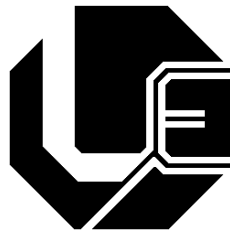


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**GROSC: UMA PROPOSTA DE
SEGMENTAÇÃO DE CARACTERES
IMPRESSOS ORIENTADA A REGIÕES
EM NÍVEIS DE CINZA**

JOSIMEIRE DO AMARAL TAVARES

**UBERLÂNDIA
2011**

JOSIMEIRE DO AMARAL TAVARES

**GROSC: UMA PROPOSTA DE
SEGMENTAÇÃO DE CARACTERES
IMPRESSOS ORIENTADA A REGIÕES
EM NÍVEIS DE CINZA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como requisito parcial para a obtenção do título de Mestre em Ciências.

Área de concentração: Processamento da Informação, Inteligência Artificial

Uberlândia, 28 de Outubro de 2011

Banca Examinadora

Prof. Dr. Keiji Yamanaka -Orientador- FEELT/UFU

Prof. Dr. Luiz Antônio Pereira Neves - UFPR

Prof. Dr. Shigueo Nomura - FEELT/UFU

Prof. Dr. Eustáquio São José de Faria - FACOM/UFU

Ao único Digno de Receber toda Honra, toda Glória,
e todo o Louvor.
Obrigada, Senhor Jesus!
Tudo o que tenho, tudo o que sou,
e tudo que vier a ser, vem de Ti.

Agradecimentos

Ao Deus Todo Poderoso pela graça, misericórdia, sabedoria, ajuda e principalmente pela Presença, sem a qual eu não sobreviveria. Sei que sem Ti, Senhor, eu não teria conseguido e nada poderia ter sido feito.

À minha mãe pelo cuidado, amor e compreensão. Sou grata a Deus por ter me presenteado com uma mãe tão maravilhosa, eu amo muito a senhora.

Aos meus irmãos e irmã pelo incentivo, palavras de força, carinho, investimentos, risos, atenção e muito amor.

Ao prof. Dr. Keiji Yamanaka pelas orientações riquíssimas, ajuda, incentivos, colaboração inestimável e principalmente pela disposição e humildade.

Aos amigos Igor Santos Peretta e Gerson Flávio de Lima e a amiga Mônica Sakuray Pais, a amizade de vocês foi fundamental para a realização deste trabalho. Agradeço, sobretudo, pelo carinho, cuidado e pelas inestimáveis ajudas e companheirismo nos trabalhos e na vida.

Ao Programa de Pós-Graduação da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia juntamente a Capes, em especial ao prof. Dr. Alexandre Cardoso, e Cinara Mattos, pela atenção, apoio financeiro e orientações durante a realização desta pesquisa.

Ao prof. Dr. José Roberto Camacho pela simpatia, conhecimentos compartilhados e disposição em ceder espaço para a realização desta pesquisa.

Aos colegas Hugo, Élvio, Fabrício, Juliana, Eduardo e Marlus, pelo apoio, companheirismo, dicas e experiências valiosas.

Ao casal Ricardo César Crosara Magnino e Andrea Lopez Gonçalves Magnino pelo amor de pais espirituais, palavras de incentivo e orações durante o mestrado. Sei que sem este apoio seria difícil prosseguir. E a todos os membros da Igreja Cristã Luz do Mundo, pelo amor, orações e carinho.

Aos meus amigos e amigas pela compreensão e amor demonstrados nos últimos anos.

“Filho meu, se o teu coração for sábio, alegrar-se-á também o meu; exultará o meu íntimo, quando os teus lábios falarem coisas retas. Não tenha o teu coração inveja dos pecadores; antes no temor do Senhor perseverarás todo dia. Porque deverás haver bom futuro; não será frustrada a tua esperança.”

(Bíblia Sagrada, Provébios de Salomão Capítulo 23:15-18)

Resumo

Os sistemas de Optical Character Recognition (OCR, ou “Reconhecimento ótico de caracteres” em português) designam genericamente tecnologias para reconhecer caracteres a partir de um arquivo de imagem que contenha textos. Permite, por exemplo, que uma folha de texto impresso, ao ser digitalizada, possa ser convertida em um arquivo de texto editável. Esses sistemas foram muito difundidos ao longo dos últimos anos, com várias versões comerciais, incluindo os incorporados em aparelhos scanner. Uma das etapas que definem o sucesso ou fracasso deste tipo de sistema é a correta segmentação das linhas de texto e dos caracteres que o constituem. Atualmente ganhou importância a correta segmentação das palavras contidas no texto, para auxiliar possíveis etapas de pós-processamento para a correção de desvios de reconhecimento. Neste trabalho são apresentados dois métodos diferenciados para segmentação de textos. O primeiro método, denominado “Segmentação de Linhas e Palavras baseado no Operador de Energia de Teager” (SLP-TEO) é utilizado na segmentação de linhas de texto e segmentação de palavras. O TEO é aplicado ao sinal abstraído da projeção linear (horizontal ou vertical) gerada através da imagem binária do texto. Um dos diferenciais do método é que o mesmo pode ser aplicado a textos impressos ou manuscritos, sem nenhum ajuste prévio. Além disso, o mesmo algoritmo é utilizado tanto na segmentação de linhas quanto na segmentação de palavras, impressas ou manuscritas. A base de dados adotada neste projeto (IAM-Database), consagrada em diversas pesquisas de OCR, possui, para todo texto impresso, transcrições manuscritas que também foram alvo do método SLP-TEO. O segundo método, denominado “gray-Region Oriented Segmentation of Characters-(gROSC)”, em português “Segmentação de Caracteres Orientado à Regiões em níveis de cinza”, é aplicado às imagens das palavras em tons de cinza com a finalidade de segmentação de caracteres. Este método baseia-se nos métodos de segmentação de regiões em que os pixels conectados são visitados e rotulados. Um dos diferenciais do método é que o mesmo deve ser aplicado às imagens em níveis de cinza das palavras segmentadas. Além disso, utilizando o limiar de Otsu e conhecendo-se

os níveis de cinza da imagem, o método determina a variação máxima entre tons de cinza que permite ou não visitar o pixel vizinho, agregando-o à região de interesse a ser segmentada. Na base de dados adotada existem três tipos básicos de caracteres problemáticos: caracteres sobrepostos, caracteres conectados e desenho de fontes. O método gROSC é aplicado às imagens de todas as palavras segmentadas anteriormente e os caracteres são segmentados igualmente sem prévia identificação dos possíveis problemas. Os experimentos e resultados encontrados em todas as etapas de segmentação dos textos com ambos os métodos são bastante relevantes e demonstram a eficiência e simplicidade dos métodos propostos.

Palavras-chave

Segmentação de Caracteres, Segmentação de Regiões, Segmentação de linhas e palavras, Operador de Energia de Teager.

Abstract

Optical Character Recognition systems (OCR) refer generically to technologies applied to recognize characters from an image file that contains text. It allows, for example, that a scanned sheet of printed text could be converted into an editable text file. These systems have been widespread over the past years, with several commercial versions, including the ones embedded within scanner devices. One of the steps that define the success or failure of this type of system is the correct segmentation of text lines and characters that constitute it. Recently, it has gained importance the correct segmentation of words in the text to assist possible post-processing steps to correct deviations of recognition. This paper presents two different methods for segmentation of texts. The first method, called “Segmentation of Lines and Words based Teager energy operator” (SLP-TEO), based on the Teager energy operator (TEO), is used in the segmentation of text lines and words. The TEO is applied to the signal abstracted from the linear projection (horizontal or vertical) generated by the binary image of the text. A unique feature of this method is that it can be applied to printed texts or manuscripts, without prior arrangements. Moreover, the same algorithm is used to segmentation of text lines and segmentation of words, no matter if they are printed or handwritten. The adopted database for this project (IAM-Database), widely used in OCR researches, has, for all printed text, handwritten transcripts that were also targeted using SLP-TEO method. The second method, called “gray-Region Oriented Segmentation of Characters” (gROSC) is applied to grayscale word images for the purpose of character segmentation. This method is based on region oriented methods where connected pixels that are visited and labeled. A unique feature of the method is that it should be applied to grayscale images of the segmented words. Moreover, using the Otsu threshold and knowing the gray levels of the image in advance, the method determines the maximum variation between shades of gray that allows or not to visit the neighboring pixel, adding it to the target region of interest. In the adopted database, there are three basic types of characters problems to solve: overlapping characters, connected characters, and font design. The gROSC method

is applied to all previously segmented words images, and the characters are equally segmented, without previous identification of possible character problems. The experiments and results in all stages of segmentation with both methods are very relevant and demonstrate the efficiency and simplicity of the proposed methods.

Keywords

Character Segmentation, Region Oriented Segmentation, Segmentation of lines and words, Teager energy operator (TEO).

Conteúdo

Conteúdo	x
Lista de Figuras	xii
Lista de Tabelas	xv
1 Introdução	1
1.1 Objetivos	4
1.2 Motivação	4
1.3 Contribuição	5
1.4 Organização da Dissertação	6
2 Fundamentação Teórica	8
2.1 Histórico	8
2.1.1 OCR	11
2.2 Conceitos Básicos sobre Imagens Digitais	13
2.2.1 Aquisição de Imagens Digitais	13
2.3 Imagens em Tons de Cinza	14
2.3.1 Representação da Imagem	14
2.3.2 Vizinhaça de um Pixel	16
2.3.3 Conectividade	17
2.3.4 Rotulação dos Componentes Conectados	17
2.3.5 Histograma	18
2.4 Segmentação de Imagens	19
2.4.1 Limiarização	20
2.4.2 Segmentação de Regiões	21
2.5 Imagens Binarizadas	22
2.5.1 Projeção Horizontal e Vertical	22

2.6	Operador de Energia Teager(TEO)	24
2.7	Trabalhos Relacionados	26
2.8	Conclusão	28
3	Solução Proposta	30
3.1	Preparação da Base de Dados	30
3.1.1	Distinção de Textos Impressos e Manuscritos	32
3.1.2	Caracteres Impressos da Base de Dados	32
3.2	Pré-processamento da Imagem	41
3.2.1	Binarização	42
3.3	Método SLP-TEO	44
3.3.1	Segmentação de Linhas	45
3.3.2	Segmentação de Palavras	49
3.4	Método gROSC	52
3.4.1	Análise da Imagem em Tons de Cinza	52
3.4.2	Método gROSC	53
3.5	Conclusão	61
4	Experimentos e Resultados	62
4.1	Textos Impressos	63
4.1.1	Segmentação de Linhas de Texto Impresso	64
4.1.2	Segmentação de Palavras Impressas	66
4.1.3	Segmentação de Caracteres Impressos	66
4.1.4	Casos Especiais	82
4.2	Estudo de Caso-Textos Manuscritos	92
4.2.1	Segmentação de Linhas de Texto Manuscritos	93
4.2.2	Segmentação de palavras em textos manuscritos	96
4.3	O Sistema Implementado	97
4.4	Conclusão	99
5	Conclusão	100
5.1	Trabalhos Futuros	102
	Referências	104

Lista de Figuras

2.1	Primeira imagem da Lua transmitida pelo ranger 7 [6]	10
2.2	Etapas do processamento digital de imagens em um sistema OCR	12
2.3	Mapa de pixel do caractere “a”	14
2.4	Convenção do sistema de coordenadas em imagem digital . . .	15
2.5	Vizinhança do pixel	16
2.6	Imagem de texto impresso da base de dados <i>IAM-DataBase</i> . .	19
2.7	Mapa de pixel de uma imagem binária gerada pela função <i>im2bw</i>	21
2.8	Projeção horizontal de um texto impresso da <i>IAM-Data Base</i>	23
2.9	Projeção vertical de um texto impresso da <i>IAM-DataBase</i> . .	24
2.10	Projeção horizontal e envelope do TEO extraídos do texto im- presso “f04-020”.	26
3.1	Formulário da <i>IAM-Database</i>	31
3.2	Imagens de textos da base de dados <i>IAM-DataBase</i>	34
3.3	Exemplo de fontes criadas por design de fontes [8].	36
3.4	Exemplo de palavra que apresentam caracteres denominados desenho de fontes.	37
3.5	Exemplo de caracteres que não podem ser segmentados como os demais.	38
3.6	Exemplo de palavras com caracteres sobrepostos.	39
3.7	Palavra “narrow” que apresenta dois caracteres conectados “a” e “r”.	39
3.8	Imagem de texto impresso da <i>IAM-DataBase</i>	44
3.9	Envelope do TEO sobre a projeção Horizontal	46
3.10	Representação de letras em matrizes.	46
3.11	Diagrama de bloco do método SLP-TEO para segmentação de linhas.	48
3.12	Diagrama de bloco do método SLP-TEO para segmentação de Palavras.	51

3.13	Imagem da palavra em tons de cinza e correspondente binarizada.	53
3.14	Pixels com mesmo tom de cinza: (1) pode ser considerado parte do objeto; (2) deve ser considerado parte do fundo. . . .	55
3.15	Imagem da letra “a” para exemplo.	58
4.1	Segmentação de linhas do texto impresso “f02-040”	65
4.2	Segmentação de linhas em texto impresso “a06-004”	65
4.3	Imagem da palavras segmentadas nas 8 linhas do texto impresso “f02-040”.	66
4.4	Segmentação da Palavra “from”.	69
4.5	Palavra que apresenta dois casos de caracteres sobrepostos. . .	70
4.6	Análise da vizinhança dos pixels da palavra “necessary”.	72
4.7	Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “necessary”.	73
4.8	Caracteres “a” e “r” segmentados pelo método gROSC.	74
4.9	Análise da vizinhança dos pixels da palavra “was”.	75
4.10	Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “was”.	76
4.11	Caracteres “a” e “s” da palavra “was” segmentados pelo método gROSC.	77
4.12	Análise da vizinhança dos pixels da palavra “afraid”.	78
4.13	Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “afraid”.	79
4.14	Análise da vizinhança dos pixels da palavra “remarkable”. . . .	80
4.15	Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “remarkable”.	81
4.16	Caracteres “k” e “a” da palavra “remarkable” segmentados pelo método gROSC.	82
4.17	Análise da vizinhança dos pixels da palavra “frequent”.	83
4.18	Caminho de decisão do método para o caractere “n”.	84
4.19	Resultados da segmentação incorreta do caractere “n” da palavra “frequent”.	85
4.20	Análise da vizinhança dos pixels da palavra “employment”. . .	86
4.21	Caminho de decisão do método para o caractere “m” da palavra “employment”.	87
4.22	Resultados da segmentação da palavra “employment”.	87
4.23	Análise da vizinhança dos pixels da palavra “remarkable”. . . .	88
4.24	Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “remarkable”.	89
4.25	Caracteres conectados “ar” da palavra “remarkable” segmentados pelo método gROSC.	90

4.26	Análise da vizinhança dos pixels da palavra “haris”.	90
4.27	Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “haris”.	91
4.28	Caracteres conectados “ar” da palavra “haris” segmentados pelo método gROSC.	92
4.29	Segmentação de linhas do texto manuscrito “r02-013”	95
4.30	Segmentação de linhas do texto manuscrito “p06-047”	95
4.31	Segmentação da 3ª linha do texto manuscrito “k07-063”	96
4.32	Segmentação da 1ª linha do texto manuscrito “linha1-j06-034”	97
4.33	Tela Principal do Sistema	97
4.34	Imagens geradas a partir do botão Imagem Linha.	98

Lista de Tabelas

3.1	Tabela dos 40 textos seleccionados da base de imagens <i>IAM-DataBase</i>	33
3.2	Caracteres que compõe a base de dados utilizada	35
3.3	Caracteres simples da base de dados	40
3.4	Níveis de Cinza únicos da imagem “a” (em negrito, os níveis permitidos para o início do gROSC; em itálico, os níveis abaixo do limiar de Otsu).	57
4.1	Textos Impressos Segmentados pelo Método SLP-TEO	63
4.2	Resultados de Segmentação de Caracteres pelo Método gROSC	68
4.3	Intervalos para transição entre pixels da palavra “necessary” .	71
4.4	Intervalos de níveis de cinza para transição entre pixels da palavra “was”	75
4.5	Intervalos para transição entre pixels da palavra “afraid” . . .	77
4.6	Intervalos para transição entre pixels da palavra “remarkable”	80
4.7	Intervalos para transição entre pixels da palavra “frequent” . .	84
4.8	Intervalos para transição entre pixels da palavra “employment”. 86	
4.9	Intervalos para transição entre os pixels dos caracteres “a” e “r” da palavra “remarkable”.	88
4.10	Intervalos para transição entre pixels da palavra “haris” na conexão dos caracteres “a” e “r”.	91
4.11	Textos manuscritos segmentados pelo método SLP-TEO . . .	94

Capítulo 1

Introdução

Uma das primeiras aplicações de técnicas de processamento de imagens data do início da década de 1920. Mas foi com o advento da computação que esta área teve seu maior avanço e recebeu o nome de processamento digital de imagem. As aplicações com imagens digitais dependem da capacidade e desempenho computacional, tanto para o seu armazenamento quanto para visualização e transmissão de dados. Segundo Gonzalez e Woods [6], “o processamento digital de imagens tem estreita relação com o desenvolvimento do computador digital”.

O processamento digital de imagem em conjunto com o desenvolvimento computacional tem beneficiado várias áreas, como por exemplo, a medicina, astronomia, sensoriamento remoto, áreas militares dentre outras. A automação de tarefas é uma aplicação de processamento digital de imagens que visa atribuir ao computador a capacidade necessária para que ele desempenhe papéis e tarefas que são executadas com facilidades pelos seres humanos. A demanda na automação de tarefas tem favorecido o desenvolvimento de sistemas que envolvem o reconhecimento de padrões como identi-

ficação de impressões digitais, identificação de faces, identificação de vasos sanguíneos, reconhecimento de placas de automóveis. O *Optical Character Recognition* - OCR é um sistema de automação de tarefas, o qual engloba vários tipos de aplicações como identificação de assinaturas, reconhecimento de textos manuscritos e impressos, reconhecimento de placas de veículos, reconhecimento de textos em Braille [3], dentre outros.

Os sistemas de OCR envolve algumas etapas como aquisição de imagens digitais, pré-processamento da imagem, segmentação, extração de características e reconhecimento. Dependendo da imagem, há a necessidade de um pré-processamento mais aprimorado, para que as outras etapas possam ser otimizadas e obtenham resultados mais satisfatórios, dentre as quais podemos citar normalização da imagem, utilização de filtros para realce da imagem, remoção de ruídos dentre outros. Em linhas gerais, a etapa de segmentação consiste em separar do fundo da imagem os objetos de interesse que constituem a mesma[6]. Em imagens de documento textuais, normalmente, é realizada a segmentação de linhas de texto, em seguida, a segmentação de palavras e por fim a segmentação de caracteres. Neste contexto, os produtos finais são os caracteres segmentados.

Nesta pesquisa realizamos a segmentação de linhas de texto e palavras em textos impressos e manuscritos utilizando um único método. O método Segmentação de Linhas e Palavras baseado no Operador de Energia de Teager (SLP-TEO). Este método se mostrou muito eficiente quando aplicado à textos impressos, por isso, resolvemos aplicá-lo também à textos manuscritos em um estudo de caso.

O processo de segmentação de linhas de texto e palavras em textos impressos, é mais simples, pois este tipo de documento apresenta um padrão de espaçamento entre linhas, entre palavras e entre caracteres. Alguns métodos se tornaram padrões para esse tipo de texto, como os métodos baseados em

projeção horizontal e vertical. Além disso, alguns trabalhos utilizam uma adaptação dos métodos baseados em projeção com a finalidade de segmentação de linhas de textos manuscritos como no trabalho [5]. Nos últimos anos o foco principal para segmentação de linhas e palavras tem sido voltado a solução deste problema em textos manuscritos, visto que ainda é um problema crítico[15], [16]. Alguns dos problemas relacionados aos textos manuscritos são inclinações de linhas, diferenças entre ângulos de linhas, vários tipos e tamanhos de letras, além de linhas sobrepostas ou fragmentadas [16].

Quanto a segmentação de caracteres, a utilização das técnicas de projeção também é tradicional. Entretanto, somente a projeção vertical é insuficiente para a solução de todos os problemas que surgem nesta etapa, por exemplo, os caracteres sobrepostos e os caracteres conectados. Estes dois tipos de caracteres não podem ser resolvidos somente com a projeção horizontal e vertical. Para solução destes problemas em caracteres sobrepostos e conectados, normalmente os autores utilizam projeção vertical e morfologia matemática [22], [25], [5].

A etapa de segmentação de caracteres, normalmente, é realizada utilizando a imagem já binarizada. No entanto, observando as imagens em tons de cinza percebemos que este tipo de imagem apresenta características importantes que devem ser consideradas para a segmentação de caracteres. Por exemplo, quando visualizamos uma imagem de palavra em tons de cinza é fácil verificar as distinções entre os caracteres constituintes daquela palavra. Mas a mesma imagem binarizada pode mostrar os caracteres como sendo um único caractere, quando na verdade se trata de dois objetos diferentes. Pensando nisso, propomos um método de Segmentação de Caracteres Orientados à Regiões em imagens em tons de cinza(gROSC) que é aplicado à imagens de palavras. Neste método é realizada uma análise local de cada pixel através da vizinhança entre pixels, e em seguida é extraída a região correspondente aos

caracteres. Desta forma, resolvemos com um único método tanto a segmentação dos caracteres simples, como os problemas de caracteres conectados e sobrepostos.

1.1 Objetivos

O objetivo deste trabalho é apresentar um sistema de segmentação de textos impressos, em que abordaremos 3 etapas:

- segmentação de linhas de textos;
- segmentação de palavras;
- segmentação de caracteres

Foram implementados dois métodos, sendo o primeiro método para segmentação de linhas de texto e palavras e o segundo método para segmentação de caracteres impressos em imagens em tons de cinza.

O método gROSC utiliza as características de níveis de cinza individuais de cada imagem da palavra a ser segmentada. Consequentemente, através de uma análise da vizinhança dos pixels, é possível mensurar a variação máxima possível para que os seus vizinhos sejam visitados pelo método. A partir deste método é possível solucionar o problema de tipos comuns de caracteres que normalmente estão presentes em sistemas de segmentação, como por exemplo, a presença de caracteres sobrepostos, conectados.

1.2 Motivação

Os trabalhos de segmentação de textos é um tema estudado a vários anos e que tem bastante aplicações. No entanto, nos últimos anos, as pesquisas

envolvendo segmentação de textos impressos estão mais escassas, visto que o alvo das pesquisas tem sido concernente a textos manuscritos e outros tipos de aplicações. Entretanto, não significa que este problema esteja totalmente resolvido, e principalmente que não possa ser melhorado.

Através da revisão bibliográfica podemos perceber que quando se trata de segmentação de caracteres, por exemplo, a grande maioria dos trabalhos abordam a segmentação de caracteres envolvendo imagens já binarizadas. Por isso, a motivação maior para o desenvolvimento deste trabalho é apresentar um método diferenciado para segmentação de caracteres utilizando imagens em níveis de cinza, utilizando as características individuais de cada pixel. Uma vantagem deste tipo de abordagem é que o método proposto neste trabalho independe dos idiomas dos textos que estão sendo segmentados.

1.3 Contribuição

Para este trabalho foi considerada uma base de texto com 40 textos impressos e manuscritos selecionados aleatoriamente da base de imagens *IAM-DataBase*[19].

Neste trabalho apresentamos dois métodos diferenciados, sendo o primeiro para segmentação de linhas de texto e palavras. E o segundo método é aplicado para a segmentação de caracteres impressos em imagens em tons de cinza.

O método de Segmentação de Linhas e Palavras baseado no Operador de Energia de Teager (SLP-TEO) foi aplicado às imagens de textos impressos para a segmentação de linhas e palavras. Este método se mostrou bastante eficiente para os textos impressos. Por isso, apesar dos textos manuscritos não serem o foco deste trabalho, resolvemos aplicar o método SLP-TEO a

este tipo de texto para a segmentação de linhas e palavras. Ao aplicarmos o método SLP-TEO aos textos manuscritos, em um estudo de caso, observamos que independente das características, totalmente distintas entre os dois tipos de textos, o método se mostra bastante eficiente para ambos os textos.

O método Segmentação de Caracteres Orientados a Regiões em imagens em tons de cinza(gROSC) contribui para a comunidade científica, visto que este método utiliza uma abordagem diferenciada para segmentação de caracteres. Uma das vantagens está no fato de que as características da imagem original são mantidas. Outro ponto a ser considerado é que um mesmo método segmenta os caracteres simples e, ao mesmo tempo soluciona os problemas de caracteres conectados e sobrepostos. Além disso, não há necessidade de se identificar os caracteres dentro das palavras para depois serem solucionados os problemas de conexão e sobreposição. Isso é feito de forma direta, eliminando assim uma etapa que seria necessária, mas não primordial. Visto que, dentro de um texto há vários tipos de caracteres e identificá-los demandaria tempo e custo computacional, além de ser uma tarefa mais complexa.

1.4 Organização da Dissertação

Esta dissertação está organizada em cinco capítulos, em que são abordados a segmentação de textos impressos nos níveis de segmentação de linhas, palavras e caracteres. Bem como, alguns dos experimentos realizados com seus respectivos resultados e um estudo de caso da aplicação do método SLP-TEO em textos manuscritos para segmentação de linhas e palavras. Neste capítulo foram apresentados os tópicos sobre a introdução do trabalho, os objetivos, a motivação, contribuições e a organização da dissertação.

No Capítulo 2 são descritos os fundamentos de imagens digitais, os conceitos básicos de processamento digital de imagens, bem como a história do conceito de OCR (*Optical Character Recognition*), representação das imagens digitais e relacionamento entre pixels e alguns tipos de segmentação.

O capítulo 3 são apresentados os dois métodos implementados nesta pesquisa, sendo o método de Segmentação de Linhas e Palavras baseado no Operador de Energia de Teager (SLP-TEO) e o método de Segmentação de Caracteres Orientado a Regiões em imagens em tons de cinza (gROSC).

No capítulo 4 mostramos os experimentos realizados, bem como, os resultados encontrados em cada etapa da segmentação dos textos impressos utilizados. Além disso, um detalhamento da aplicação do método gROSC nos casos especiais de caracteres conectados e sobrepostos na etapa de segmentação dos mesmos.

Finalmente, no capítulo 5, apresentamos as conclusões sobre as etapas desenvolvidas nesta pesquisa e os trabalhos futuros oriundos deste trabalho.

Capítulo 2

Fundamentação Teórica

Neste capítulo abordamos os fundamentos básicos aplicados em processamento digital de imagens.

2.1 Histórico

Uma das primeiras aplicações de técnicas de processamento de imagens data do início da década de 1920, em que as imagens de um jornal foram digitalizadas e enviadas por meio de um cabo submarino de Londres para Nova York. Para isso, foi implementado um sistema de transmissão de imagens denominado de *Bartlane*, esse sistema proporcionou uma redução de tempo do transporte de imagens pelo oceano Atlântico, que era equivalente a mais de uma semana para menos de três horas [6], [26]. Além disso, o sistema *Bartlane* foi capaz de codificar imagens em 5 níveis de cinza distintos sendo que essa capacidade aumentou para 15 níveis em 1929. Nessa época, também houve a introdução de sistemas para revelação fotográfica através de feixes de luz. Note que estes avanços na área de processamento de imagens não

tiveram o auxílio do computador digital, uma vez que este ainda não havia sido criado[6].

Por volta da década de 1940, originou-se o termo computador digital moderno criado por John Von Neumann. John Von Neumann definiu dois conceitos importantes para a computação: a memória para armazenamento de programas e dados e a ramificação condicional. Sendo que estes constituem os fundamentos da unidade central de processamento (CPU) até os dias atuais [6].

As aplicações de técnicas de processamento digital de imagens tiveram seu advento com a evolução da computação, e surgimento dos computadores de grande porte. Segundo Gonzales e Woods [6], “O termo processamento digital de imagens se refere ao processamento de imagens digitais por um computador digital”.

O uso de computadores digitais no processamento de imagens se iniciou com o *Jet Propulsion Laboratory* em Pasadena Califórnia em 1964, quando imagens da Lua foram transmitidas pela espaçonave americana *Range 7*. Estas imagens foram processadas por um computador digital para correção de distorções. A Figura 2.1 mostra a primeira imagem da Lua transmitida pela espaçonave *Range 7* em 31 de julho de 1964, esta imagem foi cedida pela Nasa aos autores do livro [6].

No início da década de 1970, o surgimento da tomografia axial computadorizada marcou um importante evento na aplicação de processamento digital de imagens para o diagnóstico médico [6], [26]. Desde então, a área médica tem se beneficiado com os avanços do processamento digital de imagens e evolução da computação para diversas aplicações como diagnóstico de doenças graves, avanço em pesquisas celulares, simulação de próteses, etc.

Assim como a medicina, várias áreas tem se beneficiado com as técnicas de processamento de imagens como biologia, geografia, astronomia, arqueo-

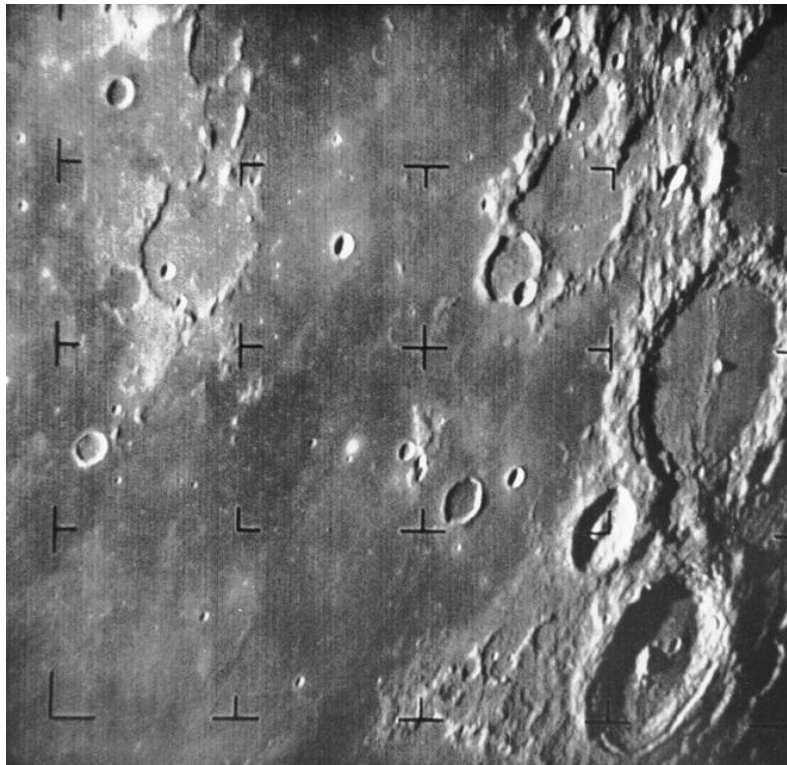


Figura 2.1: Primeira imagem da Lua transmitida pelo ranger 7 [6]

logia, física, dentre outras. Uma área que tem ganhado espaço no processamento digital de imagens é a resolução de problemas relativos a percepção por máquinas. Nesta, o interesse concentra-se em extrair informações relevantes da imagem através do processamento computacional.

O reconhecimento automático de caracteres, os sistemas industriais, os processamentos de assinaturas digitais, rastreamentos de imagens radiográficas, análises de amostras de sangue, processamento de imagens de satélites e imagens aéreas estão dentre as diversas aplicações típicas de técnicas desenvolvidas por percepção por máquinas [6].

Segundo Pedrini e Schwartz [26] “O processamento digital de imagens consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio do computador”. Essas técnicas nos permitem

extrair e identificar informações das imagens analisadas, bem como melhorar a qualidade da imagem, facilitando a interpretação automática através de máquinas.

Neste trabalho, trataremos de uma parte que consideramos de suma importância: a segmentação de imagens digitais de textos impressos.

2.1.1 OCR

O termo OCR veio a ser conhecido em 1929 quando Gustav Tauschek obteve a primeira patente de OCR na Alemanha. Depois disso, em 1933 foi concedido a Paul W. Handel uma patente americana do OCR nos Estados Unidos da América. Estes são os primeiros relatos de que se tem conhecimento quanto a origem do termo OCR [20]. O método criado por Tauschek ficou conhecido como máquina de calcular. Este método foi muito inovador, visto que o computador digital ainda não havia sido criado. No decorrer dos anos, novos métodos foram descobertos e a realidade do OCR passou a ser mais palpável. Em 1951 foi lançado o primeiro OCR comercial. Até que em 1974, Ray Kurzweil criou uma empresa de computadores e desenvolveu o primeiro sistema de reconhecimento óptico de caracteres (*Optical Character Recognition*(OCR)) utilizando computadores digitais[20]. Desde então, o termo OCR é entendido como um sistema de reconhecimento óptico de caracteres.

Segundo Laine e Nevalainen[12] um sistema de OCR envolve várias de etapas de processamento de imagens e algoritmos complicados, como pode ser visto na Figura 2.2.

A aquisição da imagem é realizada através de um digitalizador como o scanner, uma câmera fotográfica, ou webcam, por exemplo. Em seguida, é realizado um pré-processamento dessa imagem como normalização da ima-



Figura 2.2: Etapas do processamento digital de imagens em um sistema OCR

gem, calibragem, realce de cores, remoção de ruídos, suavização, adequação de resoluções quando se fizer necessário. Na etapa de pré-processamento, é realizada a conversão de cores da imagem para níveis de cinza. A próxima etapa, trata-se da binarização da imagem onde são segmentados os objetos de interesse da mesma, separando-os do fundo. Outros métodos normalmente são utilizados como filtragem da imagem para melhorar o realce e diminuir os ruídos oriundos da etapa de digitalização. Na etapa de segmentação propriamente dita, os elementos constituintes como linhas de texto, palavras e caracteres são separados do restante da imagem. Depois destas duas etapas, é feita a extração de características. No caso de um texto, espera-se que os caracteres já estejam separados completamente, e então é feito um estudo das características como medidas de reconhecimento de padrões. Estas características são utilizadas como entradas na etapa de reconhecimento, este pode ser feito utilizando técnicas de Data Mining, algoritmos de aprendizagem como redes neurais, e outros métodos. É importante ressaltar que, em muitos casos, há a necessidade de uma etapa de pós-processamento em que são realizados ajustes para uma taxa maior de reconhecimento.

Dentre as etapas de processamento de um sistema de reconhecimento óptico de caracteres, podemos dizer que a segmentação é uma das que requer mais cuidados, visto que a segmentação pode definir o sucesso ou não do reconhecimento [6].

2.2 Conceitos Básicos sobre Imagens Digitais

Nesta seção abordamos os conceitos básicos sobre imagens digitais, a representação da imagem bem como relacionamentos entre pixels.

2.2.1 Aquisição de Imagens Digitais

A aquisição de imagens é feita através de dispositivos de captura. Estes dispositivos podem ser câmeras digitais, sensores, digitalizadores como *scanners*, etc.

Os digitalizadores são dispositivos utilizados para converter a saída do dispositivo físico de sensoriamento em formato digital [6]. Existem várias modalidades de sensores para aquisição de imagens. Há aqueles que utilizam um único sensor, outros utilizam varredura matricial, e os sensores que utilizam varreduras de linhas. Os scanners de mesa utilizam esta última modalidade de sensores para aquisição de imagem.

Segundo Gonzalez e Woods [6] “a saída dos sensores consiste de uma forma de onda de tensão contínua cuja amplitude e o comportamento no espaço estão relacionados ao fenômeno físico captado pelos sensores”. Portanto, para se criar uma imagem digital, é necessário converter os dados contínuos da mesma para formato digital através dos processos de amostragem e quantização. Basicamente, uma imagem é contínua em relação às coordenadas x e y e em relação à sua amplitude. A amostragem consiste na digitalização dos valores de coordenada e a quantização é a digitalização dos valores de amplitude da imagem [6].

De acordo com Gonzalez e Woods [6], “a qualidade da imagem digital é claramente determinada, em grande parte, pelo número de amostras e de níveis discretos de intensidade utilizados na amostragem e na quantização”.

2.3 Imagens em Tons de Cinza

Denomina-se escala ou intensidade de cinza o intervalo $[Lmin, Lmax]$, onde $Lmin$ é o limite inferior positivo da imagem e $Lmax$ é o limite superior. Considerando o intervalo $[0, L - 1]$, $l = 0$ é considerado preto e $l = L - 1$ é branco. Assim, todos os valores intermediários são tons de cinza que variam do preto ao branco[6].

A Figura 2.3 mostra a imagem do caractere “a” extraído de uma palavra da base de dados utilizada neste trabalho.

230	234	227	217	215	202	185	187	183	173	175	172	167	174	196	217	224	231	238	236	231	236	233	239
234	235	231	219	186	105	56	67	70	72	70	66	55	54	95	159	192	217	224	228	231	234	231	238
233	234	223	179	91	36	29	64	135	165	165	156	105	36	31	45	94	154	180	220	221	228	231	234
229	227	211	110	34	25	23	38	123	207	213	217	194	107	38	25	31	45	104	205	220	229	231	234
229	223	205	94	28	24	23	32	111	210	222	223	218	195	112	27	22	26	73	169	220	231	228	228
233	226	217	138	50	28	30	56	146	213	223	219	215	203	122	28	22	23	33	107	213	225	216	223
231	232	227	207	141	92	97	143	194	211	214	207	209	200	124	28	21	21	33	118	212	221	221	220
230	234	233	221	210	198	194	189	198	197	168	139	143	140	97	27	19	20	31	117	208	219	225	222
227	233	230	226	217	202	169	128	116	130	86	42	61	84	72	26	20	21	29	117	203	213	221	222
236	234	222	214	178	137	72	35	36	62	88	89	138	185	149	35	21	20	28	104	204	222	222	223
231	229	211	166	80	35	29	48	79	141	196	203	206	208	159	48	23	22	28	110	206	218	221	218
225	216	175	85	31	23	30	96	180	203	214	221	221	210	167	53	22	21	28	113	208	222	222	224
221	175	88	29	22	24	58	152	208	221	226	224	221	209	147	42	22	20	28	109	210	225	227	226
211	115	34	23	21	29	110	201	219	225	225	220	219	209	129	32	21	20	29	108	207	216	224	227
212	118	29	22	21	31	116	205	221	225	226	221	219	208	131	33	22	21	28	100	202	222	223	227
219	152	54	24	22	29	123	201	218	222	223	229	216	181	125	50	24	23	29	108	197	215	222	227
229	209	141	50	26	27	73	127	177	212	212	213	178	123	126	102	28	23	26	73	131	135	173	217
220	221	208	140	81	47	31	35	73	127	138	138	108	125	189	151	56	29	28	32	98	40	101	216
228	233	228	213	202	137	78	72	76	73	78	81	129	196	214	200	144	83	79	78	83	89	135	214
231	240	235	227	225	213	199	196	194	185	190	198	211	223	229	228	218	202	196	192	200	207	214	230

Figura 2.3: Mapa de pixel do caractere “a”

2.3.1 Representação da Imagem

A digitalização da imagem através de um modelo matemático é primordial para a representação e manipulação de imagens em computador, uma vez que, a representação dessa imagem é o primeiro passo para a implementação de

algoritmos de processamento digital de imagens.

Segundo Gonzales e Woods [6], o termo imagem refere-se a uma função de intensidade luminosa bidimensional, denotada por $f(x, y)$ em que o valor ou amplitude de f nas coordenadas espaciais fornece a intensidade (brilho) ou nível de cinza da imagem naquele ponto.

A Figura 2.4, mostra uma imagem em tons de cinza denotada pela função $f(x, y)$, em que a origem $f(0, 0)$ encontra-se na parte superior esquerda da imagem. O posicionamento da origem da imagem é uma representação convencional baseada na visualização de imagens por dispositivos como monitores de TV que varrem uma imagem começando do canto superior [6]. Note que para imagens coloridas RGB, são necessários uma função para cada banda de cor.

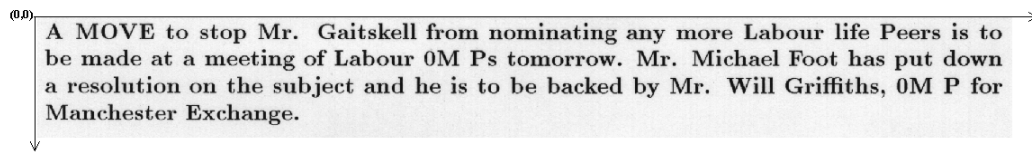


Figura 2.4: Convenção do sistema de coordenadas em imagem digital

A Equação 2.1 mostra uma imagem digital representada por uma matriz numérica, onde M e N são as dimensões dessa imagem e cada elemento dessa matriz denomina-se pixel ou elemento de imagem (abreviação de *Picture element*) [6].

$$f(x, y) \approx \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \cdots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \cdots & f(M-1, N-1) \end{bmatrix} \quad (2.1)$$

2.3.2 Vizinhança de um Pixel

Para efetuar uma manipulação em imagens digitais, é primordial que o conceito de pixel fique claro. Portanto, $\text{pixel}(p)$ é um elemento f em uma matriz bidimensional denotada por $f(x, y)$. Assim, um pixel cuja coordenada (x, y) possui quatro vizinhos verticais e horizontais, cujas coordenadas são $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ e $(x, y - 1)$. Este tipo de vizinhança é denominada vizinhança-4 de p , definida por $N_4(p)$ [6].

A vizinhança-8 do pixel (p) , denotada por $N_8(p)$ corresponde aos pixels da vizinhança-de-4 mais os quatro vizinhos diagonais ou adjacentes do pixel (p) , chamados de vizinhança-d, cuja coordenada são $(x - 1, y - 1)$, $(x - 1, y + 1)$, $(x + 1, y - 1)$ e $(x + 1, y + 1)$.

Portanto, a vizinhança-8 de p , é denotada pela união das vizinhança-4 e vizinhança-d, denotada pela equação 2.2:

$$N_8(p) = N_4(p) \cup N_d(p) \quad (2.2)$$

A figura 2.5 apresenta os três tipos de vizinhança citados acima.

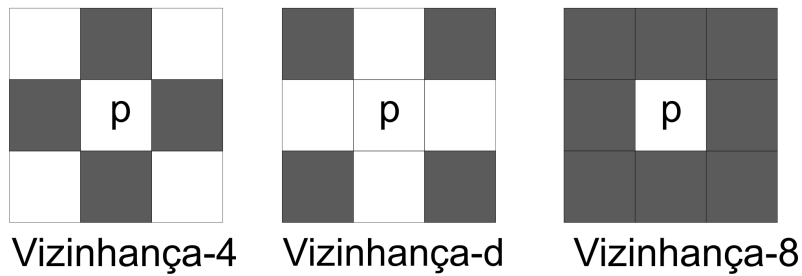


Figura 2.5: Vizinhança do pixel

O método desenvolvido para segmentação de caracteres utiliza as relações de vizinhança e conectividade que são os critérios básicos para a formação de regiões no processo de segmentação [25].

2.3.3 Conectividade

A conectividade está diretamente condicionada pela relação de vizinhança entre pixels, uma vez que dois pixels são conexos se houver uma sequência de pixel que os liga [25].

O conceito de conectividade entre pixels é importante para o estabelecimento das bordas de objetos e para determinar os componentes de regiões em imagens. Para determinar se dois pixels estão conectados é necessário que eles estejam de alguma forma adjacentes e se seus níveis de cinza satisfaçam o critério de similaridade. Assim como na vizinhança, os pixels conectados podem satisfazer a dois tipos de conexão conectados-de-4 ou conectados-de-8, obedecendo a regra de vizinhança.

Em se tratando de imagens binárias, em que os valores de pixels são 0 ou 1, dois pixels podem ser vizinhos, mas necessariamente devem possuir o mesmo valor para serem considerados conectados. No entanto, para imagens em níveis de cinza é necessário definir regras de similaridade entre os pixels para se definir a possibilidade de conectividade entre os vizinhos do pixel.

Neste trabalho utilizaremos conceitos de conectividade entre pixels considerando a similaridade dos pixels de uma imagem em níveis de cinza.

2.3.4 Rotulação dos Componentes Conectados

Na rotulação de componentes conectados, a imagem é rastreada pixel a pixel e os componentes de interesse que podem ser os componentes conectados-de-4 ou conectados-de-8 são verificados. Ao serem visitados os pixels que satisfizerem a relação de vizinhança e conectividade, recebem um rótulo que os distingue dos demais pixels que serão rastreados. No fim do processo, estes rótulos são ordenados e separados em classes de equivalência[6].

Em uma imagem binária isso é feito de forma simplificada, uma vez que

os pixels possuem valores de 0(preto) ou 1(branco). Em se tratando de imagens em tons de cinza, este processo fica um pouco mais difícil, pois a conectividade é estabelecida a partir de similaridade dos pixels.

2.3.5 Histograma

O histograma de uma imagem contém valores de intensidade de níveis de cinza, portanto ele pode ser visto como uma distribuição discreta de probabilidade dos níveis de cinza da imagem [26]. Uma vez que o número de pixels para um nível de cinza qualquer pode ser utilizado no cálculo da probabilidade de se encontrar um pixel com o mesmo valor de cinza na imagem.

Uma imagem digital $f(x, y)$ representada por uma matriz bidimensional $N \times M$ pixels composta por L níveis de cinza possui o seu histograma expresso como mostrado na equação 2.3

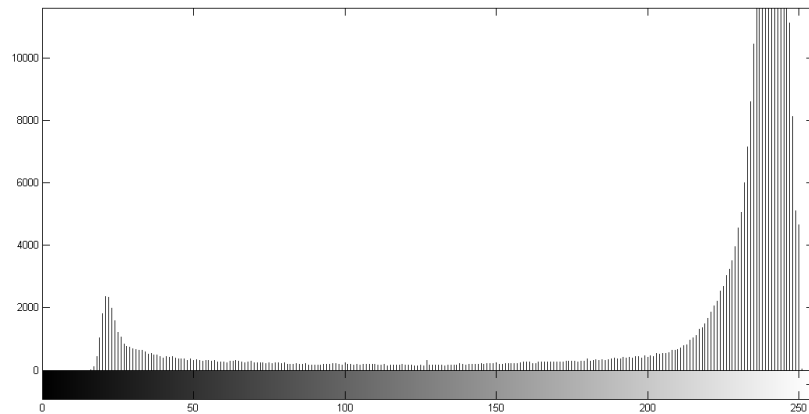
$$p_k(f) = \frac{n_k}{n} = \frac{H(k)}{NM} \quad (2.3)$$

onde $n_k = H(k)$ representa o número de ocorrências do nível de cinza k e $n = NM$ que corresponde ao total de pixels na imagem.

A Figura 2.6 apresenta o histograma da imagem em tons de cinza de um dos textos da base de dados utilizada. Neste histograma podemos observar uma distribuição maior de pixels com valores próximos 0 e 255 de níveis de cinza, caracterizando assim um histograma bimodal que normalmente é típico de imagens de documentos.

A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour 0M Ps tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, 0M P for Manchester Exchange.

(a) Imagem de texto impresso em tons de cinza.



(b) Histograma do texto em tons de cinza.

Figura 2.6: Imagem de texto impresso da base de dados *IAM-DataBase*.

2.4 Segmentação de Imagens

O processo de segmentação visa subdividir uma imagem em suas partes ou objetos constituintes, sendo que essa subdivisão dependerá do contexto da aplicação. Em sistemas de reconhecimento de texto, os objetos de interesse podem ser linhas, palavras e/ou caracteres.

Um dos requisitos fundamentais para um processo de segmentação é identificar de maneira correta a localização, a topologia e forma dos objetos que formam a imagem.

Em um primeiro estágio, as abordagens convencionais para a segmentação de imagens são baseadas nas propriedades básicas dos níveis de cinza da imagem. Esta abordagem visa detectar descontinuidade ou similaridade da imagem.

Portanto, em se tratando de imagens de texto podemos definir dois está-

gios que serão abordados, primeiro a limiarização e, em seguida, a segmentação do texto em linhas, palavras e letras/caracteres.

2.4.1 Limiarização

A limiarização é um tipo de segmentação, em que o fundo de uma imagem é separado do seu conteúdo ou objetos de interesse. Esse tipo de segmentação é normalmente feito a partir do histograma de uma imagem em tons de cinza, mas pode ser realizado com outros formatos como nos trabalhos [13], [33].

A limiarização é realizada estimando-se os intervalos de níveis de cinza que determinem os pixels correspondentes ao fundo e objetos de interesse na imagem. Assim, em uma imagem digital denotada por $f(x, y)$, é definido um limiar T , em que será definido dois intervalos: $f(x, y) > T$ e $f(x, y) \leq T$.

Esse tipo de limiarização é denominada limiarização mono-nível [18], ou global simples e o resultado é uma imagem binária. De acordo com [6], uma imagem limiarizada ou binarizada $g(x, y)$ é definida pela Equação 2.4.

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (2.4)$$

A limiarização global utiliza o valor de um único limiar (T), considerando o histograma da imagem como bimodal.

Segundo Pedrini e Schwartz [26], a forma mais direta de um valor global de limiar é através das intensidades dos pixels na imagem. Vários métodos são usados para a melhor definição do limiar. Sendo que o método de Otsu [23] é um dos mais usado na binarização de imagens de texto, tanto para textos impressos como para textos manuscritos.

O método de Otsu baseia-se no fato de que o histograma da imagem é composto por duas classes, sendo elas: os pixels do objeto e os pixels do

fundo[26].

Para o cálculo do limiar T , considera-se que cada classe ou pico da distribuição de probabilidade apresente características distintas. Tendo observado essa distinção são calculadas as medidas de tendência central do limite para as duas classes.

A segmentação é realizada visitando a imagem original pixel-a-pixel, e rotulando-os com valores 1 quando este pixel apresentar valor maior que o limiar T definido, e caso contrário o rótulo será 0.

Para a etapa de binarização da imagem foi utilizada a função *im2bw* do MATLAB®) utilizando-se o limiar *graythresh*. O resultado dessa binarização gera uma imagem como pode ser vista na Figura 2.7, onde os pixels com intensidade 0 correspondem aos objetos e os pixels com intensidade 1 correspondem ao fundo da imagem.

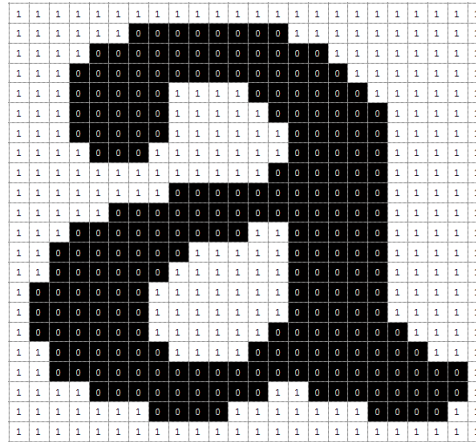


Figura 2.7: Mapa de pixel de uma imagem binária gerada pela função *im2bw*

2.4.2 Segmentação de Regiões

O fundamento básico para a segmentação de regiões é o agrupamento de pontos com as propriedades similares formando-se uma região. Dentre essas

propriedades podemos destacar a intensidade de cinza, cor, e informações semânticas ou textura.

O processo de segmentação parte do pressuposto que partes da imagem “parecidas” devem ser pertencentes ao mesmo “objeto” na imagem. Essas partes são unidas constituindo assim regiões de interesse.

Tradicionalmente, a região de uma imagem é definida por algumas propriedades [6] como se segue:

Seja R a região completa de uma imagem. A imagem submetida a segmentação em n regiões $R_1, R_2, R_3, \dots, R_n$, tal que:

1. $\bigcup_{i=1}^n R_i = R$, ou seja a segmentação deve ser completa, onde cada pixel pertence a uma região.
2. R_i é uma região conexa, $i=1,2,3,\dots,n$; os pixels da região em questão devem ser conectados.
3. $R_i \cap R_j = 0$, para todo i e j , $i \neq j$, regiões disjuntas.
4. $P(R_i) = VERDADEIRO$ para $i = 1, 2, \dots, n$, Se todos os pixels em R_i possuírem a mesma intensidade.
5. $P(R_i \cup R_j) = falso$ para $i \neq j$, mostra que as regiões R_i e R_j são diferentes no sentido do predicado P .

2.5 Imagens Binarizadas

2.5.1 Projeção Horizontal e Vertical

A projeção Horizontal de uma imagem $f(x, y)$ é definida como a soma dos pixels com valores 1, existentes em cada linha do objeto, como mostrada na equação 2.5 [26].

$$P_h(y) = \sum_{x=0}^{N-1} f(x, y) \quad (2.5)$$

A projeção linear (Horizontal e Vertical) é muito utilizada em segmentação de linhas e palavras, uma vez que através do histograma gerado é possível identificar “vales” e “picos”. Estes picos são concentrações de pixels e são facilmente identificadores de objetos dentro de uma imagem.

A Figura 2.8 mostra a projeção horizontal de um texto impresso.

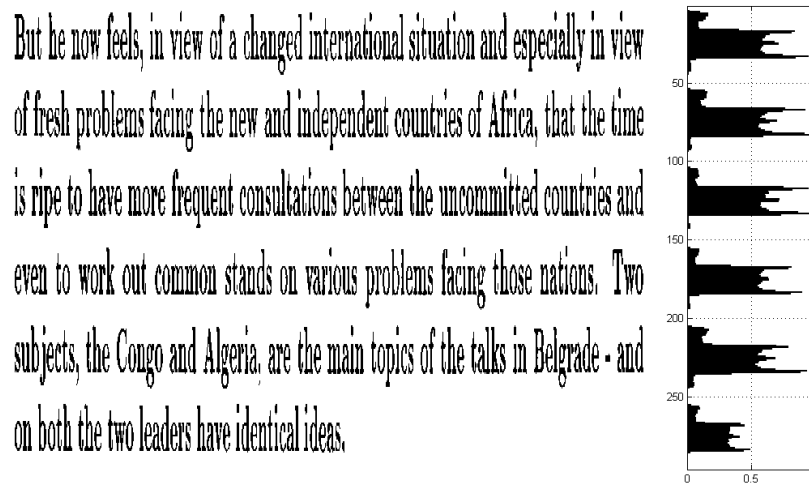


Figura 2.8: Projeção horizontal de um texto impresso da *IAM-Data Base*

Em se tratando de imagens de textos, quando se utiliza a projeção horizontal, podemos identificar que os picos apresentados pelo histograma são as linhas de textos e os vales são os espaços entre linhas. A projeção é um método simples e eficiente quando se trata de textos comportados como os textos impressos, por exemplo.

Analogamente, a equação 2.6 mostra a projeção vertical P_v que é definida como a soma dos pixels em cada coluna do objeto.

$$P_v(x) = \sum_{y=0}^{M-1} f(x, y) \quad (2.6)$$

No entanto, a segmentação de palavras é um pouco mais complexa, visto que os espaçamentos são diferenciados, entre palavra e entre caracteres de uma mesma palavra. Assim, normalmente se usa o histograma vertical juntamente com alguma medida de distância para segmentar as palavras em um texto.

A Figura 2.9 mostra a projeção vertical de uma linha de texto.

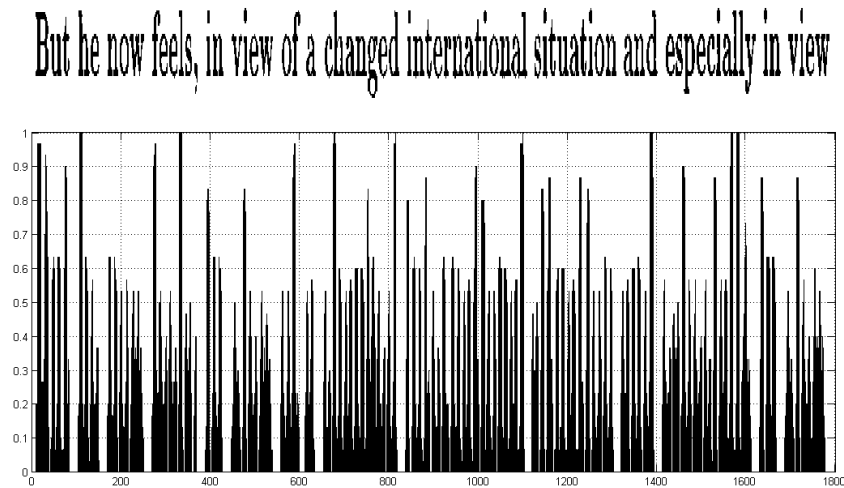


Figura 2.9: Projeção vertical de um texto impresso da *IAM-DataBase*

2.6 Operador de Energia Teager(TEO)

Teager e Teager, em um trabalho sobre modelamento não-linear da voz [32], apresentam um operador de energia que mais tarde foi denominado como operador de energia de Teager (*Teager Energy Operator* - TEO), ou operador Teager-Kaiser. Este operador, de acordo com Kaiser [10], ao ser aplicado a

um sinal composto por uma única frequência variante no tempo, é capaz de extrair a medida de energia do processo mecânico que gerou esse sinal.

O operador de energia de Teager é definido no domínio discreto [11] como:

$$\Psi [x (n)] = x_n^2 - x_{n+1} \cdot x_{n-1}, \quad (2.7)$$

onde Ψ é o operador de Teager e x_n é o valor da n -ésima amostra do sinal.

Uma importante característica do TEO analisada por Kaiser [10] é que, quando aplicado a sinais compostos por duas ou mais componentes de frequência, “seria como se o algoritmo [TEO] fosse capaz de extrair a função envelope do sinal” [tradução nossa]. Como exemplo, a dissertação de Peretta [27] utiliza esta característica do TEO, a de extrair a função envelope de um sinal, para detectar as fronteiras de um comando de voz inserido em um sinal de áudio. Outro trabalho deste grupo utiliza essa mesma abordagem [28].

Ao considerarmos os valores de projeção (histograma) de uma imagem de texto binarizada como sendo um sinal amostrado, pela teoria de séries de Fourier [4], este sinal (função arbitrária dentro de dado intervalo) pode ser expresso por uma série de senos e cossenos. Ou seja, teremos como resultado dessa abstração um sinal composto por diversas componentes de frequência. Se aplicarmos o TEO a este sinal, seremos capazes de extrair a função envelope deste sinal abstrato. Esta é a base teórica do método SLP-TEO.

A Figura 2.10 mostra a imagem do texto “f04-020” onde podemos ver a projeção horizontal e o envelope do TEO. O método SLP-TEO é aplicado aos valores de projeção horizontal da imagem do texto para segmentação de linhas de texto. Para segmentação de palavras, o método é aplicado aos valores de projeção vertical da imagem de cada linha de texto segmentada.

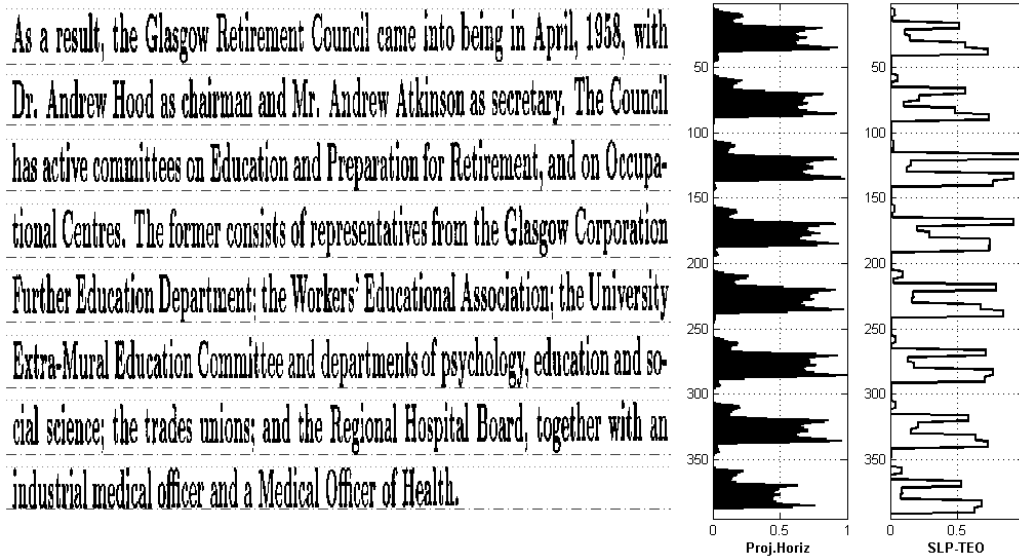


Figura 2.10: Projeção horizontal e envelope do TEO extraídos do texto impresso “f04-020”.

2.7 Trabalhos Relacionados

Nesta seção apresentamos alguns trabalhos mais recentes que englobam a segmentação de linhas e palavras em textos impressos e manuscritos, bem como a segmentação de caracteres.

Dentre as metodologias empregadas para a segmentação de linhas em documentos de texto manuscritos, podemos citar algumas categorias mais utilizadas: métodos baseados em projeção; métodos que utilizam manchas (*smearing*); e aplicações da transformada de Hough. A transformada de Hough é considerado como o estado da arte para segmentação linhas e palavras de textos manuscritos [16], [30].

No trabalho de Bar-Yosef, Hagbi e Kedem [2], são utilizados documentos degradados. A segmentação de linhas destes documentos é feita através do método de projeção. Basicamente o autor executa a soma dos valores dos pixels ao longo da direção horizontal para encontrar espaços entre linhas de texto através dos valores da projeção. Em seguida, são aplicadas técnicas

estatísticas para se encontrar o mínimo local, e algoritmos de passagem por zero para detectar as intensidades dos picos.

No trabalho de Louloudis, Gatos e Halatsis [16], a primeira etapa de processamento da imagem do texto inclui a binarização e o realce da imagem, extração de componentes conectados e estimação de altura do caractere. Na segunda etapa, uma transformada de Hough baseada em blocos é aplicada para detecção de possíveis linhas de texto. A terceira etapa visa corrigir possíveis fragmentações para detectar linhas de texto que não haviam sido detectadas previamente. Finalmente, caracteres conectados verticalmente são separados e designados às suas linhas de texto de origem.

Em outro exemplo, no trabalho de Saha et al. [30], temos uma primeira etapa de detecção de bordas através do algoritmo *Edge Sobel*. Aplicando-se a transformada de *Hough* com os parâmetros ajustados à imagem binarizada, temos as linhas de texto identificadas como componentes conectados. Essas linhas são então segmentadas por um algoritmo do tipo *Connected Component Labeling* (CCL). Cada linha de texto segmentada é então submetida a essas mesmas etapas, utilizando-se parâmetros específicos para a segmentação de palavras.

Quanto à segmentação de palavras, muitos métodos têm sido utilizados, como os métodos baseados em medidas estatísticas associados aos métodos de projeção [1]. Normalmente, os métodos baseados em projeção são associados a diferentes medidas de distâncias — *convex Hull-based*, distância Euclidiana, dentre outras. Liwicki, Scherz e Bunke [14] apresentam um método onde as palavras são segmentadas utilizando as distâncias dos componentes adjacentes de uma linha. As distâncias utilizadas nesse trabalho são a distância Euclidiana e a *convex Hull-based*.

Muitos trabalhos tem abordado a segmentação de caracteres conectados e sobrepostos em dígitos, mas poucos em alfabetos[29]. Alguns trabalhos tem

sido desenvolvidos nos últimos anos com o objetivo de leitura automática por máquinas, para isso várias estratégias tem sido investigadas para a solução dos caracteres conectados, isso por que, há uma variação muito grande entre os vários tipos de idiomas[29].

No trabalho [17] os autores falam sobre a segmentação de caracteres capturadas por câmeras. Neste trabalho é realizada a extração da linha de texto primeiramente, e verificado os componentes conectados(CC) da linha de texto. Os autores utilizam como base deste trabalho as técnicas de projeção horizontal e vertical, e criam um vetor de características extraídas dessas projeções e em seguida é realizada a classificação através do *support vector machine*(SVM).

No trabalho [21] os autores efetuam todas as etapas de segmentação de linhas, palavras e caracteres em textos históricos impressos. Para a etapa de segmentação de caracteres é utilizado um algoritmo de esqueletização e a partir dessa etapa é feito use a segmentação isolando os possíveis caracteres conectados.

Já no trabalho [9] o autor utiliza a projeção horizontal, vertical e lateral dos caracteres para extrair as características destes caracteres. O autor considera que a conexão entre caracteres não interfere na visão lateral. Assim, ele agrupa os caracteres em 13 classes distintas de acordo com as características apresentadas, e em seguida, realiza os cálculos de para verificar o melhor ponto de corte para segmentação dos caracteres.

2.8 Conclusão

Neste capítulo foram apresentados vários tópicos fundamentais para o desenvolvimento e processamento digital de imagens. Dentre os quais podemos

destacar as bases para esta pesquisa como projeção horizontal e vertical e a segmentação de regiões. No próximo capítulo detalharemos os métodos de segmentação que foram desenvolvidos nesta pesquisa.

Capítulo 3

Solução Proposta

Neste capítulo são apresentados as etapas de pré-processamento da imagem e os dois métodos desenvolvidos nesta pesquisa.

3.1 Preparação da Base de Dados

A base de imagens adotada para esta pesquisa é a *IAM-Database* [19], uma conceituada base de imagens para reconhecimento de documentos de textos impressos e manuscritos. A base de dados contém formulários de textos digitalizados a uma resolução de 300 dpi e armazenados no formato de imagem PNG com 256 níveis de cinza. Cada formulário da *IAM-Database* contém uma parte de texto impressa e a transcrição em escrita manual do mesmo texto. Contribuíram para a criação da base de dados 657 autores. Dos 1.539 formulários foram disponibilizados 1.507 páginas de texto digitalizadas.

A Figura 3.1 apresenta um dos formulários extraídos dessa base de dados utilizado neste trabalho.

Uma imagem digital é codificada usando-se formatos específicos para ar-

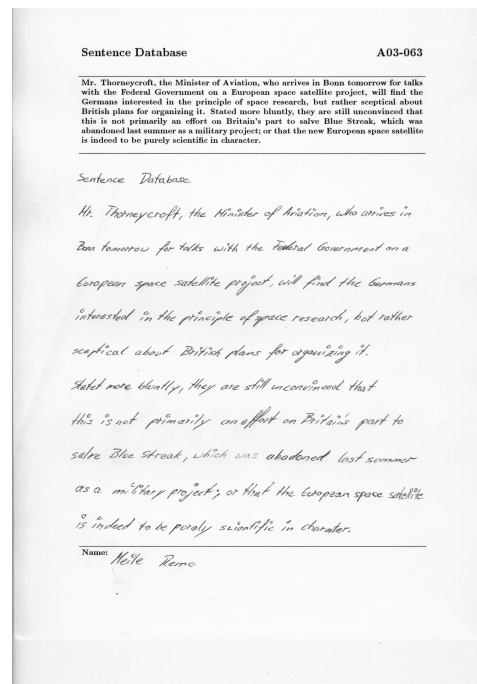


Figura 3.1: Formulário da IAM-Database

mazenamento, transmissão e decodificação de leitura para a memória. Neste trabalho optou-se por manter o formato original das imagens da base de dados que estão no formato PNG(*Portable Network Graphics*).

O formato de arquivos de imagens PNG surgiu em 1996. A versão original da especificação PNG foi escrita por *Thomas Boutell* e *Tom Lane*. Esse formato passou por várias atualizações até se tornar a especificação, *International Standard 15948:2003 – Portable Network Graphics*(PNG) recomendado pela *World Wide Web Consortium*-(W3C) em 2004.

O formato PNG apresenta várias vantagens como:

- O formato PNG independente de plataforma;
- Suporta compressão sem perda de informações;
- É utilizada para vários tipos de imagens, desde imagens pequenas como faixas ou imagens complexas como fotografias;

- Pode ser utilizado em vários programas de edição de imagens;
- É suportado por vários *Web browsers*;
- Suporta canal alfa, ou seja, define a opacidade de um pixel numa imagem;
- É um formato livre recomendado pela W3C;

Primeiramente foram selecionados aleatoriamente 40 formulários da base de imagens *IAM-Database* (com o auxílio da função *rand* do MATLAB®). A tabela 3.1 apresenta os 40 textos selecionados que constituem nossa amostra.

3.1.1 Distinção de Textos Impressos e Manuscritos

O primeiro passo foi extrair de cada formulário da base de dados utilizada as partes correspondentes aos textos impressos e manuscritos separadamente. Portanto, dois conjuntos de textos foram preparados. O primeiro conjunto foi constituído apenas por imagens de textos impressos que são o foco do nosso trabalho. O segundo conjunto é constituído por textos manuscritos utilizados neste trabalho como um estudo de caso.

Todos os experimentos foram feitos utilizando textos puramente impressos ou puramente manuscritos como mostrado na Figura 3.2

3.1.2 Caracteres Impressos da Base de Dados

A tabela 3.2 mostra o número de caracteres que apresentam problemas em cada texto da base de dados utilizada. É comum a presença de 3 tipos de caracteres que podem resultar em problemas na etapa de segmentação. Na base de imagens *IAM-DataBase*, foram observados os seguintes tipos

Tabela 3.1: Tabela dos 40 textos seleccionados da base de imagens *IAM-DataBase*

ID-Texto		Impressos			Manuscritos	
		Nº Linhas	Nº Palavras	Nº Caracteres	Nº Linhas	Nº Palavras
1	A03-011	6	78	380	9	77
2	A03-063	7	85	453	10	87
3	A04-010	6	69	408	9	68
4	A06-004	5	66	328	10	65
5	C03-000B	5	56	279	7	56
6	C04-017	5	66	297	8	66
7	D01-016	4	51	250	4	51
8	D01-052	4	57	274	8	57
9	D04-021	4	60	283	8	59
10	E04-043	5	60	275	8	60
11	E04-132	3	34	167	5	34
12	E06-026	4	54	272	10	55
13	F02-040	8	91	542	10	71
14	F04-079	7	86	420	11	88
15	G01-088	8	90	496	10	74
16	G06-011C	6	88	382	10	88
17	G06-018M	5	70	296	8	70
18	G06-042	4	51	220	6	53
19	G06-047I	5	58	275	7	58
20	G06-105	4	58	278	7	58
21	H01-000	6	77	366	11	77
22	H06-003	6	85	402	11	87
23	H07-069A	4	56	251	6	54
24	J04-035	6	69	357	10	66
25	J06-034	6	74	428	11	49
26	K04-103	5	69	326	8	68
27	K07-063A	4	62	264	7	62
28	l07-187	5	68	313	9	69
29	M01-121	5	65	299	9	66
30	M03-033	4	56	268	8	57
31	M04-000	4	57	254	8	59
32	M04-190	6	78	403	10	79
33	M06-106	6	82	404	10	82
34	N04-075	5	61	286	8	63
35	N04-084	4	58	263	7	57
36	N06-133	5	76	333	9	76
37	P06-047	4	59	277	11	60
38	R02-013	5	70	309	12	72
39	R02-131	4	61	263	6	60
40	R06-143	3	46	175	5	48
Total		202	2657	12816	341	2606

The body was examined, and it was definitely established that no sexual assault had taken place. It seemed probable that the motive had been robbery, and this was further confirmed when a check was made with the woman's relations in Hounslow. Miss Camp had been the housekeeper at the Good Intent, a public-house in Walworth. Her day off was Thursday, and before coming up to London in the evening she had called on her two sisters, one of whom lived at Hammersmith and the other at Hounslow, where she kept a shop.

(a) Imagem de texto impresso.

The body was examined, and it was definitely established that no sexual assault had taken place. It seemed probable that the motive had been robbery, and this was further confirmed when a check was made with the woman's relations in Hounslow. Miss Camp had been the housekeeper at the Good Intent, a public-house in Walworth. Her day off was Thursday, and before coming up to London in the evening she had called on her two sisters, one of whom lived at Hammersmith and the other at Hounslow where she kept a shop.

(b) Imagem de texto manuscrito.

Figura 3.2: Imagens de textos da base de dados *IAM-DataBase*.

de caracteres: Desenho de Fontes, Caracteres Sobrepostos e os Caracteres Conectados.

Os caracteres impressos totalizam 12816, sendo que destes, 64 caracteres são caracteres conectados representando 32 casos de conexão. Os caracteres sobrepostos totalizam 219 casos de sobreposição entre caracteres, totalizando 445 caracteres perdidos, visto que há alguns casos de sobreposição que envolvem mais de dois caracteres. Outro problema encontrado na base de dados é o desenho de fontes que apresentaram 46 casos, ou seja a segmentação de 93 caracteres é comprometida por este problema.

Como podemos ver através da tabela 3.2, os caracteres que apresentam problemas totalizam 602 caracteres comprometidos da base de dados.

Tabela 3.2: Caracteres que compõe a base de dados utilizada

		Caracteres Conectados		Caracteres Sobrepostos		Desenho de Fontes		
		Caracteres	Casos	Caracteres	Casos	Caracteres	Casos	Caracteres
1	A03-011	380	0	0	8	16	0	0
2	A03-063	453	0	0	4	8	3	6
3	A04-010	408	0	0	8	16	1	2
4	A06-004	328	0	0	2	4	1	2
5	C03-000B	279	1	2	5	10	1	2
6	C04-017	297	0	0	10	20	1	2
7	D01-016	250	0	0	3	6	1	2
8	D01-052	274	0	0	1	2	0	0
9	D04-021	283	0	0	2	4	0	0
10	E04-043	275	0	0	9	19	0	0
11	E04-132	167	0	0	10	20	0	0
12	E06-026	272	0	0	5	10	5	10
13	F02-040	542	1	2	5	10	3	7
14	F04-079	420	2	4	7	14	2	4
15	G01-088	496	1	2	5	10	3	6
16	G06-011C	382	3	6	13	27	1	2
17	G06-018M	296	2	4	3	6	3	6
18	G06-042R	220	1	2	4	8	0	0
19	G06-047I	275	3	6	3	6	0	0
20	G06-105	278	2	4	7	15	1	2
21	H01-000	366	0	0	6	12	1	2
22	H06-003	402	1	2	7	14	1	2
23	H07-069A	251	1	2	2	4	0	0
24	J04-035	357	0	0	2	4	2	4
25	J06-034	428	0	0	9	18	2	4
26	K04-103	326	0	0	3	6	0	0
27	K07-063A	264	2	4	3	6	0	0
28	L07-187	313	3	6	7	16	3	6
29	M01-121	299	0	0	2	4	1	2
30	M03-033	268	2	4	8	16	0	0
31	M04-000	254	0	0	5	10	0	0
32	M04-190	403	0	0	2	4	0	0
33	M06-106	404	3	6	14	30	3	6
34	N04-075	286	0	0	1	2	0	0
35	N04-084	263	0	0	3	6	2	4
36	N06-133	333	1	2	9	18	0	0
37	P06-047	277	2	4	9	18	3	6
38	R02-013	309	0	0	6	12	2	4
39	R02-131	263	0	0	4	8	0	0
40	R06-143	175	1	2	3	6	0	0
		12816	32	64	219	445	46	93

Desenho de Fontes

Projetista de fontes (*typeface design*) é um termo utilizado para os profissionais que projetam fontes [7]. Alguns exemplos dessas fontes podem ser vistos na Figura 3.3 abaixo.

*Th, Fr, Fi, de, do, da, du, er, ef, fi, fl, ff, fl, fi, ju,
ji, lo, la, le, ll, li, la, st, sp, ss, tu, tr, ti, sl, sh, si, ft, uy*

(a) 1º Exemplo de Desenho de Fonte.

***Ex & Fi Fi Th Th 1 J U
fu fu fi fi***

(b) 2º Exemplo de Desenho de Fonte.

fu fu fa fa fo fo fr fr ff ff

(c) 3º Exemplo de Desenho de Fonte.

abcçdeflqhijkl

(d) 4º Exemplo de Desenho de Fonte.

Figura 3.3: Exemplo de fontes criadas por design de fontes [8].

Observe que nos exemplos apresentados, a fonte desenhada pode conter a junção de dois ou mais caracteres, dependendo da veia artística do projetista. Estas fontes são desenhadas e representadas por símbolos específicos. Na Figura 3.4 são mostradas algumas palavras que possuem esse tipo de fonte em sua composição.

Na base de dados foram encontrados 46 junções que tipificam os caracteres que são desenho de fontes. A Figura 3.5 mostra alguns exemplos deste tipo de problema encontrado na base.

Note que as junção das letras “f” e “i”, e também das letras “f”, “f” e “i” não são apenas conexão, mas trata-se de um trabalho idealizado por artistas

Litografia

(a) 1ª Palavra que apresenta um desenho de fonte.

Prototipografia

(b) 2ª Palavra que apresenta um desenho de fonte.

Beneficial

(c) 3ª Palavra que apresenta um desenho de fonte.

fluxos

(d) 4ª Palavra que apresenta um desenho de fonte.

Figura 3.4: Exemplo de palavra que apresentam caracteres denominados desenho de fontes.

que criam fontes e não erro de digitalização. Portanto, estes caracteres não podem ser segmentados como os demais, uma vez que a separação de um dos caracteres ocasiona a degradação do outro. Consequentemente, para este caso específico, recomenda-se que o posterior reconhecimento destes caracteres seja feito de forma conjunta, ou seja, que sejam considerados apenas um caractere simples e não dois ou mais caracteres.



(a) Desenho de fonte “ffi” .



(b) Desenho de fonte “fi”.



(c) Desenho de fonte “Fr”.



(d) Desenho de fonte “Tr”.

Figura 3.5: Exemplo de caracteres que não podem ser segmentados como os demais.

Caracteres Sobrepostos

Os caracteres sobrepostos podem ser definidos como aqueles que não podem ser separados por uma linha vertical[24].

Na base de dados utilizada foram encontradas várias palavras que apresentam este tipo de caractere. Na Figura 3.6 podem ser visualizados alguns exemplos de palavras encontradas nos textos utilizados que apresentam caracteres sobreposto.

Note que na palavra “fifteen” apresentada na Figura 3.6(a), o caractere “f” sobrepõe o caractere “t”. Além disso, esta palavra também apresenta um caractere desenho de fonte “fi”. Já na imagem da palavra “nullify” o caractere “f” sobrepõe o caractere “y”.

Um método baseado em segmentação orientada à regiões utilizando imagens binarizadas seria suficiente para resolver o problema dos caracteres sobrepostos. Entretanto, se fossem utilizadas as imagens binarizadas não seria possível segmentar os caracteres conectados como veremos a seguir.



(a) Palavra “fifteen” que apresenta a sobreposição dos caracteres “f” e “t”.



(b) Palavra “nullify” que apresenta a sobreposição dos caracteres “l” e “y”.

Figura 3.6: Exemplo de palavras com caracteres sobrepostos.

Caracteres Conectados

Os caracteres conectados são originados quando dois caracteres adjacentes são escritos muito próximos e se tocam em um ou mais pontos. Sendo que alguns caracteres podem se conectar tanto horizontalmente nas direções esquerda/direita ou verticalmente se conectando tanto na parte superior quanto inferior dos caracteres envolvidos[29].

A Figura 3.7 mostra a palavra “narrow” que apresenta dois de seus caracteres conectados, o caractere “a” e o caractere “r”.



Figura 3.7: Palavra “narrow” que apresenta dois caracteres conectados “a” e “r”.

Caracteres Simples (Sem problemas)

A Tabela 3.3 apresenta o número de caracteres simples que são “salvos” dos problemas apresentados.

Tabela 3.3: Caracteres simples da base de dados

		Nº Total	Nº Caracteres	Nº Caracteres	Nº Caracteres
		Caracteres	Sem Conectados	Sem Sobrepostos	Sem Desenho Fonte
1	A03-011	380	380	364	380
2	A03-063	453	453	445	447
3	A04-010	408	408	392	406
4	A06-004	328	328	324	326
5	C03-000B	279	277	269	277
6	C04-017	297	297	277	295
7	D01-016	250	250	244	248
8	D01-052	274	274	272	274
9	D04-021	283	283	279	283
10	E04-043	275	275	256	275
11	E04-132	167	167	147	167
12	E06-026	272	272	262	262
13	F02-040	542	540	532	535
14	F04-079	420	416	406	416
15	G01-088	496	494	486	490
16	G06-011C	382	376	355	380
17	G06-018M	296	292	290	290
18	G06-042R	220	218	212	220
19	G06-047I	275	269	269	275
20	G06-105	278	274	263	276
21	H01-000	366	366	354	364
22	H06-003	402	400	388	400
23	H07-069A	251	249	247	251
24	J04-035	357	357	353	353
25	J06-034	428	428	410	424
26	K04-103	326	326	320	326
27	K07-063A	264	260	258	264
28	L07-187	313	307	297	307
29	M01-121	299	299	295	297
30	M03-033	268	264	252	268
31	M04-000	254	254	244	254
32	M04-190	403	403	399	403
33	M06-106	404	398	374	398
34	N04-075	286	286	284	286
35	N04-084	263	263	257	259
36	N06-133	333	331	315	333
37	P06-047	277	273	259	271
38	R02-013	309	309	297	305
39	R02-131	263	263	255	263
40	R06-143	175	173	169	175
		12816	12752	12371	12723

3.2 Pré-processamento da Imagem

Segundo [26], “a etapa de pré-processamento visa melhorar a qualidade da imagem por meio de aplicação de técnicas para atenuação de ruídos, correção de contraste ou brilho e suavização de determinadas propriedades da imagem”.

Dentre as técnicas para melhoria da qualidade da imagem digital, podemos destacar a suavização de ruídos a partir de filtragens no domínio da frequência — como filtros passa-baixa ideais, Butterworth ou gaussianos —, além de correções de rotação e translação da imagem [6]. Também podemos utilizar técnicas de realce de imagem — como transformação de escala de cinza, equalização ou hiperbolização de histograma, filtros no domínio espacial, dentre outras [26].

Para este trabalho, consideramos imagens bem representativas de textos, ou seja, imagens de boa qualidade que não apresentam distorções que justifiquem a aplicação das técnicas supra mencionadas. A única preocupação foi a de converter imagem coloridas (em RGB) para níveis de cinza, uma vez que as cores em um texto não agregam informação aos caracteres independentes que o constituem. Para isso, utilizamos a conhecida fórmula de cálculo de luminância¹ (ou intensidade) definida por:

$$C = 0,2989 \cdot R + 0,5870 \cdot G + 0,1140 \cdot B \quad (3.1)$$

onde C é o nível de cinza a ser atribuído ao pixel; R é o valor em sua banda *Red*; G é o valor na banda *Green*; e B é valor na banda *Blue*.

Os valores de nível de cinza resultantes para a imagem são então normalizados dentro do intervalo de 0 a 1, para o propósito dos cálculos necessários

¹Esta fórmula é parte dos cálculos para conversão do sistema RGB para o sistema NTSC de televisão.

a este trabalho.

3.2.1 Binarização

Na etapa de Binarização da imagem de texto foi utilizado o método de Otsu [23]. Esse método se mostra muito eficiente e é citado como adequado para binarização de imagens de texto.

Segundo Gonzalez e Woods [6], “O método de Otsu é ótimo no sentido de que maximiza a variância entre as classes, uma medida bem conhecida e utilizada na análise estatística discriminante”. Os histogramas das imagens de texto, normalmente, apresentam duas modas, que chamamos de classe do fundo e classe do conteúdo da imagem. Sendo assim, a ideia apresentada por Otsu é que as classes com limiares bem definidos devem ser distintas em relação aos valores de intensidade dos seus pixels. Portanto, este método de limiarização busca identificar através de uma análise estatística o melhor limiar, ou limiar ótimo, que realiza uma separação entre as duas classes em termos de valores de intensidades dos seus pixels.

Segundo Gonzalez e Woods [6], o algoritmo de Otsu pode ser resumido da seguinte maneira:

1. Calcula-se o histograma normalizado da imagem $\rho_i = \frac{x_i}{MN}$, onde x_i denota o número de pixels com intensidade i e MN é o número total de pixel da imagem.
 2. Estabelecemos um limiar $T(k) = k$, onde $0 < k < L - 1$. Este limiar é utilizado para estabelecer as duas classes da imagem de entrada C_1 e C_2 . Sendo que a classe C_1 está constituída por todos os pixels da imagem com valores de intensidade no intervalo $[0, k]$ e a classe C_2 pertence ao intervalo $[k + 1, L - 1]$. Em seguida, calcula-se a soma cumulativa para
-

estabelecer se o limiar será atribuído a classe C_1 através da equação

$$P_1(k) = \sum_{i=0}^k \rho_i$$

3. Em seguida, calcula-se a média acumulada, $m(k)$, para $k = 0, 1, 2, \dots, L-1$ através da equação

$$m(k) = \sum_{i=0}^k i\rho_i$$

4. Depois do cálculo da média acumulada para intensidade até o nível k , calcula-se a média de toda a imagem ou média global, m_g , para $k = 0, 1, 2, \dots, L-1$ através da equação

$$m_G = \sum_{i=0}^{L-1} i\rho_i$$

5. Depois de calculada a média de toda a imagem, em seguida é calcula a variância entre as classes, através da equação abaixo, sendo $k = 0, 1, 2, \dots, L-1$.

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

6. E por fim, obter o limiar de Otsu, k_* , como o valor de k para o qual é máxima. Caso a máxima variância não for única, calcula-se a média dos valores de k que correspondem aos diversos valores máximos detectados.

A Figura 3.8 mostra um texto impresso extraído da base de dados utilizada e a Figura 3.8(b) mostra o mesmo texto binarizado.

The body was examined, and it was definitely established that no sexual assault had taken place. It seemed probable that the motive had been robbery, and this was further confirmed when a check was made with the woman's relations in Hounslow. Miss Camp had been the housekeeper at the Good Intent, a public-house in Walworth. Her day off was Thursday, and before coming up to London in the evening she had called on her two sisters, one of whom lived at Hammersmith and the other at Hounslow, where she kept a shop.

(a) Imagem de texto impresso em tons de Cinza.

The body was examined, and it was definitely established that no sexual assault had taken place. It seemed probable that the motive had been robbery, and this was further confirmed when a check was made with the woman's relations in Hounslow. Miss Camp had been the housekeeper at the Good Intent, a public-house in Walworth. Her day off was Thursday, and before coming up to London in the evening she had called on her two sisters, one of whom lived at Hammersmith and the other at Hounslow, where she kept a shop.

(b) Texto binarizado pelo método de Otsu.

Figura 3.8: Imagem de texto impresso da *IAM-DataBase*.

3.3 Método SLP-TEO

Foi desenvolvido um método denominado “Segmentação de Linhas e Palavras baseado no Operador de Energia de Teager” (SLP-TEO) para as etapas de segmentação de linhas e palavras. Este método é baseado no Operador de Energia de Teager (TEO) apresentado em [32], [10]. Kaiser [10] apresenta uma importante característica do TEO, ele diz que quando este operador é aplicado a sinais compostos por duas ou mais componentes de frequência “seria como se o algoritmo [TEO] fosse capaz de extrair a função envelope do sinal” [tradução nossa]. Portanto, os valores da projeção linear (histograma) de uma imagem de texto binarizada pode ser considerado como sendo um sinal amostrado. Assim, considerando a teoria de séries de Fourier, este sinal (função arbitrária dentro de dado intervalo) pode ser expresso por um série de senos e cossenos. Ou seja, teremos como resultado dessa abstração um sinal composto por diversas componentes de frequência. Se aplicarmos o TEO a este sinal, seremos capazes de extrair a função envelope deste sinal abstrato. Como exemplo de aplicação da utilização do envelope do TEO, podemos citar a dissertação de Peretta [27]. Esse autor utilizou esta característica do TEO para extrair a função envelope de um sinal, para a detecção de fronteiras de

um comando de voz inserido em um sinal de áudio.

3.3.1 Segmentação de Linhas

Para a segmentação de linhas é importante que a imagem de texto já tenha sido binarizada, pois o algoritmo de segmentação de linhas recebe como entrada a matriz binária da imagem do texto impresso ou manuscrito. Em seguida, é realizada a projeção horizontal desta imagem.

O operador TEO é aplicado aos valores encontrados no histograma (a projeção horizontal) como se fossem parte de um sinal amostrado. Os valores calculados através do operador TEO são armazenados em um vetor, como é demonstrado em na equação 3.2.

$$\Omega(p_h(x)) = |p_h(x)^2 - (p_h(x+1) \cdot p_h(x-1))| \quad (3.2)$$

onde $\Omega(p_h(x))$ é o vetor ABS-TEO e $p_h(x)$ é a projeção horizontal.

A Figura 3.9 mostra o envelope do TEO, parte colorida do gráfico, sobreposto sobre a projeção horizontal.

A seguir, o método SLP-TEO converte os valores deste vetor em valores absolutos. Este vetor é designado como vetor ABS-TEO (Ω).

Em seguida, o ABS-TEO é normalizado no intervalo $[0, 1]$, passando a ser chamado ABS-Norm-TEO (Ω^*) como pode ser observado na Equação 3.3.

$$\Omega^*(p_h(x)) = \frac{\Omega(p_h(x))}{\max(\Omega)} \quad (3.3)$$

Depois de encontrar o valor do operador TEO para cada ponto da projeção horizontal é necessário encontrar as fronteiras das linhas de texto. Essas fronteiras são constituídas pelos limites superior e inferior de cada linha de

As a result, the Glasgow Retirement Council came into being in April, 1958, with Dr. Andrew Hood as chairman and Mr. Andrew Atkinson as secretary. The Council has active committees on Education and Preparation for Retirement, and on Occupational Centres. The former consists of representatives from the Glasgow Corporation Further Education Department; the Workers' Educational Association; the University Extra-Mural Education Committee and departments of psychology, education and social science; the trades unions; and the Regional Hospital Board, together with an industrial medical officer and a Medical Officer of Health.

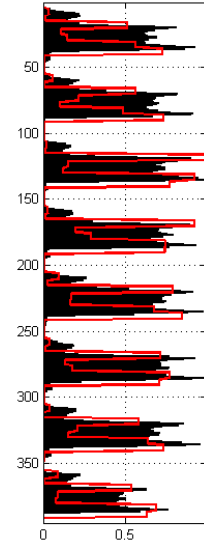


Figura 3.9: Envelope do TEO sobre a projeção Horizontal

texto contida na imagem binarizada. Para isso, é definida uma altura mínima chamada de H_{min} para uma linha de texto válida. Como pode ser observado na Figura 3.10, algumas letras necessitam de 3 pixels como altura mínima, entretanto, nem todas letras podem ser representadas por este número mínimo de pixels, como é o caso das letra “M” e “W”. Assim, atribuímos a variável H_{min} o valor 6 correspondendo a 5 pixels considerando a altura da letra propriamente dita e 1 pixels de borda.

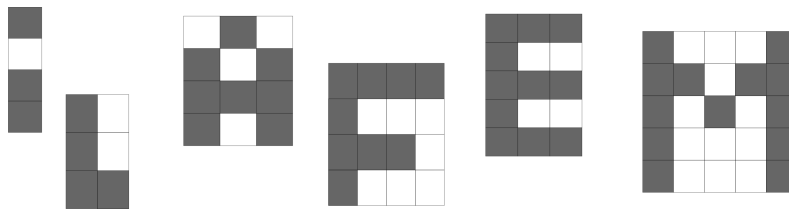


Figura 3.10: Representação de letras em matrizes.

Também foi definido um limiar que corresponde a 5% da média do ABS-Norm-TEO (Ω^*). Este limiar é utilizado na comparação com os valores individuais do vetor que constitui o ABS-Norm-TEO (Ω^*) para detecção de

linhas de texto como pode ser observado no diagrama de bloco da Figura 3.11 onde é detalhado o método SLP-TEO para segmentação de linhas.

O método SLP-TEO percorre o vetor ABS-Norm-TEO (Ω^*) à procura de fronteiras. Se o valor correspondente da linha no vetor é maior ou igual ao limiar definido e se a variável de controle é falsa, o método marca aquela linha da imagem como sendo a fronteira **superior** de uma linha de texto. Se o valor correspondente da linha no vetor é menor do que o limiar definido e se a variável de controle é verdadeira, o método marca aquela linha da imagem como sendo a fronteira **inferior** de uma linha de texto. Ao encontrar uma fronteira inferior, o método checa ainda se a altura em pixels encontrada é maior do que a altura mínima pré-determinada. Se sim, as fronteiras superior e inferior são validadas. Se não, as fronteiras encontradas são removidas.

Ajustes ao Método

1. Localizar linhas com espaço entre linhas

Percorre-se cada linha identificada por suas fronteiras buscando alguma que contenha $p_h(k) = 0$ em algum k ou seja, projeção horizontal igual a 0 pixels. Se houver, quebra-se a linha em duas.

2. Contrair as fronteiras de linhas

A cada linha detectada, verifica-se a existência de $p_h(k) = 0$ abaixo de sua fronteira inferior e acima de sua fronteira superior. Caso afirmativo, contrai as fronteiras até algum k tal que $p_h(k) \neq 0$.

3. Estender fronteiras

A cada linha detectada, verifica-se a existência de $p_h(k) \neq 0$ acima de sua fronteira superior e abaixo de sua fronteira inferior. Caso afirmativo, reajusta as fronteiras até algum k tal que $p_h(k) = 0$.

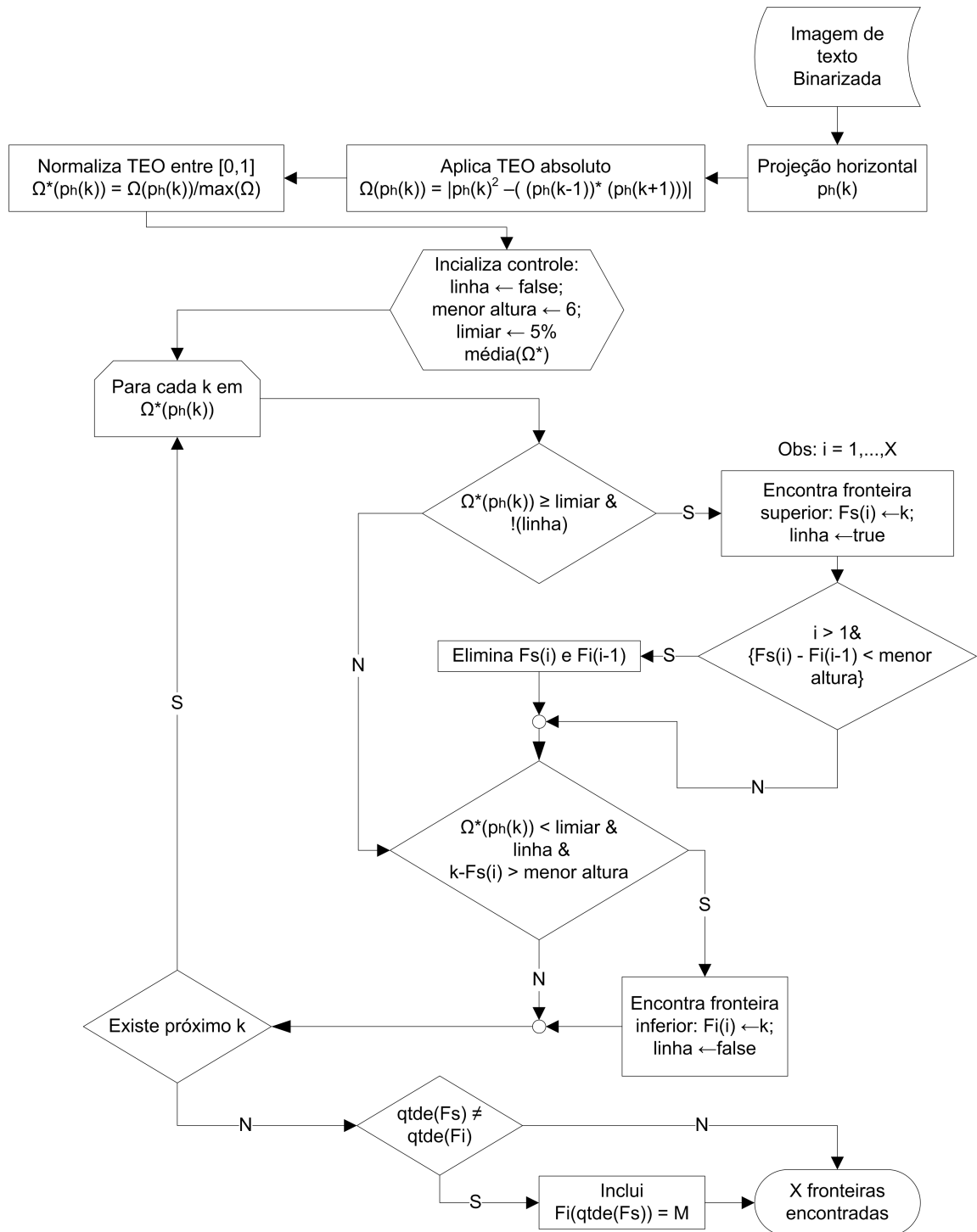


Figura 3.11: Diagrama de bloco do método SLP-TEO para segmentação de linhas.

4. Verificar consistência da linhas

Após o cálculo da altura média das linhas detectadas, é feita a verificação se duas ou mais linhas adjacentes (fronteira inferior de uma está a no máximo 1 pixel da fronteira superior da outra). Caso afirmativo, verifica se pelo menos uma das linhas possui uma altura de no máximo 90% da altura média das linhas detectadas. Se for o caso, unifica as duas linhas definindo uma nova linha com a fronteira superior da 1ª linha e a fronteira inferior da 2ª linha.

5. Eliminação de linhas irrisórias

Nesta etapa é verificado a existência de linhas. Se for encontrada 1 linha com a altura menor do que o valor-controle de menor altura (H_{min}), essa linha é eliminada.

3.3.2 Segmentação de Palavras

O algoritmo de segmentação de palavras recebe como entrada uma imagem binária de cada linha de texto previamente segmentada. A projeção (histograma) vertical desta imagem é determinada. Em seguida, o operador TEO é aplicado a esses valores como se fossem parte de um sinal amostrado composto por várias componentes de frequência, gerando um vetor com os valores resultantes desta operação. A seguir, o método SLP-TEO converte os valores deste vetor $ABS-TEO(\Omega)$ em valores absolutos e é normalizado no intervalo $[0, 1]$, passando a ser chamado $ABS-Norm-TEO (\Omega^*)$.

O método é o mesmo utilizado para segmentação de linhas de texto, apenas com as seguintes modificações:

- Ao invés de uma altura mínima (H_{min}), é definido uma largura mínima
-

(L_{min}) através da Equação 3.4;

$$L_{min} = \max(10, \lceil 0.15 \cdot H \rceil), \quad (3.4)$$

onde L_{min} é a largura mínima (em pixels) necessária ao método e H é a altura (em pixels) da imagem da linha de texto segmentada.

- O método localiza fronteiras direita e esquerda ao invés de superior e inferior.

Os mesmos ajustes realizados para a detecção de fronteiras das linhas de texto na subseção 3.3.1 são aplicados para a detecção de fronteiras das palavras a serem segmentadas, como pode ser visto no diagrama da Figura 3.12. Nesta etapa se estende uma determinada fronteira que contém uma possível palavra até o limite das fronteiras direita e esquerda, inflando-as. Primeiramente, é feito o ajuste para a esquerda, onde é analisado se a partir da fronteira detectada há uma coluna anterior à fronteira e se nela há algum pixel ativo. Assim, se a fronteira esquerda analisada não chegar à fronteira direita anterior, então se deve antecipar a fronteira. E se for a primeira coluna já antecipa-se a fronteira normalmente. O procedimento é o mesmo para os ajustes posteriores da fronteira, com detalhe ao invés de antecipar retrocede a fronteira.

Na terceira etapa, é verificada a consistência das palavras encontradas. Se entre uma fronteira esquerda e a fronteira direita exatamente anterior não houver ao menos 1 pixel de distância, então concatena-se as fronteiras encontradas.

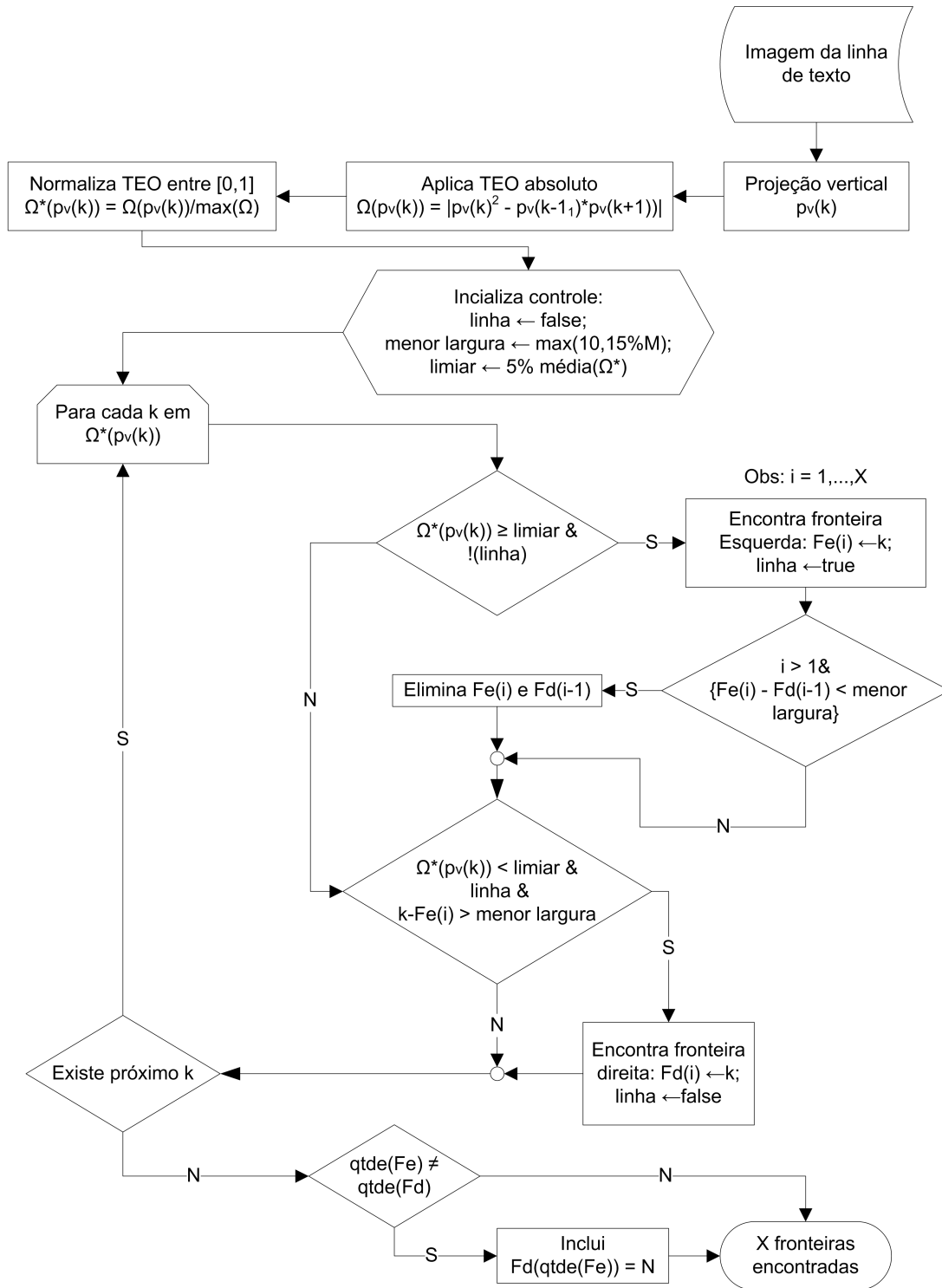


Figura 3.12: Diagrama de bloco do método SLP-TEO para segmentação de Palavras.

3.4 Método gROSC

A finalidade de um sistema de segmentação de imagens de textos é extrair as partes constituintes do texto, separando-as do fundo da imagem. Estudos indicam que a segmentação em textos manuscritos é mais efetiva quando visa o nível de segmentação de palavras [31]. Em textos impressos, entretanto, o nível de segmentação de caracteres é o mais desejado. Para uma maior eficiência em um sistema de OCR, é sugerido que os caracteres segmentados sejam relacionados à sua palavra de origem, visto que, para etapas de pós-processamento, esse tipo de informação é muito relevante.

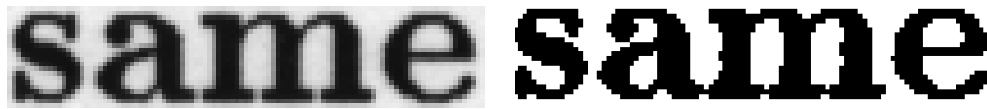
Portanto, o método aqui proposto parte da imagem da palavra segmentada para a segmentação de caracteres.

3.4.1 Análise da Imagem em Tons de Cinza

Ao observarmos a imagem em tons de cinza, percebemos que os níveis de cinza seriam determinantes para se separar corretamente os caracteres, principalmente os conectados. Assim, procuramos analisar a diferenciação feita através da visão humana quanto os níveis de cinza em um espectro de cores.

Note que, observando a imagem em tons de cinza mostrada na Figura 3.13(a). Podemos, com facilidade, dizer que as letras “a” e “m” são distintas, devido a tonalidade de cinza. Entretanto, isso não acontece na imagem binarizada, pois o método de binarização considerou o pixel entre as duas letras como um pixel constituinte do caractere e não do fundo da imagem 3.13(b).

Analisando a imagem original percebe-se que muitos pixels definidos como pixels pertencentes ao objeto de uma imagem através dos métodos de binarização, na verdade, não são parte deste caractere, mas sim do fundo da



(a) Imagem da palavra em tons de cinza.

(b) Imagem da palavra binarizada.

Figura 3.13: Imagem da palavra em tons de cinza e correspondente binarizada.

imagem.

3.4.2 Método gROSC

Considerações

Na base de dados utilizadas verificou-se a existência de três tipos de caracteres que são considerados problemas: os caracteres conectados, os caracteres sobrepostos e os desenhos de fontes.

1. Caracteres desenho de fontes:

Por ser um problema de segmentação alheio ao processo de digitalização e intrínseco às decisões do projetista da fonte utilizada, não o entendemos como um problema a ser resolvido pelo método desenvolvido. Portanto, aconselha-se que em uma etapa posterior seja realizado o reconhecimento destes caracteres conjuntamente, pois tentar separá-los ocasionaria a degradação dos caracteres envolvidos.

2. Caracteres Sobrepostos:

Um método baseado em segmentação orientada à regiões utilizando imagens binarizadas já seria suficiente para resolver este problema, visto que não há conexão entre eles. Entretanto, se fosse utilizado imagens binarizadas não seria possível segmentar os caracteres conectados.

3. Caracteres Conectados:

Os caracteres conectados é um problema típico de segmentação de caracteres. Segundo Saba [29] “as pesquisas que abordam os problemas de caracteres conectados se concentram na solução de conexão de dígitos, ou sequência de dígitos, mas poucos sobre alfabetos”. Além disso, o problema dos caracteres conectados são diferenciados de acordo com a linguagem, sendo que para cada tipo de idioma é necessário uma estratégia para se solucionar o problema em questão[29].

De acordo com Saba [29] “uma grande proporção dos resultados de reconhecimento errado são devidos a erros de segmentação”. Portanto, é primordial que nesta etapa da segmentação os resultados sejam satisfatórios.

A binarização de Otsu [23] mostrou-se insuficiente para eliminar o problema de caracteres conectados. De fato, a binarização de Otsu sem estar acompanhada de qualquer outra heurística gera esse problema, como veremos na seção seguinte.

Os métodos de binarização, em especial a binarização pela limiarização de Otsu, definem que alguns pixels pertencem à classe constituinte do objeto, quando, ao se analisar a imagem em tons de cinza, percebe-se que os mesmos pertencem à classe definida como fundo da imagem. Esse tipo de estratégia (binarização global) gera frequentemente caracteres conectados como na imagem binarizada de um texto.

A etapa de segmentação de palavras através do método SLP-TEO nos fornece como saída a imagem binarizada da palavra. O método gROSC, entretanto, parte da imagem da palavra em tons de cinza². Apoiado no limiar

²Para aplicação do método gROSC, é necessário utilizar as coordenadas geradas pelo SLP-TEO para a imagem binarizada da palavra a fim de extrair a mesma da imagem original do texto, em tons de cinza.

de Otsu, o método gROSC se propõe a não incorrer no erro de decisão local dos métodos de binarização globais, possibilitando a correta segmentação dos caracteres que seriam conectados. Note que os níveis de cinza da imagem são distribuídos entre o intervalo $[0; 1] \in \mathbb{R}$, onde 0 é o nível de cinza equivalente ao preto puro e 1 ao branco puro.

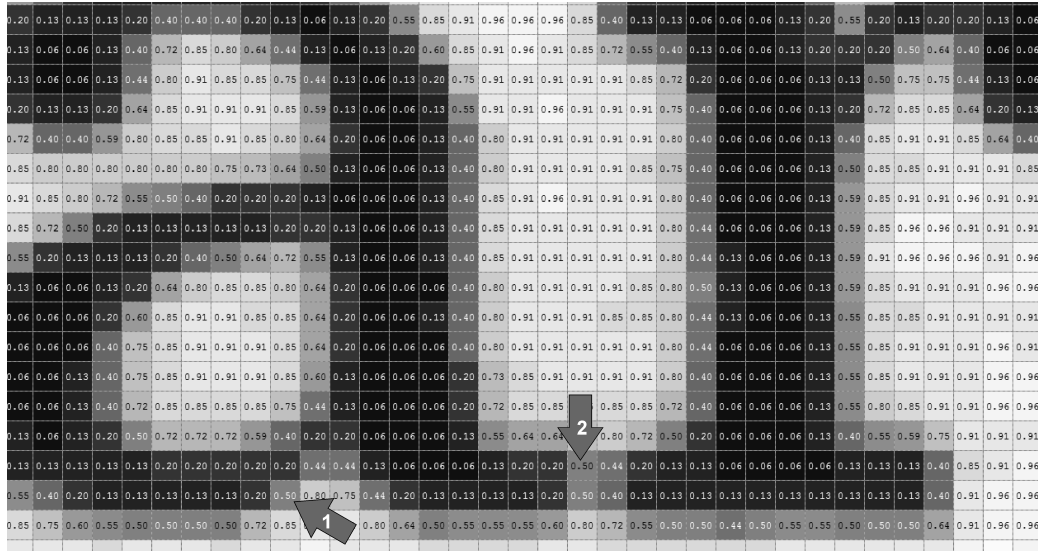


Figura 3.14: Pixels com mesmo tom de cinza: (1) pode ser considerado parte do objeto; (2) deve ser considerado parte do fundo.

O método de Otsu define um limiar de decisão para considerarmos cada nível de cinza como parte do objeto constituinte ou do fundo da imagem. Os pixels com níveis de cinza próximos ao limiar de Otsu podem pertencer à quaisquer das classes (objeto ou fundo), dependendo do local da imagem em que se encontram (ver Figura 3.14).

Para segmentar um caractere, o método gROSC depende de que o pixel inicial possua um nível de cinza que se enquadre certamente à classe constituinte do objeto³. Para isso, é necessário que todos os valores de níveis de cinza da imagem da palavra sejam conhecidos. A partir destes dados, são

³O gROSC se aproveita do fato de que o interior do caractere a ser segmentado ser mais escuro do que suas bordas.

selecionados os níveis de cinza que são únicos e em seguida ordenados em um vetor. Então são descartados do vetor todos os níveis de cinza que são definidos como fundo da imagem, com auxílio do limiar de Otsu. A seguir, é escolhida a *mediana* do vetor resultante de níveis únicos de cinza para caracterizar um pixel ativo, com relativa certeza. Esse valor encontrado é utilizado como o máximo nível de cinza aceitável para um pixel a ser procurado na imagem da palavra segmentada. O método gROSC parte deste pixel que sinaliza que ali existe um caractere (objeto) a ser segmentado.

O método gROSC teve inspiração nos métodos de segmentação por regiões [6]. Logo, após encontrarmos o pixel ativo, o método começa a visitar recursivamente os vizinhos de cada pixel identificado como pertencente à mesma região, i.e, procurando pixels que possuam um tom de cinza semelhante ao do pixel original. A cada pixel visitado é calculada a variação de tons de cinza necessária para que ocorra uma transição entre pixels vizinhos. A Equação 3.5 fornece esse nível de “salto” que o pixel observado pode dar. Ou seja, qual a diferença máxima entre tons para que um vizinho possa ser considerado parte da região do pixel observado ou não. Quanto mais próximo de zero (0) esse pixel for, ou seja, mais escuro, maior vai ser o valor encontrado para uma possível transição. Por outro lado, quanto mais o tom de cinza se aproximar do valor do limiar de Otsu, menor será o valor para uma transição.

$$\Delta g = A \cdot T \cdot \left(\frac{g - T}{g_0 - T} \right) \quad (3.5)$$

onde Δg é a variação máxima (superior ou inferior) entre tons vizinhos para que se aceite que pertençam à mesma região (objeto, ou caractere a ser segmentado); A é a inclinação da reta, definida pelo valor empírico 1, 2; T é o valor do limiar de Otsu para a imagem da palavra analisada; g é o nível de cinza do pixel observado; e g_0 é o menor valor dos níveis de cinza encontrado

na imagem da palavra analisada (pixel mais escuro).

Quando todas as possibilidades de transição forem verificadas, ou seja, quando não houver mais algum pixel com vizinhos “válidos”, a região rotulada é extraída e inserida em uma nova imagem em branco de igual tamanho. É formado assim o primeiro caractere extraído. Note que, ao contrário da imagem da palavra original, a imagem de um caractere extraído é gerada com os níveis preto e branco apenas. A imagem em tons de cinza modificada, sem o primeiro caractere, é novamente rastreada e, se encontrado um novo pixel ativo, o processo utilizado pelo gROSC para extração de caracteres inicia-se novamente.

Exemplo Prático

Observe a Figura 3.15(a) que mostra o mapa de pixels da imagem da letra “a”, extraída da base de imagens utilizada.

O método gROSC seleciona os níveis de cinza únicos da imagem da palavra e em seguida os ordena em um vetor. A Tabela 3.4 mostra os níveis de cinza da Figura 3.15(a).

Tabela 3.4: Níveis de Cinza únicos da imagem “a” (em negrito, os níveis permitidos para o início do gROSC; em itálico, os níveis abaixo do limiar de Otsu).

<i>0,0549</i>	<i>0,5176</i>	0,8078
<i>0,1333</i>	<i>0,5490</i>	0,8509
<i>0,1686</i>	0,5999	0,8666
<i>0,2000</i>	0,6313	0,9058
<i>0,2274</i>	0,6392	0,9568
<i>0,2862</i>	0,6901	0,9685
<i>0,4000</i>	0,7176	0,9725
<i>0,4353</i>	0,7489	0,9764
<i>0,4588</i>	0,7529	0,9803
<i>0,5019</i>	0,7999	0,9842



(a) Imagem exemplo.



(b) Caminho percorrido.

Figura 3.15: Imagem da letra “a” para exemplo.

A partir do valor do limiar de Otsu — neste exemplo, 0,5725 — é realizada uma seleção dos níveis de cinza menores ou iguais a este limiar (classe objeto), e são descartados os pixels que possuem valores maiores do que esse limiar (classe fundo). De posse dos níveis de cinza relevantes é calculada a mediana desses valores. Neste caso, o valor encontrado é $C_{max} = 0,2862$ (na Figura 3.15, temos 0,29).

Na primeira etapa, o método gROSC começa a visitar os pixels da imagem, começando pela coordenada superior esquerda na direção cima-baixo e esquerda-direita, como representado na Figura 3.15(b). O método busca um pixel cujo valor seja menor ou igual a C_{max} , garantindo com relativa certeza que este pixel é parte constituinte do caractere. Assim, esse primeiro pixel visitado torna-se a porta de entrada para o método começar o processo de rotulação.

Para todo pixel observado é realizada uma análise de possibilidades de transição para este pixel baseado na Equação (3.5). Para este exemplo prático, temos:

$$\Delta g = 1,2 \cdot 0,5725 \cdot \left(\frac{g - 0,5725}{0,0549 - 0,5725} \right) = -1,33 \cdot g + 0,76 \quad (3.6)$$

onde g é o valor do nível de cinza de cada pixel observado; e Δg é a variação máxima de intensidade para aceitação de um pixel vizinho como sendo da mesma região do pixel observado. Por exemplo, para $g = 0,2862$, a transição só é possível se o nível de cinza do pixel vizinho analisado estiver entre $-0,0938$ (considera-se 0, pois é menor do que o preto puro) e o limiar de Otsu. Já para $g = 0,4353$, a transição só é possível se o vizinho estiver entre 0,2532 e o limiar de Otsu.

Quando todas as possibilidades de transição forem verificadas, sem que

haja algum pixel com vizinhos “válidos”, a região rotulada é extraída da imagem da palavra original e inserida em uma nova imagem, desta vez binarizada.

Ajuste de pingos e diacríticos

O método gROSC possui uma característica intrínseca no caso de caracteres com sinais diacríticos, como os que temos na língua Portuguesa. A segmentação realizada pelo gROSC é feita orientada às regiões conectadas, de modo que os diacríticos — como os acentos circunflexo (^), agudo (´) e grave (`), bem como o til (~) ou o trema (¨) —, são segmentados isoladamente do caractere que modificam. Os pingos, como o ponto acima das letras minúsculas “i” e “j”, ou mesmo abaixo das pontuações “!” e “?”, são tratados da mesma forma que os sinais diacríticos. Já a cedilha — que é sempre ligada a uma letra — é um diacrítico que não tem essa característica.

Uma forma de lidar com essa característica é adicionar ao pós- processamento do sistema reconhecedor que adota o gROSC a lógica necessária para unir as marcações com suas respectivas letras. Neste trabalho, entretanto, foi decidido por implementar um ajuste ao método para que o mesmo una caracteres e marcações. A regra é bastante simples: se dois caracteres subsequentes possuírem coordenadas horizontais onde uma inclua a outra, o algoritmo implementado une os dois caracteres em um, respeitando as coordenadas originais de cada um.

Efeitos colaterais

Na base de dados utilizada não foram encontrados caracteres fragmentados, mas, após a aplicação do método de segmentação de caracteres alguns caracteres apresentaram este problema. Seria crítico se vários caracte-

res tivessem sido fragmentados, ou particionados, mas um número irrisório apareceu. Dentre 12816 caracteres da base de dados utilizada, somente 3 caracteres apresentaram este tipo de problema. Ou seja, não houve comprometimento da eficiência do método.

3.5 Conclusão

Neste capítulo foram discutidas as etapas da escolha da base de dados, a binarização das imagens utilizadas, bem como apresentados os métodos proposto neste trabalho: SLP-TEO e gROSC.

O método SLP-TEO é aplicado a textos impressos para segmentação de linhas e palavras, trata-se de um método simples e que no que foi proposto cumpre de forma efetiva o seu papel.

Já o método gROSC é aplicado as imagens de palavras de textos impressos em tons de cinza para segmentação de caracteres, seja ele simples, conectado ou sobreposto.

No próximo capítulo será apresentado experimentos e resultados encontrados a partir da aplicação de ambos os métodos, bem como um estudo de caso abordando aplicação do método SLP-TEO em textos manuscritos.

Capítulo 4

Experimentos e Resultados

Nesta seção mostraremos os experimentos realizados com os 40 textos selecionados da base de imagens *IAM-DataBase*. Note que as imagens de textos desta base de imagens são compostas somente por textos, são textos comportados, sem inclinações acentuadas. Essas imagens foram digitalizadas a 300 dpi e não apresentam distorções bruscas de inclinação das linhas.

Os experimentos realizados foram bem sucedidos tanto para os textos impressos quanto para os textos manuscritos. Sendo que os experimentos através dos textos manuscritos foram efetuados somente com o método SLP-TEO, visto que neste trabalho foi proposto a segmentação de linhas e palavras para este tipo de texto como parte de um estudo de caso apenas e não necessariamente por este ser o foco do trabalho. Entretanto, para os textos impressos os experimentos abordam todas as etapas de segmentação, ou seja, segmentação de linhas e segmentação de palavras através do método SLP-TEO e segmentação de caracteres através do método gROSC.

4.1 Textos Impressos

Nesta seção mostraremos alguns experimentos realizados utilizando os textos impressos da base de dados.

Na Tabela 4.1 apresentamos os resultados da etapa de segmentação de linhas e palavras em textos impressos.

Tabela 4.1: Textos Impressos Segmentados pelo Método SLP-TEO

		Nº Linhas	Linhas Segmentadas	Nº Palavras	Palavras Segmentadas
1	A03-011	6	6	78	78
2	A03-063	7	7	85	85
3	A04-010	6	6	69	69
4	A06-004	5	5	66	66
5	C03-000B	5	5	56	56
6	C04-017	5	5	66	66
7	D01-016	4	4	51	51
8	D01-052	4	4	57	57
9	D04-021	4	4	58	58
10	E04-043	5	5	60	60
11	E04-132	3	3	34	34
12	E06-026	4	4	54	54
13	F02-040	8	8	91	91
14	F04-079	7	7	86	86
15	G01-088	8	8	90	90
16	G06-011C	6	6	88	88
17	G06-018M	5	5	70	70
18	G06-042	4	4	51	51
19	G06-047I	5	5	58	58
20	G06-105	4	4	58	58
21	H01-000	6	6	77	77
22	H06-003	6	6	85	85
23	H07-069A	4	4	56	56
24	J04-035	6	6	69	69
25	J06-034	6	6	74	74
26	K04-103	5	5	69	69
27	K07-063A	4	4	62	62
28	L07-187	5	5	68	68
29	M01-121	5	5	65	65
30	M03-033	4	4	56	56
31	M04-000	4	4	57	57
32	M04-190	6	6	78	78
33	M06-106	6	6	82	82
34	N04-075	5	5	61	61
35	N04-084	4	4	58	58
36	N06-133	5	5	76	76
37	P06-047	4	4	59	59
38	R02-013	5	5	70	70
39	R02-131	4	4	61	61
40	R06-143	3	3	46	46
		202	202	2655	2655

Na primeira coluna podemos ver a identificação do texto referente aos formulários de origem da *IAM-DataBase*. Na segunda e terceira colunas pode

ser visto o total de número de linhas do texto em questão, bem como o número de linhas de texto segmentadas pelo método SLP-TEO, respectivamente. Da mesma forma, a quarta e quinta coluna da tabela apresenta o número de palavras do texto e o número de palavras segmentadas pelo método SLP-TEO. Como podemos observar 100% das linhas e palavras dos 40 textos impressos foram segmentadas corretamente pelo método SLP-TEO.

4.1.1 Segmentação de Linhas de Texto Impresso

Como foi mencionado na seção anterior, todas as linhas dos 40 textos impressos constituintes da base de dados foram segmentadas corretamente. Ou seja, 100% de um total de 202 linhas foram segmentadas sem apresentar nenhum erro de segmentação.

Para se chegar a totalidade no número de segmentação de linhas, o texto “d01-052” necessitou de um ajuste de rotação. Como neste trabalho não foi implementada a rotação automática, visto que os textos utilizados não necessitaram deste ajuste, então, para este caso especial foi realizada a rotação manual.

A Figura 4.1 mostra a segmentação de linhas do texto impresso “f02-040” através do método SLP-TEO.

Outro exemplo pode ser visto na Figura 4.2 onde pode ser visualizado o texto “a06-004”.

A segmentação em textos impressos é bem mais simples se comparada a segmentação em textos manuscritos. Neste tipo de texto as linhas se encontram uniformemente separadas, o que caracteriza a facilidade neste processo. Isso, não acontece com os textos manuscritos que são mais diversificados, pois cada um “carrega” as características da escrita do seu autor.

As a result, the Glasgow Retirement Council came into being in April, 1958, with Dr. Andrew Hood as chairman and Mr. Andrew Atkinson as secretary. The Council has active committees on Education and Preparation for Retirement, and on Occupational Centres. The former consists of representatives from the Glasgow Corporation Further Education Department; the Workers' Educational Association; the University Extra-Mural Education Committee and departments of psychology, education and social science; the trades unions; and the Regional Hospital Board, together with an industrial medical officer and a Medical Officer of Health.

Figura 4.1: Segmentação de linhas do texto impresso “f02-040”

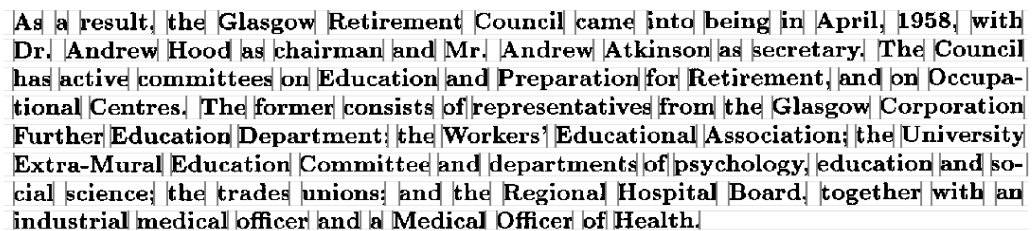
Sir Edward Boyle, Financial Secretary to the Treasury, said yesterday that the Government does not like retrospective legislation. He went on to describe the kind of people the Chancellor hopes to tax. These are people buying and selling shares within a short period, those “stagging” on a new issue and property dealers who form a chain of companies and put one property deal through each.

Figura 4.2: Segmentação de linhas em texto impresso “a06-004”

4.1.2 Segmentação de Palavras Impressas

Quanto a segmentação de palavras em textos impressos, essa etapa se mostrou muito satisfatória visto que não encontramos problemas com a segmentação deste nível em textos impressos. De um total de 2655 palavras que compõe os 40 textos selecionados todas foram segmentadas corretamente, ou seja, 100% das palavras.

A Figura 4.3 mostra as 8 linhas do texto “f02-040”, onde podemos ver as palavras que constituem essas linhas segmentadas através do método SLP-TEO.



As a result, the Glasgow Retirement Council came into being in April, 1958, with Dr. Andrew Hood as chairman and Mr. Andrew Atkinson as secretary. The Council has active committees on Education and Preparation for Retirement, and on Occupational Centres. The former consists of representatives from the Glasgow Corporation Further Education Department, the Workers' Educational Association, the University Extra-Mural Education Committee and departments of psychology, education and social science, the trades unions, and the Regional Hospital Board, together with an industrial medical officer and a Medical Officer of Health.

Figura 4.3: Imagem da palavras segmentadas nas 8 linhas do texto impresso “f02-040”.

Note que através do método é possível delimitar corretamente as palavras, sem nenhum cálculo de distâncias, mas somente utilizando o operador de energia de Teager. Cada palavra segmentada, foi em seguida submetida ao método gROSC para segmentação de seus caracteres. As coordenadas da imagem da palavra foi utilizada para obtenção da imagem em escala de cinza, pois o método gROSC utiliza este tipo de imagem.

4.1.3 Segmentação de Caracteres Impressos

Neste trabalho, a segmentação de caracteres consiste em separar cada caractere constituinte das palavras em letras, dígitos e caracteres especiais como pontuação, aspas, etc.

Para cada imagem de palavra, foi calculado o valor do limiar de Otsu(T) correspondente àquela palavra. A partir do método de Otsu assumimos que todos os pixels que possuem valores maiores que o limiar definido por este método, são considerados pixels que pertencem ao fundo da imagem. Assim, todos os pixels que apresentam valores maiores que o limiar de Otsu, não são visitados pelo método, visto que já são definidos como parte do fundo da imagem.

O método desenvolvido realiza uma análise sobre a possibilidade de transição entre níveis de cinza dos pixels da imagem de cada palavra. Com base na Equação 3.5, temos a variação máxima de nível de cinza Δg para cada pixel observado. Desta forma, é possível verificar se o valor de um pixel vizinho está ou não dentro do intervalo definido por $[g - \Delta g; g + \Delta g]$. Se o valor deste pixel se enquadrar no intervalo, este será visitado e incorporado à região a ser segmentada. Caso contrário, não será rotulado como constituinte da região de interesse, e nem terá seus vizinhos analisados. Em seguida, os vizinhos deste pixel são analisados semelhantemente.

A Tabela 4.2 mostra os 3 tipos de problemas encontrados na base de dados utilizada. Note que a base de dados contém um total de 12.816 caracteres. Sendo que destes, houve 32 casos de caracteres conectados o que compromete a segmentação de 64 caracteres, visto que a conexão entre eles envolve dois caracteres. Quanto aos caracteres sobrepostos o número de casos de sobreposição é de 219, o que corresponde a 445 caracteres prejudicados, visto que existem sobreposições de dois e três caracteres. Finalmente, os caracteres desenho de fontes totalizam 46 casos, comprometendo 93 caracteres. A partir da segmentação pelo método gROSC, observou-se que o número de caracteres conectados segmentados corretamente corresponde a 71,9% do total de caracteres conectados encontrados na base utilizada. Já os caracteres sobrepostos deixaram de existir a partir da aplicação do método gROSC.

Tabela 4.2: Resultados de Segmentação de Caracteres pelo Método gROSC

	Conectados	Sobrepostos	Desenho de Fonte
Total	64	445	93
Segmentados	46	445	0
Recall	71,9%	100,0%	0,0%

Como já havia sido mencionado anteriormente, os caracteres desenhos de fontes não são considerados, nesta pesquisa, como erro de digitalização. Portanto, como era de se esperar, para estes tipos de caracteres não houve modificações.

Houve um surgimento de um novo problema após a aplicação do método gROSC, apareceram 3 caracteres fragmentados na base de dados. Mas quando comparamos o máximo do intervalo com o mínimo da média de caracteres por texto, temos no máximo 0,06% de caracteres fragmentados. Ou seja, não compromete a eficiência do método.

Além disso, considerando o percentual de caracteres segmentados pelo método gROSC que totalizam 12703 caracteres, temos 99,10% de caracteres segmentados corretamente.

Nas seções seguintes passaremos a detalhar alguns exemplos de sucesso do método e também as limitações em alguns casos onde não houve solução do problema apresentado.

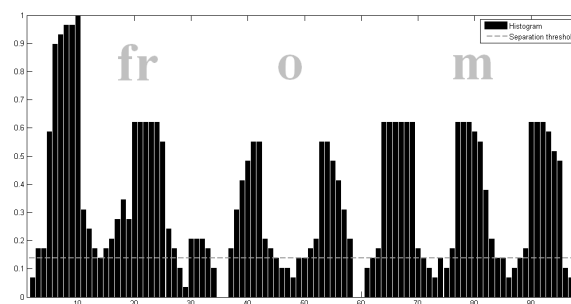
Caracteres Sobrepostos Segmentados

Os caracteres sobrepostos foram identificados a partir da projeção vertical aplicada a palavra segmentada através do SLP-TEO. Foram contabilizados 445 casos de caracteres sobrepostos em um total de 12816 caracteres da base de dados utilizada.

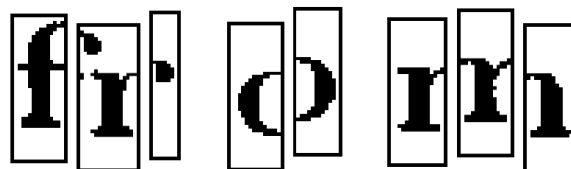
A Figura 4.4(a) apresenta a palavra “from” onde há a presença de caracteres conectados. Note que o caractere “f” se sobrepõe ao caractere “r”. Percebe-se que a projeção vertical mostrada Figura 4.4(b) detalha bem este problema. Podemos ver através do histograma apresentado que existe “vales” que correspondem aos espaços entre os caracteres e os “picos” que correspondem as letras da palavra. Note que para as letras “f” e “r” não há esta separação. Neste caso, se utilizássemos um limiar (linha pontilhada) para separar estas duas letras, assim é mostrado na imagem do histograma, conseqüentemente, a solução destes dois caracteres ocasionaria a fragmentação dos demais caracteres desta palavra. A Figura 4.4(c) simula esta segmentação através de um limiar capaz de separar os caracteres “f” do “r”.



(a) Imagem binária da palavra “from”.



(b) Projeção vertical da palavra “from”.



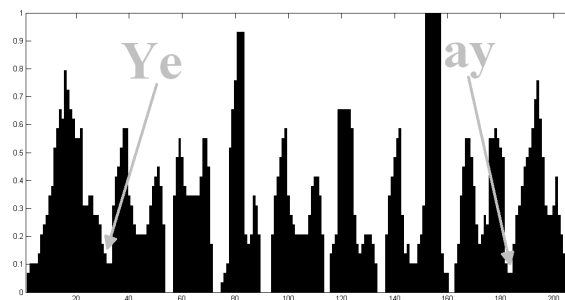
(c) Simulação de segmentação através de um limiar utilizando a projeção vertical.

Figura 4.4: Segmentação da Palavra “from”.

Outro exemplo de palavra que possui caracteres sobrepostos pode ser visto na Figura 4.5(a), onde podemos ver a palavra “Yesterday”. Esta palavra apresenta duas sobreposições, sendo a primeira envolvendo os caracteres “y” e “e” e a segunda envolve os caracteres “a” e “y”. Como pode ser observado através da projeção vertical da palavra binarizada, os espaços entre os caracteres não possuem concentração de pixels, representados pelos “vales” no gráfico da Figura 4.5(b). Já nos caracteres sobrepostos isso não acontece.

Yesterday

(a) Palavra “yesterday”.



(b) Projeção vertical da palavra “yesterday”.

Figura 4.5: Palavra que apresenta dois casos de caracteres sobrepostos.

Para este tipo de problema um método de segmentação baseado em regiões utilizando imagem binária já seria suficiente.

Através da segmentação através do gROSC foi possível resolver todos os problemas de caracteres sobrepostos que havia nos textos utilizados nos experimentos. Ou seja, 445 que seria comprometidos por este problema foram segmentados corretamente.

Caracteres Conectados

Os caracteres conectados são considerados problemas complexos e não há uma maneira única para solução de todos os casos. O método gROSC apresentou bons resultados para todos os tipos de caracteres, e também para o caso caracteres conectados. Através de uma abordagem genérica podemos resolver a maior parte dos casos presentes nos textos da base de dados utilizada.

A seguir, descreveremos alguns dos casos solucionados pelo método gROSC, fazendo um detalhamento de cada caso apresentado.

1. Primeiro Exemplo palavra “necessary”

A Figura 4.6(a) mostra a imagem binarizada da palavra “necessary”, note que os caracteres “a” e “r” aparecem conectados nesta imagem. O método de binarização considerou os pixels que estão entre estes os dois caracteres como parte de um mesmo caractere e não dois como é de fato, como pode ser visto na Figura 4.6(b). A Figura 4.6(c) mostra a posição exata no mapa de pixels da imagem em tons de cinza onde ocorre o problema de conexão entre os caracteres.

Como o método gROSC é aplicado a imagem em tons de cinza e não imagem binarizada, um estudo desta área é realizado para que não haja conexão dos caracteres “a” e “r”.

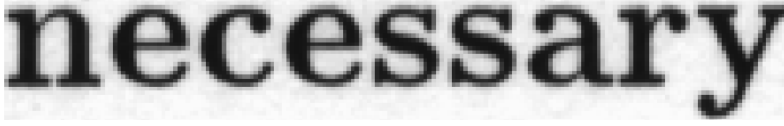
A Tabela 4.3 mostra os valores utilizados pelo método para as inferências necessárias no estudo da palavra “necessary”.

Tabela 4.3: Intervalos para transição entre pixels da palavra “necessary”

Pixel Observado	Variação Máxima (Δg)	Intervalo
0, 17	0, 55	[0, 00; 0, 57]
0, 34	0, 32	[0, 02; 0, 56]
0, 56	0, 02	[0, 54; 0, 57]



(a) Imagem binária da palavra “necessary”.



(b) Imagem em tons de cinza da palavra “necessary”.



(c) Mapa de pixel da região dos caracteres “ar” da palavra “necessary”.

Figura 4.6: Análise da vizinhança dos pixels da palavra “necessary”.

O método gROSC rastreia toda a imagem da palavra e quando acha o pixel ativo (primeiro pixel do caractere), ele então, visita todos os vizinhos e os rotula como pertencentes a região de interesse. Quando o método gROSC visita o pixel 0,34 (chamado pixel observado) são realizados os cálculos da variação máxima de níveis de cinza Δg para este pixel, neste caso, $\Delta g = 0,32$. Assim, a partir deste valor é possível definir o intervalo $[g - \Delta g; g + \Delta g]$, onde g é o menor nível de cinza da imagem. Através deste intervalo, o método verifica se é possível visitar ou não o vizinho do pixel que está sendo analisado. Se for possível ele é visitado e rotulado como parte da região do caractere.

Para a imagem da palavra “necessary”, o limiar de Otsu encontrou o

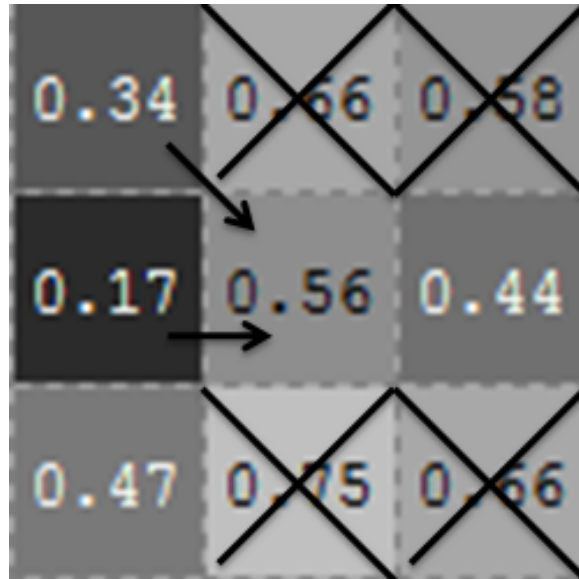


Figura 4.7: Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “necessary”.

valor de $T = 0,57$. Assim, todos os pixels nessa região que apresentam valores maiores que esse limiar não são visitados, pois o método de binarização já os considera como parte do fundo da imagem. Por isso, os marcamos com “X”, sinalizando que estes pixels não serão visitados. A Figura 4.7 mostra a análise detalhada da região, e o caminho escolhido pelo método gROSC para a solução do problema de conexão entre os pixels “a” e “r”. Veja que o pixel observado de com valor de nível de cinza 0,34 tem a possibilidade de visitar os pixels com valores 0,17 e 0,56. Quando o método visita o pixel 0,56, ele o rotula como parte da região de interesse. Por isso, quando o pixel 0,17 se torna o observado ele até pode visitar o 0,56, mas o pixel com valor 0,56 já está rotulado como visitado. Mas quando o método alcança o pixel 0,56, podemos ver que não há possibilidades de transição deste pixels para os seus vizinhos, visto que ele só pode visitar pixels que estejam delimitados pelo intervalo $[0,54; 0,57]$ e os seus vizinhos não correspondem a estes valo-

res. Assim, o método gROSC termina a execução da função recursiva e extrai o pixel “a” neste ponto.

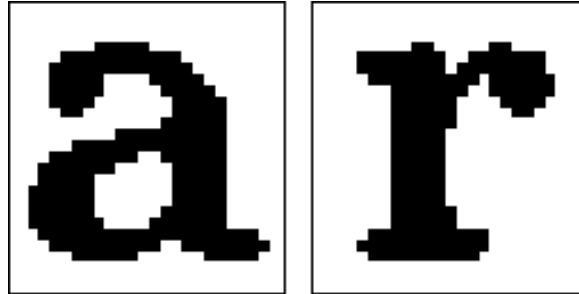


Figura 4.8: Caracteres “a” e “r” segmentados pelo método gROSC.

A Figura 4.8 mostra a segmentação de “a” e “r” da palavra “necessary” pelo método gROSC. Note que a imagem dos caracteres segmentados pelo método gROSC corresponde ao caminho escolhido pelo método no processo de segmentação destes caracteres.

2. Exemplo palavra “was”

A Figura 4.9(a) mostra a imagem binarizada da palavra “was”, note que os caracteres “a” e “s” se mostram conectados pelo método de binarização. Já na imagem em tons de cinza é possível ver que estes caracteres são disjuntos como é mostrado na Figura 4.9(b).

A Figura 4.9(c) apresenta o mapa de pixels da imagem em tons de cinza dessa área afetada. O método gROSC é aplicado a essa imagem e um estudo desta área é realizado para que não haja conexão desses dos caracteres “a” e “s”.

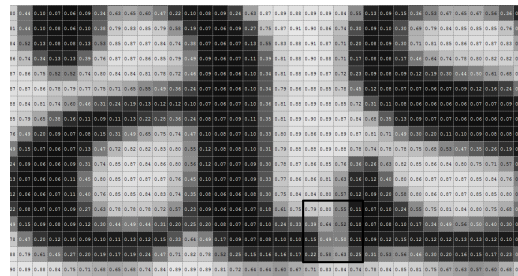
Para a imagem em questão, o limiar de Otsu possui o valor $T = 0,54$. Assim, todos os pixels nessa região que apresentam valores maiores que esse limiar não são visitados, pois o método de binarização já os considera como parte do fundo da imagem.



(a) Imagem da palavra “was” binarizada.



(b) Imagem em tons de cinza da palavra “was”.



(c) Mapa de pixel do caractere “as” da palavra “was” segmentada pelo método gROSC.

Figura 4.9: Análise da vizinhança dos pixels da palavra “was”.

Quando o método gROSC alcança o pixel 0,39(pixel observado), neste momento, são realizado os cálculos da variação máxima de tons de cinza $\Delta g = 0,19$ para este pixel. A partir desta variação máxima, é definido o intervalo de transição entre os tons de cinza.

A Tabela 4.4 mostra os valores utilizados pelo método para as inferências necessárias no estudo da palavra “was”.

Tabela 4.4: Intervalos de níveis de cinza para transição entre pixels da palavra “was”

Pixel Observado	Variação de Nível de Cinza	Intervalo Transição
0,39	0,19	[0, 20; 0, 54]
0,15	0,53	[0, 00; 0, 54]
0,49	0,06	[0, 43; 0, 54]
0,50	0,05	[0, 45; 0, 54]
0,52	0,02	[0, 49; 0, 54]

A Figura 4.10 mostra a análise detalhada da região, e o caminho escolhido pelo método gROSC para a solução do problema de conexão entre os pixels “a” e “s”.

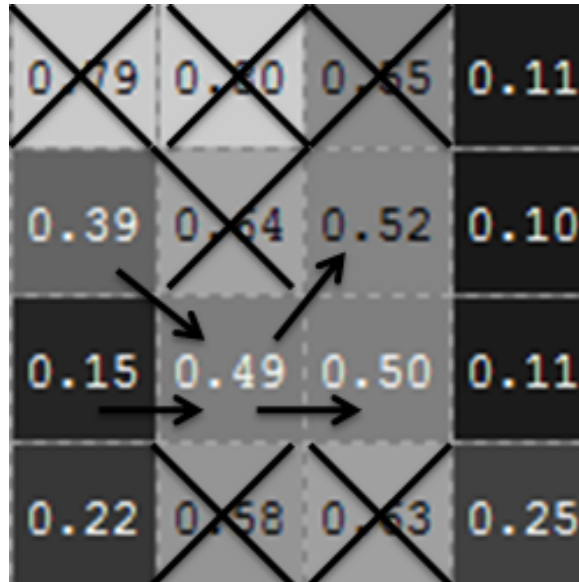


Figura 4.10: Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “was”.

Veja que o pixel observado possui valor igual a 0,39, neste caso ele poderá visitar os pixels 0,15 e 0,49. Quando o pixel observado passa a ser o pixel 0,49, este tem a possibilidade de visitar os pixels 0,50 e 0,52. Note que quando o método gROSC visita o pixel 0,52, ele tenta visitar os pixels vizinhos, mas a possibilidade de busca foi encerrada, pois os vizinhos deste pixel não se enquadram nos limites apresentados. Consequentemente, o método gROSC termina a execução da função recursiva e extrai o pixel “a” neste ponto.

A Figura 4.11 mostra o resultado da segmentação pelo método gROSC. Note que a imagem dos caracteres segmentados pelo método gROSC corresponde ao caminho escolhido pelo método no processo de segmentação destes caracteres.

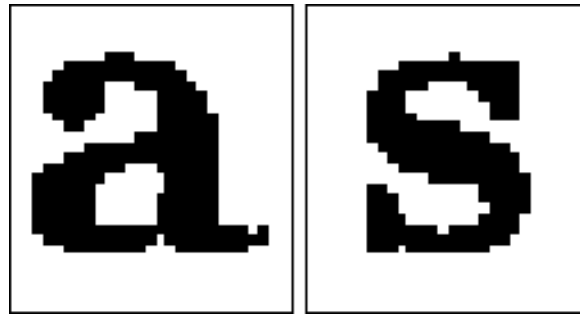


Figura 4.11: Caracteres “a” e “s” da palavra “was” segmentados pelo método gROSC.

Observe que a extensão do caractere “a” é equivalente a decisão do método gROSC.

3. Exemplo palavra “afraid”

A Figura 4.12(a) mostra a imagem binarizada da palavra “afraid”, a imagem original e o mapa de pixel da região onde ocorre a conexão entre os caracteres “a” e “f”.

A Tabela 4.5 mostra os valores utilizados pelo método para as inferências necessárias no estudo da palavra “afraid”.

Tabela 4.5: Intervalos para transição entre pixels da palavra “afraid”

Pixel Observado	Variação de Nível de Cinza	Intervalo Transição
0,29	0,3	[0, 00; 0, 54]
0,50	0,05	[0, 45; 0, 54]

Para a imagem da palavra em questão, o limiar de Otsu apresenta o valor $T = 0,54$. Quando o método gROSC visita o pixel com valor 0,29 e este se torna o pixel observado é realizado os cálculos da variação máxima de tons de cinza $\Delta g = 0,30$.

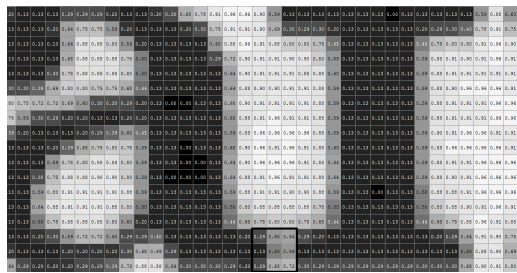
A Figura 4.13 mostra a análise detalhada da região entre os pixels “a” e “f”, e o caminho escolhido pelo método gROSC para a solução do problema de conexão entre estas duas letras.



(a) Imagem da palavra “afraid” binarizada.



(b) Imagem em tons de cinza da palavra “afraid”.



(c) Mapa de pixel dos caracteres “A” e “f” da palavra “afraid”.

Figura 4.12: Análise da vizinhança dos pixels da palavra “afraid”.

O pixel observado com valor de nível de cinza 0, 29 tem a possibilidade de visitar os pixels 0, 13 e 0, 50. Quando o método gROSC visita o pixel 0, 50, ele tenta visitar outros pixels mas a possibilidade de busca foi encerrada, pois os valores de tons de cinza dos vizinhos deste pixel não se enquadram nos limites apresentados na Tabela 4.5. Além disso, como podemos ver na Figura 4.13 os vizinhos do pixel 0, 50 são todos maiores que o valor do limiar de Otsu, ou seja já são considerados como fundo da imagem, por isso não são visitados. Consequentemente, o método gROSC termina a execução da função recursiva e extrai o pixel “a” neste ponto.

4. Exemplo palavra “remarkable”

A palavra “remarkable” apresenta duas conexões entre seus caracteres.



Figura 4.13: Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “afraid”.

Na Figura 4.14(a) podemos observar essas duas conexões através da imagem binarizada da palavra “remarkable”.

A primeira conexão envolve os caracteres “ar” e a segunda os caracteres “ka”. Aqui iremos detalhar a solução da segunda conexão, e na seção 4.1.4, será apresentada a análise da primeira conexão, pois esta não foi solucionada pelo método gROSC.

A Figura 4.14(b) mostra a imagem em tons de cinza palavra “remarkable”, e o mapa de pixel da região onde ocorre a conexão dos dois caracteres “k” e “a”. Observando a imagem em tons de cinza vemos claramente que se trata de dois caracteres facilmente separados, mas o método de binarização entendeu que se tratavam de um caractere e não dois, como de fato se verifica.

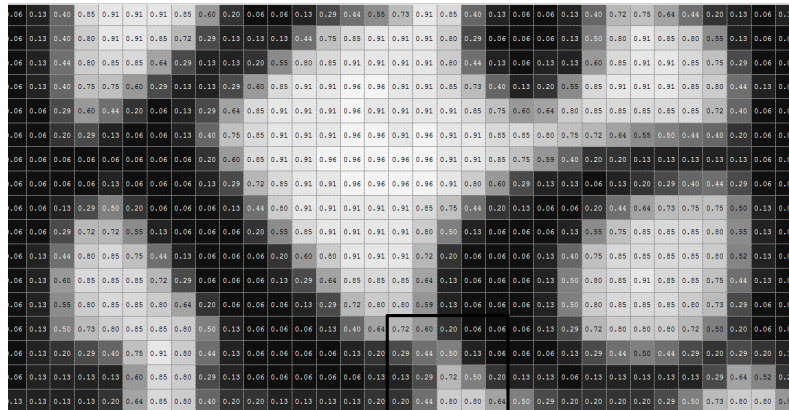
Para a palavra analisada, o valor do limiar de Otsu encontrado é $T = 0,57$. Portanto, todos os pixels nessa região que apresentam valores

remarkable

(a) Imagem binarizada da palavra “remarkable” segmentada pelo método gROSC.

remarkable

(b) Imagem em tons de cinza da palavra “remarkable” segmentada pelo método gROSC.



(c) Mapa de pixel da região do caractere “ka” da palavra “remarkable”.

Figura 4.14: Análise da vizinhança dos pixels da palavra “remarkable”.

maiores que esse limiar não são visitados.

Quando o método gROSC alcança o pixel 0,29 (pixel observado), são realizados os cálculos da variação máxima para este valor de pixel, ou seja Δg onde neste caso, $\Delta g = 0,37$.

A Tabela 4.6 mostra os valores utilizados pelo método para as inferências necessárias no estudo da palavra “remarkable”.

Tabela 4.6: Intervalos para transição entre pixels da palavra “remarkable”

Pixel Observado	Variação de Nível de Cinza	Intervalo Transição
0,29	0,37	[0, 00; 0, 57]
0,44	0,18	[0, 26; 0, 57]
0,50	0,09	[0, 41; 0, 57]

A Figura 4.15 mostra a análise detalhada da região, e o caminho escolhido pelo método gROSC para a solução do problema de conexão entre os pixels “k” e “a”. Veja que o pixel observado, de nível de cinza 0,29, tem a possibilidade de visitar os pixels 0,13, 0,29 e 0,44. Note que, tanto para o pixel 0,29 quanto 0,44 é possível visitar o pixel 0,50. Entretanto, quando o método visita o pixel 0,50, o método chega ao seu limite, visto que não encontra nos vizinhos do pixel 0,50 ninguém semelhante (que possui valor delimitado pelo intervalo de transição deste pixel). Os vizinhos do pixel 0,50 possuem valor bem próximos a 0 e como a variação máxima para o pixel 0,50 não existe a possibilidade de transição entre estes pixels. Assim, o método gROSC termina a execução da função recursiva e extrai a região interconectada ou seja, o caractere “k”

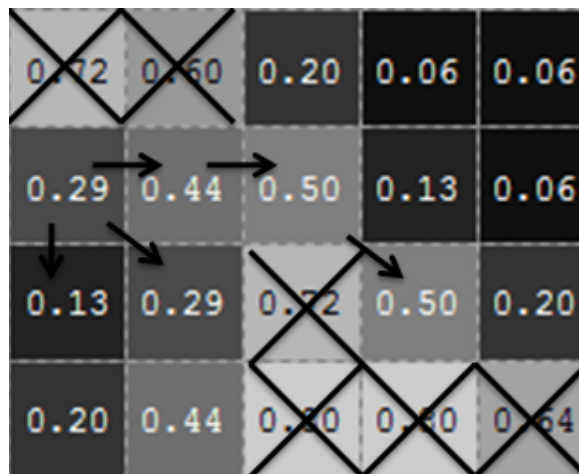


Figura 4.15: Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “remarkable”.

A Figura 4.16 mostra o resultado da segmentação pelo método gROSC. Note que a imagem dos caracteres segmentados pelo método gROSC corresponde ao caminho escolhido pelo método no processo de segmentação destes caracteres.

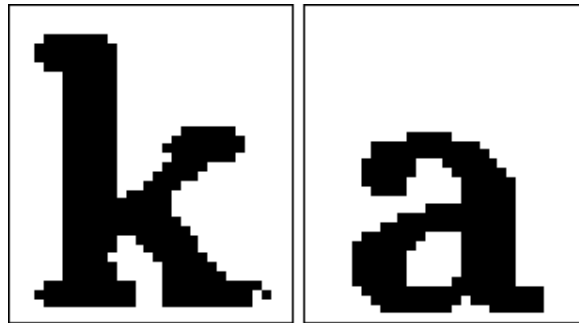


Figura 4.16: Caracteres “k” e “a” da palavra “remarkable” segmentados pelo método gROSC.

Na seção 4.1.4 detalharemos alguns casos de caracteres conectados que não puderam ser segmentados pelo método gROSC, bem como os três caracteres que sofreram fragmentação.

4.1.4 Casos Especiais

Os casos especiais descritos nesta seção trata dos caracteres que não foram segmentados corretamente pelo método gROSC. Estes caracteres sem solução totalizam 21 caracteres “perdidos” no total de 12816 caracteres da base de dados utilizada.

Assim como foi visto na seção anterior, aqui faremos um detalhamento da análise que foi executada pelo método na região dos caracteres que apresentaram estes problemas de segmentação. Primeiramente, trataremos dos casos de caracteres fragmentados e em seguida de caracteres que permaneceram conectados.

Caracteres Particionados ou Fragmentados

Considera-se caracteres particionados ou fragmentados, aqueles caracteres que, no momento da segmentação pelo método gROSC, sofreram uma perda

de características em relação ao caractere original.

No processo de segmentação pelo método gROSC, 3 caracteres foram fragmentados, portanto a seguir faremos um detalhamento de dois casos de palavras que apresentaram esses tipo de problema em seus caracteres.

1. Exemplo palavra “frequent”

A palavra “frequent” foi segmentada corretamente pelo método SLP-TEO, mas na segmentação dos seus caracteres pelo método gROSC, o caractere “n” sofreu uma fragmentação. O método dividiu o caractere “n” em dois caracteres, quando na verdade na imagem original é possível ver que este problema não existia até a aplicação do método.

A Figura 4.17(a) mostra a imagem da palavra em tons de cinza. Já na Figura 4.17(b) é apresentada o caractere “n” e a região do caractere onde ocorreu a fragmentação.



(a) Imagem em tons de cinza da palavra “frequent”.



(b) Mapa de pixel do caractere “n” da palavra “frequent”.

Figura 4.17: Análise da vizinhança dos pixels da palavra “frequent”.

A Tabela 4.7 mostra os valores utilizados pelo método para as inferências necessárias no estudo da palavra “frequent”.

Tabela 4.7: Intervalos para transição entre pixels da palavra “frequent”

Pixel Observado	Variação de Nível de Cinza	Intervalo Transição
0,40	0,23	[0, 17; 0, 57]
0,44	0,18	[0, 25; 0, 57]
0,46	0,15	[0, 31; 0, 57]

A Figura 4.18 apresenta o caminho escolhido pelo método através da análise da vizinhança dos pixels do caractere “n”.

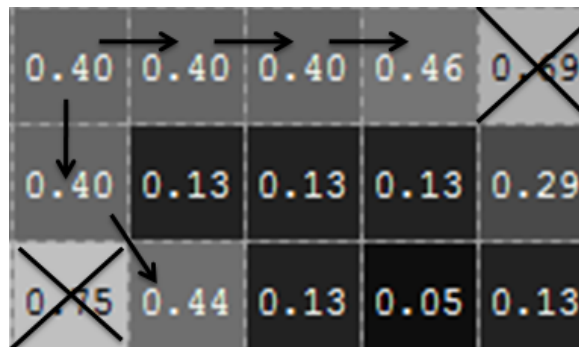


Figura 4.18: Caminho de decisão do método para o caractere “n”.

Nesta imagem o valor do limiar de Otsu é $T = 0.57$, portanto, os pixels 0.75 e 0.69 não são analisados, pois o próprio método de Otsu já os considera como parte do fundo da imagem e não do caractere. Como podemos ver, o pixel observado, com nível de cinza igual a 0.40, só tem a opção de visitar os pixels 0.44 e 0.46, devido ao intervalo de transição [0.17; 0.57]. Quando analisamos o pixel 0.46 percebemos através do intervalo encontrado [0.31; 0.57] que na vizinhança não há nenhum pixel dentro desse valor, pois os seus vizinhos são os pixels 0, 13, 0, 29 e 0, 69. Desta forma o método gROSC encerra a sua busca e extrai a região interconectada sem terminar de segmentar o caractere “n”.

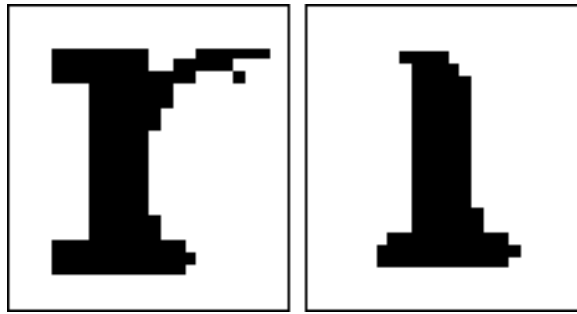


Figura 4.19: Resultados da segmentação incorreta do caractere “n” da palavra “frequent”.

Neste caso, a primeira parte extraída pelo método é a parte inicial do caractere “n”, como pode ser visto na Figura 1. Em seguida, a função recursiva do método gROSC é novamente re-iniciada, após a identificação do pixel ativo que inicia o processo de segmentação e varredura das coordenadas para formar a região, assim cada pixel é analisado e então ao final ele retorna a região correspondente a segunda parte do caractere “n”.

2. Palavra “employment”

A segmentação da palavra “employment” através do método gROSC ocasionou uma fragmentação do segundo caractere “m”.

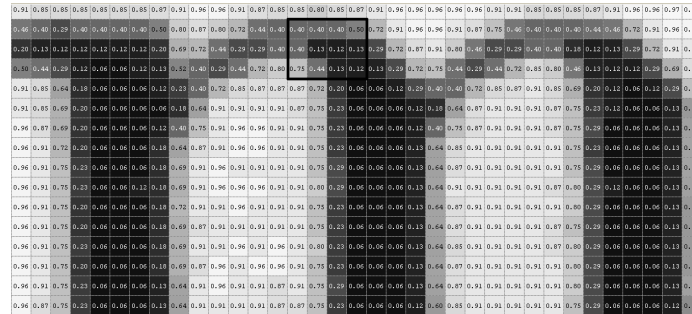
Como podemos ver na Figura 4.20(a), a imagem da palavra em tons de cinza. Já na Figura 4.20(b) é apresentado o caractere “m” e a região do caractere onde ocorreu a fragmentação.

A Tabela 4.8 mostra os valores utilizados pelo método para as inferências necessárias no estudo da palavra “employment”.

A Figura 4.21 apresenta o caminho escolhido pelo método através da análise da vizinhança dos pixels do caractere “m” da palavra “employment”.



(a) Imagem em tons de cinza da palavra “employment”.



(b) Mapa de pixel do caractere “m” da palavra “employment”.

Figura 4.20: Análise da vizinhança dos pixels da palavra “employment”.

Tabela 4.8: Intervalos para transição entre pixels da palavra “employment”.

Pixel Observado	Variação de Nível de Cinza	Intervalo Transição
0,40	0,23	[0,17 0,57]
0,44	0,19	[0,25 0,57]
0,50	0,09	[0,41 0,57]

A Figura 4.22 mostra o resultado da segmentação do caractere “m” pelo método gROSC.

Caracteres Conectados

Os caracteres conectados são os caracteres que foram segmentados unindo-se dois caracteres distintos como se fossem um único caractere. Nos experimentos houve 9 casos de caracteres conectados, por isso os caracteres comprometidos com este erro de segmentação somam 18 caracteres, visto que a conexão envolve dois caracteres.

1. Palavra “remarkable”

A Figura 4.23(a) mostra a imagem binarizada da palavra “remarkable”

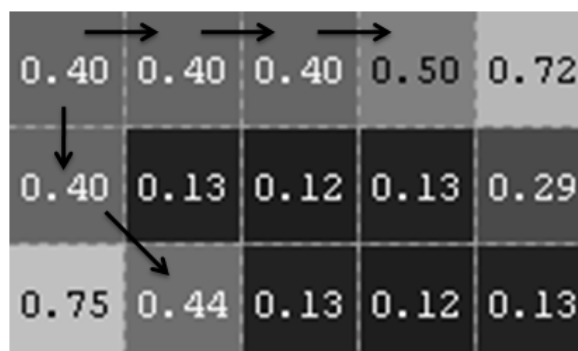


Figura 4.21: Caminho de decisão do método para o caractere “m” da palavra “employment”.

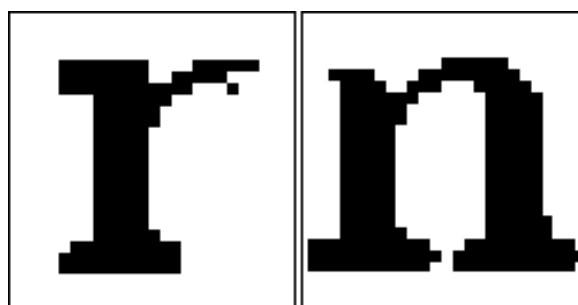


Figura 4.22: Resultados da segmentação da palavra “employment”.

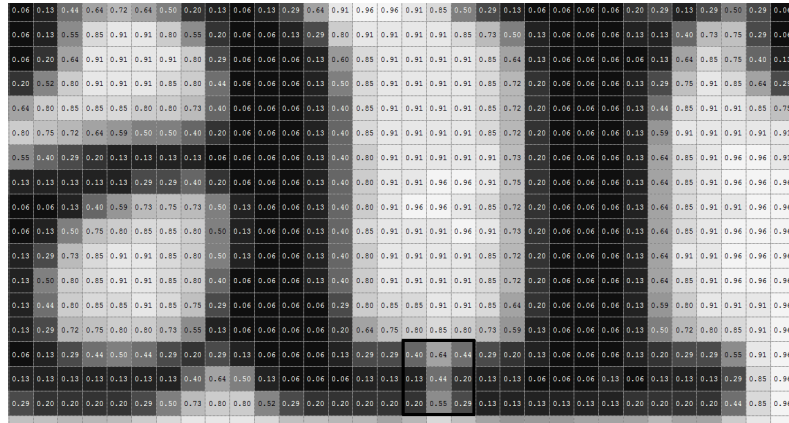
em que podem ser vistos os caracteres conectados “ar” e “ka”. Sendo que o segundo caractere conectado já foi detalhado na seção 4.1.3. Já a Figura 4.23(b) mostra o mapa de pixel da região onde ocorre a conexão dos dois caracteres “a” e “r”. Observando a imagem em tons de cinza vemos claramente que se trata de dois caracteres separados, mas o método de binarização entendeu que se tratavam de um caractere e não dois, como de fato se verifica.

Para a imagem em questão, o limiar de Otsu possui o valor $T = 0,57$. Assim, todos os pixels nessa região que apresentam valores maiores que esse limiar não são visitados, pois o método de binarização já os considera como parte do fundo da imagem.

Quando o método gROSC alcança o pixel 0,40 (pixel observado), é

remarkable

(a) Imagem binarizada da palavra “remarkable” segmentada pelo método gROSC.



(b) Mapa de pixel da região do caractere “ar” da palavra “remarkable”.

Figura 4.23: Análise da vizinhança dos pixels da palavra “remarkable”.

realizado os cálculos da variação máxima para este pixel, neste caso, $\Delta g = 0,22$. A partir da máxima variação foi calculado os intervalos de transição para este pixel.

A Tabela 4.9 mostra os valores utilizados pelo método para as inferências necessárias no estudo da palavra “Remarkable”.

Tabela 4.9: Intervalos para transição entre os pixels dos caracteres “a” e “r” da palavra “remarkable”.

Pixel Observado	Variação de Nível de Cinza	Intervalo Transição
0,40	0,22	[0, 18; 0, 57]
0,44	0,18	[0, 26; 0, 57]

Quando o método visita o pixel com tons de cinza igual a 0,40 as possibilidades de transição entre tons de cinza são entre [0, 18; 0, 57], neste caso, ele pode visitar somente o pixel 0,44. Assim, o pixel observado passa a ser o pixel com valor 0,44. Para este pixel o intervalo

encontrado é $[0, 26 - 0, 57]$, portanto o método pode visitar os pixels 0, 44 e 0, 29, só que estes pixel já fazem parte do caractere “r”. Consequentemente, todo o caractere “r” será rastreado quando um dos seus pixels são visitados e as condições para segmentação de toda a região é propícia. Por isso, que o método gROSC não solucionou este caso.

A Figura 4.24 mostra a análise detalhada da região, e o caminho escolhido pelo método gROSC para a segmentação do caractere “a”.

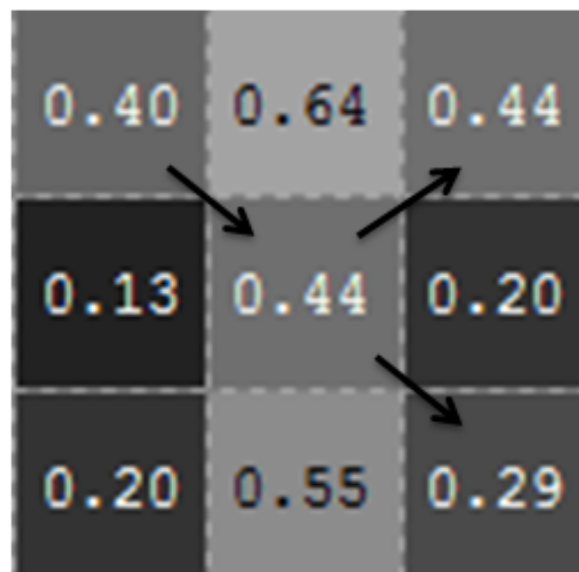


Figura 4.24: Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “remarkable”.

Portanto, o método gROSC só termina a execução da função recursiva quando ele termina de visitar os pixels correspondentes ao caractere “r” e em seguida extrai a região interconectada, neste caso os caracteres conectados “ar” como mostrado na Figura 4.25.

2. Exemplo da palavra Haris

A Figura 4.26(a), mostra a imagem em tons de cinza da palavra “haris”.

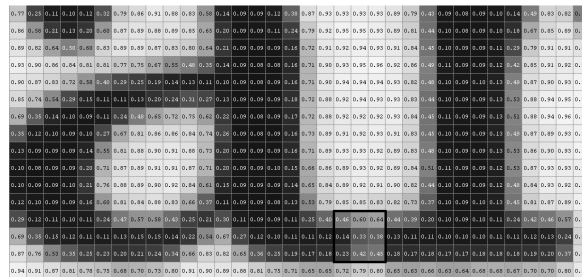
Quando o método gROSC alcança o pixel 0, 14(pixel observado) é rea-



Figura 4.25: Caracteres conectados “ar” da palavra “remarkable” segmentados pelo método gROSC.



(a) Imagem binarizada da palavra “haris” segmentada pelo método gROSC.



(b) Mapa de pixel da região do caractere “ar” da palavra “haris”.

Figura 4.26: Análise da vizinhança dos pixels da palavra “haris”.

lizado o cálculo da máxima variação entre tons de cinza para o pixel, neste caso $\Delta g = 0,57$.

A Tabela 4.10 mostra os valores utilizados pelo método para as inferências necessárias no estudo da palavra “haris”.

A Figura 4.27 mostra a análise detalhada da região, e o caminho escolhido pelo método gROSC para a segmentação do caractere “a”.

Quando o método visita o pixel 0,14 a função recursiva deveria parar

Tabela 4.10: Intervalos para transição entre pixels da palavra “haris” na conexão dos caracteres “a” e “r”.

Pixel Observado	Variação de Nível de Cinza	Intervalo Transição
0,14	0,57	[0,00; 0,55]
0,33	0,3	[0,03; 0,55]
0,42	0,18	[0,24; 0,55]

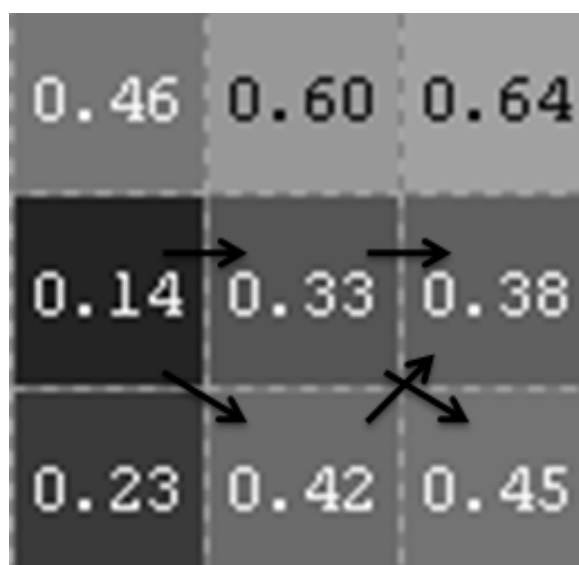


Figura 4.27: Caminho escolhido pelo método gROSC para segmentação dos caracteres da palavra “haris”.

a busca por vizinhança para que o caractere “a” fosse segmentado corretamente, mas isso não acontece. Como podemos observar na Tabela 4.10 o pixel que apresenta nível de cinza igual a 0,14 possibilita que o método visite os seus vizinhos de nível de cinza que se enquadram no limite $[0,00; 0,55]$. Portanto, o pixel com nível de cinza igual a 0,33 é alcançado pelo método. Considerando o pixel 0,33, sua variação de níveis de cinza possíveis é de $[0,03; 0,55]$. Assim, os pixels pertencentes ao caractere “r” também são visitados. Por isso, o método gROSC só termina a execução da função recursiva quando ele termina de visitar os pixels correspondentes ao caractere “r” e em seguida extrai a região interconectada, neste caso, os caracteres conectados “ar”, como mostrado na Figura 4.28.



Figura 4.28: Caracteres conectados “ar” da palavra “haris” segmentados pelo método gROSC.

4.2 Estudo de Caso-Textos Manuscritos

Nesta seção descreveremos os testes realizados com os textos manuscritos. Estes testes fazem parte de um estudo de caso envolvendo a aplicação do método SLP-TEO. Como mencionamos anteriormente, os resultados encontrados para os textos impressos foram bastante relevante, portanto, resolvemos aplicar o mesmo método sem nenhuma alteração prévia ao textos

manuscritos para segmentação de linhas e palavras.

A Tabela 4.11 mostra os resultados da segmentação dos textos manuscritos em linhas de texto e palavras. Note que os textos manuscritos da base utilizada apresentam linhas bem comportadas, por isso somente dois textos apresentaram problemas na segmentação de linhas através do SLP-TEO. No entanto, isso não acontece da mesma forma com as palavras. Podemos ver que todos os textos apresentaram mais de 50% de suas linhas segmentadas corretamente, mas o texto “G01-088” apresentou um percentual de 43,24% de palavras segmentadas corretamente. Por outro lado, podemos perceber que alguns textos apresentaram 100% de aproveitamento, o que não é muito frequente em segmentação de palavras.

Cada texto tem a sua singularidade e devido a isso, a segmentação neste tipo de texto tende a ser mais complexa que nos textos impressos. Isso se deve a fatores como diferentes escritores, inclinação de linhas, palavras escritas separadamente, conexões e sobreposição entre linhas superiores e inferiores.

4.2.1 Segmentação de Linhas de Texto Manuscritos

A Figura 4.29 mostra uma imagem de texto manuscrito cujas linhas foram segmentadas pelo método SLP-TEO. Neste caso, mesmo que as linhas de texto apresentem diferentes extensões verticais, o método SLP-TEO consegue delimitá-las corretamente.

Em um outro exemplo o texto segmentado apresenta algumas conexões entre linhas o que dificultaria a segmentação através do método de projeção somente. Através do operador TEO foi possível encontrar as concentrações de pixels e decidir sobre as fronteiras através dos ajustes do método SLP-TEO. A Figura 4.30 mostra o texto “p06-047” segmentado em linhas e a projeção do histogramas das linhas com o envelope do TEO em vermelho no gráfico.

Tabela 4.11: Textos manuscritos segmentados pelo método SLP-TEO

	Texto	Nº Linhas	SLP-TEO	Recall	Nº Palavras	SLP-TEO	Recall
1	A03-011	9	9	100,00%	77	73	94,81%
2	A03-063	10	10	100,00%	87	57	65,52%
3	A04-010	9	9	100,00%	68	55	80,88%
4	A06-004	10	10	100,00%	65	65	100,00%
5	C03-000B	7	7	100,00%	56	46	82,14%
6	C04-017	8	8	100,00%	66	62	93,94%
7	D01-016	4	4	100,00%	51	37	72,55%
8	D01-052	8	8	100,00%	57	56	98,25%
9	D04-021	8	8	100,00%	59	42	71,19%
10	E04-043	8	8	100,00%	60	34	56,67%
11	E04-132	5	5	100,00%	34	30	88,24%
12	E06-026	10	6	60,00%	55	34	61,82%
13	F02-040	10	10	100,00%	71	64	90,14%
14	F04-079	11	11	100,00%	88	76	86,36%
15	G01-088	10	10	100,00%	74	32	43,24%
16	G06-011C	10	10	100,00%	88	88	100,00%
17	G06-018M	8	8	100,00%	70	64	91,43%
18	G06-042	6	6	100,00%	53	53	100,00%
19	G06-047I	7	7	100,00%	58	53	91,38%
20	G07-105	7	7	100,00%	58	42	72,41%
21	H01-000	11	11	100,00%	77	64	83,12%
22	H06-003	11	11	100,00%	87	73	83,91%
23	H07-069A	6	6	100,00%	54	53	98,15%
24	J04-035	10	10	100,00%	66	46	69,70%
25	J06-034	11	11	100,00%	49	47	95,92%
26	K04-103	8	8	100,00%	68	58	85,29%
27	K07-063A	7	7	100,00%	62	56	90,32%
28	L07-187	9	9	100,00%	69	69	100,00%
29	M01-121	9	9	100,00%	66	49	74,24%
30	M03-033	8	8	100,00%	57	50	87,72%
31	M04-000	8	8	100,00%	59	51	86,44%
32	M04-190	10	10	100,00%	79	62	78,48%
33	M06-106	10	10	100,00%	82	82	100,00%
34	N04-075	8	8	100,00%	63	46	73,02%
35	N04-084	7	7	100,00%	57	55	96,49%
36	N06-133	9	9	100,00%	76	70	92,11%
37	P06-047	11	11	100,00%	60	47	78,33%
38	R02-013	12	12	100,00%	72	49	68,06%
39	R02-131	6	6	100,00%	60	59	98,33%
40	R06-143	5	5	100,00%	48	48	100,00%
		341	337	98,83%	2606	2197	84,31%

The train drew into another station and Cecil, with a further six stops to go, was left almost alone in the coach. He fumed as he recollected the long correspondence he had had with the Inland Revenue in an effort to obtain tax relief for a jacket used solely in the office. "If the jacket is a condition of your employment," the inspector had written, "it may qualify for relief."

Figura 4.29: Segmentação de linhas do texto manuscrito "r02-013"

'I will leave you the necessary prescriptions for the sedatives. You will contact your new doctor when you reach your parents' house.' She watched him flick his fountain-pen from his waistcoat pocket, take out his ~~prescription~~ prescription pad, and scribble on it. Then, with a gesture she knew so well, he tore off the leaflet and

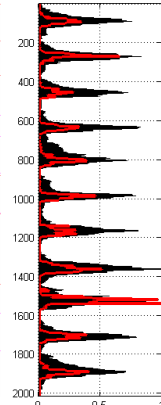


Figura 4.30: Segmentação de linhas do texto manuscrito "p06-047"

4.2.2 Segmentação de palavras em textos manuscritos

Na base de dados utilizada havia um total de 2.606 palavras, sendo que destas 2.197 foram segmentadas corretamente pelo método, o que representa um percentual de 84,3% das palavras manuscritas segmentadas corretamente pelo método SLP-TEO.

A Figura 4.31 apresenta a imagem da terceira linha do texto manuscrito “k07-063” e sua respectiva projeção vertical com o envelope do TEO. Note que a sexta palavra, “thrown” possui seus caracteres com uma distância significativa, mas o método através dos estudos de junção de palavras foi capaz de identificar corretamente a extensão da palavra e não a particionou.

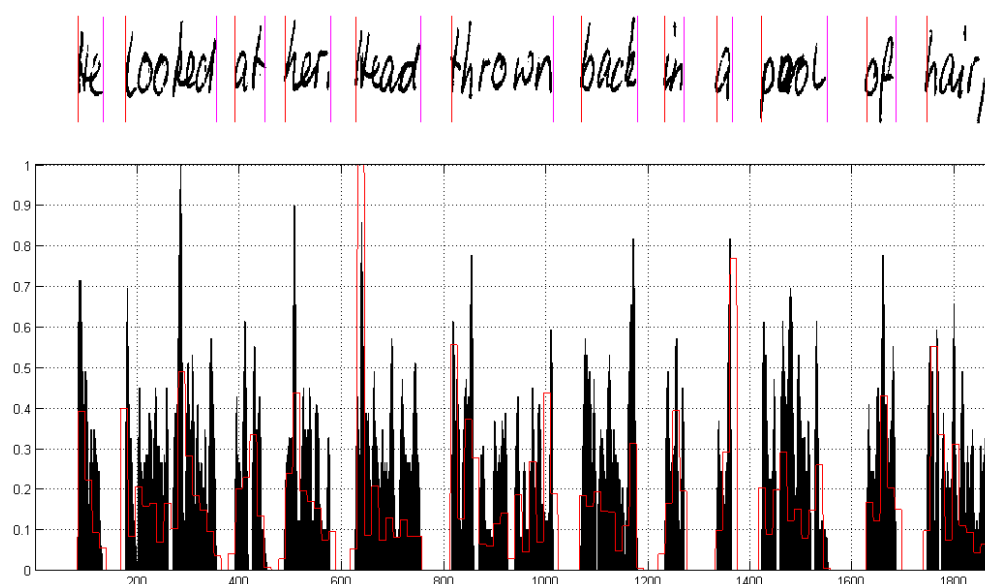


Figura 4.31: Segmentação da 3ª linha do texto manuscrito “k07-063”

A Figura 4.32 mostra outra linha segmentada pelo método SLP-TEO, observe a imagens das palavras “mechanisms” e “majority”. Se observarmos o histograma veremos que há um espaço considerável entre os caracteres desta palavra. Mas note que o método através dos ajustes realizados consegue segmentá-las corretamente.

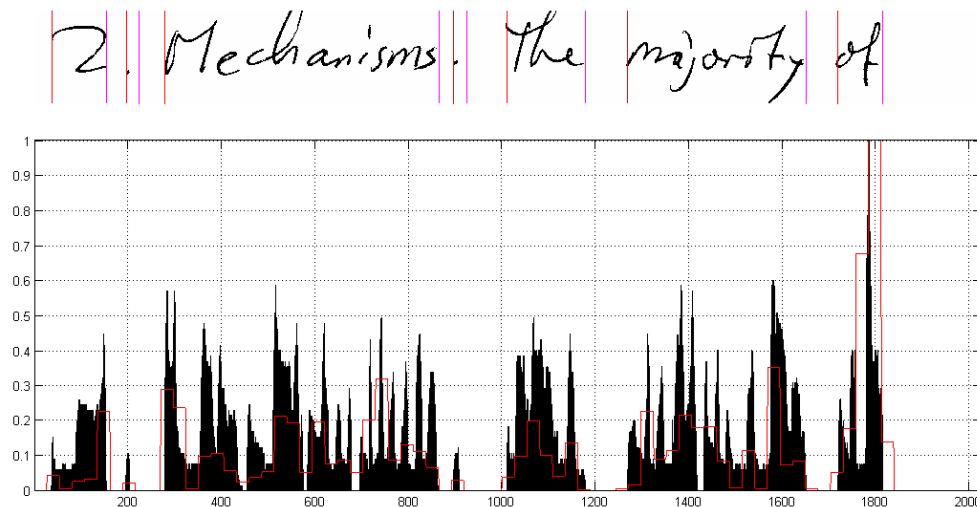


Figura 4.32: Segmentação da 1ª linha do texto manuscrito “linha1-j06-034”

4.3 O Sistema Implementado

O sistema de segmentação foi implementado no ambiente de desenvolvimento MATLAB® R2008b em um notebook com processador Intel(R) Core (TM) 2Duo CPU T6400 2.00GHz e memória RAM de 3,00GB e Sistema Operacional Windows 7 Ultimate de 64 bits .

A Figura 4.33 apresenta a interface do Sistema de Segmentação desenvolvido.

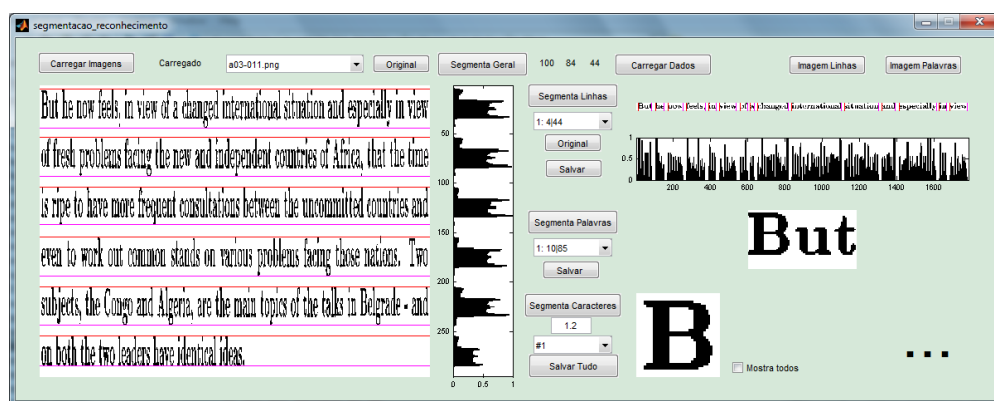
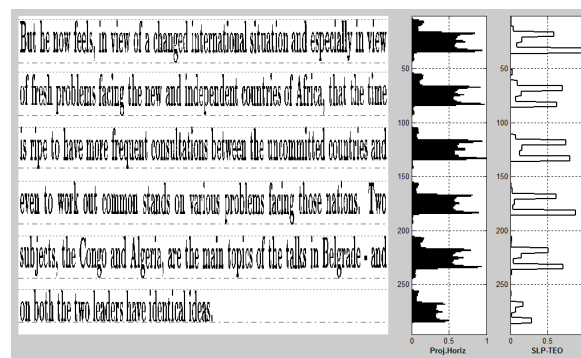


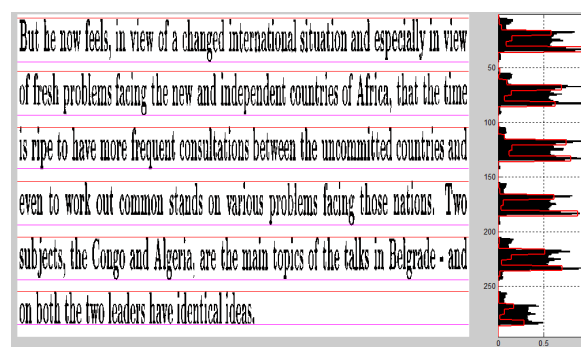
Figura 4.33: Tela Principal do Sistema

Como podemos ver a tela do sistema implementado possibilita a visualização de todas as etapas do processo de segmentação dos textos. Podemos ver na parte central da tela o texto segmentado em linhas, do lado direito deste espaço é possível verificar o histograma da imagem segmentada. E na parte superior direita é possível verificar as palavras segmentadas em suas respectivas linhas, logo abaixo o histograma da linha segmentada. Bem como, cada palavra e caracteres segmentados logo abaixo.

A partir do botão Imagem Linha, que se encontra na parte superior direita da tela é possível gerar algumas imagens da segmentação com o histograma da imagem e o envelope do TEO como pode ser observado na Figura 4.34(a) ou a sobreposição do Teo sobre o Histograma 4.34(b).



(a) Imagem da linha segmentada com o histograma e o envelope do Teo.



(b) Imagem da linha segmentada com a sobreposição do envelope do Teo sobre o histograma.

Figura 4.34: Imagens geradas a partir do botão Imagem Linha.

Da mesma forma, é possível gerar imagens das palavras através do botão Imagens Palavras. Estas imagens são importantes pois podemos analisar o que tem gerado problemas e fazer possíveis ajustes ao método.

4.4 Conclusão

Neste capítulo foram apresentados os experimentos realizados com os textos impressos e um estudo de caso do textos manuscritos. Além disso, foram explicados detalhadamente alguns casos especiais na segmentação de caracteres e suas soluções para os problemas apresentados. No capítulo seguinte realizaremos a conclusão do trabalho realizado.

Capítulo 5

Conclusão

Neste trabalho, foram desenvolvidos dois métodos que supriram as expectativas propostas no início deste trabalho. Podemos observar que os métodos desenvolvidos contribuem para a pesquisa na área de processamento digital de imagens, principalmente nas etapas de segmentação de imagens de textos impressos.

Baseado na capacidade do operador de energia de Teager (TEO) de extrair a função envelope do sinal com múltiplas componentes de frequência e aplicando este operador ao “sinal” gerado com os valores das projeções lineares de uma imagem de texto, o método SLP-TEO foi desenvolvido e utilizado para a segmentação de linhas e palavras de textos impressos e também aplicado à textos manuscritos. O método SLP-TEO apresentou resultados bastante interessantes quanto a segmentação de linhas tanto para textos impressos como para textos manuscritos. Um dos diferenciais do método está no fato de que um mesmo algoritmo é utilizado para ambos os tipos de texto. Além disso, o método SLP-TEO obteve sua eficiência na segmentação de palavras para textos impressos ou manuscritos, independente de qualquer mensuração

de distâncias entre palavras e/ou caracteres.

O método SLP-TEO teve um bom desempenho em relação a segmentação de palavras em textos manuscritos. Além disso, segundo Li [7], não é fácil estender um algoritmo de segmentação utilizado em textos impressos adequando-o para segmentação de textos manuscritos. Neste trabalho, mostramos que um mesmo algoritmo pode ser utilizado tanto para segmentação de linhas quanto para palavras, independentemente da natureza do texto, seja ele impresso ou manuscrito.

Alguns ajustes ao método ainda são necessários. O método SLP-TEO realiza uma separação linear das linhas, “cortando” componentes conectados em um ponto aproximadamente equidistante das concentrações de pixels de duas linhas adjacentes.

Para segmentação de caracteres o método gROSC se apresentou suficiente para solução na segmentação de caracteres simples ou para os problemas de caracteres conectados e sobrepostos. O grande diferencial do método é que este utiliza imagens em tons de cinza e as inferências utilizadas são baseadas nas características únicas do valor de nível de cinza do pixel analisado. Associado ao método de limiarização de Otsu, o método gROSC delimita as possibilidades de transição entre pixels estabelecendo um degrau disponível de acordo com o valor do pixel na imagem da palavra.

O método gROSC além de ter se mostrado eficiente, é simples e por não depender de outras abordagens como projeção linear, ou mesmo cálculo de distância entre caracteres para fazer suas inferências, pode ser considerado rápido. Além disso, não há a necessidade de se encontrar os caracteres problemáticos dentro do texto para depois ser aplicado o método. O método gROSC trata todos os caracteres da mesma maneira, independente se o caractere está ou não sobreposto ou conectado.

5.1 Trabalhos Futuros

A partir deste trabalho novos horizontes são vislumbrados. Sabemos que todo trabalho pode ser melhorado e visa dar suporte para a evolução de novos estudos e métodos. Assim, propomos alguns trabalhos que poderão ser feitos a partir deste trabalho.

- Implementar um sistema de reconhecimento utilizando os caracteres segmentados neste trabalho.
- Aplicar o método SLP-TEO a outras bases de dados e adaptar o método para imagem com problemas de rotação.
- Adequar o método SLP-TEO, para que este apresente uma melhor precisão na identificação de linhas de textos e palavras em textos manuscritos.
- Realizar pesquisa sobre as inferências utilizadas no método gROSC, para automatizar o valor do parâmetro A que corresponde a inclinação da reta definido na equação 3.5.
- Desenvolver novos algoritmos utilizando as características de imagens em tons de cinza para segmentação de outros tipos de imagens como imagens médicas, paisagem, dentre outras.

Publicações

- Tavares J., Peretta I. S., Lima, G. F. M., Yamanaka K., Pais M. S. **Algoritmo Baseado no Operador de Energia de Teager (TEO) para Segmentação de Linhas e Palavras em Textos Manuscritos e Impressos.** VII Workshop de Visão Computacional - WVC 2011, Curitiba-PR, Maio 2011, P. 219-224.
- Tavares J., Peretta I. S., Lima, G. F. M., Yamanaka K., Pais M. S. **A Recursive heuristic based on connected component labeling applied to segmentation for characters overlapping in printed text.** Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais. (aceito)

Bibliografia

- [1] ATAER, E. & DUYGULU, P. Retrieval of Ottoman documents. In: *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, Santa Barbara, California, USA, Oct 2006, MIR '06, ACM, p. 155–162.
- [2] BAR-YOSEF, I.; HAGBI, N.; KEDEM, K. & DINSTEIN, I. Line Segmentation for Degraded Handwritten Historical Documents. In: *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, Washington, DC, USA, 2009, ICDAR '09, IEEE Computer Society, p. 1161–1165.
- [3] BEZERRA, C. M. C. BR BRAILLE: Programa Tradutor de Textos Braille Digitalizados para Caracteres Alfanuméricos em Português. Dissertação de Mestrado, Universidade Estadual de Campinas, Campinas, Janeiro 2003.
- [4] DIRICHLET, P. G. Sur la convergence des séries trigonométriques qui servent à représenter une fonction arbitraire entre des limites données. *Journal für die reine und angewandte Mathematik*, vol. 4, no., 1829.
- [5] DOS SANTOS, R.; CLEMENTE, G.; REN, T. I. & CAVALCANTI, G. Text Line Segmentation Based on Morphology and Histogram Projec-

- tion. In: *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, july 2009, p. 651 –655.
- [6] GONZALEZ, R. C. & WOODS, R. E. *Processamento Digital de Imagens*. São Paulo : Pearson Prentice Hall, 3^a ed., 2010.
- [7] HEITLINGER, P., Ed. *Cadernos de Tipografia 1*. Paulo Heitlinger, 2007.
- [8] HEITLINGER, P., Ed. *Cadernos de Tipografia e Design - Caligrafias*. No. 16. maio 2010.
- [9] JUNG, M. Character segmentation using side view feature in machine-printed optical characters recognition. In: *Department of Computer System Engineering sangmyung*, Out 2010, Department of Computer System Engineering sangmyung.
- [10] KAISER, J. On a simple algorithm to calculate the 'energy' of a signal. In: *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, Abril 1990, p. 381–384.
- [11] KAISER, J. Some useful properties of Teager's energy operators. In: *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, apr 1993, vol. 3, p. 149 –152.
- [12] LAINE, M. & NEVALAINEN, O. A Standalone OCR System for Mobile Cameraphones. In: *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, sept 2006, p. 1 –5.
- [13] LEEDHAM, G.; YAN, C.; TAKRU, K.; TAN, J. H. N. & MIAN, L. Comparison of Some Thresholding Algorithms for Text/Background Segmentation in Difficult Document Images. In: *Proceedings of the Seventh International Conference on Document Analysis and Recognition*,
-

- Washington, DC, USA, 2003, vol. 2 of *ICDAR '03*, IEEE Computer Society.
- [14] LIWICKI, M.; SCHERZ, M. & BUNKE, H. Word Extraction from On-Line Handwritten Text Lines. 2008.
- [15] LOULLOUDIS, G.; GATOS, B. & HALATSIS, C. Text Line Detecting in Unconstrained Handwritten Documents Using a Block- Based Hough Transform Approach. In: *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, 2007, vol. 2, p. 599–603.
- [16] LOULLOUDIS, G.; GATOS, B.; PRATIKAKIS, I. & HALATSIS, C. Text line and word segmentation of handwritten documents. *Pattern Recognition*, vol. 42, no. 12, 2009. New Frontiers in Handwriting Recognition.
- [17] LUE, H.-T.; WEN, M.-G.; CHENG, H.-Y.; FAN, K.-C.; LIN, C.-W. & YU, C.-C. A novel character segmentation method for text images captured by cameras. *ETRI Journal*, vol. 32, no. 5, Oct 2010.
- [18] MARANA, A. N. & BREGA, J. R. F. *Técnicas e Ferramentas de Processamento de Imagens Digitais e Aplicações em Realidade Virtual e Misturada*. Bauro/SP : Canal6Editora, 2008.
- [19] MARTI, U. & BUNKE, H. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, vol. 5, no., 2002. 10.1007/s100320200071.
- [20] MORI, S.; SUEN, C. & YAMAMOTO, K. Historical review of OCR research and development. In: *Proceedings of the IEEE*, 1992, vol. 80, p. 1029 – 1058.
-

-
- [21] NIKOLAOU, N.; MAKRIDIS, M.; GATOS, B.; STAMATOPOULOS, N. & PAPAMARKOS, N. Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths. *Image Vision Comput.*, vol. 28, no., April 2010. Newton, MA, USA.
- [22] NOMURA, S.; YAMANAKA, K.; KATAI, O.; KAWAKAMI, H. & SHIOSE, T. A novel adaptive morphological approach for degraded character image segmentation, pattern recognition. *Pattern Recognition*, vol. 38, no., Nov 2005.
- [23] OTSU, N. A Threshold Selection Method from Gray-Level Histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, jan. 1979.
- [24] PAPAMARKOS, N. & KOUTALIANOS, T. Separation of overlapping characters. In: *Electronics, Circuits and Systems, 1999*, sep 1999, vol. 2, 6th IEEE International Conference, p. 867–870.
- [25] PECCINI, G. & ORNELLAS, M. C. Segmentação de Imagens por Watersheds: Uma Implementação Utilizando a Linguagem Java. *Revista Eletrônica de Iniciação Científica*, vol. IV, no., 2005.
- [26] PEDRINI, H. & SCHWARTZ, W. R. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. São Paulo : Thomson Learning, 2008.
- [27] PERETTA, I. S. A Novel Word Boundary Detector based on the Teager Energy Operator for Automatic Speech Recognition. Dissertação de Mestrado, Universidade Federal de Uberlândia, Dezembro 2010.
-

-
- [28] PERETTA, I. S.; LIMA, G. F. M.; TAVARES, J. A. & YAMANAKA, K. A Spoken Word Boundaries Detection Strategy for Voice Command Recognition. *Learning and Nonlinear Models*, vol. 8, no. 3, 2010.
- [29] SABA, T.; SULONG, G. & REHMAN, A. A Survey on Methods and Strategies on Touched Characters Segmentation. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, vol. 1, no., June 2010.
- [30] SAHA, S.; BASU, S.; NASIPURI, M. & BASU, D. K. A Hough Transform based Technique for Text Segmentation. *Journal of Computing*, vol. abs/1002.4048, no., Feb 2010.
- [31] SILVA, L. F. Distinção Automática de Texto Impresso e Manuscrito em uma Imagem de Documento. Dissertação de Mestrado, Universidade Federal Fluminense, Niterói, RJ, mar 2009.
- [32] TEAGER, H. M. & TEAGER, S. M. Evidence for Nonlinear Production Mechanisms in the Vocal Tract. In: *NATO Advanced Study Institute on Speech Production and Speech Modeling*, Julho 1989.
- [33] THILLOU, C.; FERREIRA, S. & GOSSELIN, B. An embedded application for degraded text recognition. *EURASIP J. Appl. Signal Process.*, January 2005.
-