seção 2 são usadas para atingir o objetivo de agrupar palavras extraídas de documentos históricos digitalizados.

Todos os documentos históricos utilizados neste trabalho foram retirados da Biblioteca Nacional Digital¹. A base de dados constituiu de cerca de 30 imagens voltadas a testar principalmente os algoritmos de processamento de imagem, ou seja, cartas e textos curtos em diversas condições de conservação, diferentes estilos de escrita e de diferentes formatos. Assim que a parte de processamento de imagem se consolide, no lugar das cartas e textos curtos serão utilizados livros para dar maior enfoque a parte de agrupamento de palavras.

¹Biblioteca Nacional Digital: <http://bndigital.bn.br/>. Acessado em 07 de Novembro de 2012.

3.2 Processamento de Imagem

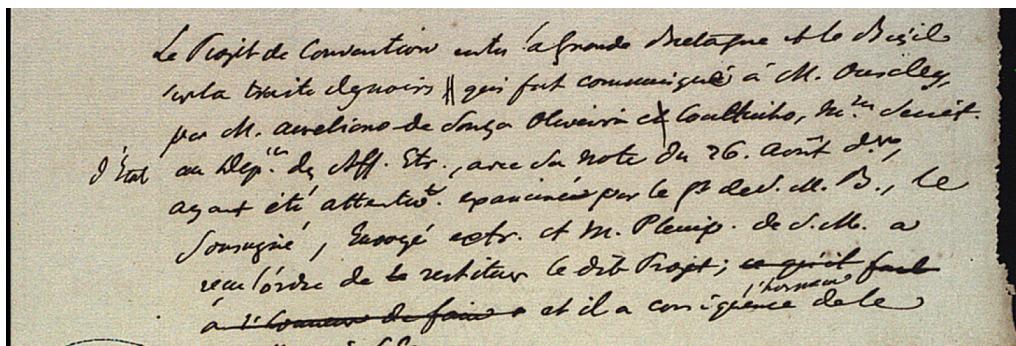
3.2.1 Esqueletização e Tratamento de Ruído

É comum em processamento de imagem aplicar um método de tratamento de ruídos após a aplicação de um processo de segmentação, pois, como argumentado no projeto de graduação anterior [Cavalcante, 2012], é preferível um método de segmentação que gere mais ruído mas mantenha grande parte do traçado da escrita do que um método que gera pouco ruído mas perde muito do traçado da escrita, em especial no traçado que liga os caracteres, pois afeta diretamente a capacidade de determinação das palavras por analisar as componentes conexas. Sendo assim, não foi incomum encontrar documentos que precisariam de uma técnica de tratamento de ruído, como pode ser visto nas Figuras 3.1, 3.2 e 3.3.

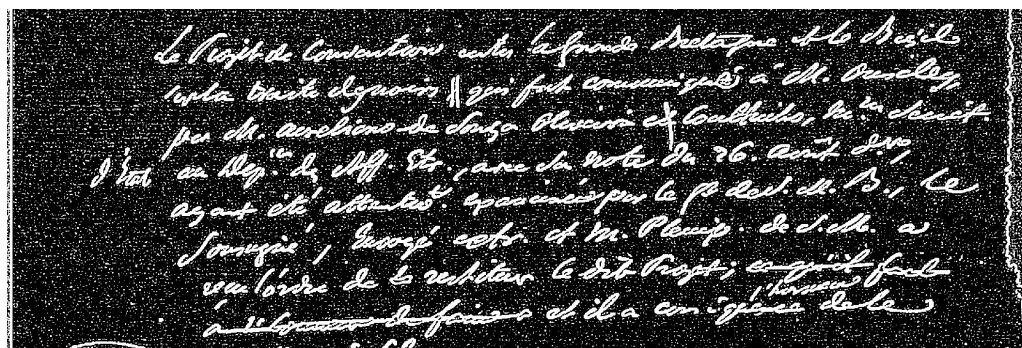
Uma técnica comum para tratamento de ruído em processamento de imagem é aplicar a operação morfológica de erosão. No entanto, mesmo com um dos menores elementos estruturantes possíveis (3×3), não foi possível remover o ruído sem grandes perdas no traçado da escrita, pois em inúmeros documentos a digitalização não tem resolução o suficiente para deixar o traçado com mais de 3 pixels de espessura. Por ora ignoremos este problema e vamos passar a esquelitização da imagem.

Para aplicar o processo de esqueletização não é preciso definir nenhum parâmetro. No entanto, o processo de *prunning*, para remoção de artefatos, necessita da informação de qual é o tamanho máximo de um artefato, que neste trabalho foi estabelecido em 4 pixels.

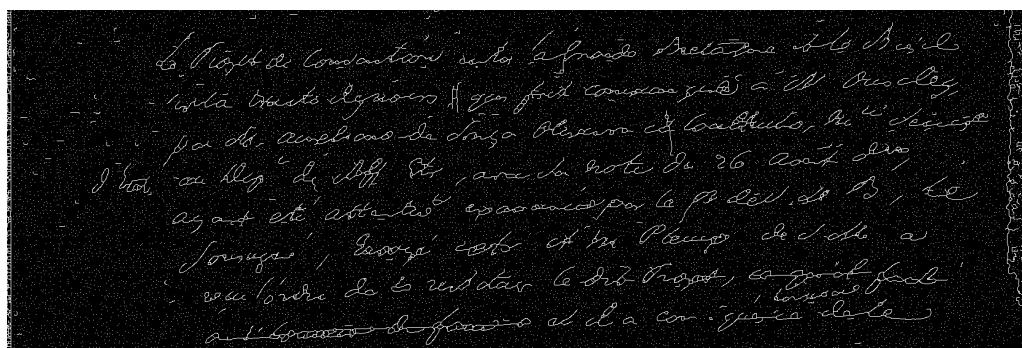
Após a execução da esqueletização e do processo de *prunning*, ocorreu um efeito inesperado. Antes destes processos não era incomum encontrar ruídos compostos por um algomeroado significativo de pixels, mas após a execução destes processos, as componentes conexas dos ruídos passaram a ter menos de 7 pixels, valor que foi definido como o *threshold* para detecção de ruído. Assim, todas essas componentes foram retiradas, e este se tornou o método para tratamento de ruído. Além deste limiar de 7 pixels na componente conexa, também foram impostos outros, como uma largura e altura mínima e uma altura máxima. Um exemplo deste processo pode ser visto na Figura 3.1.



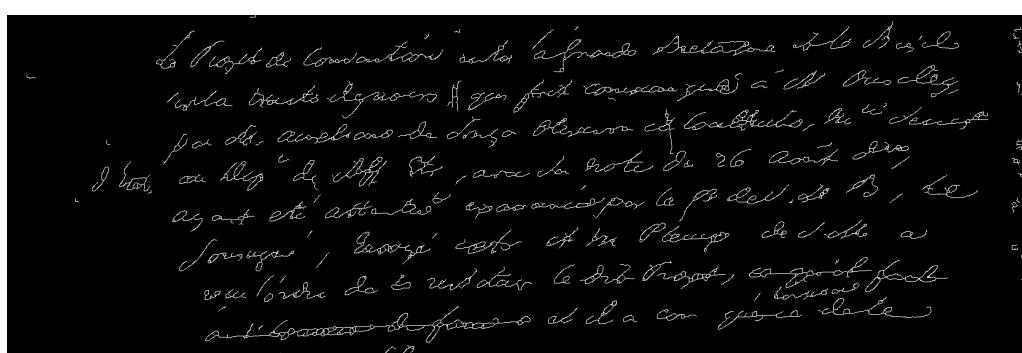
(a) Parte do documento original.



(b) Segmentação da Figura 3.1(a).



(c) Esqueletização e pruning da Figura 3.1(b).



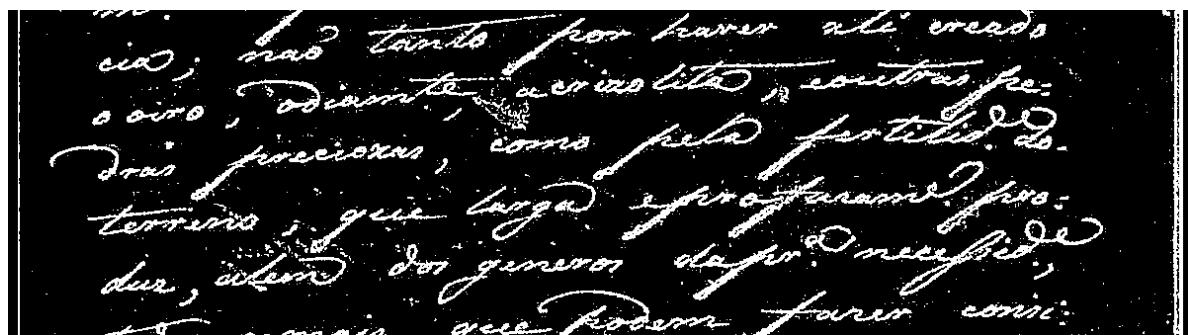
(d) Imagem final após a eliminação do ruído da Figura 3.1(c).

Figura 3.1: Ilustração dos processos de processamento de imagem em um documento com ruído.



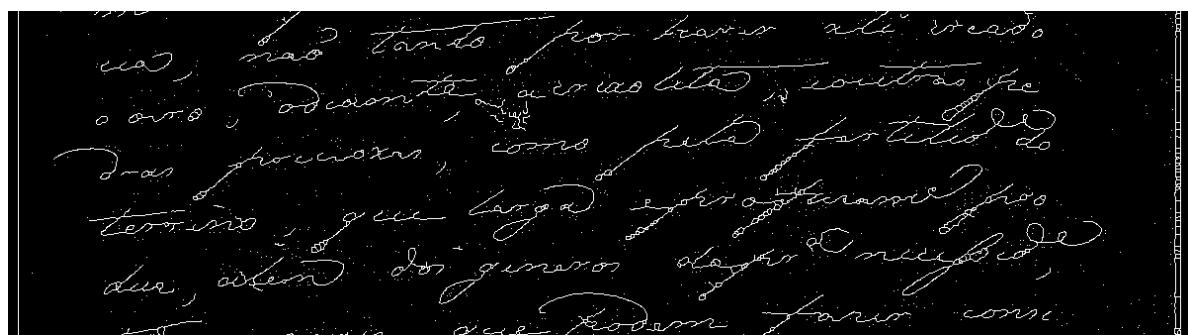
m.
cia; não tanto por haver ali creado
o ouro, adiante acinzelado, outras pe-
dras preciosas, como pela fertilidade do
terreno, que larga e profusamente pro-
duz, além dos gêneros da província,
as mais que podem fazer comi-

(a) Parte do documento original.



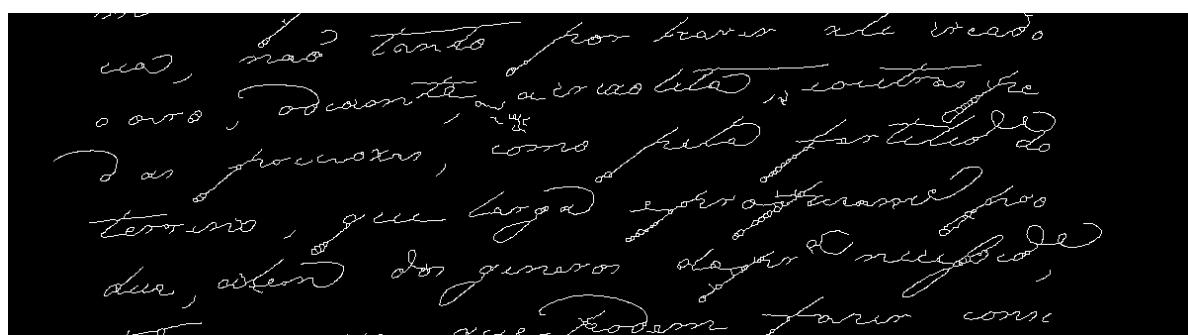
m.
cia; não tanto por haver ali creado
o ouro, adiante acinzelado, outras pe-
dras preciosas, como pela fertilidade do
terreno, que larga e profusamente pro-
duz, além dos gêneros da província,
as mais que podem fazer comi-

(b) Segmentação da Figura 3.2(a).



m.
cia; não tanto por haver ali creado
o ouro, adiante acinzelado, outras pe-
dras preciosas, como pela fertilidade do
terreno, que larga e profusamente pro-
duz, além dos gêneros da província,
as mais que podem fazer comi-

(c) Esqueletização e pruning da Figura 3.2(b).



m.
cia; não tanto por haver ali creado
o ouro, adiante acinzelado, outras pe-
dras preciosas, como pela fertilidade do
terreno, que larga e profusamente pro-
duz, além dos gêneros da província,
as mais que podem fazer comi-

(d) Imagem final após a eliminação do ruído da Figura 3.2(c).

Figura 3.2: Ilustração dos processos de processamento de imagem em um documento com ruído.

The Project of a Convention between S. B. & Brazil
on Slave Trade, wh. was com. to Mr. Bradley by J. Morelino
Ref^o o Ofici. Conf. M. S. of State for the Dep. of S. Aff.,
in his note of the 26th Aug. last, having received the affiance

(a) Parte do documento original.

The Project of a Convention between S. B. & Brazil
on Slave Trade, wh. was com. to Mr. Bradley by J. Morelino
Ref^o o Ofici. Conf. M. S. of State for the Dep. of S. Aff.,
in his note of the 26th Aug. last, having received the affiance

(b) Segmentação da Figura 3.3(a).

The Project of a Convention between S. B. & Brazil
on Slave Trade, wh. was com. to Mr. Bradley by J. Morelino
Ref^o o Ofici. Conf. M. S. of State for the Dep. of S. Aff.,
in his note of the 26th Aug. last, having received the affiance

(c) Esqueletização e pruning da Figura 3.3(b).

The Project of a Convention between S. B. & Brazil
on Slave Trade, wh. was com. to Mr. Bradley by J. Morelino
Ref^o o Ofici. Conf. M. S. of State for the Dep. of S. Aff.,
in his note of the 26th Aug. last, having received the affiance

(d) Imagem final após a eliminação do ruído da Figura 3.3(c).

Figura 3.3: Ilustração dos processos de processamento de imagem em um documento com ruído.

É importante ressaltar que, nas Figuras 3.1, 3.2 e 3.3, pode parecer que parte do traçado foi perdido, mas isto é somente o resultado do fato de que as imagens foram re-dimensionadas para caber nesta folha.

3.2.2 Detecção de Linhas

Como mencionado na Seção 2.1.3, alguns documentos apresentam linhas no papel que foi utilizado para a escrita. Estas linhas, por estarem próximas a escrita, podem criar uma “ponte” de pixels segmentados, unindo palavras que não deveriam estar conectadas. Sendo assim, é preciso detectar estas linhas para possível remoção.

A detecção é feita pela transformada de Hough, discretizando θ em 0.01 radianos e ρ em 1 pixel. Além disso, a quantidade necessária de pixels na reta para ela ser considerada interessante foi definido em 100, e também foi imposta uma restrição sobre a variação da altura da reta nas extremidades da imagem para extrair somente as retas horizontais ($\Delta y \leq 30$). O resultado desta execução sobre a imagem da Figura 3.4 pode ser visto na Figura 3.5.

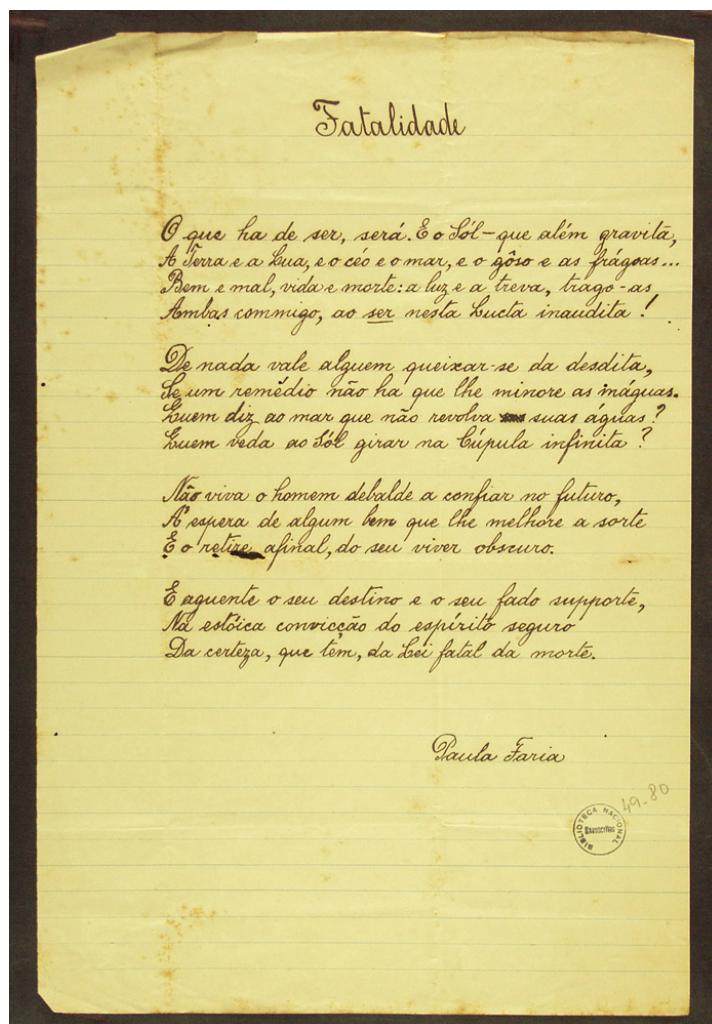


Figura 3.4: Documento com linhas ao fundo.

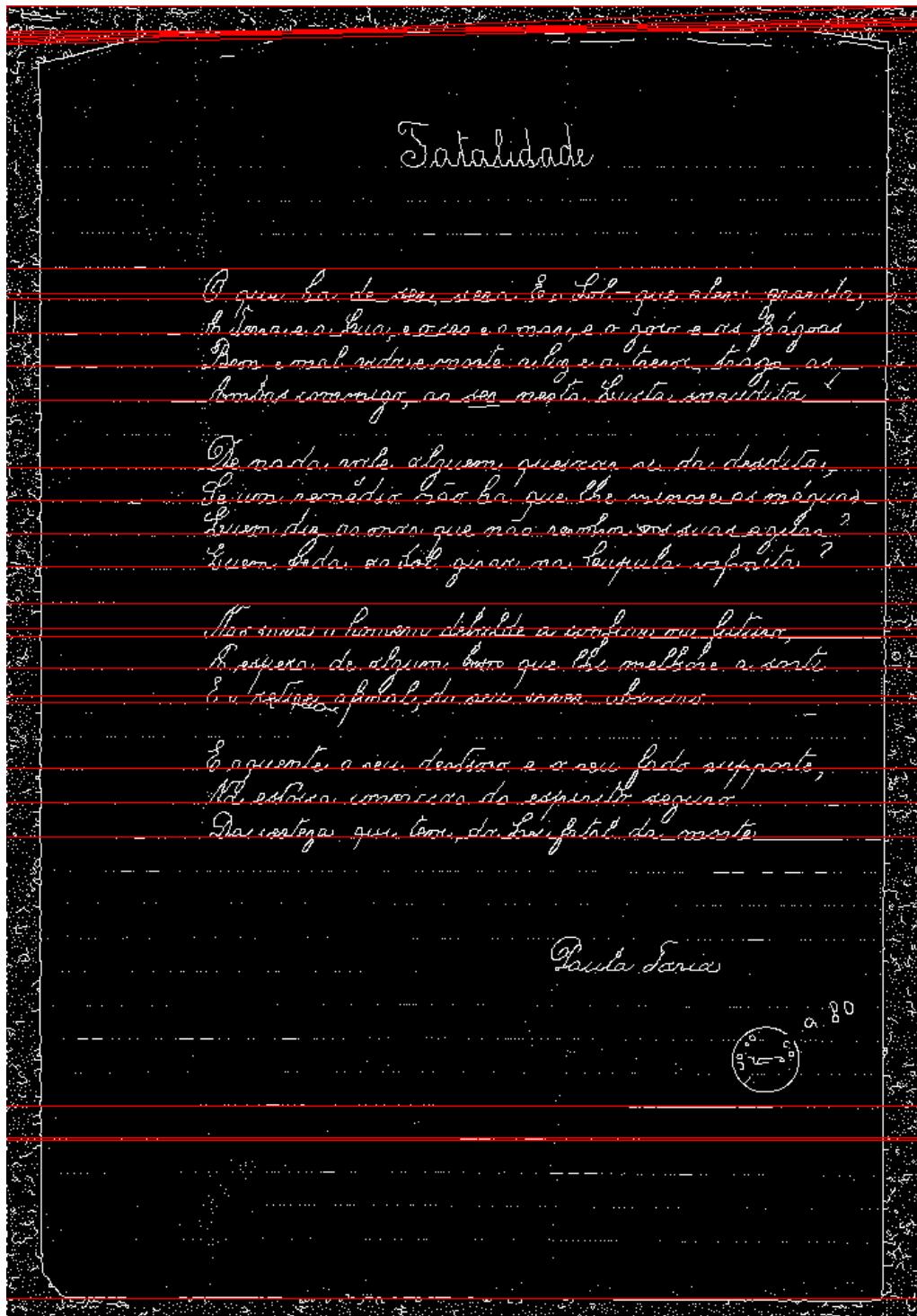


Figura 3.5: Em vermelho as linhas detectadas de 3.4 pela transformada de Hough.

Apesar de conseguir detectar as retas, ainda não foi definido um método para a remoção das mesmas, isto será um dos trabalhos futuros.

3.2.3 Agrupamento de *tokens*

Um *token* é uma componente conexa, como previamente definido. Em geral um *token* não é uma palavra, pois o autor pode escrever os caracteres de forma espaçada ou a segmentação pode ter desfeito o traçado entre os caracteres, sendo assim há o problema de unir os *tokens* para construir uma palavra. Esta etapa será um dos focos dos trabalhos futuros. Por ora, para ser capaz de testar a DTW, o algoritmo de junção de *tokens* se resume a uma série de regras que visam unir *tokens* que estejam próximos, garantindo que eles não fiquem grandes demais.

As Figuras 3.6, 3.7 e 3.8 mostram através de *bounding boxes* o resultado após a execução do algoritmo de junção de *tokens*, ou seja, a localização das palavras.

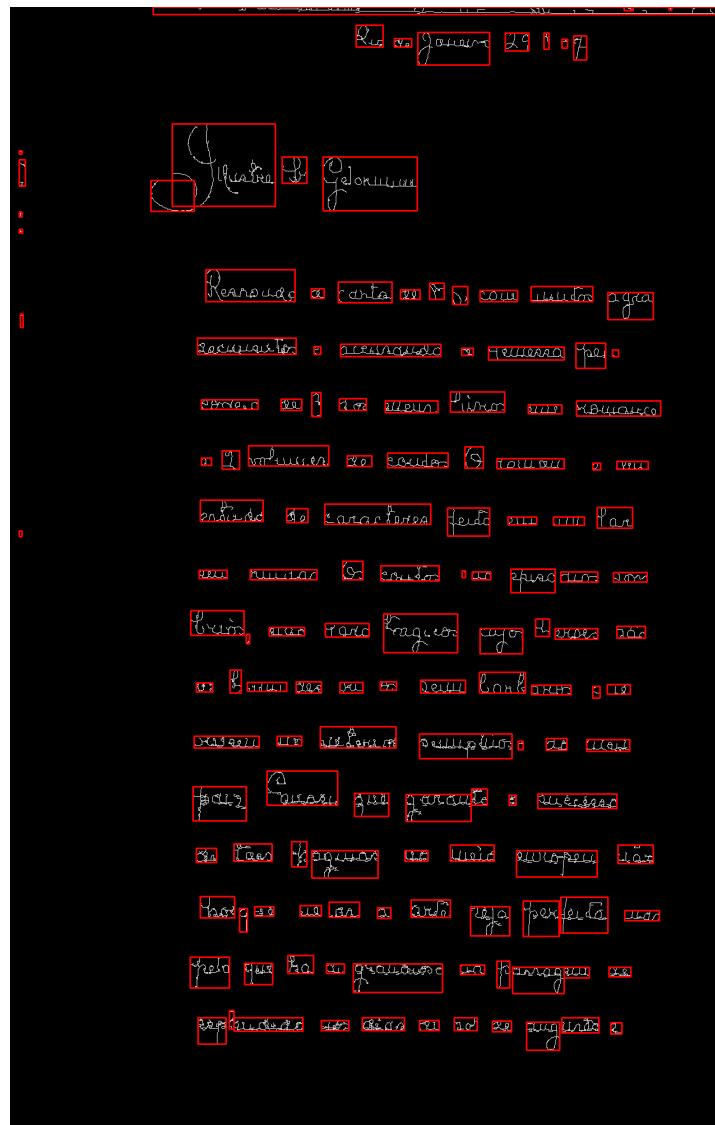


Figura 3.6: Os *bounding boxes* indicam a localização das palavras.

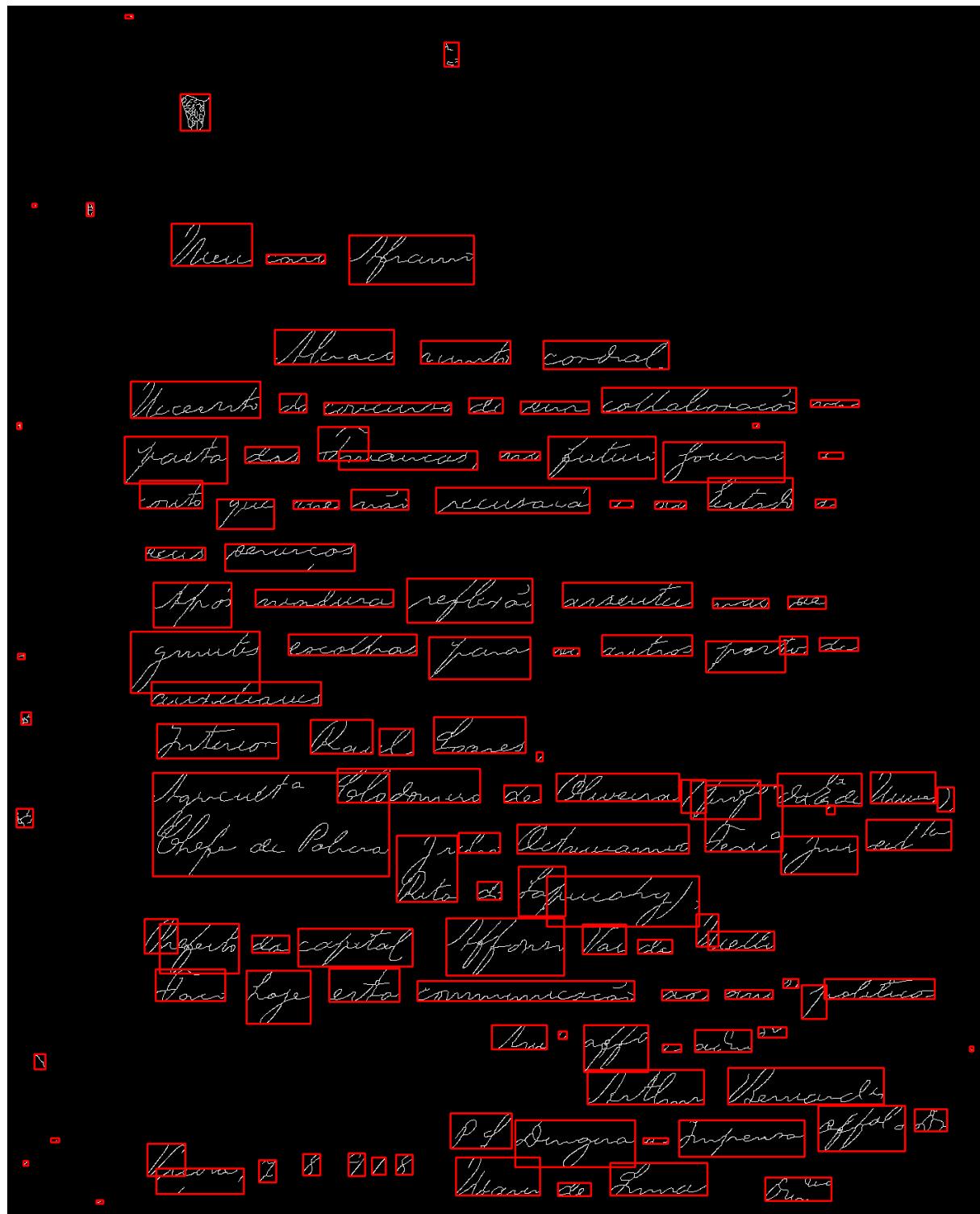


Figura 3.7: Os *bounding boxes* indicam a localização das palavras.

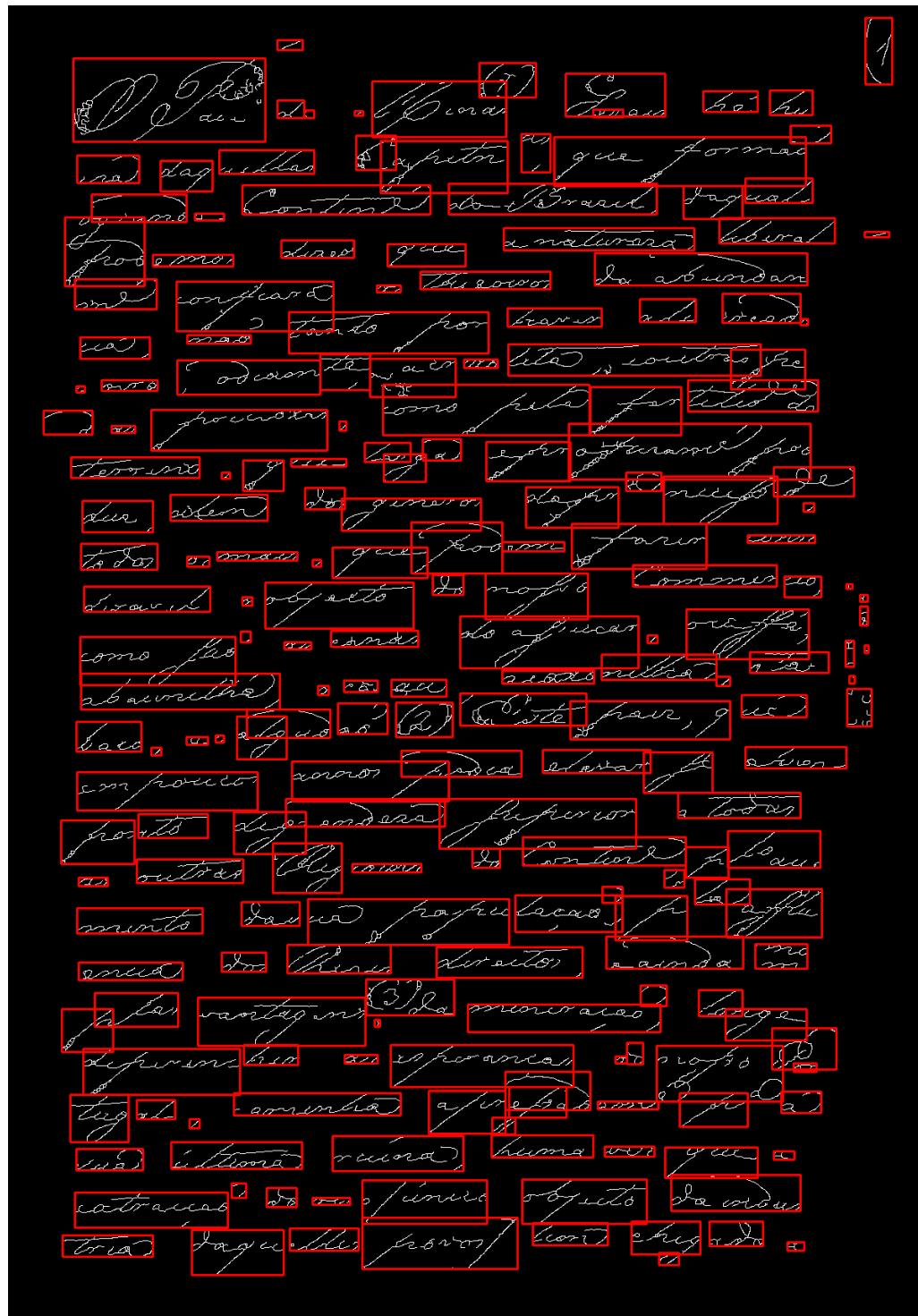


Figura 3.8: Os *bounding boxes* indicam a localização das palavras.

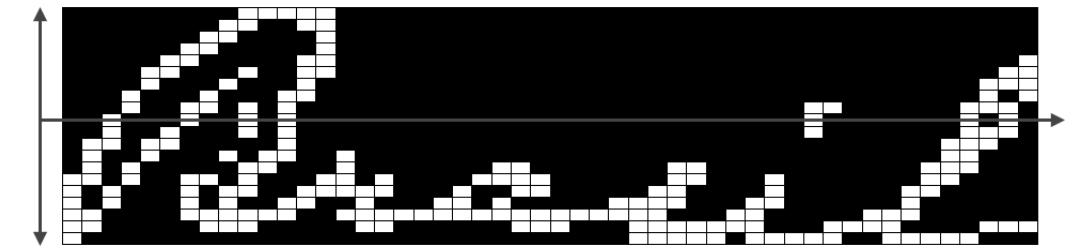
3.3 Inteligência Computacional

Nesta seção, é definida como as séries temporais são formadas a partir das imagens, bem como a função de distância para uso na DTW.

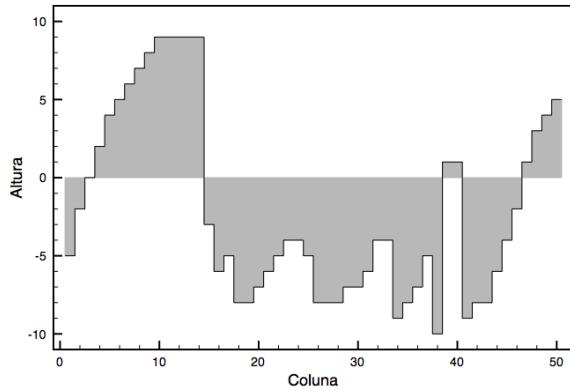
3.3.1 Geração de Séries Temporais

De cada palavra é extraído um vetor de características. Esse vetor tem tamanho igual ou menor à largura (quantidade de colunas) da *bounding box*. Para cada coluna são extraídas três características: f_1 , que é a altura do pixel segmentado mais próximo do limitante superior da *bounding box* em relação à metade da altura da palavra; f_2 , que é a altura do pixel segmentado mais próximo do limitante inferior da *bounding box* em relação a metade da altura da palavra; e f_3 , que é a quantidade de transições de *background* para *foreground* e vice-versa. Para um exemplo, veja a Figura 3.9(a) que representa a palavra “Brasil”. A Figura 3.9(b) corresponde aos valores de f_1 , a Figura 3.9(c) corresponde aos valores de f_2 e a Figura 3.9(d) corresponde aos valores de f_3 .

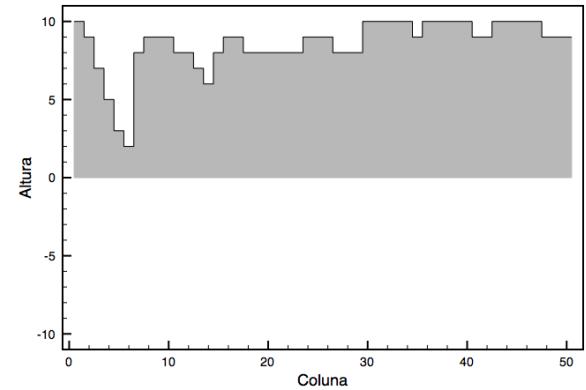
A razão pela qual a altura zero está na metade da altura da *bounding box* (ver Figura 3.9(a)) é para tentar não favorecer a magnitude de nenhum dos dois contornos, para que eles tenham valores semelhantes. Quando em uma coluna não há um pixel segmentado, esta coluna é ignorada e nenhuma característica dela é coletada.



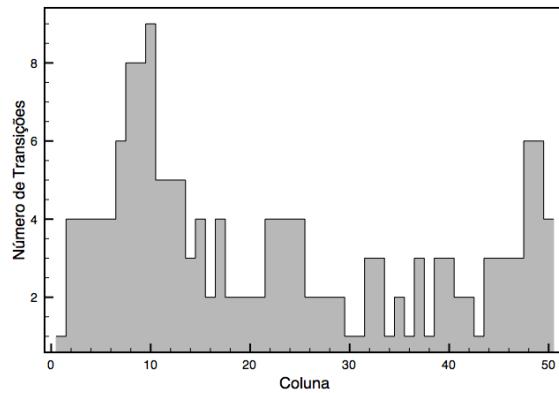
(a) Palavra a ser transformada em série temporal. A altura zero é no eixo que passa na horizontal, metade da altura da *bounding box*.



(b) Contorno superior da palavra.



(c) Contorno inferior da palavra.



(d) Número de transições.

Figura 3.9: Extração da série temporal de uma palavra.

3.3.2 Dynamic Time Warping

Para a execução da DTW, é preciso definir a função de distância, que neste trabalho é a soma da distância euclidiana ao quadrado das duas características, sendo assim a Equação 2.1 se transforma na 3.1.

$$pd[i, j] = \sum_{k=1}^3 d(s_i(f_k), t_j(f_k)) + \min\{pd[i-1, j], pd[i][j-1], pd[i-1][j-1]\} \quad (3.1)$$

3.4 Resultados

Nesta seção, é mostrado o resultado de todos os elementos deste trabalho. Primeiramente, a imagem passa por uma etapa de segmentação, então é aplicado o processo de *skeletonizing* e de *prunning*. Após o *prunning*, é feito o tratamento de ruídos, para então agrupar os *tokens* e determinar as palavras. De cada palavra é então extraída a sua série temporal, da qual é construída a matriz de distância das palavras pela medida de distância *dynamic time warping*. Por fim, é feito o agrupamento por *average linkage*, elaborado com auxílio do software R¹. Todo este processo pode ser visto na Figura 3.10.

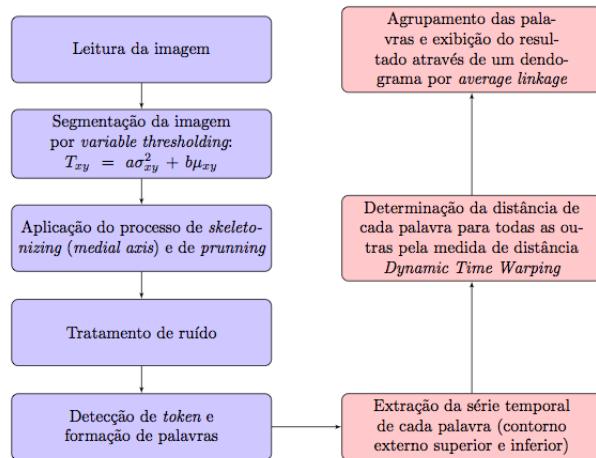


Figura 3.10: Todas as etapas para a obtenção do agrupamento das palavras.

Para ilustração do resultado final de todas essas etapas, foram escolhidos dois documentos (Figura 3.11 e 3.12), no qual os *bounding boxes* indicam as palavras escolhidas para efetuar o agrupamento (os números no *bounding boxes* são para fácil associação com o dendrograma).

¹<http://www.r-project.org/>. Acessado em 07 de Novembro.

1079

Projeto de **Letra**, para o Hino Nacional.

1022 1064
 Queremos olo **esperança** aos meus pais
 De sempre acima o brado do **Brasante**,
 E o sol da **Liberdade**, que ramos fulgidos,
P. Que no eco de países seceder nos tempos

1019

1169
 Se o povo desse aguardando
 Conquemos conquistar esse brado **forte**,
 Pelo amor da **liberdade**
 Despida o mundo pelo o **projecção** **morte**.

1118 1156

1108 1138
 O **Brasil** amado 1154
 Terra da **saude** 1131
 Saber, **labore**

1008 1059
 Brasil, com **sombra** seteindo, sacerdote mudo
 De amor e de esperança à terra disse
 Encendo em teu céu avel, redondo e fulgido,
 A magia do **Brazileiro** resplandece.

1123

1070
 Gigante pela **projecção** montanha,
 É belo, é grande, resplendo solaroso,
 E o teu fruto espelha essa grandezza.

1111 1133
 Terra adorada 1155
 Letra **ordem** 1109 1132 1139
 E te, **Brasil**, 1146
 O **Brasil** amado 1086
 Nos **futuros** de teu **flambo** es vere goitado
 Para os **secreto**, **Brasil**, 1159 1125

Figura 3.11: Palavras escolhidas para agrupar da primeira página do esboço do hino nacional.

I

Deus é o que é e é um Deus exaltado,
E no aniversário da morte e o céu profundo,
Fazendo, o Brasil, juntar de maneira
A felicidade ao P de Deus adorado!

De que a terra mais jardim,
Tão resplandecentes campos têm mais flores,
"Nós os longos tempos viveremos,"
Nossa viva voz fala dous, um e amores!

2125
D / Palma exaltação, 2157
2111
Ecolatridos, 2150
Salvo, salvo

2008 2100
Fim é signo de auro esterno símbolo
e primitivas que sentem estrelada
é signo o verde lindo desse planeta
que os futuros e gloriosos nos festejado!

2162
Mais em justas erguendo a clava forte,
Vêns que seu filho tem nos feito a bacia,
Nem temere, que te adoro, a programar morte

2168
2169
2106 2139
Eressa a dorosa, 2137 2164
Entre outras real, 2148
E tem Brasil, 2114
D / Portaria amada, 2154
2086
2096 2169
Dois filhos se tem flamedo que o vento, 2169
Outros amada, Brasil

Outubro

Opção Diferente Entendendo

709

Figura 3.12: Palavras escolhidas para agrupar da segunda página do esboço do hino nacional.

Antes de mostrar o dendrograma, é preciso introduzir um fator de correção que foi introduzido na matriz de distância da qual é gerada o dendrograma. Quando duas palavras tem séries pequenas (largura pequena), por mais que sejam muito diferentes, a sua distância tende a ser baixa, pois haverá poucas somas. Quando a palavra é grande, mesmo que sejam iguais ou semelhantes, a distância tende a ser grande, pois há muitas somas de poucas diferenças. Para contornar este problema foi introduzido um fator de correção que multiplica a distância entre as séries. Este fator de correção divide a maior série em comparação pela maior série de todas as séries:

$$DTW(i, j) \times \frac{\max_{k=1, \dots, n} \{comprimento(k)\}}{\max \{comprimento(i), comprimento(j)\}}$$

Por fim, para executar a DTW, é necessário definir a largura da faixa de Sakoe e Chiba. Neste trabalho, foram testadas as larguras de 10%, 20%, 30% e 40%. Para este caso, o melhor resultado obtido foi com a largura de Sakoe e Chiba em 30% e com o fator de correção. O dendrograma pode ser visto na Figura 3.13.

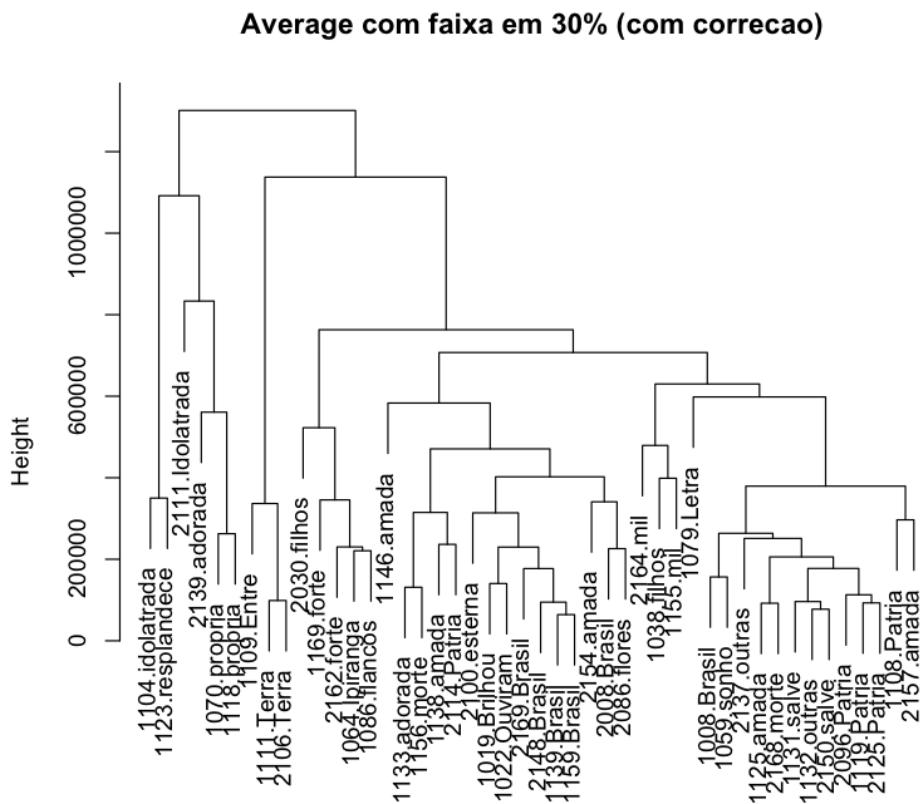
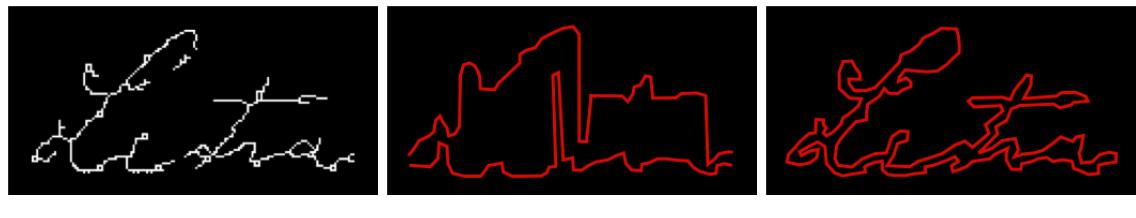


Figura 3.13: Agrupamento das palavras da Figura 3.11 e 3.12.

No dendograma é possível observar vários grupos corretos como: {1070.propria, 1118.propria}, {1111.Terra, 2106.Terra}, {1169.forte, 2162.forte}, {2169.Brasil, 2148.Brasil, 1139.Brasil, 1159.Brasil} e {2069.Patria, 1119.Patria, 2125.Patria}; e alguns com alguma confusão como o grupo {1131.salve, 1132.outras, 2150.salve}. Além disso, palavras como 1104.idolatrada, 1123.resplandece, 1146.amada e 1079.Letra seriam consideradas *singletons* (formam um grupo de um único elemento) por terem uma distância muito alta em relação as outras palavras.

Apesar de grupos corretos terem sido obtidos, este dendrograma ainda apresenta baixa qualidade. Um ponto a ser estudado para ganhar qualidade está no contorno. Apesar do contorno externo apresentar grande informação, há suspeita de que ele não está sendo utilizado em sua melhor forma. Seja o caso da palavra “Letra”: nota-se que o ‘L’ esconde o ‘e’ e o mesmo acontece com os caracteres ‘ra’ que são encobertos pelo ‘t’ (ver Figura 3.14(b)). Uma proposta de contorno desejado é ilustrado na Figura 3.14(c).



(a) Palavra “Letra”. (b) Contorno extraído atualmente. (c) Contorno desejado.

Figura 3.14: Ilustração do contorno extraído atualmente e do contorno desejado.

Para obter este contorno desejado, a técnica de contorno ativo está sob consideração [Nixon and Aguado, 2008]. Também, sob atual consideração, está o uso da silhueta [Kaufman and Rousseeuw, 1990], que é um critério de validade relativo, como indicador do ponto de corte no dendrograma para obtenção dos grupos. Uma vez que se tenha os grupos, uma proposta de quantificação dos resultados está no uso do índice de Jaccard.

4 *Conclusão*

4.1 Considerações Finais

Este trabalho teve como objetivo validar a possibilidade de uso de séries temporais na determinação da ocorrência de todas as instâncias de uma mesma palavra em documentos históricos de escrita cursiva. Caso este problema venha a ter uma resolução satisfatória a transcrição de livros históricos poderá ser feita em uma fração do total que seria despendido em um método que não tenha esse auxílio, pela suposição de que em longos documentos as palavras se repetem inúmeras vezes. Com o resultado exibido no dendograma confirma-se o indicativo de viabilidade desta abordagem, que requer diversas técnicas de processamento de imagem como segmentação, esqueletização e tratamento de ruído; e técnicas que envolvem séries temporais e agrupamento.

4.2 Considerações sobre o Curso de Graduação

A diversidade de ramos da computação apresentada durante o curso é interessante. No entanto, a obrigatoriedade de aprofundar o conhecimento em todos esses ramos (pelas matérias que possuem sequência) é danosa para os estudantes, pois torna a carga horária grande demais, e isso afeta e inibe projetos extracurriculares. Além disso, não são todos que desejam se aprofundar em todos os ramos, recomendando um desenvolvimento mais individual do aluno, ou seja, que haja mais matérias optativas e mais cedo. Sobre as matérias optativas, acredito também que seria interessante estimular o aluno a fazer optativas livres, afinal eu acredito que o futuro está na intersecção de várias áreas. Por fim, minha última crítica ao curso, é que ele deixa a desejar na matemática voltada a programação, por exemplo na análise de complexidade e prova de corretude de algoritmos, algo que só aprendi por ser membro do Grupo de Estudos da Maratona de Programação.

4.3 Trabalhos Futuros

Na parte de processamento de imagem, a pesquisa será feita em relação a etapa de junção de *tokens* para formação de palavras e da extração do contorno. Na parte de inteligência computacional, o trabalho se focará na determinação de características que descrevam melhor a palavra. Além disso, também é necessário pesquisar métodos para determinação dos *clusters* que determinam todas as instâncias de uma mesma palavra, e também uma forma de quantificar a qualidade destes *clusters*.

Referências Bibliográficas

- [Arica and Yarman-Vural, 2002] Arica, N. and Yarman-Vural, F. T. (2002). Optical character recognition for cursive handwriting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:801–813. Citado na pna [5](#).
- [Blum, 1967] Blum, H. (1967). A Transformation for Extracting New Descriptors of Shape. In Wathen-Dunn, W., editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge. Citado na pna [11](#).
- [Cavalcante, 2012] Cavalcante, L. S. (2012). Extração e análise pelo contorno de palavras de textos históricos. Monografia de Conclusão de Curso. Citado nas pnas [7](#) e [23](#).
- [Fernández et al., 2011] Fernández, D., Lladós, J., and Fornés, A. (2011). Handwritten Word Spotting in Old Manuscript Images Using a Pseudo-structural Descriptor Organized in a Hash Structure. In Vitrià, J., Sanches, J., and Hernández, M., editors, *Pattern Recognition and Image Analysis*, pages 628–635. Springer Berlin / Heidelberg. Citado na pna [5](#).
- [Gonzalez and Woods, 2007] Gonzalez, R. C. and Woods, R. E. (2007). *Digital Image Processing*. Prentice Hall, 3 edition. Citado nas pnas [11](#), [12](#), e [13](#).
- [Hough, 1962] Hough, P. (1962). Method and Means for Recognizing Complex Patterns. U.S. Patent 3.069.654. Citado nas pnas [6](#) e [16](#).
- [Itakura, 1975] Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):67–72. Citado na pna [19](#).
- [Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. (1990). *Finding groups in data: an introduction to cluster analysis*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley. Citado na pna [38](#).
- [Keogh and Ratanamahatana, 2005] Keogh, E. and Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3). Citado na pna [18](#).
- [Niels and Vuurpijl, 2005] Niels, R. and Vuurpijl, L. (2005). Using Dynamic Time Warping for intuitive handwriting recognition. In *Proc. IGS2005, 2005. In*, pages 217–221. Citado na pna [5](#).
- [Nixon and Aguado, 2008] Nixon, M. and Aguado, A. S. (2008). *Feature Extraction & Image Processing for Computer Vision*. Academic Press, 2 edition. Citado na pna [38](#).
- [Rath and Manmatha, 2003] Rath, T. M. and Manmatha, R. (2003). Word Image Matching Using Dynamic Time Warping. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:521. Citado nas pnas [5](#), [17](#), e [20](#).

[Sakoe and Chiba, 1978] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49. Citado na pna 19.

[Serra, 1983] Serra, J. (1983). *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA. Citado na pna 11.

[Tomai et al., 2002] Tomai, C., Zhang, B., and Govindaraju, V. (2002). Transcript mapping for historic handwritten document images. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 413–418. IEEE Computer Society. Citado na pna 5.