



Pontifícia Universidade Católica do Paraná

Curso de Pós-Graduação em Informática Aplicada

Dissertação

apresentada à PUC-PR como requisito para a obtenção do título de

Mestre em Informática Aplicada

por

Fabricio Gabriel Olivo

Segmentação de Imagens de Palavras Manuscritas

Banca Examinadora:

Orientador:

Prof. Dr. Mounim El Yacoubi

PUC-PR

Examinadores:

Prof. Dr. Edouard Lethelier

PUC-PR

Prof. Dr. Edson Costa de B. Carvalho F.

UFPE

Prof. Dr. Lee Luan Ling

UNICAMP

Curitiba, outubro de 2000

FABRICIO GABRIEL OLIVO



SEGMENTAÇÃO DE IMAGENS DE PALAVRAS MANUSCRITAS

Dissertação apresentada à Pontifícia
Universidade Católica do Paraná para a
obtenção do título de Mestre em
Informática Aplicada.

Área de concentração:
**Análise e Reconhecimento de
Documentos**

Orientador:
Prof. Dr. Mounim El Yacoubi

Curitiba
2000



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO
PARANÁ
Pró-Reitoria de Pós-Graduação, Pesquisa e Extensão

ATA DA SESSÃO PÚBLICA DE EXAME DE DISSERTAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA DA PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ.

Exame de dissertação nº 023

Aos 15 dias do mês de setembro de 2000, realizou-se a sessão pública de defesa de dissertação “**Segmentação de Imagens de Palavras Manuscritas**”, apresentada por Fabricio Gabriel Olivo, ano de ingresso 1997, para obtenção do título de Mestre em Ciências. A Banca Examinadora foi composta pelos seguintes professores:

MEMBROS DA BANCA	ASSINATURA
Presidente: Prof. Dr. Abdenaim El-Yacoubi (PUCPR)	
Prof. Dr. Edouard Lethelier (PUCPR)	
Prof. Dr. Edson Costa de B. Carvalho Filho (UFPE)	
Prof. Dr. Lee Luan Ling (UNICAMP)	

De acordo com as normas regimentais a Banca Examinadora deliberou sobre os conceitos a serem atribuídos e que foram os seguintes:

MEMBROS DA BANCA	CONCEITOS
Presidente: Prof. Dr. Abdenaim El-Yacoubi (PUCPR)	Aprovado
Prof. Dr. Edouard Lethelier (PUCPR)	APROVADO
Prof. Dr. Edson Costa de B. Carvalho Filho (UFPE)	Aprovado
Prof. Dr. Lee Luan Ling (UNICAMP)	APROVADO
Conceito Final	Aprovado

Observações da Banca Examinadora

Prof.º Júlio Cesar Nievola
Coordenador do Programa de Pós-Graduação em Informática Aplicada-PUC-PR

Dedico este trabalho à minha família Tarciana
Elisa e João Gabriel, à minha mãe Inês Maria, e aos
meus irmãos Fabiola, Graciela, Maristela e Pedro.

Agradecimentos

Ao Prof. Dr. Mounim El Yacoubi, pela orientação desta dissertação e principalmente por confiar a mim a realização deste projeto.

Aos professores Dr. Edouard Lethelier, Dr. Edson Costa de B. Carvalho F. e Dr. Lee Luan Ling, pela disponibilidade de participar da banca examinadora deste trabalho.

Ao amigo Prof. Dr. Jacques Facon, pelo apoio prestado desde o início da realização de minhas pesquisas.

Aos professores Dr. Celso A. A. Kaestner e Dr. Marco A. B. Cândido, pela colaboração prestada.

Aos companheiros de pesquisa Alessandra, Andréia, Miguel, Paulo e Raquel, pelos momentos de trabalho e lazer que passamos juntos.

À minha esposa Tarciana e meu filho João Gabriel, pelo sacrifício, compreensão e motivação.

À minha mãe Inês Maria, e aos meus irmãos Fabiola, Graciela, Maristela e Pedro, pelo suporte e incentivo.

Finalmente, a todos aqueles que não foram mencionados e que contribuíram, de forma direta ou indireta, na realização deste trabalho.

F.G.O.

Sumário

Agradecimentos	i
Sumário	ii
Lista de Figuras	vi
Lista de Tabelas	x
Resumo	xi
Abstract	xii
Capítulo 1	
Introdução	1
1.1 Visão geral	1
1.2 Segmentação no Processo de Reconhecimento	1
1.3 Motivação e Objetivos	4
1.4 Sistema Proposto	5
1.5 Organização da Dissertação	7
Capítulo 2	
Estado da Arte	8
2.1 Sistemas de Reconhecimento de Texto Manuscrito	8
2.2 Técnicas de Reconhecimento	11
2.2.1 Técnicas Globais.....	12
2.2.2 Técnicas Baseadas em Segmentação.....	13
2.3 Tipos de Palavras Manuscritas	15
2.3.1 Escrita em Letras de Forma.....	16
2.3.2 Escrita Cursiva.....	17
2.4 Características que Influenciam na Segmentação.....	18
2.5 Segmentação de Escrita em Letras de Forma	20

2.5.1	Segmentação Usando Características do Contorno	20
2.5.2	Segmentação Usando Distâncias	21
2.6	Segmentação de Escrita Cursiva.....	22
2.6.1	Segmentação Explícita da Palavra	22
2.6.2	Segmentação Implícita da Palavra	27
2.7	Conclusão	31

Capítulo 3

Análises e Correções na Imagem da Palavra **32**

3.1	Pré-Processamento	32
3.2	Obtenção dos Contornos da Palavra	33
3.2.1	Contorno por Morfologia.....	33
3.2.2	Contorno pela Seqüência da Informação (Chain Code).....	34
3.3	Cadastramento dos Contornos da Palavra.....	35
3.4	Correção do ângulo de Inclinação das Letras.....	36
3.4.1	Conceitos da Estimativa da Inclinação.....	36
3.4.2	Método da Projeção	36
3.4.3	Método da Seqüência de Informação (Chain Code).....	36
3.4.4	Conceitos da Correção da Inclinação	37
3.5	Suavização da Escrita	38
3.5.1	Suavização por Morfologia	38
3.5.2	Suavização pela Rotação da Máscara	38
3.6	Cálculo da Espessura Média da Escrita	39
3.7	Análise da Imagem da Palavra	40
3.7.1	Análise do Contorno.....	41
3.7.2	Análise do Perfil.....	41
3.7.3	Segmentação em Caixas Disjuntas	42

Capítulo 4

Segmentação Baseada em Grapheme **43**

4.1	Reconhecimento por <i>Grapheme</i>	43
4.2	Definição de <i>Graphemes</i>	44
4.3	Extração de <i>Grapheme</i>	45
4.4	Detecção das Ligações.....	47
4.5	Detecção de ligações como pré-segmentação	48
4.6	Tipologia de Ligações	50
4.7	Processo de Pré-Reconhecimento.....	51
4.7.1	Análise da Concavidade	51
4.7.2	Análise da Extensão.....	52

4.7.3	Análise da Forma e Tamanho	53
-------	----------------------------------	----

Capítulo 5

Metodologia da Pesquisa 54

5.1	Conceitos da Metodologia	54
5.1.1	Etapas do Processo de Segmentação	54
5.2	Estudo Bibliográfico	56
5.3	Base de Dados de Validação	56
5.4	Pré-processamento	57
5.4.1	Conceitos	57
5.4.2	Suavização da Escrita	58
5.4.3	Obtenção dos Contornos da Palavra	59
5.4.4	Cadastramento dos Contornos da Palavra	60
5.4.5	Estimativa da Espessura Média da Escrita	61
5.4.6	Estimativa e Correção da Inclinação das Letras da Palavra	63
5.5	Pré-segmentação	66
5.5.1	Conceitos	66
5.5.2	Detecção do Corpo da Palavra	67
5.5.3	Detecção dos Pontos de Máximo e de Mínimo Locais	69
5.5.4	Divisão do Contorno Externo em Bordas Superior e Inferior	73
5.5.5	Detecção das Ligações	74
5.6	Segmentação	77
5.6.1	Conceitos	77
5.6.2	Quebra da Palavra em Ligações e Graphemes	78
5.6.3	Quebra dos Graphemes pelos Pontos de Mínimo	78
5.6.4	Reagrupamento dos Graphemes	85
5.7	Pós-processamento	88
5.7.1	Conceitos	88
5.7.2	Quebra das Ligações por Interseção	88
5.7.3	Reagrupamento das Ligações por Interseção	91
5.7.4	Reagrupamento das Ligações Altas	93
5.8	Conclusão	94

Capítulo 6

Análise dos Resultados 96

6.1	Introdução	96
6.2	Análise por Letra	97
6.2.1	Estatísticas	97
6.2.2	Observações	98

6.3	Análise Geral	106
6.3.1	Tempo de Processamento	106
6.3.2	Problemas que Causam Erros	106
6.4	Segmentação pelo Processo Tradicional	109
6.5	Conclusões	112

Capítulo 7

Conclusões Gerais	113
--------------------------	------------

Bibliografia	116
---------------------	------------

Lista de Figuras

Figura 1 : Imagem de carta postal com texto e elementos gráficos	2
Figura 2 : Estrutura geral de um sistema de reconhecimento off-line	9
Figura 3 : Fluxograma de um sistema que reconhece texto manuscrito [Kim98]	10
Figura 4 : Escrita em (a) letras de forma, (b) cursiva e (c) misturada	15
Figura 5 : Categorias de palavras escritas em letras de forma	17
Figura 6 : Exemplo de escrita cursiva pura e escrita cursiva misturada	17
Figura 7 : Exemplo do algoritmo de Casey e Horne [Casey92]. (a) Duas letras manuscritas tocando-se e os pontos de quebra candidatos P1, P2 e P3. (b) O modelo gráfico de (a) [Lu96].	20
Figura 8 : Exemplo do cálculo das distâncias entre a informação e o fundo [Lu96]. (a) Imagem binária. (b) Informação e fundo. (c) Distância de x para A^c . (d) Distância de x para A .	21
Figura 9 : Segmentação por Bosinovic e Srihari. (a) Imagem original. (b) Três zonas demarcadas. (c) Linhas verticais correspondentes aos PSPs. (d) Os PSPs após o processo de compactação.	24
Figura 10 : Exemplo de segmentação usando <i>axis</i> e <i>tarsi</i> [Lu96]. (a) Imagem original; (b) a imagem afinada; (c) o <i>axis</i> abaixo e o <i>tarsi</i> no topo.	27
Figura 11 : Contornos por morfologia. (a) Por dilatação e (b) por erosão	33
Figura 12 : Definição da seqüência de informação (chain code) com oito vizinhos.	34
Figura 13 : Contornos pela seqüência da informação (Chain Code). (a) Imagem original e (b) contorno da palavra	35
Figura 14 : Exemplo ilustrativo do cálculo da inclinação da palavra [Shridhar97]	37
Figura 15 : Correção da inclinação: (a) imagem da palavra de entrada, (b) pixels das bordas verticais e próximas a verticais, (c) imagem da palavra após a correção da inclinação [Shridhar97]	38
Figura 16 : Máscara de suavização por rotação	38
Figura 17 : Divisão do contorno externo. (a) Contornos interno e externo, e (b) borda superior e inferior.	39
Figura 18 : Estimativa da espessura média da escrita. (a) Calculo das y-distâncias para cada coordenada x. (b) Análise das ocorrências das espessuras. [Kim97]	40
Figura 19 : Pontos de pré-segmentação [Shridhar97]	41
Figura 20 : Perfil superior da palavra	42

Figura 21 : Segmentação em (a) caixas disjuntas e (b) ideal [Shridhar97]	42
Figura 22 : Partes significantes e não significantes da palavra	43
Figura 23 : Imagem da palavra dividida em sub-imagens. (a) original e (b) sub-imagens	44
Figura 24 : Extração de características por técnicas globais. (a) palavras diferentes com representações semelhantes e (b) mesma palavra com representações diferentes [Lecolinet91]	46
Figura 25 : Projeção e concavidades laterais [Lecolinet91]	48
Figura 26 : Segmentação. (a) Imagem original. (b) Representação do contorno externo e interno após a correção da inclinação. (c) Divisão dos contornos superior e inferior. (d) Ligações baseadas na espessura média. (e) Concavidades/convexidades. (f) Pontos de segmentação. [Kim97]	49
Figura 27 : Detecção dos pontos de mínimo do perfil do contorno superior [Lecolinet91]	50
Figura 28 : Tipologia de ligações [Lecolinet91]	50
Figura 29 : Ciclos abertos e fechados [Lecolinet91]	52
Figura 30 : Fluxograma do processo de segmentação proposto	56
Figura 31 : Suavização pela método "Rotação da Máscara". (a) Imagem original e (b) imagem suavizada.	59
Figura 32 : Cadastramento da lista principal e da lista secundária. (a) Detecção da linha média da palavra. (b) Pontos da lista secundária.	60
Figura 33 : Estimativa da espessura média pela projeção vertical	61
Figura 34 : Estimativa da espessura média da escrita. (a) e (b) Imagens de palavras. (c) e (d) Histogramas das imagens (a) e (b) respectivamente.	62
Figura 35 : Ângulos acumulados para a estimativa da inclinação. (a) Sistema original. (b) Sistema alterado para esta pesquisa.	64
Figura 36 : Correção da inclinação das letras da palavra. (a) Palavras originais. (b) palavras corrigidas.	66
Figura 37 : Divisão da palavra em <i>zona superior</i> , <i>zona média</i> e <i>zona inferior</i>	67
Figura 38 : Suavização do histograma de transição da palavra da Figura 37. (a) Histograma suavizado. (b) Histograma original.	68
Figura 39 : Detecção dos pontos de máximo e mínimo.	71
Figura 40 : Detecção dos pontos de mínimo e máximo do contorno externo de uma palavra.	71
Figura 41 : Exclusão dos pontos de mínimo e dos pontos de máximo redundantes.	72
Figura 42 : Divisão do contorno externo em bordas superior e inferior. (a) Pontos extremos. (b) Borda superior. (c) Borda inferior.	73
Figura 43 : Exemplo da detecção das ligações em uma palavra. (a) Contorno interno e externo da palavra. (b) Ligações detectadas pelo contorno externo.	75

- Figura 44 : Escolha do ponto base para filtrar ligação. (a) Ponto de mínimo pertence à porção média da ligação. (b) Ponto de mínimo não pertence à porção média da ligação. 76
- Figura 45 : Filtragem das ligações. (a) Pela inclinação. (b) Ligações pequenas. 76
- Figura 46 : Problema na detecção de ligação . 77
- Figura 47 : Exemplo da quebra da palavra em ligações e graphemes. (a) Imagem original. (b) Imagem normalizada. (c) Ligações detectadas. (d) Quebra da palavra. 78
- Figura 48 : Exemplo da quebra da palavra em ligações e graphemes. (a) Pontos de máximo e mínimo. (b) Quebra da palavra 79
- Figura 49 : Exclusão dos pontos de mínimo e máximo próximos. (a) Máximos e mínimos detectados. (b) Máximos e mínimos filtrados. 81
- Figura 50 : Situações que podemos encontrar no filtro da distância. (a) Distância entre as bordas superior e inferior é menor que a distância limiar. (b) Distância entre as bordas superior e inferior é maior ou igual à distância limiar. (c) Distância entre as bordas superior e inferior é maior que a altura do corpo da palavra e a coordenada vertical da borda superior é menor que a linha de base inferior da palavra. 82
- Figura 51 : Alteração do ponto de mínimo pelo filtro da distância entre as bordas superior e inferior. (a) Ponto de mínimo original. (b) Ponto de mínimo alterado. 83
- Figura 52 : Exemplo de distância muito grande entre as bordas superior e inferior. 83
- Figura 53 : Exclusão de pontos de mínimo pelo ponto correspondente ser da borda superior. 84
- Figura 54 : Alteração no ponto de mínimo após verificação de ponto intermediário. (a) Ponto de mínimo original. (b) Ponto de mínimo alterado 85
- Figura 55 : Obtenção do tamanho e da altura do grapheme. 86
- Figura 56 : Reagrupamento de graphemes. 88
- Figura 57 : Interseção entre ligação e graphemes. (a) Interseção à esquerda. (b) Interseção à direita. 90
- Figura 58 : Quebra da ligação por interseção. (a) Detecção das ligações. (b) Quebra das ligações com interseção. 90
- Figura 59 : Reagrupamento das ligações por interseção. (a) Interseção entre as entidades. (b) Quebra da letra. (c) Reagrupamento por Interseção. 92
- Figura 60 : Exemplos de Reagrupamento por interseção. (a) Detecção das ligações. (b) Quebra das ligações por interseção. (c) Reagrupamento das ligações por interseção. 92
- Figura 61 : Reagrupamento por ligações altas e por interseção. (a) Pontos de mínimo. (b) Zonas da palavra e linha média. (c) Quebra das ligações por interseção. (d) Reagrupamento por interseção e por ligações altas. 93

- Figura 62 : Exemplos de quebra de letras com ciclo no corpo da palavra. (a) Letra segmentada inteira. (b) Letra segmentada em duas entidades. (c) Letra segmentada em três entidades. (d) Letras segmentadas em quatro entidades. 99
- Figura 63 : Exemplos de quebra de letras que não tem ponto de mínimo na borda superior. (a) Letra “e” segmentada em uma e em duas entidades. (b) Letra “i” segmentada em uma entidade. (c) Letra “l” segmentada em uma entidade. 100
- Figura 64 : Exemplos de quebra das letras com concavidade para a direita. (a) Letra “c” segmentada em uma e duas entidades. (b) Letra “E” segmentada inteira. (c) Letra “t” segmentada em uma, duas e três entidades. (d) Letra “L” segmentada em uma e duas entidades 101
- Figura 65 : Exemplos de quebra de letras com concavidade para baixo. (a) Letra “m” segmentada em três, quatro, cinco, seis, sete e oito entidades. (b) Letra “n” segmentada em duas, três, quatro, cinco e seis entidades. (c) Letra “r” segmentada em uma, duas e três entidades. (d) Letra “s” segmentada em uma, duas e três entidades. 102
- Figura 66 : Exemplos de quebra de letras com concavidade para cima. (a) Letra “M” segmentada em duas e três entidades. (b) Letra “N” segmentada em duas, três e quatro entidades. (c) Letra “u” segmentada em duas, três e quatro entidades. (d) Letra “v” segmentada em duas, três e quatro entidades. (e) Letra “V” segmentada em duas e três entidades. 103
- Figura 67 : Exemplo de quebra de outras letras. (a) Letra “A” segmentada em uma, três e quatro entidades. (b) Letra “R” segmentada em uma entidade. (c) Letra “S” segmentada em uma, duas, três e quatro entidades. (d) Letra “Z” segmentada em uma, duas e três entidades. (e) Letra “z” segmentada em uma e três entidades. 105
- Figura 68 : Exemplo de letras que se tocam. (a) Parte da imagem original. (b) Parte da imagem segmentada. 107
- Figura 69 : Exemplos de problemas encontrados na segmentação. (a) Detecção dos pontos de mínimo e das ligações. (b) Quebra da palavra. 107
- Figura 70 : Exemplo de problemas encontrados na segmentação. (a) Detecção de pontos de mínimo e das ligações. (b) Apontamento para mínimo não detectado e mínimo excluído. 108
- Figura 71 : Exemplos de problema não tratado na segmentação. (a) Detecção de ligações sem o filtro por inclinação. (b) Segmentação após aplicado todos os filtros da detecção das ligações. 108
- Figura 72 : Segmentação pelo processo tradicional e pelo processo proposto nesta pesquisa. (a) Segmentação por pontos de mínimo. (b) Segmentação por pontos de mínimo e por ligações. 110

Lista de Tabelas

Tabela 1 : Palavras do dicionário de validação	57
Tabela 2 : Estatísticas do particionamento das letras pelo sistema proposto.	98
Tabela 3 : Estatísticas do particionamento das letras pelo sistema tradicional.	111

Resumo

O reconhecimento de texto manuscrito tem sido alvo de intensa pesquisa nos últimos anos. Neste campo, existem dois tipos de técnicas que são usualmente adotadas: *técnicas globais* e *técnicas baseadas em segmentação* (ou analítica). As técnicas globais tentam reconhecer a palavra como uma unidade usando características globais, tais como, arcos, buracos, ascendentes, descendentes, ciclos, etc. As técnicas baseadas em segmentação podem usar modelos que tentam particionar a palavra em letras individuais ou modelos que tentam particionar a palavra em entidades que podem ser parte de uma letra, uma letra individual ou mais de uma letra (*grapheme*). As técnicas de segmentação que particionam a palavra em letras isoladas são muito difíceis na prática devido à ambigüidade das letras manuscritas. Já as técnicas de segmentação baseadas em *grapheme* consistem em produzir um alto número de pontos de segmentação livremente, oferecendo desta maneira várias opções de combinação para o sistema de reconhecimento. Podemos encontrar ainda, um modelo de reconhecimento que utilize características globais e analíticas para realizar sua tarefa. Estas técnicas são usualmente chamadas de *técnicas híbridas*.

A pesquisa relacionada com esta tese de mestrado concentrou-se no desenvolvimento de uma técnica de segmentação baseada em *grapheme*. Um *grapheme* é definido como sendo uma entidade produzida pelo processo de segmentação, ou seja, pode ser uma letra inteira (*correct-segmentation*), parte de uma letra (*over-segmentation*) ou mais que uma letra (*under-segmentation*).

A característica de destaque deste algoritmo de segmentação é a habilidade de segmentar a palavra não somente em *graphemes*, como a maioria das técnicas baseadas em segmentação o faz, mas também segmentar as pontes de ligação entre as letras. A detecção e segmentação das ligações tendem a diminuir a variabilidade dos *graphemes*. Esta característica contribuiu para o desenvolvimento de um sistema *omni-scriptor* de segmentação de manuscritos.

Resumo

O reconhecimento de texto manuscrito tem sido alvo de intensa pesquisa nos últimos anos. Neste campo, existem dois tipos de técnicas que são usualmente adotadas: *técnicas globais* e *técnicas baseadas em segmentação* (ou analítica). As técnicas globais tentam reconhecer a palavra como uma unidade usando características globais, tais como, arcos, buracos, ascendentes, descendentes, ciclos, etc. As técnicas baseadas em segmentação podem usar modelos que tentam particionar a palavra em letras individuais ou modelos que tentam particionar a palavra em entidades que podem ser parte de uma letra, uma letra individual ou mais de uma letra (*grapheme*). As técnicas de segmentação que particionam a palavra em letras isoladas são muito difíceis na prática devido à ambigüidade das letras manuscritas. Já as técnicas de segmentação baseadas em *grapheme* consistem em produzir um alto número de pontos de segmentação livremente, oferecendo desta maneira várias opções de combinação para o sistema de reconhecimento. Podemos encontrar ainda, um modelo de reconhecimento que utilize características globais e analíticas para realizar sua tarefa. Estas técnicas são usualmente chamadas de *técnicas híbridas*.

A pesquisa relacionada com esta tese de mestrado concentrou-se no desenvolvimento de uma técnica de segmentação baseada em *grapheme*. Um *grapheme* é definido como sendo uma entidade produzida pelo processo de segmentação, ou seja, pode ser uma letra inteira (*correct-segmentation*), parte de uma letra (*over-segmentation*) ou mais que uma letra (*under-segmentation*).

A característica de destaque deste algoritmo de segmentação é a habilidade de segmentar a palavra não somente em *graphemes*, como a maioria das técnicas baseadas em segmentação o faz, mas também segmentar as pontes de ligação entre as letras. A detecção e segmentação das ligações tendem a diminuir a variabilidade dos *graphemes*. Esta característica contribuiu para o desenvolvimento de um sistema *omni-scriptor* de segmentação de manuscritos.

Abstract

Over the last years, research on handwriting recognition has been the subject of much attention. The main used techniques in this field can be classified into three categories: *Global* techniques, *Analytic (or segmentation-based)* techniques and *Hybrid* techniques. Global techniques try to recognize a word as a whole by using global features like arcs, concavities, ascenders, descenders, loops, etc. Analytic techniques usually split words into letters or graphemes. A grapheme may consist of an entire letter, a letter piece or more than a letter. Techniques that try to segment a word into whole letters are very difficult to implement in practice due to the high ambiguity of handwritten words and letters. To tackle this problem, the grapheme-based segmentation techniques try to produce a deliberately high number of segmentation points, offering in this way various segmentation options, the best one to be hopefully validated during recognition. Hybrid techniques use global features and features obtained from word segments to recognize this word.

The objective of this master thesis is the development of a segmentation technique to segment words into graphemes. A grapheme is defined as any entity produced by the segmentation algorithm. Hence, it may consist of an entire letter (correct segmentation), a letter piece (over-segmentation), or more than a letter (under-segmentation)

The main contribution of this work lies in the ability of the proposed algorithm to segment not only the graphemes as is usually the case, but also the ligatures when they exist. Hence, in our case, the segmented word consists of a sequence of graphemes and ligatures. The detection and the segmentation of the ligatures permit the reduction of the variability of the graphemes, which may significantly alleviate the job of an omni-scriptor handwriting recognizer.

Abstract

Over the last years, research on handwriting recognition has been the subject of much attention. The main used techniques in this field can be classified into three categories: *Global* techniques, *Analytic (or segmentation-based)* techniques and *Hybrid* techniques. Global techniques try to recognize a word as a whole by using global features like arcs, concavities, ascenders, descenders, loops, etc. Analytic techniques usually split words into letters or graphemes. A grapheme may consist of an entire letter, a letter piece or more than a letter. Techniques that try to segment a word into whole letters are very difficult to implement in practice due to the high ambiguity of handwritten words and letters. To tackle this problem, the grapheme-based segmentation techniques try to produce a deliberately high number of segmentation points, offering in this way various segmentation options, the best one to be hopefully validated during recognition. Hybrid techniques use global features and features obtained from word segments to recognize this word.

The objective of this master thesis is the development of a segmentation technique to segment words into graphemes. A grapheme is defined as any entity produced by the segmentation algorithm. Hence, it may consist of an entire letter (correct segmentation), a letter piece (over-segmentation), or more than a letter (under-segmentation)

The main contribution of this work lies in the ability of the proposed algorithm to segment not only the graphemes as is usually the case, but also the ligatures when they exist. Hence, in our case, the segmented word consists of a sequence of graphemes and ligatures. The detection and the segmentation of the ligatures permit the reduction of the variability of the graphemes, which may significantly alleviate the job of an omni-scriptor handwriting recognizer.

Capítulo 1

Introdução

1.1 Visão geral

A meta mais discutida no campo da inteligência artificial, já que este campo toma a inteligência humana como alvo principal de sua pesquisa, é de simplesmente habilitar computadores para realizar tarefas que são naturais às pessoas. Deste modo, os computadores devem ser hábeis para interagir com as pessoas e atuar no mundo de maneira cada vez menos restrita. Estas máquinas de computação saíram dos laboratórios e se tornaram indispensáveis nos lares e nos escritórios. Foi através da popularização do computador que surgiu a possibilidade de se substituir o papel pelo meio digital podendo-se atribuir a este meio de comunicação muitas vantagens de uso. Mas, junto com esta possibilidade veio a necessidade de tornar o computador uma máquina hábil a tratar de informações manuscritas.

O texto manuscrito é um meio natural de comunicação que quase todo mundo aprende desde cedo. Por isso, os pesquisadores estão preocupados em encontrar maneiras de se manipular texto manuscrito em computadores ao nível de reconhecimento e interpretação.

1.2 Segmentação no Processo de Reconhecimento

No geral, ao se processar automaticamente a imagem de um documento digital tem-se como objetivo reconhecer e interpretar determinados elementos deste documento para se extrair algum tipo de informação. Estes elementos normalmente são do tipo caractere e/ou elementos gráficos.

Porém, realizar o reconhecimento de um documento formado por vários elementos de diferentes naturezas (texto, números, gráficos, ...) é uma tarefa difícil. Os sistemas de classificação (OCR), freqüentemente se deparam com documentos de texto que consistem de linhas de palavras e algumas vezes elementos gráficos. No geral, a primeira tarefa a ser realizada em um documento digital é a de separar os elementos gráficos dos caracteres. Esta separação forma duas categorias distintas de informação que devem ser tratadas de modo diferente. A Figura 1 mostra um exemplo de imagem de um documento composto por elementos de diferentes naturezas.



Figura 1 : Imagem de carta postal com texto e elementos gráficos

Mesmo para tratar de cada uma das categorias citadas acima (elementos gráficos e caracteres) é necessário que existam algoritmos especializados em trabalhar com os diferentes tipos de elementos que possivelmente exista dentro da categoria analisada. Podemos perceber facilmente esta idéia se levarmos em consideração, por exemplo, que a categoria de elementos do tipo caractere pode ser constituída de dígitos numéricos e/ou texto, e a subcategoria de texto pode ser constituída de texto pré-impreso e/ou texto manuscrito, e a subcategoria de texto manuscrito pode ser constituída de uma escrita cursiva e/ou em letras de forma. Deste modo, o sucesso dos processos de análise, reconhecimento e interpretação do documento está diretamente relacionado com o sucesso da segmentação do documento em seus elementos básicos (texto, dígitos numéricos, figuras, etc.).

Cada subcategoria previamente segmentada deve servir de entrada para módulos especializados em analisar esta subcategoria em específico,

utilizando em sua análise regras que sejam capazes de obter informações e produzir estimativas sobre a imagem analisada.

Deve-se notar que ruídos e elementos que não se encaixam em nenhuma subcategoria especializada, devem ser percebidos e tratados de alguma maneira.

Nestes estudos o processo pelo qual as pessoas reconhecem caracteres, palavras e documentos manuscritos, tem sido o assunto de intenso interesse e investigação por pesquisadores de diversos campos. No geral, um bom entendimento do mecanismo do reconhecimento humano de documento manuscrito tem um impacto significativo no desenvolvimento de máquinas capazes de reconhecer e interpretar tais documentos. Porém, o processo de reconhecimento humano é muito complexo e incorpora informações extraídas de diferentes níveis: letras, palavras inteiras, palavras-chave e processamento contextual. A eficiência do reconhecimento humano de texto manuscrito pode ser atribuído a integração efetiva de múltiplas insinuações a respeito do texto analisado [Shridhar97].

O reconhecimento de texto manuscrito é um problema importante na prática. Suas aplicações incluem cheques bancários, endereços de cartas postais, formulários, documentos de escritório, sistemas de informação de biblioteca, organizações pessoais e muitas outras.

Nos sistemas de reconhecimento de texto manuscrito, existem duas categorias principais de técnicas usadas: *técnicas globais*, *técnicas baseadas em segmentação*. As técnicas globais (ver seção 2.2.1) utilizam características da forma da palavra extraídas da análise global de sua imagem e tenta reconhecer a palavra de entrada através destas características. As técnicas baseadas em segmentação (ver seção 2.2.2) podem ser subdivididas em *segmentação explícita*, *segmentação implícita* e *segmentação baseada em grapheme*. As técnicas de segmentação explícita e segmentação implícita tentam particionar a imagem da palavra em letras individuais. Se uma segmentação da palavra é feita antes do reconhecimento, esta é chamada de *segmentação explícita*. Se o processo de segmentação acontecer ao mesmo tempo em que ocorre o reconhecimento, esta é chamada de *segmentação implícita*. Já a técnica de segmentação baseada em grapheme tenta particionar a imagem da palavra em entidades que podem ser parte de uma letra, uma letra individual ou mais de uma letra. As técnicas baseadas em grapheme

extraem as características da palavra de forma seqüencial e identificam a palavra analisando esta seqüência. Estas técnicas tendem a produzir um grande número de pontos de segmentação. Para realizar o reconhecimento, o HMM tem sido usado amplamente em conjunto com as técnicas baseadas em grapheme. A utilização de alguns conceitos de ambas as técnicas de reconhecimento também gera um modelo de reconhecimento chamado de *modelo híbrido de reconhecimento* [Shridhar97].

Como visto acima, pode-se usar uma técnica de reconhecimento de palavra manuscrita baseada em um modelo de reconhecimento que utiliza uma segmentação explícita da palavra baseada em grapheme.

O assunto desta pesquisa está concentrado na fase de segmentação de uma técnica de reconhecimento cujo modelo utilize uma segmentação explícita da palavra baseada em grapheme. Esta proposta de segmentação da imagem de uma palavra tenta produzir explicitamente várias entidades elementares chamadas *graphemes* (ver seção 4.2). É a partir destas entidades que se realiza o reconhecimento da palavra. Para que se possa obter um bom resultado no processo de reconhecimento da palavra manuscrita, é preciso que a fase de segmentação tenha sido realizada com sucesso, ou seja, é preciso que a imagem da palavra tenha sido segmentada em letras isoladas e/ou partes de letras.

1.3 Motivação e Objetivos

Apesar do grande interesse e investigação dos pesquisadores em encontrar uma técnica eficaz e robusta de reconhecimento de texto manuscrito, ainda nos deparamos com problemas neste processo que não foram tratados ou não foi obtido um avanço suficiente para que o processo de reconhecimento de texto manuscrito fosse realizado com alto grau de confiança.

Este trabalho foi motivado pelo avanço que a segmentação de palavras manuscritas ainda pode alcançar. Em nossos estudos foram analisados vários conceitos e regras de segmentação de palavras manuscritas relatadas pela comunidade científica envolvida neste assunto. Nesta análise, observamos que os modelos propostos poderiam ser complementados entre si, ou seja, a solução para a deficiência de alguns trabalhos estava relatada em trabalhos similares.

O objetivo principal deste trabalho é o de desenvolver um sistema de segmentação que segmente imagens de palavras manuscritas em entidades elementares, ou seja, em letras e/ou partes de letras. Isto é compreensível se notarmos que, neste modelo de segmentação, a segmentação final fica a cargo do algoritmo de reconhecimento que deve ser robusto o suficiente para realizar a melhor combinação entre os segmentos gerados pelo processo de segmentação. Como já insinuado anteriormente, o processo de segmentação deve evitar entidades constituídas de mais de uma letra para que o modelo de reconhecimento não perca a liberdade de combinação dos segmentos.

O sistema de segmentação proposto, também deve ser capaz de “observar” algumas características da palavra para que, desta forma, possa adaptar-se a ela e consiga tratar da variância que ocorre nos estilos da escrita. Com isto, o sistema de segmentação poderá tratar qualquer tipo de escrita.

As entidades finais do processo de segmentação devem servir de entrada a um modelo de reconhecimento que seja capaz de realizar uma segmentação final combinando estas entidades para representarem uma hipótese de uma palavra.

1.4 Sistema Proposto

Este sistema de segmentação propõe gerar uma quantidade razoável de segmentos da imagem de uma palavra manuscrita. Isto é compreensível a partir do momento em que podemos passar estes segmentos ao módulo de reconhecimento da palavra e este módulo se encarregará de produzir a seqüência mais provável de uma palavra existente.

O projeto deste sistema de segmentação está baseado em 4 processos principais: pré-processamento, pré-segmentação, segmentação e pós-processamento. Através destes processos monta-se uma estrutura hierárquica de segmentação onde cada uma das fases do processo de segmentação é dedicada a uma tarefa específica.

Na fase de pré-processamento o sistema trata a imagem da palavra de forma a deixá-la adequada ao processo que irá recolher as informações sobre as características desta palavra. Os principais processos realizados nesta fase são:

1. estimativa e correção da inclinação das letras (ver seção 3.4),

2. suavização dos contornos das palavras e preenchimentos de pequenas falhas na escrita (ver seção 3.5),
3. cadastramento do texto da imagem a ser analisada (ver seção 3.3), e
4. estimativa da espessura média da escrita (ver seção 3.6).

Ainda na fase de pré-processamento é realizada uma verificação do número de cores da imagem e se a imagem não for binária (duas cores) o sistema converte esta em imagem para binária.

No passo de pré-segmentação, o texto da imagem já passou por um processo de cadastramento e está pronto para ser analisado. A partir da realização desta análise, são obtidas algumas informações necessárias para que se possa realizar uma futura segmentação da palavra. Os principais processos realizados nesta etapa de análise são:

1. detecção do corpo da palavra,
2. detecção dos *pontos de pico* (máximos locais) e *pontos de vale* (mínimos locais), e
3. detecção das ligações existentes entre as letras.

Deste modo, na etapa de pré-segmentação o sistema busca por possíveis pontos de segmentação. Ainda nesta etapa o sistema tenta anular a variância entre estilos de escrita. Isto ocorre porque o sistema obtém informações a respeito das ligações que existem entre as letras da palavra e, através destas informações, tenta reduzir a variabilidade entre os estilos de escrita. Assim, quando falamos sobre os dados de entrada deste sistema, estamos nos referindo a texto manuscrito em letras de forma, texto manuscrito cursivo e texto manuscrito misturado, podendo ser utilizado qualquer estilo de escrita.

No passo de segmentação o sistema utiliza-se das informações obtidas de uma análise da imagem palavra para formar regras. Estas regras são usadas para encontrar possíveis pontos de segmentação da imagem analisada. Através do corte da imagem nos pontos de segmentação pode-se obter letras inteiras individuais, segmentos de letra e letras sobrepostas.

O passo de pós-processamento foi projetado para validar as entidades resultantes do processo de segmentação. Este processo atua nas hipóteses de

hipóteses de *ligações* quebrando e reagrupando ligações baseado no contexto local.

Os dados de saída do sistema proposto, ou seja, os *graphemes* e as *ligações*, devem servir como dados de entrada a um sistema que seja capaz de realizar o reconhecimento de uma palavra através do agrupamento destas entidades (letras ou partes de letra). Por este motivo, quanto mais eficaz e certo for o processo de segmentação da palavra maior será a chance de sucesso de um módulo de reconhecimento.

Este modelo será comparado ao modelo mais tradicional que segmenta a palavra sem a informação extraída da detecção das ligações.

1.5 Organização da Dissertação

Este capítulo descreveu o assunto desta pesquisa de modo a introduzir o leitor no ambiente deste trabalho. A organização do restante desta dissertação é a seguinte:

- O capítulo 2 descreve brevemente alguns conceitos das técnicas de reconhecimento de texto manuscrito e os avanços que a comunidade científica obteve na pesquisa sobre o processo de segmentação de texto manuscrito.
- O capítulo 3 descreve a análise que as técnicas de segmentação devem realizar para a obtenção de informações básicas sobre a imagem do texto manuscrito.
- O capítulo 4 descreve o processo de segmentação de texto manuscrito utilizando o conceito de *grapheme*. O processo de segmentação proposto utiliza os conceitos de *grapheme* para gerar as entidades de saída.
- O capítulo 5 descreve a metodologia utilizada para analisar e solucionar o problema da segmentação de texto manuscrito baseada na técnica proposta.
- O capítulo 6 descreve alguns resultados obtidos através deste trabalho.
- O capítulo 7 descreve as conclusões sobre a metodologia utilizada para realizar o processo de segmentação de texto manuscrito.

Capítulo 2

Estado da Arte

2.1 Sistemas de Reconhecimento de Texto Manuscrito

A maior divisão que os sistemas de reconhecimento de palavras manuscritas podem sofrer é feita entre dois grandes tipos: *sistemas on-line* e *sistemas off-line*. Os sistemas ditos *on-line* tentam converter e reconhecer os caracteres interativamente, ou seja, recebendo os dados diretamente de algum tipo de dispositivo ou caneta ligada ao computador. Já os sistemas ditos *off-line* tentam reconhecer o texto que foi digitalizado de um pedaço de papel, ou seja, realizam o reconhecimento a partir da imagem digital estática do texto.

Os sistemas de reconhecimento discutidos neste trabalho são do tipo *off-line*. Em geral, para vocabulários extensos, estes sistemas requerem uma segmentação da palavra para realizar o reconhecimento. O passo de segmentação, nestes casos, é um processo de decisão. Assim, nestes tipos de sistemas existem, no geral, 3 (três) componentes lógicos [Casey96][Srihari]:

1. Digitalização da imagem;
2. Software e hardware de reconhecimento;
3. Interface de saída;

A Figura 2 exibe a relação entre os componentes lógicos de um típico sistema OCR (Reconhecedor Óptico de Caractere).

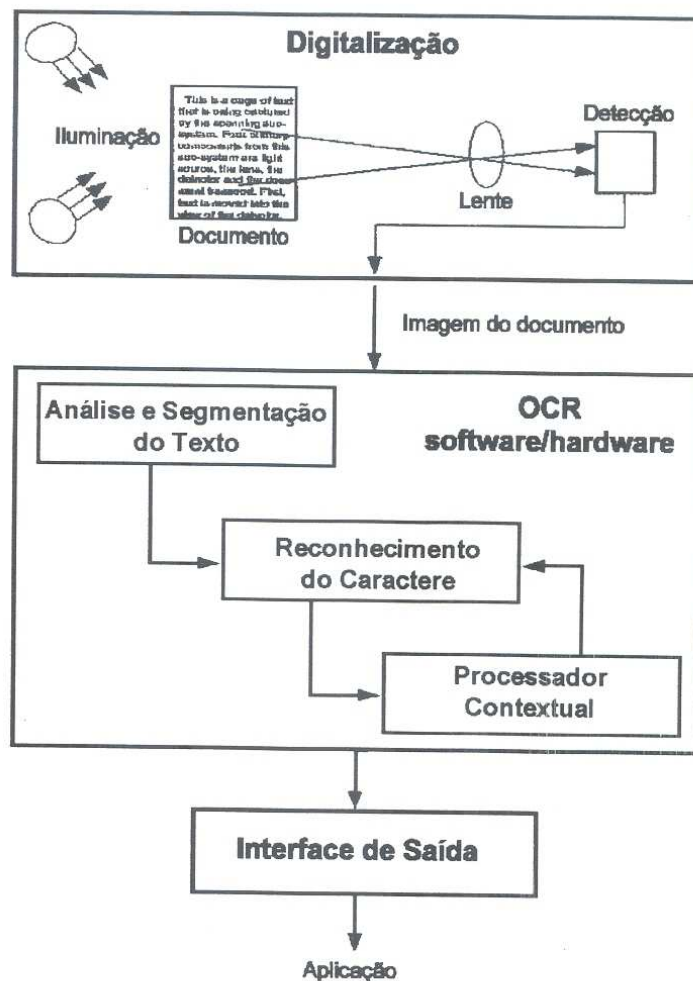


Figura 2 : Estrutura geral de um sistema de reconhecimento off-line

O documento digitalizado retém a imagem do texto a ser reconhecido. A imagem do texto é processada com o software e hardware do sistema de reconhecimento (OCR). Este processo envolve três operações: análise e segmentação da imagem do texto contido no documento digital (extração das características da imagem da palavra), o reconhecimento do texto analisado (identificação da palavra baseada em suas características), e o processamento contextual (corrige classificações incorretas do algoritmo de reconhecimento). A interface de saída é responsável pela comunicação entre a saída do sistema de reconhecimento e a aplicação.

Dois componentes essenciais em um algoritmo de reconhecimento de caracteres são o extrator de características e o classificador. A análise das características determina os descritores, ou conjunto de características, usado para descrever todos os caracteres. Dada a imagem de um caractere, o extrator de características extrai as características que o caractere possui. As

características extraídas são usadas como entrada para o classificador de caracteres [Srihari].

A pesquisa sobre o reconhecimento de palavras manuscritas, sejam cursivas ou não, teve um progresso impressionante nos últimos anos e muitas técnicas foram desenvolvidas para tentar resolver este problema [Lu96].

Para que um sistema de reconhecimento de palavras manuscritas tenha funcionalidade e eficácia a imagem da palavra analisada, cuja seqüência das letras será reconhecida, deverá passar por um módulo de segmentação que segmente esta imagem em uma seqüência de entidades elementares. A Figura 3 mostra o fluxograma de um sistema de reconhecimento de palavras manuscritas.

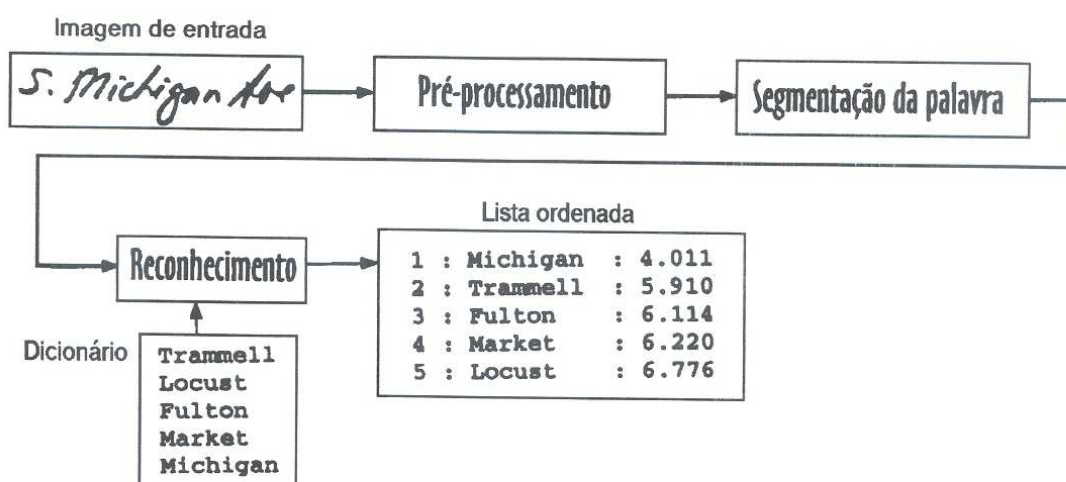


Figura 3 : Fluxograma de um sistema que reconhece texto manuscrito [Kim98]

A segmentação e reconhecimento da palavra são dependentes da qualidade do documento original e da qualidade da imagem registrada (digitalizada) pelo sistema. Alguns sistemas tentam compensar a má qualidade da imagem original e/ou a má qualidade da imagem digitalizada incluindo processos de realce da informação, de remoção de linhas de base, e remoção de ruídos.

O sistema exibido na Figura 3 recebe como entrada uma frase formada por múltiplas palavras e um dicionário de palavras simples. A saída deste sistema é uma lista ordenada que armazena as melhores comparações entre a imagem de entrada e as palavras previstas no dicionário.

Para que o sistema de reconhecimento baseado no fluxograma da Figura 3 realize sua tarefa, é necessário que a imagem da palavra de entrada

tenha sido processada por um algoritmo de segmentação. Esta segmentação tenta fragmentar a palavra em entidades elementares (letras ou partes de letras). Para os fragmentos constituídos por mais de uma letra usam-se algoritmos mais sofisticados (pós-processamento). Em imagens de palavras não uniformes ou de baixa qualidade, até mesmo os algoritmos sofisticados podem não extrair corretamente as entidades desejadas e, deste modo, a ocorrência de erros no reconhecimento é inevitável.

A meta do algoritmo de segmentação de imagens de palavras é de particionar uma imagem de uma palavra em regiões, onde cada região contenha uma letra isolada e completa, ou contenha parte de uma letra da palavra. Para texto impresso a máquina, a imagem de uma palavra é, muitas vezes, segmentada em letras isoladas que, em geral, tornam-se entrada para um módulo de reconhecimento de letras. No entanto, é extremamente difícil segmentar uma palavra manuscrita em entidades elementares sem o suporte de um algoritmo de reconhecimento. Desta maneira, ao contrário do problema de se trabalhar com letras impressas a máquina, o reconhecimento e a segmentação de letras manuscritas são muitas vezes dependentes um do outro [Lu96].

2.2 Técnicas de Reconhecimento

Em textos manuscritos podemos notar uma variabilidade significativa nos estilos de escrita. Por causa desta extrema variabilidade, o reconhecimento de texto manuscrito é um problema de difícil solução. No entanto, temos que tratar de alguma forma este tipo de problema, pois o texto manuscrito continua sendo um meio universal de comunicação. Que meios poderíamos usar para reconhecer a escrita de pessoas em uma mesma categoria? Experimentos psicológicos mostraram que para lermos nós fixamos o texto e realizamos uma rápida tradução (interpretação da imagem). Desta forma, a leitura de um texto deve ser segmentada. Mas como podemos realizar esta segmentação? Em que nível? Nós devemos segmentar para reconhecer ou reconhecer para segmentar [Lecolinet91]?

No geral, a meta dos sistemas de reconhecimento de texto manuscrito é de desenvolver máquinas que sejam capazes de transcrever automaticamente documentos manuscritos tentando adotar o processo de reconhecimento humano [Shridhar97].

Nos *Leitores Ópticos de Caracteres* (OCR), existem duas categorias principais de técnicas de reconhecimento de palavras: *técnicas globais* e *técnicas baseadas em segmentação*. No entanto, alguns modelos de reconhecimento usam características de ambas as técnicas citadas anteriormente que são geralmente chamadas de *técnicas híbridas*.

2.2.1 Técnicas Globais

Estas também são chamadas de técnicas *holísticas*. Estas técnicas não tentam distinguir as diferentes letras da palavra. Elas são geralmente baseadas em uma análise global da palavra ou em uma análise do conjunto de características extraídas da palavra inteira. As características de classificação que são extraídas da imagem da palavra podem ser o número de interrupções verticais, ascendentes, descendentes e ciclos, e uma decisão é tomada de acordo com os classificadores da palavra [Lecolinet91][Lu96].

A principal vantagem destes métodos é que eles superam o difícil problema de segmentação da palavra em letras separadas. Deste modo, as palavras devem ser reconhecidas até mesmo quando alguma letra for omitida ou mal escrita [Lecolinet91].

Sua principal desvantagem é que estas técnicas são hábeis somente para tratar com um vocabulário predefinido. Uma fase de treinamento é necessária para reconhecer um vocabulário novo ou uma nova palavra adicionada ao dicionário predefinido (algumas técnicas baseadas em segmentação necessitam de um dicionário, mas somente de uma transcrição ASCII: nenhum treinamento é requerido). Este método só é viável quando devemos reconhecer pequeno número de palavras (no geral, menos que 100) [Lecolinet91][Shridhar97].

Outra desvantagem é que o tempo computacional e o uso da memória podem ser proibitivos se o tamanho do dicionário for grande, porque é necessário comparar uma palavra desconhecida com todas as possíveis palavras do dicionário [Lecolinet91].

Além disso, do ponto de vista conceitual, pode-se dizer que a requerida fase de treinamento das palavras implica num tipo de dependência do estilo da escrita: existe um alto número de possíveis variações no estilo de se escrever palavras e nunca foi provado que alguma fase de treinamento leve em conta todas estas variações. Isto faz com que estas técnicas fiquem mais confiáveis

para o reconhecimento *mono-scriptor* (um único escritor) e *multi-scriptor* (poucos escritores) do que para o reconhecimento *omni-scriptor* (quaisquer escritores) [Lecolinet91].

Como visto, as técnicas globais podem ser usadas principalmente em duas situações [Shridhar97]:

- Quando as palavras a serem reconhecidas são pré-definidas e a quantidade destas palavras é pequena, e;
- Quando o objetivo principal é a redução do tamanho do dicionário pela eliminação de comparações óbvias, permitindo deste modo, uma maior eficiência. Depois é usada uma técnica de reconhecimento mais intensa no final do processo.

2.2.2 Técnicas Baseadas em Segmentação

A imagem da palavra é particionada em uma seqüência de segmentos. Estes segmentos são reconhecidos pelo classificador de letras e então, uma vez reconhecidas, os segmentos são comparados com possíveis palavras [Lu96].

Estas técnicas podem ser livres de contexto ou direcionadas a um determinado dicionário. O reconhecimento de números de CEPs, ou quantias de cheques bancários, e etc., é sempre baseada em técnicas livre de contexto [Shridhar97].

Podemos ainda subdividir as técnicas baseadas em segmentação em três grupos: técnicas com *segmentação explícita* (ou “externa”), técnicas com *segmentação implícita* (ou “interna”) e técnicas *baseadas em graphemes*.

A segmentação explícita é realizada explicitamente em um estágio prévio, onde o sistema de reconhecimento produz uma seqüência ótima de letras e um dicionário é usado apenas como pós-processamento para selecionar a melhor hipótese.

A segmentação implícita realiza a segmentação e o reconhecimento ao mesmo tempo (a segmentação da palavra em letras é então o “resultado” do reconhecimento), onde a imagem da palavra pré-segmentada é comparada com todas as palavras do dicionário para se obter a melhor hipótese [Lecolinet91][Shridhar97].

A segmentação baseada em grapheme pode estar presente tanto nas técnicas de reconhecimento que realizam uma segmentação explícita quanto nas técnicas que realizam uma segmentação implícita. Nos modelos de reconhecimento que usam técnicas baseadas em graphemes para realizar seu

trabalho, geralmente estão dispostos algoritmos de reconhecimento com alta capacidade de análise. Estes algoritmos devem ter esta característica pois, no geral, os modelos de reconhecimento que utilizam este tipo de técnica tendem a particionar a imagem da palavra gerando muitas entidades a serem combinadas e reconhecidas. Estas entidades podem ser constituídas de letras individuais, partes de letra ou letras sobrepostas (graphemes). Assim, cabe ao algoritmo de reconhecimento organizar as entidades geradas pelo processo de segmentação em uma seqüência que seja compatível (comparável) com uma seqüência prevista pelo sistema (ver seção 2.1).

Como estas técnicas são projetadas para reconhecer letras e agrupamentos de partes de letras (graphemes), a principal vantagem de todas estas técnicas é que a maior parte delas não necessitam de uma fase de treinamento preliminar das palavras. Algumas destas técnicas são até mesmo capazes de reconhecer um vocabulário ilimitado (nenhum dicionário é necessário). Outros métodos necessitam de um dicionário arbitrário (tipicamente uma lista ASCII de possíveis palavras), mas não uma aprendizagem específica das amostras de dados das palavras.

As técnicas baseadas em segmentação têm desvantagens importantes:

- Segmentar a palavra antes de reconhecê-la pode ser problemático. Qual tipo de critério pode ser usado por técnicas explícitas para evitar os numerosos erros que podem ser o resultado de um processo de segmentação arbitrária [Lecolinet91]?
- Segmentação baseada no reconhecimento também pode ser problemático: palavras contendo letras ilegíveis não podem ser identificadas porque a segmentação da palavra falhará se algumas letras não puderem ser reconhecidas. Isto limita o uso das técnicas implícitas a palavras manuscritas cursiva bem escritas [Lecolinet91].
- Se considerarmos palavras contendo letras como “w”, “m”, “d” e pares de letras como “rn”, “nr”, “un”, “iv”, etc., é evidente que muitas possibilidades de segmentação para identificar a letra são possíveis. Ainda existem problemas como diferenciar a letra “o” da letra “a”, especialmente quando as ligações estão envolvidas [Shridhar97].

- Se uma letra da imagem de uma palavra não é reconhecida corretamente pode ser que a palavra inteira não seja reconhecida corretamente. Este é um problema geralmente encontrado em uma seqüência de letras que não correspondem a uma palavra escrita corretamente. Nestes casos, é necessário incorporar um passo de pós-processamento para selecionar as palavras contidas no dicionário usando técnicas de comparação mais avançadas [Shridhar97].
- A disponibilidade de um dicionário de palavras que contenha somente palavras corretas é crucial para desenvolver técnicas eficientes de reconhecimento de palavras manuscritas. Felizmente, em sistemas como o processamento de cheques e reconhecimento de endereços pode-se gerar um dicionário satisfatório. Nos casos de seqüências numerais, o dicionário nem sempre está disponível [Shridhar97].

Também é necessário observar que a segmentação não é um processo local. A extração de uma letra da imagem de uma palavra manuscrita é bastante dependente da letra extraída anteriormente (e sua identificação) e da provável letra que está após a letra corrente. Contextualmente, é claro que se a letra corrente é um “q” (esta letra pode ser facilmente reconhecida como “g”), então é maior a probabilidade de que a letra anterior seja uma vogal e a letra seguinte é a letra “u” com probabilidade 1 (um) [Shridhar97].

Na maior parte dos sistemas de OCR existentes, os algoritmos de reconhecimento de palavras se enquadram nas categorias vistas acima, em que a segmentação das palavras em letras e/ou partes de letras ainda é a parte crítica do trabalho.

2.3 Tipos de Palavras Manuscritas

Palavras manuscritas podem ser divididas em duas categorias: escrita cursiva e escrita em letras de forma. Na prática, a combinação das duas categorias provavelmente ocorra em uma imagem de uma palavra [Lu96]. A Figura 4 mostra um exemplo de escrita em letras de forma e escrita cursiva.

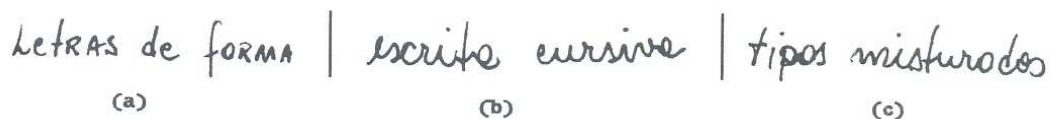


Figura 4 : Escrita em (a) letras de forma, (b) cursiva e (c) misturada

No caso da escrita em letras de forma, ilustrada na Figura 4a, a segmentação é uma tarefa relativamente simples (ver seção 2.3.1). No caso da

escrita cursiva e escrita misturada, ilustradas na Figura 4b e na Figura 4c respectivamente, a segmentação é um processo mais difícil (ver seção 2.3.2).

No geral, em imagens de texto manuscrito, podemos encontrar vários componentes que fazem parte de uma mesma palavra. Por exemplo, podemos encontrar componentes que podem ser uma letra separada da palavra, letras juntas que são parte de uma palavra quebrada, uma palavra inteira, ou ainda, pontuações e acentos da escrita. Estes componentes podem ser observados na Figura 4.

As técnicas de segmentação de palavras manuscritas, geralmente aplicam um passo de pré-processamento para tratar os traços da escrita pertencentes às pontuações, acentos, ou qualquer outro artefato que não pertença ao traço principal da escrita. Este tratamento é realizado para evitar que o processo de segmentação seja prejudicado pela análise de artefatos que não estão previstos no modelo.

Muitas técnicas estão sendo exploradas na tentativa de facilitar o reconhecimento de texto manuscrito (ver seção 2.6). Estas técnicas, geralmente são projetadas para realizar seu trabalho em um dos seguintes tipos de palavras: *escrita cursiva* ou *escrita em letras de forma*. Deste modo, geralmente, as palavras são primeiro classificadas em uma categoria de tipo (cursiva ou letras de forma) e subseqüentemente diferentes esquemas são aplicados dependendo do tipo da escrita [Kim97].

2.3.1 Escrita em Letras de Forma

O texto nesta categoria é produzido supondo-se que ao fim de cada letra o escritor levanta sua caneta e então inicia outra letra. As letras podem ser divididas nas seguintes categorias:

- a. Letras encaixotadas;
- b. Letras espaçadas;
- c. Letras tocando-se;

A palavra nas duas primeiras categorias não apresenta dificuldades em sua segmentação. O maior problema na segmentação da palavra escrita em letras de forma é separar as letras que se tocam. Quando a escrita está em letras de forma, o contato pode ocorrer devido ao estilo da escrita ou devido ao

aglomerado das letras [Lu96]. A Figura 5 mostra um exemplo de cada categoria.

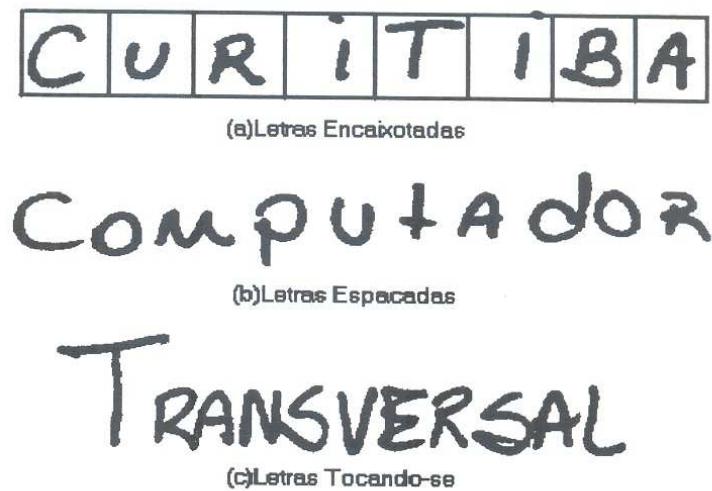


Figura 5 : Categorias de palavras escritas em letras de forma

2.3.2 Escrita Cursiva

Palavras manuscritas cursivas podem conter apenas letras cursivas ou uma mistura de letras de forma com letras cursivas (ver Figura 6). É difícil de segmentar e reconhecer o texto em ambas as categorias. A dificuldade existe devido aos seguintes fatores [Lu96]:

- as letras na escrita cursiva são conectadas em sua maioria,
- as letras individuais em uma palavra cursiva são muitas vezes escritas de maneira que não podem ser identificadas como letras isoladas, e
- a variância no estilo das escritas.



Figura 6 : Exemplo de escrita cursiva pura e escrita cursiva misturada

Uma palavra é, em sua essência, uma seqüência de letras, e isso fez com que uma trajetória natural da pesquisa realizada para o reconhecimento de palavras estivesse baseado na segmentação da palavra em letras e no reconhecimento destas letras [Kim97]. Na maioria das aplicações de reconhecimento é natural assumir que um dicionário está à disposição. O dicionário pode ser estático ou gerado dinamicamente. A tarefa de

interpretação de endereço em uma carta é um exemplo de uma aplicação onde o dicionário é gerado dinamicamente. O CEP no endereço providencia todos os possíveis nomes de cidade ou nomes de ruas como um dicionário. Diferentes CEPs geram dicionários diferentes. Um exemplo de uma aplicação que utiliza um dicionário estático é o leitor de quantias em cheques bancários. Neste caso, o dicionário é fixado em mais ou menos 40 palavras.

2.4 Características que Influenciam na Segmentação

Quando trabalhamos com texto manuscrito, não é difícil de encontrarmos palavras, frases, ou até mesmo, sentenças completas sublinhadas, emolduradas ou inclinadas. Em tais casos, é preciso incorporar um passo de pré-processamento (processamento que antecede a fase de extração das características – veja seção 3.1) formado por técnicas que tratam destes problemas. O tratamento dos casos citados acima deve ser realizado com eficiência, pois a segmentação e o reconhecimento da imagem do texto que exhibe as características citadas podem ser prejudicados (ver seção 3.1).

Em geral, o texto manuscrito pode ter uma ou mais das seguintes características [Shridhar97]:

- **Inclinação das letras:** é muito comum encontrar uma inclinação diferente na escrita da maioria dos humanos. A inclinação é uma medida angular com relação a um eixo vertical de referência. Os ângulos de inclinação estão geralmente no intervalo de -45° a 45° . Muitas vezes, a inclinação não é uniforme complicando a análise do documento. Esta inclinação também pode ocorrer devido a inclinação da linha horizontal do documento.
- **Inclinação da linha horizontal:** Este tipo de inclinação ocorre quando as linhas das palavras formam um ângulo com relação ao eixo horizontal. Isto ocorre muito durante o processo de digitalização de um documento mal alinhado ou quando o escritor introduz este tipo de inclinação por sua inabilidade de escrever sob as linhas de referência que muitas vezes não estão presentes fisicamente.
- **Texto sublinhado:** É muito comum encontrar texto sublinhado em muitos documentos manuscritos. O sublinhado que é muitas vezes ondulado e não reto é destinado para enfatizar algumas características chaves do documento.

- **Linhas sobrepostas:** Este é um problema sério no reconhecimento de manuscrito. Devido ao espaço limitado e o freqüente hábito das pessoas de se escrever rapidamente, palavras de uma linha se intrometem em linhas adjacentes, muitas vezes interceptando-se com palavras das linhas superiores ou inferiores. Tal interferência é tipicamente causada por letras ascendentes (l, t, b, h, etc.) e descendentes (f, g, q, y, etc.) nas palavras.
- **Pontuação imprecisa:** Em documentos manuscritos, é comum a ocorrência de pontuação mal feita no texto. Deste modo, uma vírgula pode ser confundida com um pedaço de letra em seu tamanho e localização.
- **Letras quebradas:** Isto é uma ocorrência muito comum em documentos manuscritos. Deste modo, a barra horizontal de "T" pode estar fisicamente separada de seu membro vertical. Isto também ocorre para letras como "A", "B", "D", "H", "R", etc.
- **Palavras com formas similares:** Em palavras manuscritas, muitas vezes é muito difícil distinguir palavras similares. Por exemplo, distinguir a palavra "clean" da palavra "dean" quando escrita de modo cursivo. Uma interpretação contextual será necessária para resolver este tipo de confusão.
- **Ligações:** Novamente devido à diversidade de estilos de escrita, ligações longas e desenhadas conectando as letras adjacentes muitas vezes confundem a identificação da palavra. Isto é especialmente problemático com palavras que contenham as letras "w", "v", "u", etc. Outros exemplos incluem conexões não desejadas como em "tt", "ff". A obtenção eficaz das informações que fazem referencia a ligações existentes entre as letras de uma palavra é uma ótima maneira de se obter independência do estilo de escrita.
- **Sobreposição de letras:** Uma outra ocorrência comum é a sobreposição de letras em uma palavra. Deste modo, um "t" ou um "T" com a barra horizontal longa muitas vezes sobrepõe as letras das posições adjacentes, dificultando a tarefa de extrair as letras. Tais sobreposições também ocorrem em combinações como "gh", "gy", etc. especialmente quando a parte cíclica inferior das letras "g" ou "y" é muito larga.

É no contexto das observações feitas acima que a maioria das técnicas baseiam-se para desenvolver máquinas capazes de ler documentos manuscritos. É importante notar que mesmo sem a maioria dos problemas citados, o reconhecimento de texto manuscrito ainda é um formidável desafio.

2.5 Segmentação de Escrita em Letras de Forma

O texto nesta categoria é produzido assumindo-se que ao final de cada letra o escritor levanta a caneta para iniciar outra letra. Porém, esta suposição não impede que estas letras se toquem. Os tipos de texto manuscrito em letras de forma são discutidos na seção 2.3 e 2.3.1.

2.5.1 Segmentação Usando Características do Contorno

Casey e Horne [Casey92][Lu96], sugeriram que quando duas letras estão se tocando deve haver uma concavidade no ponto de contato. Desta forma, o algoritmo examina a seção média da imagem das letras que se tocam, analisando as concavidades desta porção da imagem.

A curvatura local da borda é estimada pela direção do contorno. O contorno é obtido por algum algoritmo de análise de contorno. Um ponto de quebra é estimado se existir uma concavidade no contorno e esta não estiver direcionada verticalmente, e se a distância entre o possível ponto de quebra com um ponto no lado oposto do mesmo contorno respeitar um dado limiar. Os pontos de quebra de uma dada imagem formam um conjunto de escolhas de segmentação admissíveis. Uma segmentação da imagem pode ser especificada pela escolha de algum subconjunto destes pontos de quebra. A Figura 7a mostra um exemplo de letras que se tocam e os pontos de quebra candidatos.

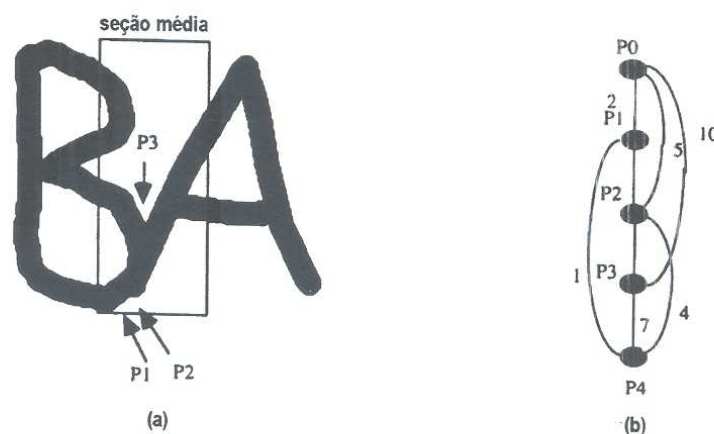


Figura 7 : Exemplo do algoritmo de Casey e Horne [Casey92]. (a) Duas letras manuscritas tocando-se e os pontos de quebra candidatos P1, P2 e P3. (b) O modelo gráfico de (a) [Lu96].

Casey e Horne usaram um modelo em grafo para descrever o controle de um processo de segmentação-reconhecimento de texto manuscrito em

letras de forma. Os nodos internos ao grafo representam os pontos de quebra, o primeiro e último nodos representam os lados esquerdo e direito da imagem e cada ramo do grafo é associado a um custo. A Figura 7b exibe o modelo em grafo correspondente ao exemplo ilustrado na Figura 7a. Os pontos P0 e P4 representam os lados esquerdo e direito da imagem que contém as letras se tocando. Os pontos P1, P2 e P3 são candidatos a pontos de quebra da imagem da palavra exibida na Figura 7a. Cada arco existente entre dois nodos do grafo representam a possível segmentação das letras através da linha formada pelos pontos que formam este arco.

2.5.2 Segmentação Usando Distâncias

Gader et al [Gader92][Lu96], propuseram um algoritmo baseado em distâncias para segmentar palavras manuscritas em letras de forma. O algoritmo consiste de 3 (três) passos principais: segmentação inicial, divisão e comparação. O passo de segmentação inicial primeiro usa o método do componente conectado para calcular os segmentos iniciais das letras. A partir deste ponto, ele detecta e remove todas as marcas de pontuação, e agrupa as letras quebradas conectando as barras horizontais destas letras. A saída do passo de segmentação deve conter uma ou mais letras em cada segmento. O passo de divisão é baseado na técnica da transformação da distância. Este método é clássico em técnicas de processamento de imagens digitais binárias. Esta técnica assume que A é o conjunto de pixels com informação e A^c é o conjunto de pixels de fundo em uma imagem I . Dois tipos de transformações de distância podem ser definidos dependendo da aplicação: para cada pixel x em I é designado um número que é a distância entre cada x e A^c ou x e A . A Figura 8 exibe um exemplo do cálculo das distâncias.

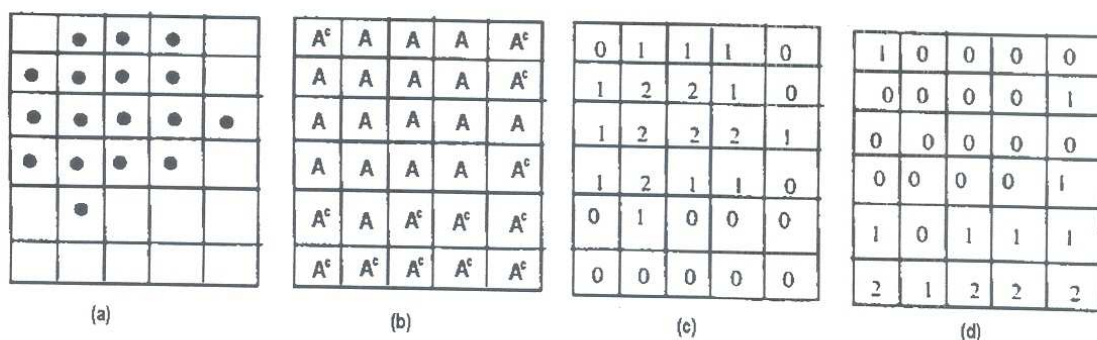


Figura 8 : Exemplo do cálculo das distâncias entre a informação e o fundo [Lu96]. (a) Imagem binária. (b) Informação e fundo. (c) Distância de x para A^c . (d) Distância de x para A .

O algoritmo transforma os pixels de fundo para os valores da distância entre estes pixels e os pixels com informação. Depois da realização da transformação da distância, os pixels, cujo valor da distância é pequeno, formam os possíveis caminhos da divisão. Heurísticas são usadas para definir o ponto de início da divisão e para modificar a função de distância. O resultado do processo de segmentação é uma seqüência de primitivas isoladas, onde cada primitiva contém uma letra ou parte de uma letra. Finalmente, uma rede neural é usada para comparar a seqüência de primitivas do processo de segmentação com as palavras de um dicionário.

2.6 Segmentação de Escrita Cursiva

No geral, não é tarefa fácil tratar com texto manuscrito cursivo. Este tipo de texto pode apresentar características que confundem o sistema de segmentação-reconhecimento no momento da análise da imagem. Este tipo de texto é discutido na seção 2.3 e 2.3.2.

Os algoritmos de segmentação desta categoria, na maioria das vezes, estão envolvidos com uma fase de pré-processamento que precede a fase de segmentação da palavra. Este pré-processamento inclui a remoção de ruídos e correção da inclinação das palavras e das letras.

Uma característica deste tipo de algoritmo é que, no geral, eles tentam segmentar o texto em entidades que podem ser parte de letra, letra inteira ou mais de uma letra.

2.6.1 Segmentação Explícita da Palavra

A segmentação de palavra cursiva geralmente consiste de dois passos principais: *pré-segmentação* e *segmentação*. Existem duas técnicas no passo de pré-segmentação. Enquanto ambas as técnicas tentam segmentar a imagem de uma palavra em uma seqüência de pré-segmentos, uma assegura que cada segmento não contenha mais que uma letra e a outra assegura que cada segmento contenha pelo menos uma letra. No passo de segmentação, os pré-segmentos são novamente processados para alcançar resultados mais confiáveis, ou seja, normalmente o algoritmo deve assegurar que cada segmento contenha apenas uma letra ou parte de uma letra [Lu96].

A característica que estes algoritmos tem em comum é que todos eles têm métodos simples de segmentação e algoritmos sofisticados de

reconhecimento. Estes algoritmos de reconhecimento são robustos o bastante pra tolerar os erros gerados na fase de segmentação. Um exemplo deste tipo de técnica é o algoritmo de reconhecimento de palavras baseado em Modelos Escondidos de Markov (HMM). A maior parte das técnicas baseadas em HMM primeiro segmenta a imagem da palavra em uma seqüência de entidades que podem ser letras ou partes de letras e então usa o HMM para reconhecer estas entidades. Normalmente, o modelo de Markov é robusto o suficiente para reparar erros de segmentação. Desta forma, as técnicas de segmentação usadas neste tipo de algoritmo são simples, usando características observadas na borda superior e borda inferior do contorno externo dos componentes analisados para extrair os melhores pontos de segmentação. Porém, este tipo de técnica pode trabalhar somente com um pequeno vocabulário [Lu96].

Segmentação Baseada em Pontos de Pré-Segmentação (PSP)

Bosinovic e Srihari [Bosinovic89][Lu96], assumem que a imagem da palavra de entrada do sistema não está inclinada, ou seja, a orientação das letras deve ser a vertical, e que o contorno inferior da palavra deve ser contínuo, ou seja, as letras que compõem a palavra devem estar todas conectadas. O algoritmo de pré-segmentação consiste de dois passos principais: estabelecer pontos de pré-segmentação (PSP) e eliminar pontos de pré-segmentação indesejados (compactação). A operação de pré-segmentação corta a imagem horizontalmente em entidades, chamadas de pré-segmentos (PS), onde cada um destes contém letras inteiras ou parciais. O método de pré-segmentação foi baseado na seguinte análise do traço da escrita:

- a imagem de uma palavra é dissecada em três zonas: superior, média e inferior. Veja a Figura 9b;
- nos pontos de mínimo local pertencente à borda inferior do contorno externo da imagem da palavra comumente ocorrem ligações nas proximidades deste mínimo na zona média;
- as ligações vindas de letras com ciclos na zona inferior como “g”, “j” e “f”, correspondem à porção do traço que retém o mínimo local na zona inferior da letra que as precedem, porém;
- esses mínimos podem marcar pontos médios nas seguintes letras: “m”, “u”, “a”, “o”, “b”, “d” e “r”;

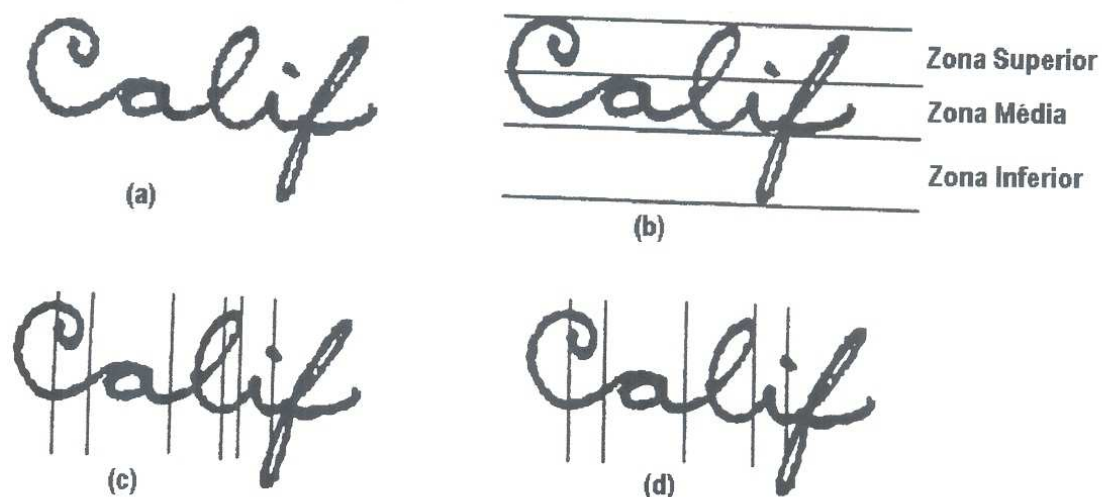


Figura 9 : Segmentação por Bosinovic e Srihari. (a) Imagem original. (b) Três zonas demarcadas. (c) Linhas verticais correspondentes aos PSPs. (d) Os PSPs após o processo de compactação.

Para calcular os PSPs, o algoritmo primeiro busca o mínimo local ao longo da borda inferior do contorno externo da palavra. Para cada mínimo local, o algoritmo “olha” para a esquerda e para a direita dentro de um certo limite para localizar zonas em que PSPs possam ser colocados. Cada uma destas zonas é caracterizada por uma seqüência contínua de “transições simples” na projeção vertical e por uma densidade da escrita (acúmulo de pixels relevantes) menor que um limite predeterminado. Se uma zona é encontrada deste modo, um PSP é colocado no meio dela.

O processo acima pode resultar nos seguintes casos indesejados:

- múltiplos PSPs em uma letra geram letras parciais. Por exemplo a letra “C” na Figura 9;
- em ligações longas é encontrado dois PSPs. Por exemplo, a ligação entre as letras “l” e “i” na Figura 9.

Para remediar o segundo problema, um processo de compactação foi aplicado. O processo busca por seqüências de PSPs cuja distância máxima é menor que um limiar. Se tal seqüência for encontrada, ela é substituída por um PSP dentro da seqüência que tenha o menor valor de densidade do traço nesta seqüência.

O método proposto por Bosinovic e Srihari, utiliza um processo de pré-segmentação que gera entidades do tipo *over-segmentation* (ver seção 4.2). A principal diferença está no método proposto para eliminar os pontos de corte não desejados.

Sayre [Sayre73], provavelmente expôs as primeiras informações sobre os conceitos de *grapheme*, usado em um sistema de reconhecimento off-line de escrita cursiva. Neste trabalho, uma pré-segmentação da palavra em *graphemes* foi realizada baseado na detecção das características da área da imagem. As sub-imagens de saída do processo de pré-segmentação não correspondiam necessariamente a letras, mas a entidades que poderiam ser segmentadas com confiança (tipicamente uma combinação de letras ou um pedaço de letra).

Holt, Beglou e Datta [Holt92], propuseram uma segmentação baseado na análise do contorno superior e em um conjunto de regras baseada na direção do contorno, na detecção de fechamentos e na localização de zonas.

Lecolinet e Crettez [Lecolinet91], propuseram uma solução alternativa com um sistema de reconhecimento *omni-scriptor* e *off-line*. Para a segmentação, eles propuseram um tipo de reconhecimento (chamado de pré-reconhecimento) que distingue dentro das palavras as entidades significantes (nomeadas *graphemes*) das partes não significantes (as ligações). Eles mostraram que podemos obter resultados interessantes no reconhecimento de manuscrito cursivo utilizando um método híbrido de identificação baseado na extração e identificação de entidades significativas de alto nível chamadas *graphemes* (ver seção 4.2). A parte crítica desta técnica é a segmentação da imagem da palavra em *graphemes*. Para isso, eles utilizaram um algoritmo que é baseado em processos competitivos (utilizando informações de ligações para evitar erros de segmentação) e introduziram o conceito de pré-reconhecimento que é um tipo de reconhecimento primitivo usado somente para detectar a presença de caracteres sem ter que reconhecê-lo.

Segmentação Usando Singularidades e Regularidades

Chen et al [Chen92][Lu96], propuseram um algoritmo de segmentação baseado em *singularidades* e *regularidades* que muitas vezes são chamados de *ilhas* (possíveis letras) e *pontes* (possíveis ligações). Estas características são extraídas da imagem de uma palavra depois de serem aplicados uma seqüência de pré-processos, tais como, correção da inclinação, filtragem morfológica e estimativa da largura da ponte. A filtragem morfológica consiste de uma operação de fechamento e uma de abertura com um elemento

estruturante 3×3 (ver apêndice A). O objetivo da filtragem morfológica é remover ruídos, preencher buracos, e suavizar os contornos.

Após a filtragem morfológica, cada pixel de informação é considerado pertencente a uma *ilha* ou *ponte*. As ilhas correspondem a possíveis letras e pontes são os pontos de conexão entre as ilhas. Uma operação morfológica de abertura com elemento estruturante na vertical, cujo tamanho é estimado pela largura da escrita (ver seção 3.6) é aplicado para obter-se as ilhas e então uma operação morfológica de fechamento (ver apêndice A) com um elemento estruturante 5×5 na horizontal (apenas um pixel de largura) é usado para agrupar as ilhas que estão próximas uma da outra. As pontes são candidatas a pontos de segmentação e são encontradas pelo complemento das ilhas. Os pontos de segmentação devem existir somente nas pontes. Os segmentos são encontrados percorrendo-se a imagem da palavra de entrada da ilha mais a esquerda para a ilha mais a direita usando o algoritmo *Viterbi* [Forney73]. Através de uma matriz de custos obtém-se o custo mínimo para percorrer o caminho da segmentação contendo pontos de uma ponte. É claro que cada entidade produzida por esta técnica contém uma letra, ou parte de uma letra ou mais de uma letra. A saída do algoritmo de segmentação deve satisfazer os seguintes critérios:

- cada letra completa pode ser segmentada em no máximo três partes, e
- cada segmento contém no máximo duas letras completas.

Os segmentos resultantes do processo de segmentação são então enviados para um sistema de reconhecimento de palavras baseado em HMM.

Segmentação Usando Características Globais e Locais

Simon [Simon92][Lu96], descreve uma segmentação que utiliza características globais e locais. Primeiro é obtida a esqueletização das letras da imagem binária de uma palavra através do processo de afinamento. A segmentação de uma palavra é realizada encontrando-se o *axis* e o *tarsi* da palavra. O *axis*, também chamado de informação regular, é definido como o caminho mais curto de uma extremidade (esquerda) para outra (direita) que se encontra no corpo da palavra. O *tarsi* é definido como sendo o complementar do *axis* para formar a palavra após o processo de afinamento. A Figura 10 ilustra este conceito.

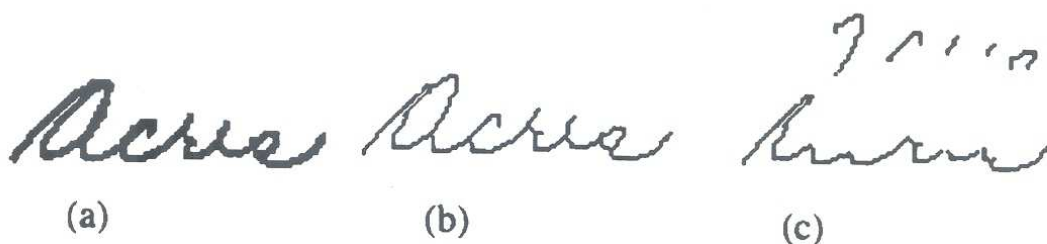


Figura 10 : Exemplo de segmentação usando *axis* e *tarsi* [Lu96]. (a) Imagem original; (b) a imagem afinada; (c) o *axis* abaixo e o *tarsi* no topo.

As singularidades da imagem de uma palavra incluem bifurcações ou cruzamentos, extremidades da linha e ciclos. A segmentação da imagem de uma palavra pode ser realizada combinando características do *axis* e do *tarsi*. Todas as características extraídas incluindo o *axis* e o *tarsi*, os ascendentes e descendentes, as concavidades e convexidades do contorno da imagem, etc., são representados através de uma corrente de símbolos descritivos (SDC – Symbolic Description Chain) para a realização do reconhecimento da palavra. Os detalhes do reconhecimento podem ser encontrados na referência [Simon92].

2.6.2 Segmentação Implícita da Palavra

Estas técnicas são comuns no aspecto em que todas elas dividem a imagem da palavra manuscrita em pré-segmentos que constituem letras, e todas elas usam dicionários de conhecimento no processo de segmentação-reconhecimento. Deste modo, podemos dizer que estas técnicas de segmentação são dependentes de um passo de reconhecimento das letras.

Comparação entre Pré-segmentos e Palavras do Dicionário

Ouladj et al [Lu96], realizaram o processo de segmentação-reconhecimento através da comparação entre os pré-segmentos gerados e as palavras pertencentes a um dicionário. O processo de segmentação está embutido no processo de reconhecimento. A imagem de uma palavra é representada por uma seqüência de códigos que representam as características das letras da palavra analisada. Cada código da seqüência faz referência a quatro características observadas na letra capturadas ao longo da direção da escrita. Durante o processo de captura, cada ocorrência de transição de um máximo para um mínimo foi considerada um possível ponto de segmentação. A seqüência de pontos de segmentação até o ponto corrente

foi comparada com palavras de uma base de dados, que são organizadas em uma árvore de decisão. Um nodo na árvore de decisão representa uma letra do alfabeto e os nodos do caminho que vai da raiz da árvore para uma folha representa uma palavra. Cada letra foi codificada por um par de primitivas (P_1 , P_2), onde P_1 era a seqüência de códigos direcionais definindo a evolução temporal de uma letra e P_2 representando um conhecimento prévio da forma da letra introduzida na base de dados. P_1 foi usado para a classificação. Valores designados para P_2 são iguais a 1, 2, 3 ou 4 que representam respectivamente 4 (quatro) tipos de letras que são: não exista nenhuma linha vertical como em "e", exista um ascendente como em "h", exista um ascendente e um descendente como em "j" ou exista um descendente como em "g". Se uma comparação for possível, a primitiva P_1 é detectada e uma letra associada com o par (P_1 , P_2) é prognosticado. Depois de uma letra ter sido identificada, um ponto de segmentação é obtido, e a próxima letra na palavra de entrada é procurada. Todos os pontos de segmentação são armazenados de maneira que quando uma letra for rejeitada ou uma letra encontrada for idêntica a uma folha do dicionário, o processo pode retroceder para tentar prognosticar outras letras do ultimo ponto de segmentação retido. Deste modo, o algoritmo conduz para uma lista de palavras candidatas, discriminadas por um critério definido como a adição dos pesos de reconhecimento designada para cada letra identificada.

Kimura e Shridhar [Kimura93], propuseram um algoritmo que usa um dicionário para conduzir a segmentação e o reconhecimento da palavra.

A imagem de entrada deve ser binária e o processo começa com a correção da inclinação das letras. A segmentação em letras consiste de um passo de pré-segmentação e um passo de segmentação. Durante o passo de pré-segmentação, o algoritmo primeiro detecta os pontos de mínimo do contorno superior da imagem de entrada. Os pontos de pré-segmentação são mínimos locais que estão no fundo dos máximos locais adjacentes. Se um ponto de mínimo não tem uma abertura vertical para cima, o algoritmo altera o ponto de mínimo para a esquerda ou para a direita tentando encontrar um ponto ótimo considerando uma função de avaliação, e escolhe este ponto ótimo como ponto de pré-segmentação. A imagem da palavra é então fisicamente dividida por linhas verticais passando pelos pontos de pré-segmentação. Os componentes conectados e seus limites (acima, abaixo, à

esquerda e à direita) são calculados na imagem da palavra dividida. Cada componente conectado, referente a um segmento primitivo, é parte de uma letra. Os limites, que são normalmente disjuntos e não incluem partes de outros componentes conectados, são ordenados da esquerda para a direita de acordo com a localização de seus centróides. Se o limite de dois ou mais componentes mostrarem a mesma coordenada x como centróide, eles são ordenados de cima para baixo. Para cada palavra do dicionário, os segmentos primitivos da palavra de entrada são agrupados e comparados com as letras das palavras do dicionário até que a probabilidade de ser a letra correta seja maximizada.

O problema da segmentação torna-se um processo simples de Markov onde sua tarefa é de calcular os limites das letras. Para calcular a probabilidade total do reconhecimento da letra, uma rápida rotina de extração de características é aplicada a cada conjunto de segmentos primitivos ao longo de uma letra. Um vetor de características é construído utilizando os limites dos segmentos primitivos e as características intermediárias da palavra extraídas antes do processo de segmentação e reconhecimento. Devido a normalização da altura para o cálculo da probabilidade, uma verificação de consistência da altura relativa da letra é usada em um pós-processamento no resultado da segmentação e reconhecimento.

Os processos de segmentação-reconhecimento e o pós-processo são repetidos para todas as palavras do dicionário e as palavras são organizadas de acordo com a combinação linear da probabilidade das letras serem pertencentes àquela determinada palavra.

Segmentação Baseada na Busca em Grafo

Nohl et al [Nohl92][Lu96], propuseram técnicas baseadas na busca em grafo para segmentar-reconhecer a imagem de uma palavra. Antes de se realizar a busca em grafo, alguns passos de pré-processamento foram aplicados na imagem da palavra de entrada para normalizar a imagem.

Após a normalização, a imagem da palavra foi primeiro dividida em um conjunto de cortes que não se interceptavam e que faziam referência a uma seqüência de células que representavam porções da imagem da palavra candidatas a letra. Os cortes foram numerados de 1 até N da esquerda para a direita, onde a borda do lado esquerdo da imagem da palavra é numerada

como 0 (zero) e a borda do lado direito é numerada como $N+1$. A segmentação da imagem de uma palavra é uma seqüência de segmentos, onde cada segmento consiste de uma ou mais células adjacentes.

Os três métodos para encontrar o caminho da segmentação correta foram:

- encontrar a melhor pontuação de segmentação para cada entrada do dicionário;
- combinar as entradas do dicionário com segmentações alternativas para formar um simples problema de caminho ótimo, onde exista uma correspondência de um-para-um entre os caminhos do grafo e uma segmentação legítima das entradas do dicionário; e
- encontrar a K melhor segmentação-interpretação independente do dicionário e então realizar pós-processos para eliminar interpretações não pertencentes ao dicionário.

O primeiro método empregou o algoritmo *Viterbi* [Forney73] para encontrar a melhor segmentação e a interpretação da palavra foi restringida à comparação do resultado da segmentação com entradas de um dicionário particular. Este processo foi realizado uma vez para cada palavra do dicionário produzindo uma pontuação de reconhecimento para cada palavra. A partir deste ponto, um segmento consistia de pelo menos uma célula, onde o processo de segmentação tentava encontrar o melhor encaixe entre as células. Um segmento da imagem da palavra foi identificado pela especificação de um par $L-R$, onde L era o índice do ponto de corte do lado esquerdo do segmento e R o índice do ponto de corte do lado direito do segmento. Para ajudar no cálculo dos segmentos, um grafo foi construído.

Assim, cada segmentação da imagem da palavra teve um caminho correspondente no grafo. A pontuação para uma segmentação particular era o produto da pontuação associado com os nodos ao longo do caminho correspondente, onde a melhor segmentação foi calculada usando o algoritmo de *Viterbi* para maximizar a pontuação. O sistema de reconhecimento da palavra avaliou cada palavra do dicionário e produziu uma pontuação para cada uma destas. A palavra do dicionário com a maior pontuação foi designada como a interpretação mais provável. A vantagem desta técnica é que ela providencia uma pontuação para cada palavra do dicionário, e o grafo construído é relativamente pequeno. A principal desvantagem é a ineficiência computacional quando o sistema se depara com um dicionário muito grande.

O segundo método integra as limitações impostas pelo dicionário de forma que ao encontrar a maior pontuação de segmentação e a interpretação desta segmentação comparada com o dicionário forem satisfatórias a segmentação ótima foi atingida. Este método produz somente a uma possibilidade (a mais provável) de segmentação e interpretação, porém, ele pode melhorar significativamente o tempo computacional quando o algoritmo de reconhecimento das letras for confiável.

O terceiro método calcula os K melhores caminhos sem nenhum tipo de restrição. Este método é mais atrativo para aplicações onde a segmentação e o reconhecimento são confiáveis, ou onde os dicionários são muito grandes no espaço de possíveis interpretações.

2.7 Conclusão

O problema de segmentação de palavras manuscritas em letras de forma é relativamente fácil e muitas técnicas para segmentar texto impresso a máquina em letras podem ser aplicadas para este problema [Lu95].

A segmentação e reconhecimento de palavras manuscritas cursivas são problemas mais difíceis de se tratar e a pesquisa que é feita para estes problemas é relativamente recente. No reconhecimento de palavras cursivas, o processo de segmentação é muitas vezes dependente do reconhecimento com visto na seção 2.6.2. O intercâmbio entre um processo de segmentação e um processo de reconhecimento é relevante desde que temos consciência de que para reconhecer uma letra temos que saber onde ela começa e onde ela termina, levando em conta que para isolar uma letra precisamos reconhecê-la primeiro.

Acredita-se que por causa da complexidade deste problema, um bom sistema de reconhecimento de palavras manuscritas deve ser hábil para analisar a palavra localmente, onde, por exemplo, deve-se observar os pontos de pico e vale juntamente com traços que possivelmente sejam ligações entre letras, e analisar a palavra globalmente, onde, por exemplo, deve-se observar os ciclos, as concavidades, os ascendentes e descendentes (ver seção 2.2).

Capítulo 3

Análises e Correções na Imagem da Palavra

3.1 Pré-Processamento

Em muitos pontos deste trabalho citamos a necessidade de se realizar uma fase de pré-processamento antes que o processo de análise e segmentação seja aplicado.

No geral, é na fase de pré-processamento que o sistema prepara a imagem da palavra para futuros processamentos. Um dos processos que constam nesta fase é o de estimar e corrigir a inclinação tanto da imagem do texto quanto a inclinação das letras que compõe o texto de um imagem (ver seção 3.4). Isto é necessário porque, desta forma, podemos prever o corte vertical nos pontos de segmentação definidos por um algoritmo de segmentação de palavras [Shridhar97].

Outro processo muito utilizado na fase de pré-processamento é a suavização (ver seção 3.5). A captura digital de imagens pode introduzir ruídos na imagem através do dispositivo de captura. Além disso, após a correção da inclinação das letras da palavra, podem surgir deformações no contorno da palavra que são considerados ruídos. Tanto os ruídos da captura quanto os ruídos da correção da inclinação precisam ser tratados. Operações de suavização são usadas para eliminar os ruídos introduzidos durante a captura da imagem e correção da inclinação [Kim97].

Ainda na fase de pré-processamento, pode-se realizar algumas estimativas necessárias para a fase de análise e segmentação. Uma estimativa

interessante, para ser usada como informação em processos futuros, é o cálculo da espessura média dos traços da escrita (ver seção 3.6). Este tipo de informação é geralmente usado na detecção de ligações entre letras e, deste modo, usado na detecção de possíveis pontos de segmentação.

Como visto acima, o objetivo destes pré-processos é de ajudar o algoritmo de segmentação a ser mais eficiente e preciso. Isto ocorre porque a fase de pré-processamento prepara a imagem da palavra para futuros processamentos.

3.2 Obtenção dos Contornos da Palavra

As bordas caracterizam os limites entre a escrita e o fundo. Pontos de borda podem ser encontrados através da localização dos pixels que mudam abruptamente de cor. Um exemplo ideal é obtido através de uma imagem binária onde a fronteira entre os objetos e o fundo é uma mudança de um pixel branco para preto ou vice-versa. Este conceito é discutido em mais detalhes nas seções 3.2.1 e 3.2.2.

3.2.1 Contorno por Morfologia

A obtenção dos contornos da palavra através da morfologia é uma operação bastante simples e rápida. Basicamente esta operação pode ser obtida através de dois processos morfológicos [Facon98]: *erosão* ou *dilatação*. Se a operação morfologia usada for a *erosão*, o contorno da imagem é o resultado da imagem erodida subtraída da imagem original. Caso a operação morfológica usada for a *dilatação*, o contorno da imagem é o resultado da imagem original subtraída da imagem dilatada. O resultado do processo de obtenção do contorno da Figura 13a pela operação de *erosão* e pela operação de *dilatação* e exibido na Figura 11.

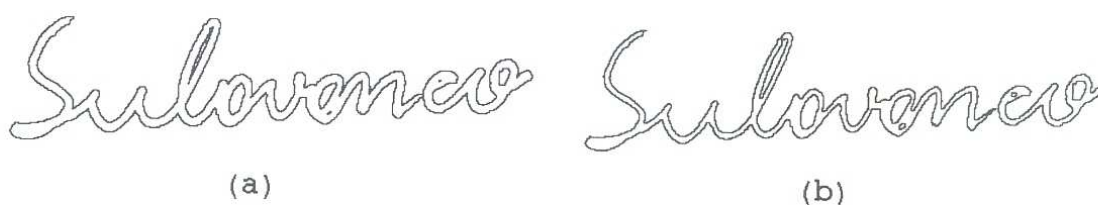


Figura 11 : Contornos por morfologia. (a) Por dilatação e (b) por erosão

3.2.2 Contorno pela Seqüência da Informação (Chain Code)

Quando queremos analisar algum tipo de informação em uma imagem digital, como uma região ou conjunto de pontos, algumas representações compactas estão disponíveis para facilitar a manipulação e análise desta informação. Desta maneira, podemos representar a região com a informação relevante descrevendo seu contorno.

A meta deste processo é encontrar os pontos da imagem analisada que definem a região limite entre a informação da escrita e o fundo da imagem. É através dos pontos do contorno da palavra que o algoritmo de segmentação analisa uma palavra.

Esta representação é baseada no trabalho de Freeman [Freeman78]. O processo segue os pontos do contorno da região analisada obedecendo a uma direção horária para contorna-la. Este processo mantém o rastro das direções que foram utilizadas na seqüência dos pontos quando passamos de um ponto do contorno para o próximo ponto. O código associado com as possíveis direções do sistema *chain code* é exibido na Figura 12, onde "X" representa a posição do ponto corrente no contorno.

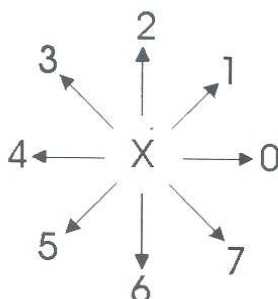


Figura 12 : Definição da seqüência de informação (chain code) com oito vizinhos.

O método *chain code* permite a codificação de configurações geométricas arbitrarias. A linha reta que permite a mudança entre dois pontos adjacentes do contorno é chamada de vínculo, e uma cadeia é formada como uma sucessão ordenada dos códigos destes vínculos.

A seqüência de informação (chain code) apresenta algumas propriedades:

- Os códigos pares {0,2,4,6} correspondem a direções horizontais e verticais. Os códigos ímpares {1,3,5,7} correspondem às direções diagonais.

- Cada código pode ser considerado como uma direção angular, em múltiplos de 45° , que nós temos que mover para ir de um ponto de contorno para o próximo ponto.
- A coordenada absoluta (m,n) do primeiro pixel de contorno (ou seja, o ponto que está na extrema esquerda e no topo do conjunto de pontos) junto com o código da seqüência do contorno representa uma descrição completa do contorno da região.
- Quando há uma mudança entre dois códigos de seqüência sucessivos, então o contorno mudou de direção. Este ponto é definido como um *canto*.

Após a obtenção do contorno da palavra, o sistema cadastra estes pontos para representarem a imagem de entrada. Este processo é discutido na seção 3.3. A Figura 13a ilustra uma palavra cursiva de entrada do sistema e a Figura 13b ilustra os pontos do contorno cadastrados através do método da seqüência da informação (chain code).



Figura 13 : Contornos pela seqüência da informação (Chain Code). (a) Imagem original e (b) contorno da palavra

3.3 Cadastramento dos Contornos da Palavra

Todo o processamento é baseado nas informações obtidas através de um método que extrai as coordenadas dos pontos do contorno das palavras contidas na imagem. Este é um método compacto de representação da imagem que reduz os dados a serem analisados e, conseqüentemente, reduz o tempo de processamento do sistema [Kim96]. Este método resulta no cadastramento linear das informações observadas na trajetória traçada pelos elementos que delimitam as informações relevantes do fundo da imagem. O fundo da imagem é caracterizado por pontos que não pertencem a nenhum traço da escrita contido na imagem. Nós podemos dizer que o cadastramento da palavra é a criação de uma lista que contém as coordenadas dos pontos dos contornos da palavra. A obtenção destes contornos pode ser realizada através de um dos métodos descritos na seção 3.2. As informações cadastradas estão na forma

de coordenadas obedecendo a estrutura (x,y) , onde x corresponde ao posicionamento da informação no eixo horizontal e y corresponde ao posicionamento da informação no eixo vertical.

3.4 Correção do ângulo de Inclinação das Letras

3.4.1 Conceitos da Estimativa da Inclinação

A inclinação das letras de uma palavra é obtida analisando-se o ângulo que existe entre o eixo que corresponde à direção média das letras e o eixo vertical. Através da estimativa do ângulo de inclinação das letras de uma palavra pode-se aplicar um processo de correção que torne o eixo correspondente à inclinação média das letras o mais próximo possível do eixo vertical.

Existem duas técnicas principais para estimar a inclinação das letras de uma palavra [Shridhar97]: *método da projeção* e *método da seqüência da informação*. A seguir, está uma breve descrição destas técnicas.

3.4.2 Método da Projeção

A inclinação média das letras em uma palavra ou em uma linha é estimada pela análise de projeções verticais (histogramas) com inclinação em vários ângulos. A inclinação média é encontrada através da busca da maior derivada positiva entre todas as projeções feitas [Shridhar97].

3.4.3 Método da Seqüência de Informação (Chain Code)

A inclinação média das letras em uma palavra é facilmente estimada usando segmentos da borda (contorno) da palavra [Lee93]. A inclinação média é dada por

$$\theta = \tan^{-1} \left(\frac{n_1 + n_2 + n_3}{n_1 - n_3} \right) \quad (3.1)$$

onde n_i é o número de elementos dispostos a um ângulo de i vezes 45° (/ ou | ou \). Uma transformação de ajuste, discutida na seção 3.4.4, é então aplicada para corrigir a inclinação das letras.

Para estimar a inclinação média das letras, somente as bordas verticais e quase verticais são usadas, ou seja, segmentos da borda cuja inclinação

encontra-se entre 45° e 135° . Os segmentos da borda que estão fora deste intervalo são descartados. Se os segmentos horizontais da palavra são removidos, as bordas das letras são separadas em segmentos verticais e segmentos que tendem a verticais dentro do intervalo especificado. Deste modo, a inclinação de cada um destes segmentos é calculada pela equação (3.1), e a média global também é calculada pela mesma equação.

A Figura 14 ilustra o cálculo da inclinação média dos segmentos verticais e quase verticais usando a equação (3.1).

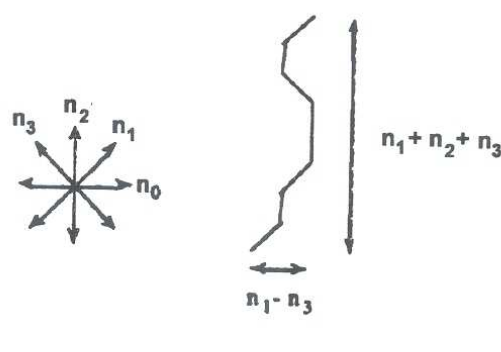


Figura 14 : Exemplo ilustrativo do cálculo da inclinação da palavra [Shridhar97]

3.4.4 Conceitos da Correção da Inclinação

A etapa de correção da inclinação das letras de uma palavra é requisito em sistemas de segmentação, pois a maior parte dos estilos de texto manuscrito é formado por letras inclinadas. Esta inclinação é prejudicial ao processo de segmentação-reconhecimento e por isso deve ser tratado.

Como visto na seção 3.4.1, existem alguns métodos para se estimar o ângulo de inclinação das letras de uma palavra. Uma vez que o ângulo global θ da inclinação das letras de uma palavra foi estimado, podemos aplicar um modelo de correção a esta inclinação.

A tangente do ângulo θ , é usada para corrigir a inclinação destas letras [Kim97]. A equação (3.2) mostra o ajuste feito nos parâmetros da coordenada de um ponto que faz referência ao eixo horizontal (coordenada x) e eixo vertical (coordenada y) da imagem,

$$\begin{aligned} x_n &= x - y \cdot \tan(\theta) \\ y_n &= y \end{aligned} \tag{3.2}$$

onde x e y representam as coordenadas horizontal e vertical originais do ponto analisado, e x_n e y_n representa as coordenadas horizontal e vertical corrigidas do ponto analisado.

A Figura 15 mostra um exemplo de correção da inclinação das letras de uma palavra.



Figura 15 : Correção da inclinação: (a) imagem da palavra de entrada, (b) pixels das bordas verticais e próximas a verticais, (c) imagem da palavra após a correção da inclinação [Shridhar97]

3.5 Suavização da Escrita

A suavização é geralmente utilizada para eliminar pequenos ruídos e/ou para tornar o contorno da imagem mais regular em sua seqüência.

Muitos modelos de suavização foram propostos pela comunidade científica. As seções abaixo descrevem duas destas técnicas.

3.5.1 Suavização por Morfologia

A suavização por morfologia realiza uma operação morfológica de *fechamento* seguido por uma operação morfológica de *abertura* usando um elemento estruturante adequado para remover ruídos de um determinado tipo [Facon98].

3.5.2 Suavização pela Rotação da Máscara

A suavização através da rotação de uma máscara percorre a imagem da esquerda para a direita e de baixo para cima utilizando uma máscara 3 X 3 como exibido na Figura 16.

X	X	X
=	T	=
=	=	=

Figura 16 : Máscara de suavização por rotação

O pixel no centro da máscara é marcado. Todos os pixels que estiverem sobrepostos pelo “X” da máscara são descartados, enquanto que os que estão marcados com “=” são analisados da seguinte maneira: se todos os pixels tiverem os mesmos valores, ou seja, se todos forem “1” ou se todos forem “0”, então o pixel central é forçado a receber o valor igual ao da maioria, caso contrário, o valor do pixel central não é alterado. Esse processo é realizado 4 (quatro) vezes, uma para cada possível rotação da máscara. O resultado dessa suavização é que pequenas endentações e pequenas saliências são removidas.

Para obter mais informações sobre este processo de suavização, procure pela referência [Stratry93].

3.6 Cálculo da Espessura Média da Escrita

A espessura da escrita de uma palavra pode variar localmente dependendo do dispositivo usado para escrever e do papel em que se está escrevendo. Assim, este tipo de informação pode minimizar os efeitos de se usar dispositivos diferentes para escrever. Esta informação pode ser utilizada por procedimentos de processamento de imagem de uma maneira adaptativa (detecção de componentes pequenos e etc.).

Para estimar a espessura média da escrita, geralmente o contorno externo da imagem da palavra é dividido horizontalmente em borda superior e borda inferior. A Figura 17 mostra este conceito.

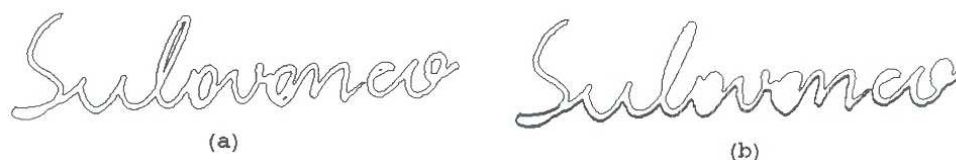


Figura 17 : Divisão do contorno externo. (a) Contornos interno e externo, e (b) borda superior e inferior.

Em [Kim97], Gyeonghwan e Venu propuseram um modelo de estimativa da espessura do traço da escrita que analisa o contorno externo do texto do ponto mais a esquerda ao ponto mais a direita. Deste modo, as seguintes distâncias são calculadas para cada coordenada x :

- Distância entre a borda superior e inferior do contorno externo;
- Distância entre a borda superior do contorno interno e a borda superior do contorno externo;

- c. Distância entre a borda inferior do contorno interno e a borda inferior do contorno externo;

Através desta análise consegue-se obter a espessura do traço da escrita para cada coordenada horizontal do sistema de coordenadas da imagem da palavra. A espessura média do traço da escrita é determinada para ser o valor da espessura que mais aparecer no traço da escrita da palavra, ou seja, o valor da espessura com maior ocorrência na escrita será denominado como sendo o valor da espessura média. A Figura 18 ilustra a estimativa da espessura média do traço da imagem de uma palavra.

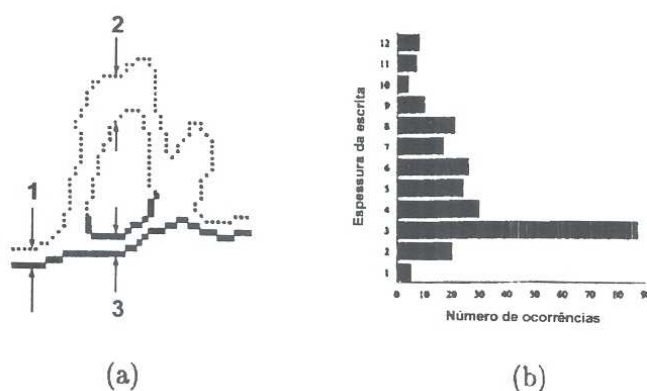


Figura 18 : Estimativa da espessura média da escrita. (a) Cálculo das y-distâncias para cada coordenada x. (b) Análise das ocorrências das espessuras. [Kim97]

O histograma na Figura 18b ilustra o número de ocorrências de cada valor das distâncias encontradas no traço da escrita da imagem de uma palavra. O cume do histograma fornece uma estimativa da espessura média através da equação (3.3).

$$\begin{aligned} N_d &= \text{número_de_ocorrências_do_valor_de_distância_d} \\ P_d &= \text{valor_máximo}(N_d) \end{aligned} \quad (3.3)$$

3.7 Análise da Imagem da Palavra

No geral, é necessário que a imagem da palavra que será alvo de análises tenha passado por uma fase de pré-processamento para corrigi-la (ver seção 3.1). Os processos de correção da imagem original, que antecedem o processo de análise, geralmente deixam a imagem da palavra apta para que se possa realizar uma busca pelos prováveis pontos de segmentação desta palavra. Estes pontos de segmentação (ou os limites da letra) são então detectados para que a divisão seja realizada. Muitas vezes, as técnicas de

segmentação são projetadas para gerar imagens com pedaços de letra (*over-segmentation*). Os segmentos de letra são agrupados em uma letra inteira se o processo de reconhecimento da letra ou o reconhecimento da palavra for realizado com sucesso [Shridhar97].

A detecção de pontos de segmentação é baseada na análise das formas da imagem da palavra. Esta detecção comumente baseia-se na *análise do contorno* e/ou na *análise do perfil* da palavra [Shridhar97].

3.7.1 Análise do Contorno

O método da análise do contorno é aceitável para obter imagens com pedaços de letra (*over-segmentation*). Possíveis pontos de segmentação são detectados pela análise do extremo local do contorno superior da imagem da palavra. Entre os mínimos locais, aqueles que são próximos a um ponto de máximo local adjacente são removidos.

No geral, para se obter componentes isolados da imagem de uma palavra, os pontos de segmentação determinados no passo anterior devem ser deslocados horizontalmente, para a esquerda ou para a direita, como segue: se o ponto mínimo não está em uma abertura que represente um vale na borda superior, o ponto é deslocado para a direita ou para a esquerda deste ponto dependendo de onde a espessura do traço analisado é mínimo (veja a seção 3.6). A Figura 19 ilustra este processo.



Figura 19 : Pontos de pré-segmentação [Shridhar97]

3.7.2 Análise do Perfil

Ao invés de analisar o contorno superior de uma palavra, este método analisa o perfil superior. O perfil superior é definido como sendo o conjunto dos pixels com relevância mais ao topo em cada coluna. Um pós-processamento é requerido para encontrar pontos de segmentação que não foram encontrados por causa de segmentos de letras como a barra do "t". A Figura 20 ilustra este conceito.



Figura 20 : Perfil superior da palavra

3.7.3 Segmentação em Caixas Disjuntas

Letras e segmentos de letras são manipulados mais facilmente se seus limites (que formam uma caixa envolvendo a entidade analisada) são mutuamente disjuntos (sem interseção). Se for permitido *over-segmentation*, a sobreposição horizontal de segmentos de letras é solucionada por um algoritmo simples. A imagem de uma palavra é dividida verticalmente em cada ponto de pré-segmentação e é separada em zonas sem sobreposição. Uma análise de componentes conectados é aplicada na imagem dividida para detectar as caixas incluindo cada componente conectado. Estas caixas são usualmente disjuntas e não incluem partes de outros componentes conectados. A Figura 21 ilustra esta análise.

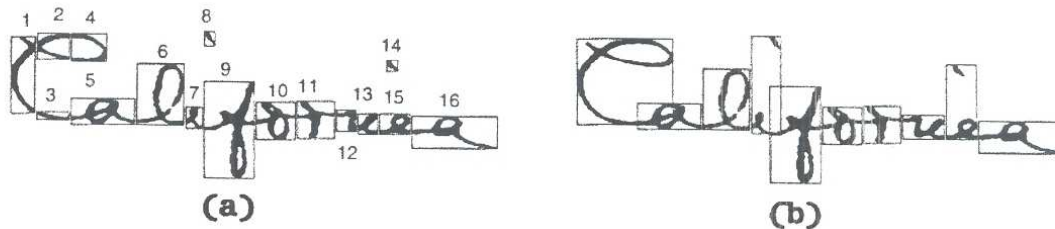


Figura 21 : Segmentação em (a) caixas disjuntas e (b) ideal [Shridhar97]

Capítulo 4

Segmentação Baseada em Grapheme

4.1 Reconhecimento por *Grapheme*

Uma propriedade específica de palavras manuscritas cursiva é que os componentes da escrita não têm o mesmo nível de significado [Lecolinet91]:

- Ligações são somente partes não significantes do estilo;
- Enquanto que as letras são consideradas como a parte informativa da palavra.

Além disso, algumas letras podem ser muito mais significantes que outras para um dado contexto como mostra a Figura 22, que contém dois exemplos da palavra “cinquenta”. Embora a segunda palavra tenha sido escrita sem algumas letras, ela deve ser facilmente reconhecida por um leitor automático brasileiro (pelo menos no contexto de quantias literais de cheques). Isto pode ser explicado da seguinte forma: como o número total de possíveis palavras é pequeno para o dado contexto, o reconhecimento de somente algumas letras da palavra (por exemplo, as letras “c”, “q”, “t” e “a”) é suficiente para um módulo de reconhecimento identificar esta palavra [Lecolinet91].

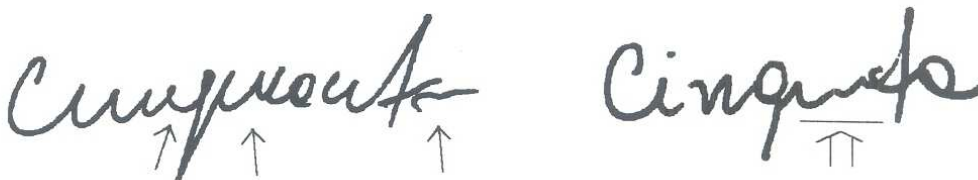


Figura 22 : Partes significantes e não significantes da palavra

Um sistema de identificação *omni-scription* deve ser hábil para focar o reconhecimento nas partes mais significantes da palavra e descartar os dados

não significantes como as ligações. O sistema também deve realizar algum tipo de interpretação global de todas essas entidades significantes para permitir a identificação da palavra mesmo que algumas letras estejam omitidas, com erros ou mal escritas [Lecolinet91].

Todas estas considerações nos levam aos conceitos da técnica de identificação híbrida (ver seção 2.2), ou seja, um método que utiliza tanto os conceitos da técnica analítica (baseada em segmentação), por ser hábil na extração das entidades significantes da palavra (durante a fase de segmentação), quanto os conceitos da técnica global, como o processo de identificação final feito de uma maneira global [Lecolinet91].

Este conceito enfatiza o papel crucial de um processo de segmentação inteligente. A fase de segmentação deve acontecer antes do reconhecimento das entidades extraídas (segmentação explícita). Finalmente, a palavra deve ser identificada globalmente por meio de um estágio de comparação contextual, comparando a lista completa das sub-imagens da palavra (os graphemes) com as prováveis palavras do dicionário (ver seção 2.1). O método não deve tentar reconhecer com precisão todas as letras, e sim colecionar bastante informações da palavra inteira [Lecolinet91].

4.2 Definição de Graphemes

Uma alternativa para uma técnica de segmentação é dividir a imagem de uma palavra de entrada em sub-imagens que não são necessariamente letras individuais, como mostra a Figura 23. É desejado que as formas preliminares destas sub-imagens, chamadas de *graphemes* ou *pseudo-letras*, sejam enquadradas em classes prontamente identificáveis [Casey96].



Figura 23 : Imagem da palavra dividida em sub-imagens. (a) original e (b) sub-imagens

Por definição, *grapheme* é a saída do processo de segmentação de uma palavra. *Graphemes* são aquelas partes da palavra que “não olhamos como” uma ligação, ou seja, “olhamos como” uma parte da informação da palavra (letra ou parte da letra). Na prática, *graphemes* são entidades de alto nível que

se assemelham com letras, mas não podem ser denominadas letras [Lecolinet91].

Provavelmente, as primeiras informações sobre este conceito foram expostas em [Sayre73], onde é feito um relato sobre um sistema de reconhecimento off-line de escrita cursiva. Neste trabalho, uma pré-segmentação da palavra em *graphemes*, foi realizada baseado na detecção das características da área da imagem. As sub-imagens de saída do processo de pré-segmentação não correspondiam necessariamente a letras, mas em formas específicas que poderiam ser segmentadas confiantemente (tipicamente uma combinação de letras ou um pedaço de letra) [Casey96].

Um *grapheme* é definido como uma entidade produzida por um processo de segmentação, ou seja, a segmentação da imagem de uma palavra pode produzir entidades que retêm uma letra inteira (correct-segmentation), parte de uma letra (over-segmentation) ou mais que uma letra (under-segmentation).

4.3 Extração de Grapheme

Assim como em [Sayre73], o conceito de grapheme (ver seção 4.2) foi muito aplicado em pesquisas sobre escrita cursiva. A segmentação de uma palavra em graphemes é referenciada como pré-segmentação, pois a segmentação final é atribuída ao processo de reconhecimento da palavra. As técnicas de pré-segmentação de escrita cursiva são baseadas em regras heurísticas derivadas de observações visuais. Não existem regras “mágicas” e não é possível segmentar todas as palavras manuscritas em letras perfeitamente separadas na ausência de reconhecimento. Assim, as sub-imagens resultado do processo de pré-segmentação da palavra não são somente letras inteiras, mas também partes ou combinações de letras (graphemes). As relações entre letras e graphemes devem permanecer simples o bastante para permitir um estágio eficiente de pré-reconhecimento. Na prática, isso significa que uma letra deve ser decomposta em poucos graphemes e um único grapheme represente no máximo uma letra completa. Os graphemes que constituem mais de uma letra deve ser percebido (ver seção 4.5) e tratado (ver seção 4.6) [Casey96].

Uma propriedade importante existente no conceito de grapheme é que estas entidades consideram um pré-segmento da palavra como objeto de

avaliação. As características de um grapheme são mais invariantes que características de entidades geradas a partir da análise baseada em técnicas globais que consideram ciclos, curvas, pontos de cruzamento, etc., como objetos de avaliação. Além disso, o reconhecimento através de graphemes é menos sensível aos ruídos criados pelas ligações. Isto é ilustrado na Figura 24 que mostra a instabilidade e a inexatidão de uma representação uniforme baseada nas entidades geradas por uma técnica de reconhecimento global [Lecolinet91]:

- no primeiro caso (Figura 24a), duas palavras diferentes (neuf=nove & sept=sete em francês) tem representações muito similares, mas como a ligação permanece não detectada a informação que diferenciaria as palavras é perdida.
- no segundo caso (Figura 24b), a mesma palavra (trente=trinta) escrita em diferentes estilos (cursivo em um caso e em letras de forma no outro) resulta em representações muito diferentes porque a ligação entre as letras causa um “ruído de fundo” que aparece mais importante que a “mensagem escrita”.



Figura 24 : Extração de características por técnicas globais. (a) palavras diferentes com representações semelhantes e (b) mesma palavra com representações diferentes [Lecolinet91]

Por outro lado, a segmentação e o reconhecimento de graphemes fazem decrescer a influência das ligações porque as ligações podem ser detectadas e consideradas como sendo entidades não importantes [Lecolinet91].

Um dos últimos estudos sobre técnicas baseadas no princípio de detectar ligações foi baseado em dois processos [Casey96]:

- Detecção de possíveis zonas de segmentação (detecção das ligações);
- Detecção da letra (ou o uso de um algoritmo de pré-reconhecimento, cuja meta não é a de reconhecer a letra, mas avaliar a probabilidade de uma sub-imagem ser uma letra válida).

As zonas de segmentação são detectadas pela análise das bordas superior e inferior do contorno externo da palavra e concavidades abertas das palavras. A tentativa de se criar um caminho possível de segmentação é definido como sendo a tentativa de se dividir a palavra em graphemes isolados. Estes caminhos são escolhidos respeitando algumas regras heurísticas que expressam constrangedores de continuidade e conectividade. De qualquer maneira, esta pré-segmentação só é validada se ela for consistente com as decisões do algoritmo de reconhecimento [Casey96].

O objetivo do processo de detecção de letras é detectar as partes da escrita que provavelmente são letras para evitar a descoberta de características que podem ser consideradas como ligações pelo algoritmo de segmentação mas que são partes de alguma letra. É interessante notificar que não se trata de um algoritmo de reconhecimento de letras, e desta maneira este algoritmo não é hábil para decidir qual a letra escrita, mas somente detectar se é provável que uma parte da escrita pertença a uma letra. Como o algoritmo realiza alguns reconhecimentos básicos, embora ele não consiga reconhecer a letra, é chamado de pré-reconhecimento [Lecolinet91].

4.4 Detecção das Ligações

Os segmentos de linha que formam as conexões entre as letras na escrita cursiva são conhecidos como “ligações” (veja seção 4.5). Deste modo, algumas técnicas de pré-segmentação para escrita buscam “ligações baixas”, ou conexões próximas da linha base da palavra que muitas vezes ligam as letras. De qualquer forma, a qualidade da segmentação ainda está muito dependente da efetividade do processo de pré-segmentação que produz os graphemes. Um modo simples de localizar ligações é detectar um ponto de mínimo do contorno superior da palavra. Infelizmente, este método traz alguns problemas não resolvidos [Casey96]:

- As letras “o”, “b”, “v” e “w” são normalmente seguidas por ligações altas;
- As letras “u” e “w” contêm ligações internas, ou seja, uma sub-parte destas letras não podem ser diferenciadas de uma ligação sem o auxílio de contexto;
- Ligações algumas vezes causam erro na segmentação;

A detecção das ligações também pode ser baseada na detecção dos vales do contorno superior da imagem da palavra. Considere dois tipos diferentes de vales:

- A *projeção dos vales* no contorno superior, que pode ser alcançada pela projeção vertical, é caracterizada pelo mínimo local, cercado por dois máximos locais (um em cada lado), e com uma profundidade relativa suficiente entre este mínimo e os dois máximos que o cercam;
- As *concauidades laterais* do contorno superior, que não pode ser alcançado diretamente por causa da presença de quebras horizontais ou por causa da inclinação da escrita. Estas são detectadas quando existe um espaço vazio entre dois pontos sucessivos do contorno superior como no exemplo da Figura 25 (entre “d” e “e”). Estas concauidades são sistematicamente examinadas por meio do algoritmo *cursor móvel* que tenta encontrar o ponto mais próximo da concauidade, enquanto evita um mínimo local não significativo. Um exemplo do caminho de um cursor móvel é também mostrado na Figura 25;

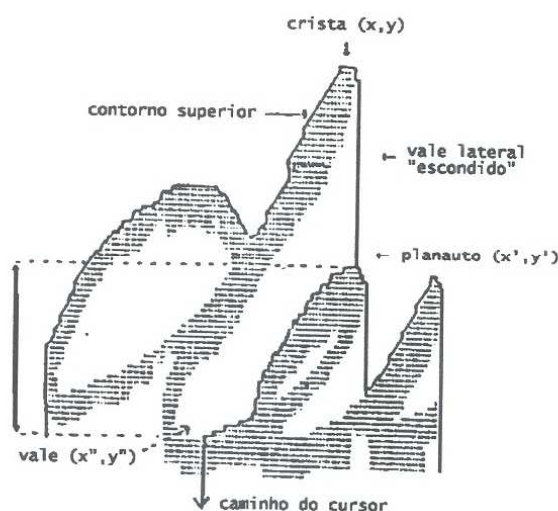


Figura 25 : Projeção e concauidades laterais [Lecolinet91]

4.5 Detecção de ligações como pré-segmentação

Um algoritmo de segmentação de texto manuscrito deve ser geral e robusto para manipular os vários estilos e espessuras do traço da escrita [Kim97]. Deve-se considerar o seguinte:

- O número de segmentos por letra deve ser limitado a um valor máximo (por exemplo, uma letra só pode ser particionada em quatro graphemes), e

- Todas as letras que se tocam devem ser separadas.

As palavras manuscritas são geradas a partir do movimento da caneta da esquerda para a direita ao longo de um eixo na direção horizontal, dando origem ao que chamamos de ligações entre letras. As ligações são fortes candidatas para pontos de segmentação na escrita cursiva e podem ser extraídas da seguinte forma: se a distância entre as coordenadas y da borda superior e borda inferior do contorno externo para uma dada coordenada x for menor ou igual a espessura média da escrita (ver seção 3.6), então a coordenada x é marcada como sendo um elemento de uma ligação. Este processo é repetido para todas as coordenadas x . As ligações nas extremidades podem ser eliminadas para reduzir o número de pontos de segmentação. A Figura 26d mostra as ligações obtidas para uma imagem de exemplo [Kim97].

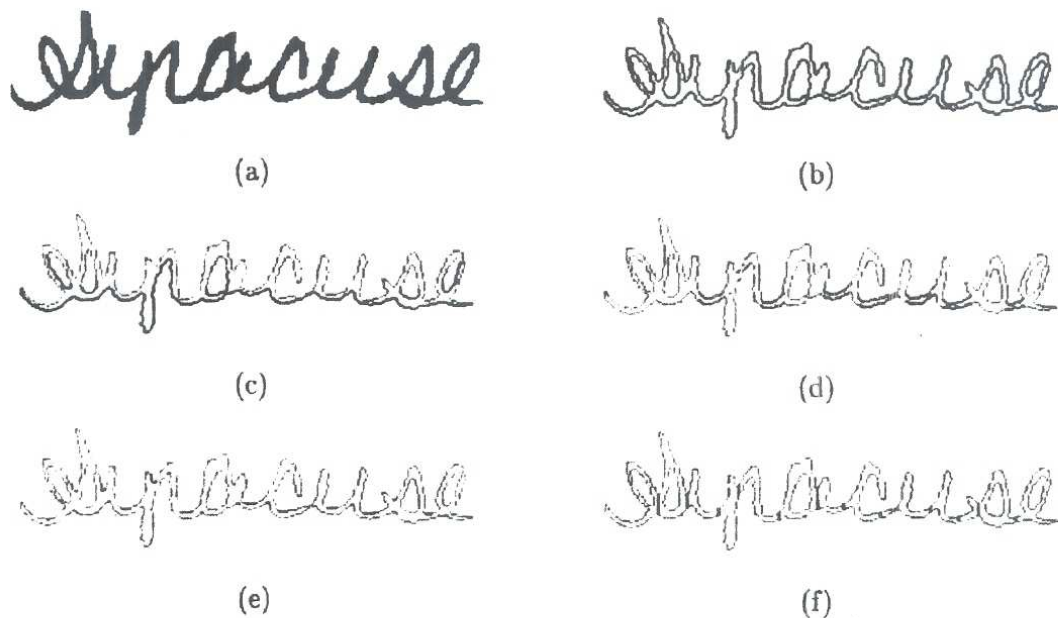


Figura 26 : Segmentação. (a) Imagem original. (b) Representação do contorno externo e interno após a correção da inclinação. (c) Divisão dos contornos superior e inferior. (d) Ligações baseadas na espessura média. (e) Concavidades/convexidades. (f) Pontos de segmentação. [Kim97]

Por outro lado, pontos de segmentação em uma escrita cujas letras se tocam não podem ser estimadas apenas por ligações. Alternativamente, características de concavidade no contorno superior e convexidades no contorno inferior são usados em conjunto com as ligações (ver Figura 26e) para realizar a segmentação. Heurísticas são aplicadas para reduzir o número de pontos de segmentação, ou seja, se as ligações e concavidades estão sobrepostas, a concavidade é ignorada, e se a concavidade e convexidade estão

sobrepostas em uma coordenada x, um ponto de segmentação é designado para a coordenada x [Kim97].

4.6 Tipologia de Ligações

A experiência mostra que muitas das ligações entre letras cursivas interceptam o ponto de mínimo local (ou “vales”) do perfil da borda superior da palavra, como mostra a Figura 27.

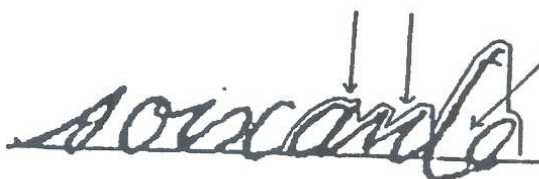


Figura 27 : Detecção dos pontos de mínimo do perfil do contorno superior [Lecolinet91]

Porém, como mostram as indicações da figura acima, existem algumas exceções para esta regra:

- Nem todos os traços que interceptam o ponto de mínimo (os vales da borda superior) são ligações:
 - Alguns vales podem ser enganosos quando ciclos não estão fechados (Figura 28a);
 - Alguns vales estão dentro da letra. Isto ocorre principalmente com as letras “u”, “v”, “w” ... , onde uma parte destas letras parece com uma ligação, mas é uma parte real da letra (Figura 28b);
- Algumas ligações não têm vales na borda superior:
 - Algumas letras estão muito juntas e/ou muito difíceis de se distinguir (Figura 28c);
 - Algumas ligações correspondem ao cume do contorno inferior. Este caso ocorre principalmente depois das letras “o”, “b”, “v” e “w” (por exemplo “oi” etc.) (Figura 28d);

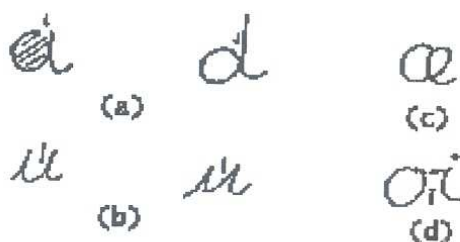


Figura 28 : Tipologia de ligações [Lecolinet91]

O pré-reconhecimento deve evitar que os vales detectados no caso da Figura 28a sejam considerado como ligações, pois eles acontecem dentro da letra.

O caso da Figura 28b e Figura 28c produzem respectivamente um segmento com parte de uma letra e um segmento com duas letras porque não existe modo de prevenir isto sem a ajuda do contexto: a estrutura da letra é de tal forma que ela pode ser confundida com duas letras juntas (por exemplo um “u” ser visto como dois “i” sem o ponto e juntos). Como isto é um problema contextual que geralmente requer a análise da palavra inteira, é resolvido no estágio final da comparação contextual entre a lista de *graphemes* extraídos e a referência ASCII das palavras no dicionário. Durante este estágio, os *graphemes* podem ser combinados por meio de regras contextuais de re-associação.

O caso da Figura 28d requer um procedimento complementar especial para detectar e segmentar corretamente este tipo de ligação. O princípio é o mesmo da detecção e segmentação de vales e contornos superiores.

4.7 Processo de Pré-Reconhecimento

Como exposto anteriormente, o processo de pré-reconhecimento é usado para evitar a segmentação de certas partes da letra que pode ser identificado erroneamente como ligações. Este processo é baseado na análise da concavidade e alguns processos complementares.

4.7.1 Análise da Concavidade

Este processo é realizado para detectar ciclos que não estão totalmente fechados (por causa da palavra ter sido escrito apressadamente ou com problema na tinta ...). Este processo permite a extração de informações relevantes (a presença de ciclos) e evita alguns erros de segmentação.

Um algoritmo específico foi desenvolvido para resolver este difícil problema (o algoritmo deve evitar a detecção de concavidades que não são ligações, mas ele deve também ser hábil para detectar a maior parte dos “ciclos abertos”). Este algoritmo trabalha de um modo complementar:

- O “*outside*” das letras é estimado (a área de fundo que envolve a letra);

- As “*concavidades*” (as áreas com possíveis ciclos) são obtidas por subtração (cada um destes é um conjunto de pixels conectados que pertencem ao fundo, mas não pertencem ao “*outside*”);
- Estas *concavidades* são classificadas considerando sua área e seu contorno:
 - Se esta área é muito pequena, a *concavidade* é somente um artefato insignificante;
 - Se N_c , o número de pontos do contorno que são conectados por um pixel do “*outside*” é nulo, a *concavidade* é um ciclo fechado;
 - Caso contrário, N_c é comparado com N_t , que representa o número total dos pontos do contorno, para determinar se a *concavidade* é um “*ciclo aberto*” ou uma mera “*concavidade*”

Mais precisamente, esta comparação é melhorada levando em conta a *direção* dos pontos do contorno. Isto é mostrado na Figura 29, onde a relação N_c/N_t é quase o mesmo para a *concavidade* dos dois caracteres, mas totalmente diferente se a direção (por exemplo, as direções horizontal e vertical) é levada em conta. Este último critério faz uma melhor avaliação das propriedades tipológicas das *concavidades* e isto deve corresponder à noção intuitiva de um “*ciclo*”.

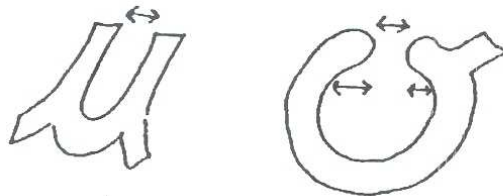


Figura 29 : Ciclos abertos e fechados [Lecolinet91]

4.7.2 Análise da Extensão

Este processo permite invalidar certas hipóteses de segmentação por causa de sua *y*-localização, pois as ligações são supostamente fechadas na “*parte central*” da escrita cursiva. A parte central da palavra (ou seja, a parte horizontal entre as extensões superior e inferior) é determinada calculando-se uma linha base superior e inferior por meio da técnica do histograma. Então, ascendentes e descendentes são extraídos, e como é suposto que eles fazem parte de alguma letra, a segmentação é evitada diante destas características.

4.7.3 Análise da Forma e Tamanho

Este processo (que também é usado na detecção de caracteres numéricos) leva em conta o número de ciclos, o tamanho relativo, etc., para estimar se o componente analisado é provavelmente uma letra real, parte de uma letra ou uma combinação de letras fundidas. Neste caso, ele é usado para refinar o resultado de estágios anteriores e para realizar um re-agrupamento útil quando algumas letras são quebradas em várias partes.

Capítulo 5

Metodologia da Pesquisa

5.1 Conceitos da Metodologia

Como exposto no Capítulo 1, a comunidade envolvida com a pesquisa na área de análise e reconhecimento de imagens de palavras manuscritas tem relatado muitas técnicas que abordam o problema de segmentação e reconhecimento de texto manuscrito.

Este estudo considerou os conceitos das técnicas reportadas nos capítulos anteriormente mencionados para desenvolver um algoritmo de segmentação de imagens de palavras manuscritas.

Deste modo, dedicamos nossa pesquisa para o desenvolvimento de uma técnica de segmentação de palavras manuscritas que analise uma imagem de uma palavra manuscrita de entrada e retorne as entidades primitivas que formam esta palavra. Cada entidade primitiva pode pertencer a uma das seguintes categorias: *graphemes* ou *ligações*.

Este trabalho será validado sobre uma base de dados formada por imagens do extenso de valores numéricos, podendo ser expandido para um sistema de segmentação *omni-scriptor*.

5.1.1 Etapas do Processo de Segmentação

No geral, o algoritmo de segmentação deve ser baseado, mas não limitado, aos seguintes passos:

- Detecção dos contornos interno e externo;
- Detecção dos pontos de mínimo e máximo dos contornos superior e inferior da palavra;

- Segmentação baseada em regras como:
 - Presença de ciclos;
 - Histograma de transição na direção da segmentação;
 - Espessura média do traço da escrita;
- Estimativa da direção da segmentação (quebra);
- Corte das ligações baseadas em:
 - Histograma de transição;
 - Histograma de densidade;
 - Comparação contextual entre entidades vizinhas;
- Pós-processamento
 - Estimativa da largura média dos segmentos;
 - Divisão de segmentos largos;

Deve-se observar que os passos de segmentação listados acima podem estar inseridos ou não em um determinado modelo de segmentação. Alguns modelos de segmentação ainda podem trabalhar com características não previstas pelo modelo exposto acima.

Esta variância entre modelos de segmentação de texto manuscrito é diretamente dependente do modelo de reconhecimento que se propõe utilizar para analisar as entidades geradas no passo de segmentação. As técnicas de reconhecimento de texto manuscrito podem ser observadas na seção 2.2, onde são expostas as características que cada modelo de reconhecimento pode assumir.

A Figura 30 ilustra as etapas que o modelo de segmentação proposto realiza para quebrar a palavra.

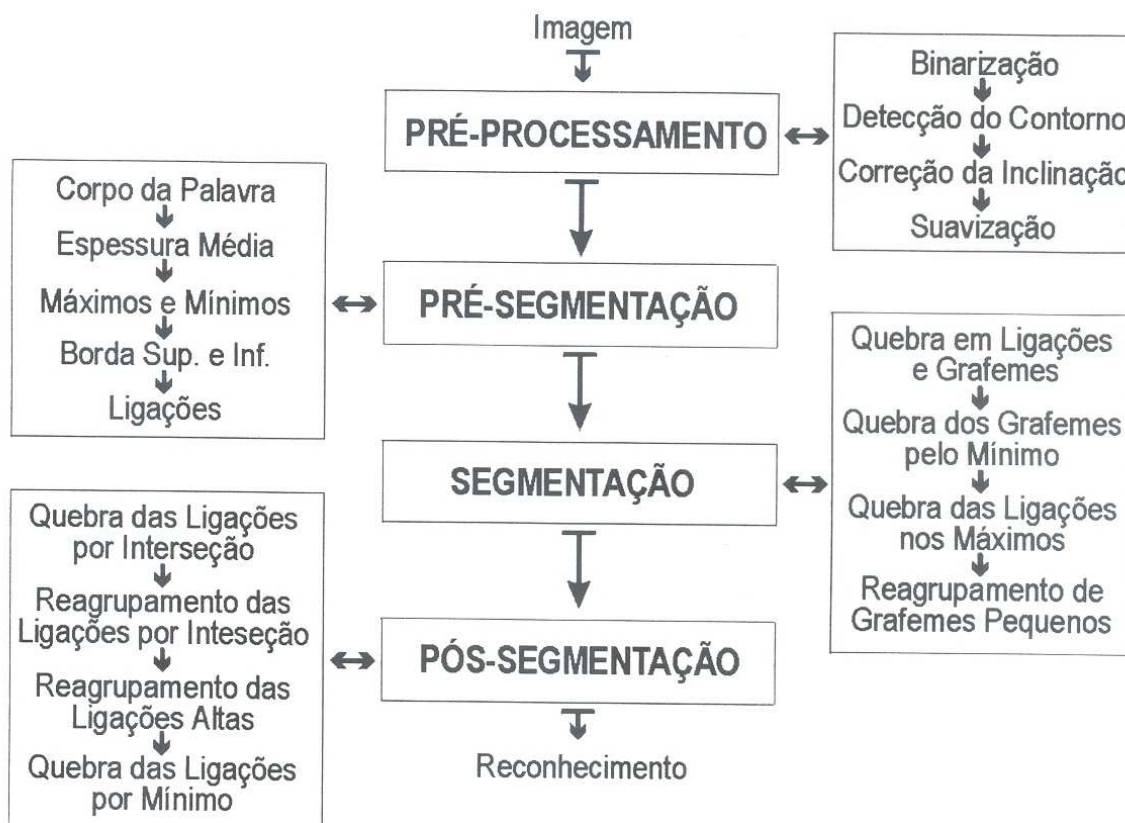


Figura 30 : Fluxograma do processo de segmentação proposto

5.2 Estudo Bibliográfico

Este trabalho iniciou-se através de um estudo bibliográfico com o propósito de reter o estado da arte no que diz respeito à segmentação de palavras manuscritas cursivas (ver seção 2.6). Esta é a etapa em que buscamos o auxílio da literatura para direcionar nosso trabalho e elaborar uma proposta consistente e promissora, colocando em discussão as técnicas e métodos de segmentação existentes.

5.3 Base de Dados de Validação

A base de dados usada na validação deste trabalho foi criada por estudantes de doutorado da Pontifícia Universidade Católica do Paraná – PUC/PR. Esta base foi criada no ano de 1998 com o propósito de validar uma pesquisa de doutorado desenvolvida na instituição. Esta base é formada por aproximadamente 12.000 palavras de extensos de valores numéricos de cheques bancários brasileiros. A base está dividida em 40 outras bases que

representam os extensos possíveis de aparecer em cheques bancários brasileiros, ou seja, o extenso das palavras exibidas na tabela tal.

Tabela 1 : Palavras do dicionário de validação

um	nove	dezesete	setenta	seiscentos
dois	dez	dezoito	oitenta	setecentos
três	onze	dezenove	noventa	oitocentos
quatro	doze	vinte	cem	novocentos
cinco	treze	trinta	duzentos	mil
seis	quatorze	quarenta	trezentos	reais
sete	quinze	cinquenta	quatrocentos	centavos
oito	dezesesseis	sessenta	quinhentos	cento

Em uma primeira fase, a validação desta pesquisa de mestrado foi efetuada sob um subconjunto de 736 palavras retiradas da base citada acima. Esta redução na base de validação é necessária por se tratar de um processo visual de análise, ou seja, a eficiência do algoritmo de segmentação é estimada *analisando-se visualmente todas as imagens processadas*. Desta maneira, optamos por validar a pesquisa com uma base de dados que tenha amostras o suficiente e que consuma um tempo reduzido. As palavras que formam a base de validação foram escolhidas da seguinte forma: para cada categoria de palavras existente, que neste caso são 42, foram escolhidas em média 40 palavras aleatoriamente. Assim, garantimos que a base fora formada com imagens de palavras que tenham características de escrita que se assemelham com as escritas do mundo real.

A base de dados de origem foi montada com imagens no formato *bitmap padrão*, ou seja, com extensão BMP. O sistema de segmentação foi adequado a este formato de imagem gráfica.

5.4 Pré-processamento

5.4.1 Conceitos

O processo de segmentação de palavras, seja em suas unidades básicas (letras) ou em entidades de alto nível como *graphemes*, geralmente está relacionada com algumas técnicas de pré-processamento.

Neste trabalho, antes que a palavra seja analisada, a imagem de entrada passa pelos seguintes pré-processamentos (ver seção 3.1):

- Suavização da imagem da palavra (ver seção 5.4.2),
- obtenção dos contornos externo e interno da palavra (ver seção 5.4.3),
- cadastramento dos pontos de contorno da palavra (ver seção 5.4.4),
- estimativa da espessura média dos traços da escrita (ver seção 5.4.5), e
- correção da inclinação global das letras da palavra (ver seção 5.4.6).

Os processo de suavização e correção da inclinação são realizados para normalizar a imagem de modo a deixá-la apta a uma etapa de análise.

Os processo de obtenção e cadastramento dos contornos da palavra são realizados para conseguirmos uma redução nos pontos de informação da imagem a serem analisados, deixando somente os pontos de interesse para serem alvo de análises. Esta etapa é realizada de forma que todos os pontos que se encontram no limite da palavra com o fundo (pontos de fundo são pontos que estão fora da escrita) são armazenados seqüencialmente em uma lista de pontos.

A espessura do traço da escrita é uma informação valiosa neste trabalho, pois é através desta informação que podemos estimar parâmetros de correção e análise que tenha relação com o tipo de palavra que está sendo processada. É também através desta informação, que estimamos traços da escrita com grande probabilidade de ser ligações que existam entre as letras e, deste modo, aumentar o grau de confiança na estimativa dos pontos de segmentação. Este processo é discutido na seção 5.4.5.

Este trabalho analisa imagens binárias. Deste modo, o primeiro processo realizado na fase de pré-processamento é o de verificar se a imagem da palavra é binária. Caso a imagem de entrada esteja em um formato diferente o sistema transforma esta para o formato adequado.

5.4.2 Suavização da Escrita

Como visto na seção 3.5, o processo de suavização da imagem normalmente é realizado com a finalidade de eliminar pequenos ruídos e de tornar o contorno da escrita mais regular e suave.

O método utilizado neste trabalho para suavizar o contorno dos traços da palavra também é hábil para realizar o preenchimento de pequenas falhas

que possam aparecer nestes traços. Este método é nomeado como “Suavização pela Rotação da Máscara” e está descrito na seção 3.5.2 e na referência [Stratry93].

As principais características deste tipo de suavização são a eliminação de endentações no contorno da imagem da palavra e a eliminação de pequenos ruídos que possam aparecer no traço da escrita ou no fundo da imagem. O resultado deste processo pode ser observado na Figura 31.

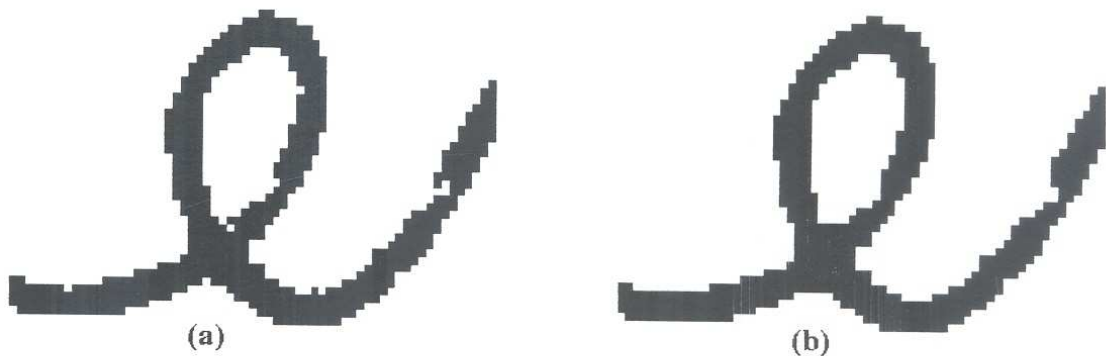


Figura 31 : Suavização pela método "Rotação da Máscara". (a) Imagem original e (b) imagem suavizada.

A imagem da letra “e” cursiva exibida na Figura 31a faz parte de uma das imagens binárias de palavras que constam na base de validação desta pesquisa.

O processo de suavização dos traços da imagem da palavra é aplicado na imagem original de entrada e na imagem cuja inclinação foi corrigida. Desta maneira, conseguimos uma suavização mais intensa e com melhores resultados.

Através deste modelo de suavização, obtivemos resultados satisfatórios tanto no objetivo de sua tarefa quanto no tempo de processamento.

5.4.3 Obtenção dos Contornos da Palavra

Como visto na seção 3.2, o contorno da palavra de entrada do sistema é uma forma compacta de se representar a informação relevante para o processo de segmentação.

Neste trabalho, o processo de obtenção dos contornos interno e externo da palavra são realizados pelo método da seqüência de informação (chain code), descrito na seção 3.2.2.

Este método utiliza códigos direcionais para decidir quais pontos pertencem ao contorno da palavra.

Após a obtenção do pontos do contorno da palavra, o sistema cadastra estes pontos em uma lista seqüencial para representar a imagem de entrada, como descrito na seção 5.4.4.

5.4.4 Cadastramento dos Contornos da Palavra

Quase toda a análise realizada na imagem da palavra é feita sob a *palavra cadastrada*. O termo “palavra cadastrada” refere-se a uma lista ordenada que retém as coordenadas dos pontos dos contornos da palavra de entrada do sistema. Os contornos externo e interno da palavra são obtidos através do modelo descrito na seção 5.4.3. Após a obtenção dos pontos que fazem parte do contorno da palavra é criada uma lista através do cadastramento seqüencial das coordenadas destes pontos.

O cadastramento dos pontos de contorno da palavra gera duas listas de pontos: *lista principal* e *lista secundária*. Na lista principal são cadastrados os pontos do contorno externo dos componentes do texto de entrada que estão em contato com a linha média da palavra. A detecção da linha média da palavra é descrita na seção 5.5.2 e é ilustrada pela Figura 32a. Na lista secundária são cadastrados os pontos do contorno interno da palavra dos componentes que tocam a linha média da palavra e os pontos do contorno interno e externo dos componentes que não tocam a linha média da palavra. A ilustração da Figura 32b indica os pontos que fazem parte da lista secundária

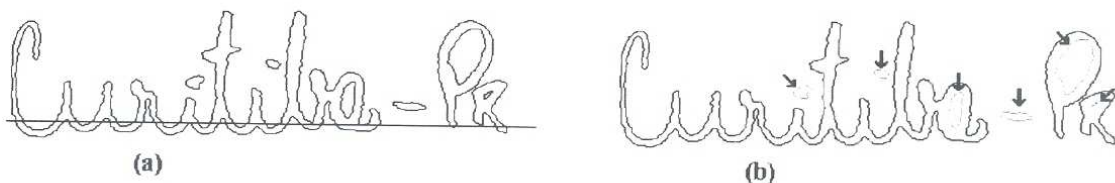


Figura 32 : Cadastramento da lista principal e da lista secundária. (a) Detecção da linha média da palavra. (b) Pontos da lista secundária.

O termo *palavra cadastrada* refere-se aos pontos que interessam a esta pesquisa, ou seja, os pontos da lista principal.

Como dito acima, nem todos os processos dependem da palavra cadastrada. Somente os processos de suavização, detecção do corpo da palavra e detecção do contorno, dispensam o uso da palavra cadastrada por utilizarem as informações retiradas diretamente da imagem.

5.4.5 Estimativa da Espessura Média da Escrita

Como visto na seção 2.3, ao trabalharmos com texto manuscrito nos deparamos com uma grande variabilidade de estilos e dispositivos de escrita. Esta variabilidade pode gerar erros no processo de análise dos traços da palavra caso o sistema não observe e não se adapte à nova escrita que está sendo processada.

O modelo de segmentação proposto neste trabalho utiliza o valor da espessura média como uma informação bastante valiosa, pois é através desta estimativa que os procedimentos de segmentação da imagem da palavra se adaptam às mudanças da palavra quanto ao seu estilo e dispositivo de escrita.

Na seção 3.6, abordamos um modelo de estimativa da espessura média que para realizar sua tarefa precisava separar os contornos interno e externo da palavra em bordas superior e inferior do contorno interno, e borda superior e inferior do contorno externo. Neste trabalho, conseguiu-se obter as mesmas informações de uma maneira menos trabalhosa e mais rápida. Isto foi realizado através da projeção vertical da palavra observando-se as transições entre o fundo e os traços da escrita. Assim, a espessura de cada traço da escrita para cada posição horizontal é estimada.

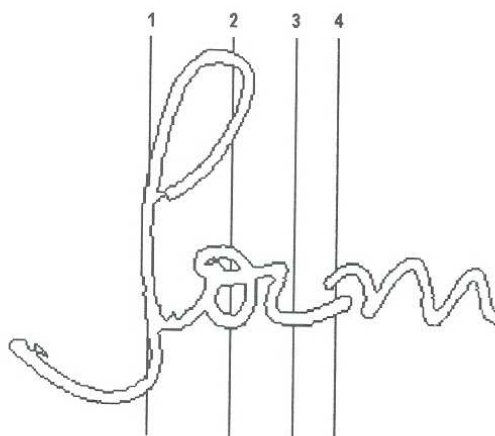


Figura 33 : Estimativa da espessura média pela projeção vertical

Na Figura 33 pode-se observar quatro exemplos de projeções verticais. Nestes exemplos, cada interseção da linha de projeção com o traço da escrita gera um valor para uma espessura local do traço.

A linha vertical “1”, contém duas interseções com o traço da escrita, onde uma destas superestima a espessura do traço da escrita por se tratar de uma estimativa em um traço vertical.

A linha vertical “2”, intercepta o traço da escrita cinco vezes e produz cinco valores de espessura local do traço. A linha vertical “3”, intercepta o traço da escrita uma única vez e produz somente um valor de espessura local. A linha vertical “4”, intercepta o traço da escrita duas vezes produzindo dois valores de espessura local.

Este modelo de estimativa da espessura média da escrita projeta verticalmente os traços da escrita e, desta maneira, cria um histograma onde cada posição horizontal do histograma representa um valor de espessura e cada barra vertical indica o número de ocorrências dos traços da escrita que tem aquela espessura.

Através desta análise consegue-se obter a espessura do traço da escrita para cada coordenada horizontal da imagem da palavra. A espessura média do traço da escrita é determinada como sendo o valor da espessura que mais aparecer no traço da escrita da palavra, ou seja, o valor da espessura com maior ocorrência na escrita.

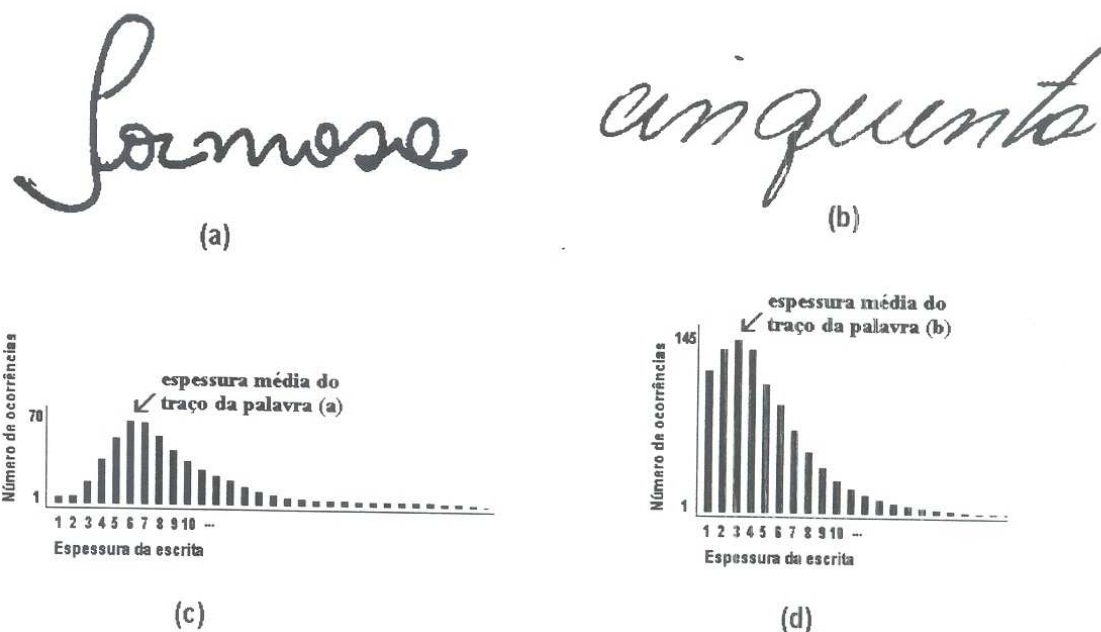


Figura 34 : Estimativa da espessura média da escrita. (a) e (b) Imagens de palavras. (c) e (d) Histogramas das imagens (a) e (b) respectivamente.

A Figura 34 ilustra a estimativa da espessura média do traço da imagem de duas palavras. Os histogramas na Figura 34c e Figura 34d ilustram o número de ocorrências de cada valor das espessuras encontradas no traço da escrita da imagem da imagem das palavras da Figura 34a e Figura 34b respectivamente.

O cume do histograma fornece uma estimativa da espessura média através da equação (5.1).

$$\begin{aligned} N_d &= \text{número_de_ocorrências_do_valor_de_distância_d} \\ \text{Espessura_Média} &= \text{valor_máximo}(N_d) \end{aligned} \quad (5.1)$$

5.4.6 Estimativa e Correção da Inclinação das Letras da Palavra

Estimativa do Ângulo de Inclinação

Em uma imagem podemos encontrar vários componentes que compõem uma palavra. Um componente pode ser uma letra separada da palavra, letras juntas que são parte de uma palavra quebrada, uma palavra inteira, ou ainda, pontuações e acentos da escrita (ver seção 2.3). A análise que estima o ângulo de inclinação global das letras de uma palavra é realizada em cada componente que compões a escrita da imagem.

Para realizar o cálculo do ângulo global de inclinação das letras da escrita, o sistema realiza uma análise nas coordenadas dos pontos da lista que caracteriza a imagem cadastrada. Como visto na seção 3.3, o cadastramento da imagem disponibiliza uma lista seqüencial com as coordenadas dos pontos dos contornos interno e externo da palavra.

Um modelo de estimativa do ângulo de inclinação das letras da palavra foi abordado na seção 3.4.3, onde a estimativa da inclinação foi realizada acumulando-se os ângulos formados pelas linhas retas compreendidas entre dois pontos da lista seqüencial, ou seja, acumulando-se os ângulos formados pelos pontos atual e próximo da lista seqüencial. Neste modelo, podemos obter somente quatro tipos de ângulos: os ângulos horizontais 0° e 180° , os ângulos de inclinação à direita 45° e 225° , os ângulos de inclinação à esquerda 135° e 315° e os ângulos verticais 90° e 270° . A Figura 14 ilustra este processo.

Neste trabalho, foi utilizado um modelo semelhante ao descrito no parágrafo anterior. Desta maneira, para estimar o ângulo de inclinação global das letras da palavra analisou-se as coordenadas dos pontos da imagem cadastrada.

Um ângulo de inclinação é estimado a partir de dois pontos: ponto atual, referenciado por um índice i da lista de pontos, e ponto $i+\text{espessura_média}$, referenciado pelo índice correspondente à soma do índice i com o valor obtido através do cálculo da espessura média (ver seção 5.4.5).

Através da inserção do valor da espessura média no cálculo da distância entre os pontos a serem analisados, conseguimos um processo de estimativa a um tempo de processamento adaptável ao traço da escrita sem comprometer a eficiência do processo.

Todos os valores de ângulo obtidos através da estimativa de cada segmento local (definido pelos dois pontos analisados) são acumulados. Este processo é realizado até que o ponto final da lista seja atingido.

A equação (5.2) exhibe o cálculo realizado para obtermos o ângulo formado entre a reta definida por cada par de pontos e a linha vertical,

$$\theta = \arctan\left(\frac{x_i - x_{i+espessura_média}}{y_i - y_{i+espessura_média}}\right) \quad (5.2)$$

onde x_i e y_i representam as coordenadas do ponto atual, e $x_{i+espessura_média}$ e $y_{i+espessura_média}$ representam as coordenadas do ponto que está o valor da espessura média do traço da escrita distante do ponto atual na lista de pontos. Somente ângulos menores ao valor $\pi/3$ em radianos são acumulados, ou seja, acumula-se somente os ângulos compreendidos entre o intervalo de 30° à 150° , como ilustra a Figura 35a. Isto pode ser facilmente entendido após verificarmos que os ângulos pertencentes a este intervalo são os ângulos que ditam a inclinação vertical da palavra.

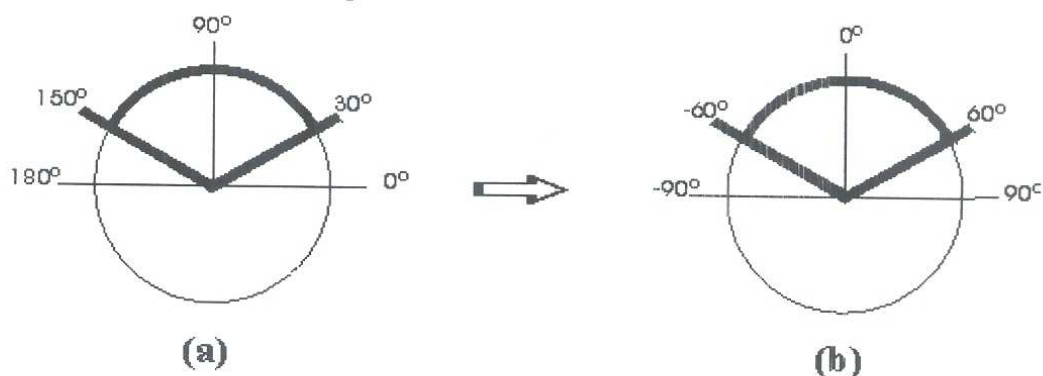


Figura 35 : Ângulos acumulados para a estimativa da inclinação. (a) Sistema original. (b) Sistema alterado para esta pesquisa.

Neste processo de estimativa, o ângulo reto 90° , é considerado o ângulo 0° ou sem inclinação, os ângulos que estão à esquerda do ângulo reto 90° , são ângulos ou inclinações considerados negativos, e os ângulos que estão à direita do ângulo reto 90° , são ângulos ou inclinações considerados positivos, como ilustra a Figura 35b.

O processo de estimativa do ângulo de inclinação das letras, ignora as inclinações verticais acumulando somente as inclinações positivas e negativas.

O acúmulo das inclinações positivas é realizado separado do acúmulo das inclinações negativas. Neste ponto, obtemos o valor total do acúmulo subtraindo o valor de acúmulo dos ângulos negativos do valor de acúmulo dos ângulos positivos.

O ângulo global é estimado dividindo-se o acúmulo total dos ângulos locais, obtidos através da equação (5.2), pela quantidade total de ângulos acumulados. Se comparado ao algoritmo exposto na seção 3.4.3, o algoritmo que estima o ângulo global da inclinação das letras de uma palavra utilizado neste trabalho tem a vantagem de se adaptar ao tipo de escrita sendo o valor da espessura média do traço da escrita mais rápido computacionalmente sem comprometer o resultado final da operação.

Correção da Inclinação

Uma vez que o ângulo global θ da inclinação das letras de uma palavra foi estimado, é aplicado um modelo de correção a esta inclinação.

O modelo que corrige a inclinação das letras de uma palavra aplicado neste trabalho é o mesmo modelo discutido na seção 3.4.4, onde a tangente do ângulo θ , é usada para realizar esta correção.

Ao realizar a correção da inclinação, o sistema gera uma nova imagem da palavra com suas letras corrigidas. Este sistema ainda observa os limites horizontais da nova palavra criada para que não ocorra perda de informações dos traços da escrita. Isto pode ocorrer quando as novas coordenadas de um ponto de informação (um ponto do traço da escrita) estiverem fora dos limites da imagem.

A equação (5.3), exhibe a transformação ocorrida nas coordenadas dos pontos de informação da imagem,

$$\begin{aligned}x_n &= x - y \cdot \tan(\theta) \\y_n &= y\end{aligned}\tag{5.3}$$

onde x e y representam as coordenadas originais do ponto analisado, e x_n e y_n representam as coordenadas corrigidas do ponto analisado. A Figura 36 ilustra o processo de correção da inclinação das letras de algumas palavras efetuado pelo modelo usado neste trabalho.

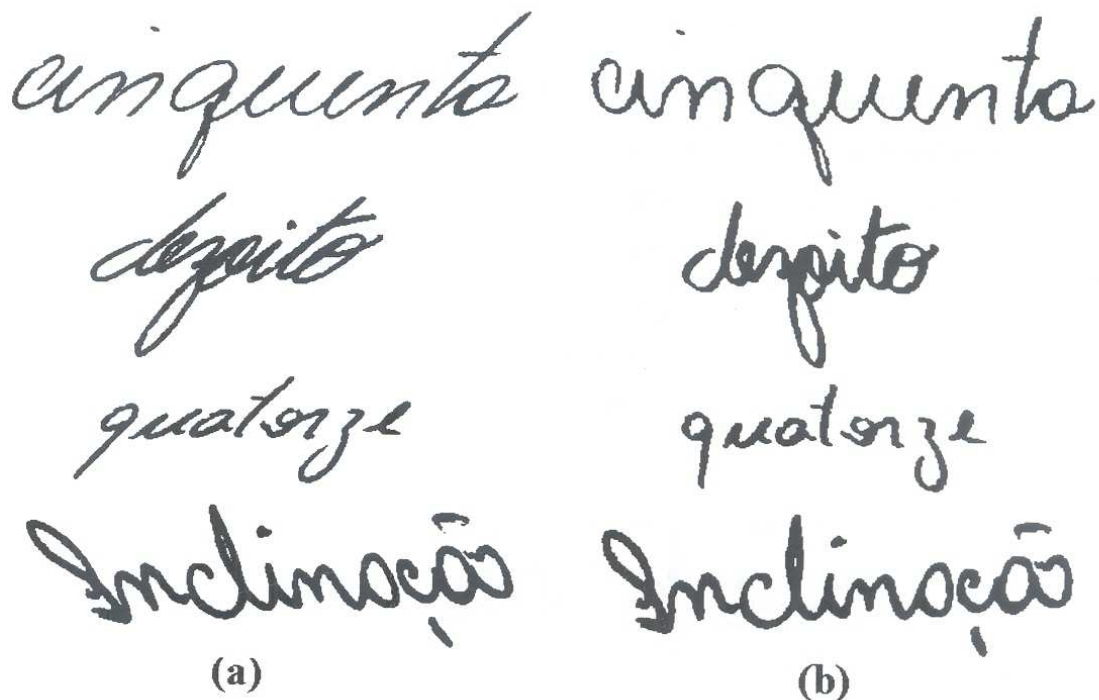


Figura 36 : Correção da inclinação das letras da palavra. (a) Palavras originais. (b) palavras corrigidas.

5.5 Pré-segmentação

5.5.1 Conceitos

Após a normalização e cadastramento da imagem (ou seja, após a realização dos pré-processos), a próxima tarefa é extrair e analisar informações locais, como:

- a detecção das *ligações* (ver seção 5.5.5) e
- a detecção dos *pontos de mínimo e máximo* (ver seção 5.5.3),

e informações globais, como:

- a detecção do *corpo da palavra* (ver seção 5.5.2) e
- a divisão do contorno externo da palavra em *bordas superior e inferior* (ver seção 5.5.4).

Alguns destes processos são realizados sob a palavra cadastrada como visto na seção 5.4.4.

A etapa de extração de informações importantes sobre a palavra preocupa-se em observar as características da palavra em análise para que estas informações sejam a base do processo de segmentação.

5.5.2 Detecção do Corpo da Palavra

Uma das informações usadas na detecção dos pontos de mínimo e máximo (ver seção 5.5.3) do traço da escrita, e detecção das ligações (ver seção 5.5.4) entre as letras da palavra analisada são os limites superior e inferior do corpo da palavra. Estes limites são encontrados realizando-se uma projeção horizontal das informações da imagem por transição. O resultado desta projeção é um histograma que acumula as transições entre o traço da escrita e o fundo, realizado para cada linha da imagem.

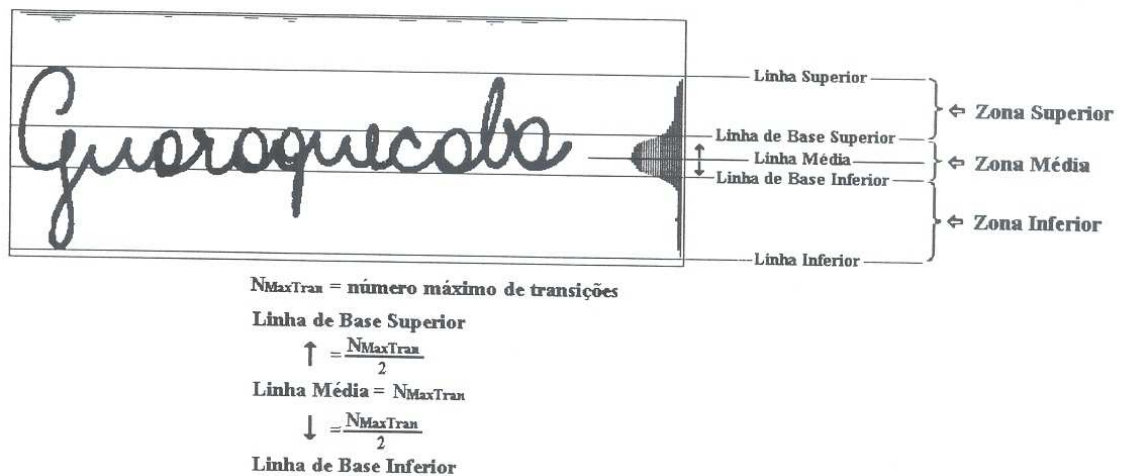


Figura 37 : Divisão da palavra em *zona superior*, *zona média* e *zona inferior*

A Figura 37 ilustra o processo realizado para obtermos os limites do corpo da imagem. O primeiro passo é projetar horizontalmente as transições entre o traço da escrita e o fundo da imagem. Este passo cria um histograma com o acúmulo das transições. Em seguida, é realizada uma suavização no histograma de transições. Esta suavização é feita para equilibrar os valores do histograma, evitando, assim, que valores muito altos ou muito baixos de acúmulo de transições (ou seja, ruídos) interfiram no processo de detecção do corpo da palavra.

A suavização do histograma da palavra ilustrada na Figura 37 é exibida na Figura 38, e é realizada como segue:

- Para cada valor de acúmulo de transições, é adicionado os valores de acúmulo dos três vizinhos anteriores e dos três vizinhos posteriores. O valor resultante é dividido pelo número de vizinhos que foram adicionados. O resultado deste processo é o valor de acúmulo suavizado.

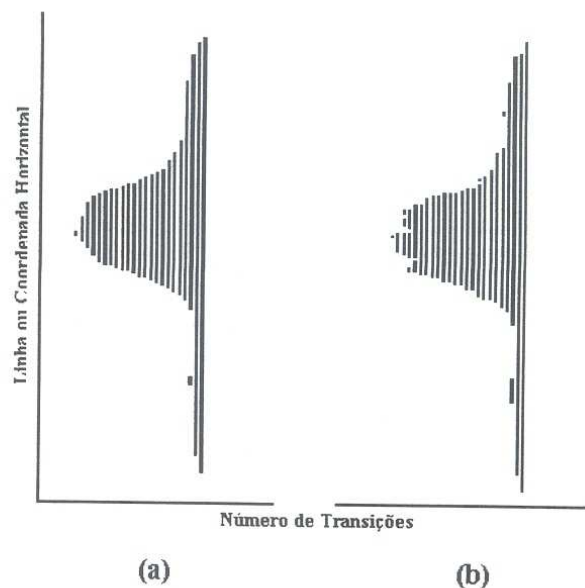


Figura 38 : Suavização do histograma de transição da palavra da Figura 37. (a) Histograma suavizado. (b) Histograma original.

Analisando o histograma de transições, estimamos as seguintes informações (ver Figura 37):

- Linha Média – Observando o cume do histograma de transições, que representa uma linha na imagem, obtemos a coordenada vertical cujo número de transições é o maior dentre os estimados. Esta linha é denominada de *linha média* do corpo da palavra.
- Linha de Base Superior – Depois de encontrada a coordenada vertical que representa a linha média da palavra, podemos estimar a coordenada vertical que representa a *linha de base superior* do corpo da palavra. Esta linha é obtida analisando os acúmulos das transições das linhas que estão acima do acúmulo máximo do histograma de transições. Isto é feito da seguinte forma: partindo-se da linha média, é realizada uma busca, acima no histograma, pela linha cujo valor de acúmulo de transições seja *menor ou igual à metade* do valor de acúmulo da linha média.
- Linha de Base Inferior – Da mesma forma como foi estimada a linha de base superior é estimada a *linha de base inferior* da palavra, só que agora a busca é realizada nos acúmulos de transições abaixo do acúmulo máximo do histograma de transições. Assim, esta linha é obtida a partir da linha média, onde é realizada uma busca, abaixo no histograma, pela linha cujo valor de acúmulo de transições seja *menor ou igual à metade* do valor de acúmulo da linha média.
- Linha Superior – A primeira linha com informação no histograma de transições é denominada de *linha superior* da palavra.

- Linha Inferior – A última linha com informação no histograma de transições é denominada de *linha inferior* da palavra.

Caso exista mais de uma linha do histograma que expresse uma das linhas citadas acima, a linha escolhida é a que está no meio deste conjunto.

Após encontrarmos as linhas que representam os limites de base superior, de base inferior, superior e inferior da imagem da palavra, pode-se dividir esta palavra em três zonas (ver Figura 37):

- Zona Média – A região compreendida entre a linha de base superior e a linha de base inferior da palavra, é pertencente à *zona média* ou *corpo* da palavra.
- Zona Superior – Da mesma maneira, designamos que a região compreendida entre a linhas de base superior e a linha superior da palavra, é pertencente à *zona superior* da palavra.
- Zona Inferior – Por último, designamos que a região compreendida entre a linha de base inferior e a linha inferior da palavra, é pertencente à *zona inferior* da palavra.

Através destas informações, pode-se estipular em qual região da palavra é permitido detectar ligações, detectar pontos de máximo e mínimo, e até mesmo, qual região pode-se realizar uma quebra.

5.5.3 Detecção dos Pontos de Máximo e de Mínimo Locais

Uma das informações mais úteis em processos de segmentação é a localização dos pontos de máximos (picos) e pontos de mínimos (vales) do texto analisado. Isto pode ser dito por se tratar de pontos cujas características podem revelar pontos de segmentação em potencial. Apesar destes pontos serem ótimos candidatos para a segmentação quando usados neste tipo de processo sem o auxílio de outras técnicas que validem estes pontos como sendo pontos de segmentação deve-se observar algumas características para evitar quebras não válidas, como visto na seção 4.6.

Um dos benefícios da detecção dos pontos de máximo e mínimo, é que estas informações podem validar as ligações (ver seção 5.5.4) que serão detectadas posteriormente. Na verdade, este trabalho cruza as informações obtidas na detecção dos pontos de máximo e mínimo com as informações obtidas na detecção das ligações para obter um alto grau de confiança na determinação de quais porções do traço da escrita realmente são ligações e quais graphemes podem ser quebrados.

No geral, para se estimar pontos segmentação através dos pontos de mínimo e máximo, o modelo deve encontrar a seguinte situação:

- deve haver um ponto de mínimo local, cercado por dois máximos locais (um em cada lado), e com uma profundidade relativa suficiente entre este mínimo e os dois máximos que o cercam.

Neste trabalho, a obtenção dos pontos de máximo e mínimo é realizada sob a palavra cadastrada (ver seção 5.4.4). O modelo de busca percorre toda a lista gerada através do cadastramento das coordenadas dos pontos de contorno da palavra fixando o ponto atual e analisando os três pontos anteriores e os três pontos posteriores ao ponto atual da lista. Para ilustrar esta análise, suponha que P_{i-k} e P_{i+k} sejam pontos da lista a ser analisada e que dependendo do valor atribuído à k estes pontos estejam a k pontos antes do ponto atual e a k pontos depois do ponto atual respectivamente. Sob esta perspectiva devemos fazer as seguintes observações:

- Quando $k=0$, estamos nos referenciando ao ponto atual.
- Quando $k=1$, os pontos P_{i-k} e P_{i+k} devem estar a um ponto de distancia antes e a um ponto de distancia após o ponto atual respectivamente.
- Quando $k=2$, os pontos P_{i-k} e P_{i+k} devem estar a dois pontos de distancia antes e a dois pontos de distancia após o ponto atual respectivamente. A mesma análise deve ser feita a todos os valores atribuídos a k ;

Através destas observações, podemos “visualizar” as seguintes situações:

- a. Quando $k=1$, e os pontos P_{i-k} e P_{i+k} estão acima ou na mesma horizontal de quando $k=0$, e quando $k=2$, e os pontos P_{i-k} e P_{i+k} estão acima ou na mesma horizontal de quando $k=1$, e assim por diante. A Figura 39a ilustra esta situação.
- b. Quando $k=1$, e os pontos P_{i-k} e P_{i+k} estão abaixo ou na mesma horizontal de quando $k=0$, e quando $k=2$, e os pontos P_{i-k} e P_{i+k} estão abaixo ou na mesma horizontal de quando $k=1$, e assim por diante. A Figura 39b ilustra esta situação.
- c. Quando para todo o k , os pontos P_{i-k} e P_{i+k} estão na mesma horizontal de quando $k=0$. A Figura 39c ilustra esta situação.

Quando encontrada a situação (a), marca-se o ponto atual como sendo um *ponto de mínimo*. Quando encontrada a situação (b), marca-se o ponto atual como sendo um *ponto de máximo*. Quando encontrada a situação (c),

caminham-se mais pontos antes e depois do ponto atual até que se encontre o primeiro ponto, em ambas as direções, que não esteja na mesma horizontal que o ponto atual. Neste caso, se os pontos extremos estiverem acima do ponto atual, este é marcado como ponto de mínimo (ver Figura 39c2), e se os pontos extremos estiverem abaixo do ponto atual, este é marcado como ponto de máximo (ver Figura 39c1).

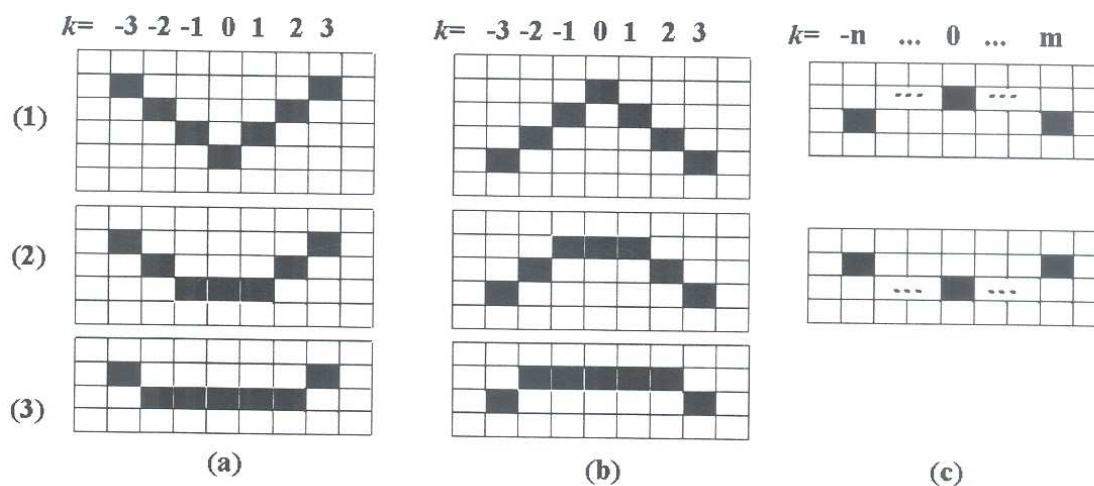


Figura 39 : Detecção dos pontos de máximo e mínimo.

Note que na situação ilustrada na Figura 39c, o sistema pode percorrer distâncias diferentes em ambas as direções até encontrar um ponto que esteja em uma coordenada vertical diferente da coordenada vertical do ponto atual. A Figura 40 ilustra a detecção dos pontos de máximo e mínimo na imagem de uma palavra.

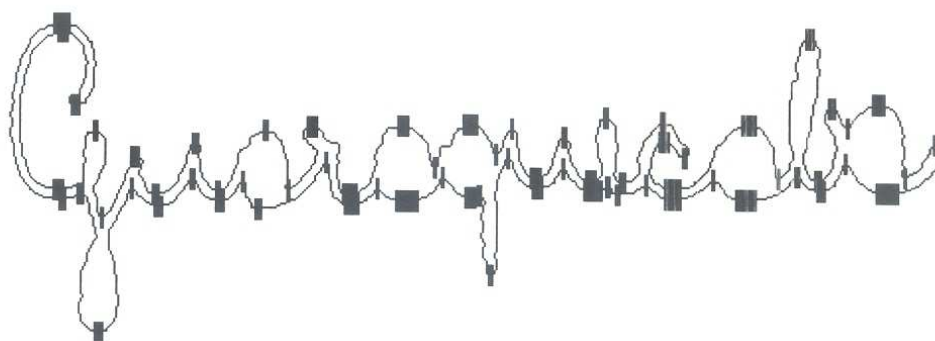


Figura 40 : Detecção dos pontos de mínimo e máximo do contorno externo de uma palavra.

A ilustração da Figura 40 exhibe todos os pontos de máximo e mínimo detectados em uma palavra. Note que a maior parte dos pontos detectados são vizinhos de pontos na mesma concavidade. Se em uma concavidade for encontrado mais de um ponto de máximo ou mais de um ponto de mínimo que

faz referência à esta mesma concavidade, então existem pontos redundantes que devem ser eliminados.

Pontos redundantes são encontrados nas seguintes situações:

- quando existe uma seqüência de pontos de máximo que fazem referência a um mesmo pico (concavidade para baixo), ou
- quando existe uma seqüência de pontos de mínimo que fazem referência a um mesmo vale (concavidade para cima).

Em qualquer destas situações citadas acima, o ponto que representará toda a seqüência é obtido da seguinte forma: conta-se a quantidade de pontos na seqüência, e o ponto que for referenciado com a metade da quantidade total de pontos da seqüência será o ponto que representará toda a seqüência.

A Figura 41 ilustra a exclusão de pontos de mínimo e pontos de máximo redundantes.

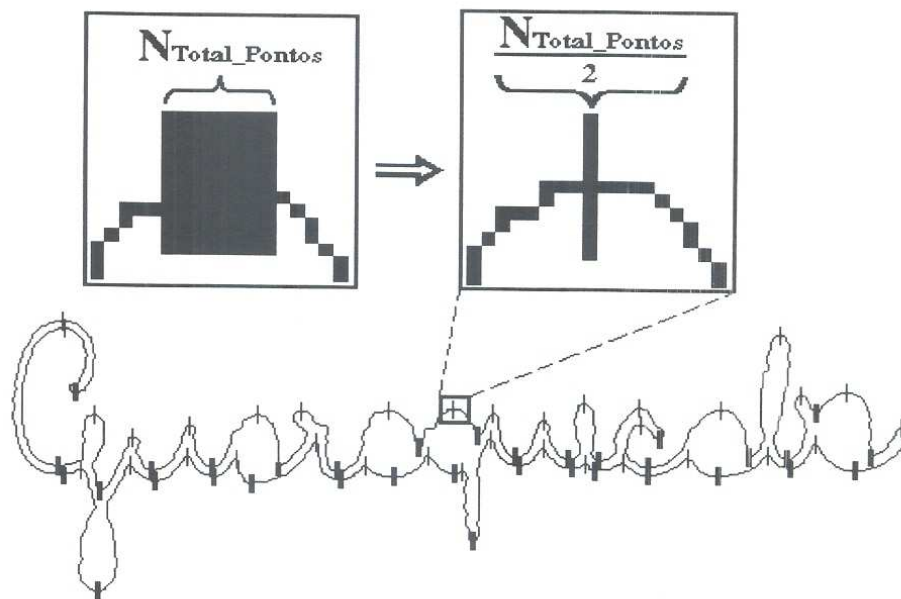


Figura 41 : Exclusão dos pontos de mínimo e dos pontos de máximo redundantes.

Note que para diferenciar os pontos de máximo e mínimo na Figura 41, realizou-se uma dilatação lateral nos pontos de mínimo.

Alguns filtros ainda são aplicados sob os pontos de mínimo e máximo detectados com a finalidade de excluir ou modificar pontos cujas características sejam consideradas inválidas para a análise realizada neste trabalho. Estes filtros são discutidos na seção 5.6.3.

5.5.4 Divisão do Contorno Externo em Bordas Superior e Inferior

Este processo é vital para o bom desempenho do algoritmo de detecção das ligações da palavra (ver seção 5.5.5). A detecção das ligações é inteiramente baseada em pontos cuja localização pode estar na borda superior ou na borda inferior do contorno externo da palavra.

O modelo de divisão do contorno externo é baseado em uma lista que retém as coordenadas dos pontos divisórios à esquerda e a direita de cada componente que compõe o texto. Estes pontos divisórios são obtidos da seguinte forma: os pontos que estiverem na extrema esquerda e extrema direita do componente analisado, e estiverem internos ao limite do corpo da palavra (ver seção 5.5.2) serão designados para serem os pontos divisórios à esquerda e à direita deste componente respectivamente. A Figura 42 ilustra este processo.

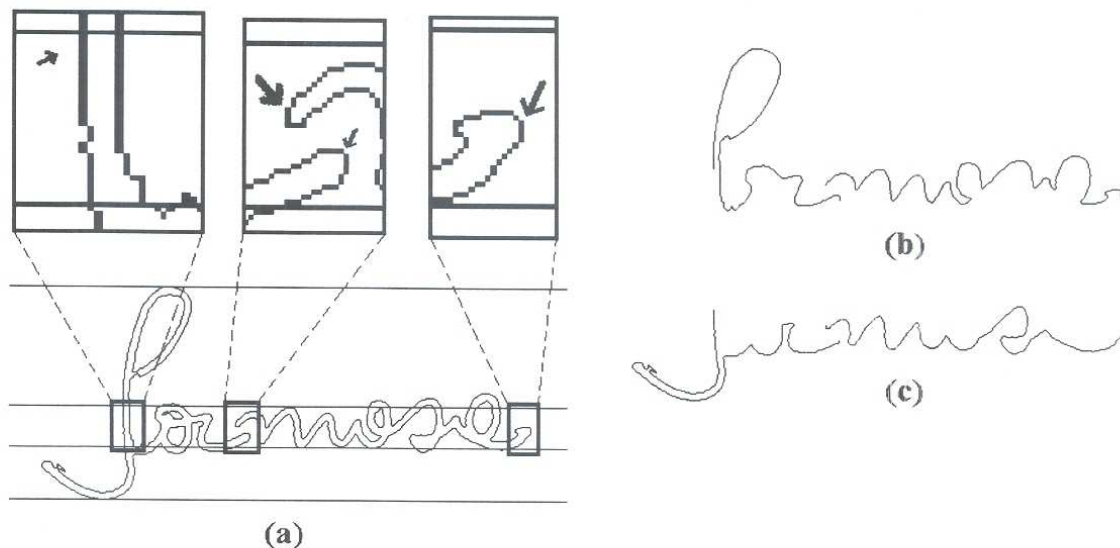


Figura 42 : Divisão do contorno externo em bordas superior e inferior. (a) Pontos extremos. (b) Borda superior. (c) Borda inferior.

Na ilustração da Figura 42a, é exibido os pontos que compõe a lista de pontos divisórios do componente. Neste exemplo, a palavra está composta por dois componentes e, portanto, a lista de coordenadas terá quatro elementos: as coordenadas da extrema esquerda e extrema direita do primeiro componente e as coordenadas da extrema esquerda e extrema direita do segundo componente.

Na ilustração da Figura 42b e Figura 42c, é exibido as bordas superior e inferior do contorno externo da palavra respectivamente.

5.5.5 Detecção das Ligações

Como discutido nas seções 4.4 e 4.5, se conseguirmos detectar as ligações existentes entre as letras de uma palavra serão grandes as chances de que a segmentação desta palavra, baseada em suas ligações e pontos de máximo e mínimo, também seja realizada com eficiência.

Na seção 4.4, foram abordados dois modelos de detecção de ligações. Um destes é baseado nos pontos de mínimo (ver seção 5.5.3) da borda superior do contorno da palavra. O outro é baseado nas projeções dos vales do contorno superior. Mas como visto na seção 4.6, estes modelos de detecção de ligações geralmente produzem ligações inconsistentes causando problemas na segmentação.

Neste trabalho, foi utilizado um modelo de detecção de ligações semelhante ao descrito na seção 4.5, onde se caracteriza ligações através da espessura média (ver seção 5.4.5) do traço da escrita. Deste modo, após estimada a espessura média do traço analisado, o sistema realiza a seguinte tarefa:

- busca por partes do traço da escrita cuja espessura seja menor ou igual a duas vezes o valor da espessura média e seja maior ou igual a um. Deste modo, podemos adaptar a busca por ligações ao tipo de traço de escrita que a palavra analisada possui. Se uma parte do traço da escrita for encontrada desta maneira, esta será marcada como uma parte de uma ligação. Esta detecção é realizada sob o contorno externo da palavra.

O principal objetivo da detecção e segmentação das ligações que existem entre as letras das palavras é o de diminuir a variabilidade encontrada nas letras da palavra. Este fato pode realmente colaborar com o processo de reconhecimento da palavra.

Para realizar o processo de detecção das ligações, o sistema primeiro particiona o contorno externo da palavra em borda superior e borda inferior, como descrito na seção 5.5.4.

Após o processo de divisão do contorno externo da palavra em borda superior e borda inferior ser realizado, o algoritmo de detecção realiza a seguinte análise: se a distância entre as coordenadas y_s da borda superior e y_i da borda inferior do contorno externo para uma dada coordenada x for *menor ou igual* a *duas vezes* a espessura média da escrita e for maior ou igual a 1, então os pontos cujas coordenadas correspondam a (x, y_s) e (x, y_i) são marcados

como sendo elementos de uma ligação. Estes valores limiaries mostraram-se satisfatórios na detecção das ligações por ignorar traços muito espessos e por abranger partes do traço da escrita com grandes chances de ser ligação.

Este processo é repetido para todas as coordenadas x . A Figura 43 ilustra este processo.



Figura 43 : Exemplo da detecção das ligações em uma palavra. (a) Contorno interno e externo da palavra. (b) Ligações detectadas pelo contorno externo.

Após a detecção das possíveis ligações entre as letras da palavra, o sistema realiza uma filtragem nestas ligações para preencher falhas ou eliminar parte ou toda a ligação. A lista abaixo, exhibe alguns filtros que são utilizados:

1. Eliminar ligações que não tem ponto de mínimo e não tem ponto de máximo;
2. Eliminar ligações que estão totalmente fora do corpo da palavra;
3. Eliminar parte das ligações analisando o ângulo de inclinação destas;
4. Eliminar Ligações pequenas;

O filtro que analisa o ângulo de inclinação das ligações usa como ponto base da análise o ponto de mínimo ou de máximo da borda superior. Caso não exista pontos de mínimo ou máximo na borda superior, o sistema opta pelo ponto de mínimo ou máximo que venha existir na borda inferior. Ainda assim, o sistema verifica se o ponto escolhido pertence à porção central da ligação que retém metade da ligação. Caso não pertença, o ponto escolhido como base será o ponto médio da ligação. A ilustração da Figura 44a exhibe um ponto de mínimo da borda superior que pertence à porção média da ligação e, portanto, é escolhido para ser o ponto base. Nesta ilustração existe ainda a indicação dos pontos extremos da ligação. A ilustração da Figura 44b exhibe um ponto de mínimo da borda superior que não pertence à porção média da ligação e, portanto, o ponto base escolhido é o ponto médio da ligação. Nesta ilustração existe ainda a indicação dos pontos extremos da ligação.

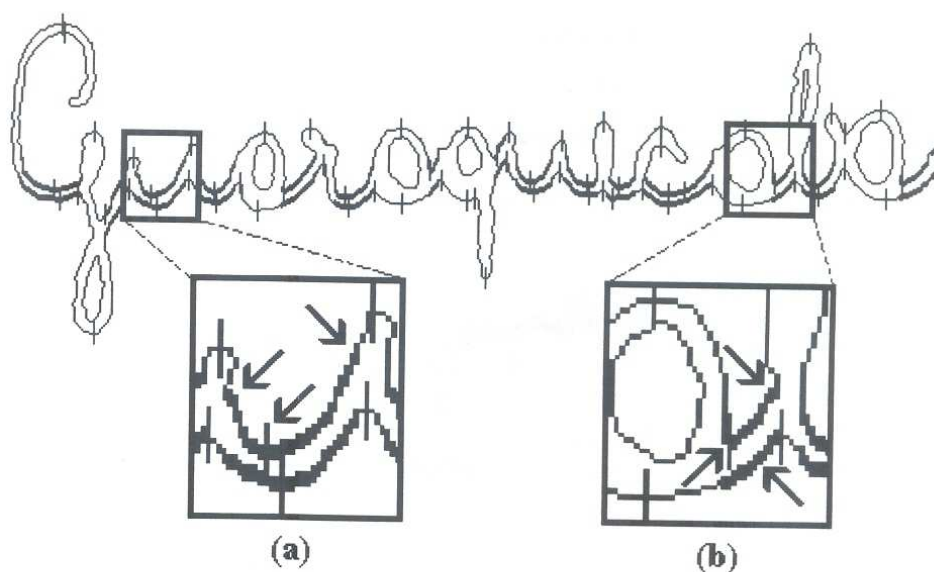


Figura 44 : Escolha do ponto base para filtrar ligação. (a) Ponto de mínimo pertence à porção média da ligação. (b) Ponto de mínimo não pertence à porção média da ligação.

Após a realização da escolha do ponto de base, o sistema calcula o ângulo de inclinação que existe entre o ponto base e os pontos à extrema esquerda e à extrema direita da ligação. Caso este ângulo seja maior ou igual a 30° o ponto extremo da ligação passa ser parte do grapheme. Este processo se repete até que o ângulo de inclinação seja menor que 30° ou até que a distância do extremo analisado ao ponto base seja menor que 3 pontos. A Figura 45a ilustra o processo de filtragem das ligações pelo processo da inclinação das ligações.

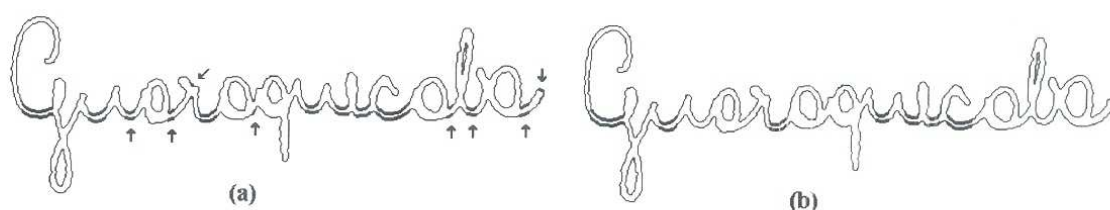


Figura 45 : Filtragem das ligações. (a) Pela inclinação. (b) Ligações pequenas.

O último dos processos de filtragem das ligações é a filtragem de ligações pequenas. Isto é feito porque quando a ligação é muito pequena possivelmente esta porção do traço da escrita pertença a um grapheme. Neste caso, estas ligações são eliminadas deixando uma possível quebra neste traço para o ponto de mínimo, se houver. Este processo é realizado como segue:

- Se a ligação for formada por seis ou menos pontos (estas ligações são apontadas na Figura 45a), então esta ligação é considerada pequena e é eliminada, como mostra a transição da Figura 45a para a Figura 45b.

A utilização destes filtros faz aumentar o grau de confiança no processo de detecção das ligações, mas, no entanto, ainda surgem problemas quando a escrita é muito corrida, como ilustra a Figura 46.

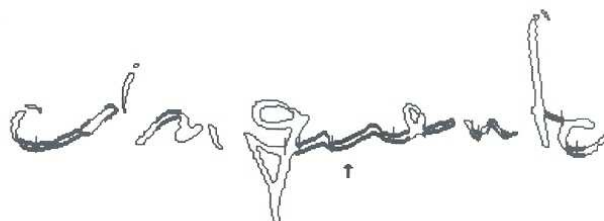


Figura 46 : Problema na detecção de ligação .

5.6 Segmentação

5.6.1 Conceitos

Nas etapas anteriores, foram aplicados processos de normalização, cadastramento, extração e análise da palavra. Os resultados destes processos são informações importantes sobre a palavra que passará pelo processo de segmentação. Deste modo, após a obtenção das informações necessárias sobre a palavra que está sendo analisada, o algoritmo de segmentação é acionado.

Muitas vezes, a palavra ou o texto que deve ser segmentado é composto de mais de um componente. Um componente pode ser uma palavra inteira, parte de uma palavra ou, até mesmo, duas palavras que estão conectadas. Este algoritmo realiza o processo de segmentação em todos os componentes que compõem a imagem da palavra. O processo é dividido nos seguintes passos:

1. Quebra da palavra em elementos de *ligação* e *graphemes*, baseada nas ligações e pontos de corte;
2. Quebra dos graphemes que tem ponto de corte no contorno superior;
3. Reagrupamento dos graphemes considerados pequenos;

Cada um dos passos listados acima tem uma meta específica no processo de segmentação. Estes passos são explorados em sua respectiva seqüência nos tópicos abaixo.

5.6.2 Quebra da Palavra em Ligações e Graphemes

Neste passo do processo de segmentação da palavra o sistema quebra os componentes, que formam o texto da imagem, em dois tipos de entidades: *ligações* e *graphemes*. Como visto na seção 5.5.4, estas entidades são determinadas para ser ligações ou graphemes baseado em análises realizadas no traço da escrita.

Através da separação das ligações e dos graphemes, pretendemos diminuir o máximo possível a variabilidade das letras que compõem a palavra. Isto é compreendido quando percebemos que as ligações são partes do traço da escrita não pertencentes às letras, tornando-se ruídos que devem ser detectados como visto na seção 5.5.5.

A Figura 47d ilustra a quebra de um texto em ligações e graphemes.

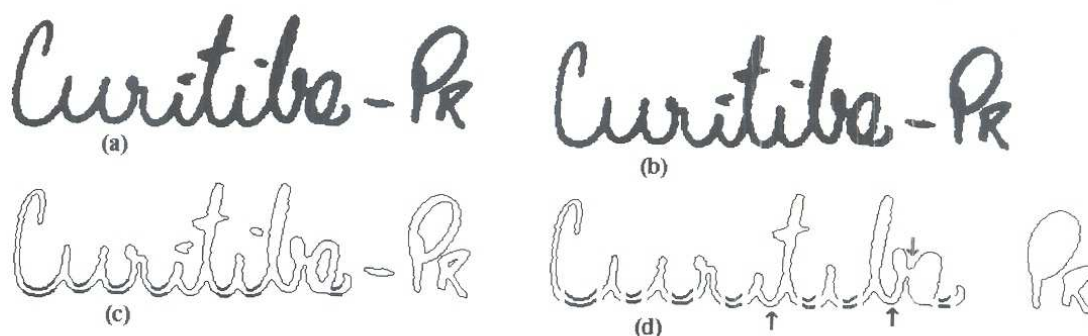


Figura 47 : Exemplo da quebra da palavra em ligações e graphemes. (a) Imagem original. (b) Imagem normalizada. (c) Ligações detectadas. (d) Quebra da palavra.

Note que este processo somente separa as ligações dos graphemes, pois a quebra só é realizada nos trechos onde foram detectadas ligações. A ilustração da Figura 47d exhibe três indicações que apontam para pontos que deveriam ter sido quebrados também mas suas respectivas ligações foram filtradas (ver seção 5.5.5). O processo posterior, tende a resolver estes casos através da quebra dos graphemes pelos pontos de mínimo da borda superior do componente.

5.6.3 Quebra dos Graphemes pelos Pontos de Mínimo

Após quebrarmos a palavra em ligações e graphemes, podemos observar, na ilustração da Figura 47d, que alguns dos graphemes da palavra ainda são constituídos de mais de uma letras. Isto ocorre por causa da filtragem que é realizada nas ligações.

Para solucionar este problema, é realizada uma quebra nos graphemes observando os pontos de mínimo da borda superior do grapheme. Esta quebra é feita a partir do ponto de mínimo encontrado na borda superior e que tenha as características de um *ponto de corte* válido. Um ponto de corte é definido como segue:

- **Ponto de Corte:** um ponto de mínimo pode ser dito como sendo um ponto de corte se o próximo ponto abaixo deste ponto de mínimo pertencer à borda inferior do contorno do mesmo componente a que o ponto de mínimo analisado pertence.

Para que não ocorram quebras inválidas, é feita uma filtragem nos pontos de mínimo do contorno superior da palavra com a finalidade de validar pontos de mínimo como pontos de corte, e invalidar pontos de mínimo cujas características não se enquadrem a um ponto de corte válido.

Este processo pode executar uma das seguintes operações nos pontos de mínimo analisados: *excluir* o ponto analisado ou *modificar* a localização deste ponto. Esta filtragem só é realizada nos pontos de mínimo da borda superior do contorno da palavra, pois é somente através destes pontos que uma possível quebra poderá ocorrer.

A Figura 48a ilustra os pontos de mínimo detectados pelo processo discutido na seção 5.5.3 e as ligações detectadas pelo processo discutido na seção 5.5.5. A Figura 48b ilustra a quebra da palavra utilizando as informações das ligações e dos pontos de corte válidos.

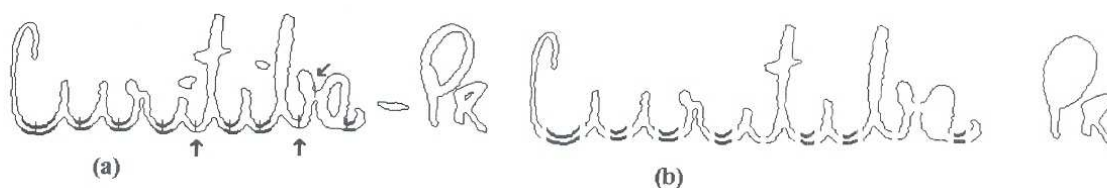


Figura 48 : Exemplo da quebra da palavra em ligações e graphemes. (a) Pontos de máximo e mínimo. (b) Quebra da palavra

Note que as partes dos traços da escrita detectadas como ligação são descartados do processo de quebra dos graphemes por ponto de corte válido. As entidades analisadas são constituídas apenas por graphemes. A ilustração da Figura 48a exibe as indicações para os pontos de mínimo na borda superior que são pontos de corte válidos.

As seguintes regras são usadas na análise realizada para filtrar os pontos de mínimo inválidos:

- ponto de mínimo próximo a ponto de máximo,

- distância entre as bordas superior e inferior do mesmo componente,
- ponto de quebra correspondente ao mínimo da borda superior pertencente à borda superior, e
- ponto de quebra correspondente ao mínimo da borda superior pertencente ao contorno interno.

Estes filtros são discutidos mais detalhadamente nos próximos subtópicos.

Ponto de Mínimo Próximo de Ponto de Máximo

Quando um ponto de mínimo está muito próximo a um ponto de máximo, assumimos que estes pontos não caracterizam pontos mínimo ou máximo relevantes para esta pesquisa. Nestes casos, optamos por eliminar ambos os pontos que tenham estas características. Este processo é realizado da seguinte forma:

- para cada ponto de mínimo da borda superior do contorno externo, verifica-se se um dos pontos vizinhos, a uma distância de quatro pontos antes e depois do ponto de mínimo, é ponto de máximo. Caso um dos pontos percorridos seja um ponto de máximo, ambos os pontos são excluídos.

Desta forma, tentamos evitar que pontos de mínimo inválidos façam parte do processo de segmentação da palavra. A Figura 49 ilustra o processo de eliminação de pontos de mínimo e um pontos de máximo pelo filtro da proximidade.

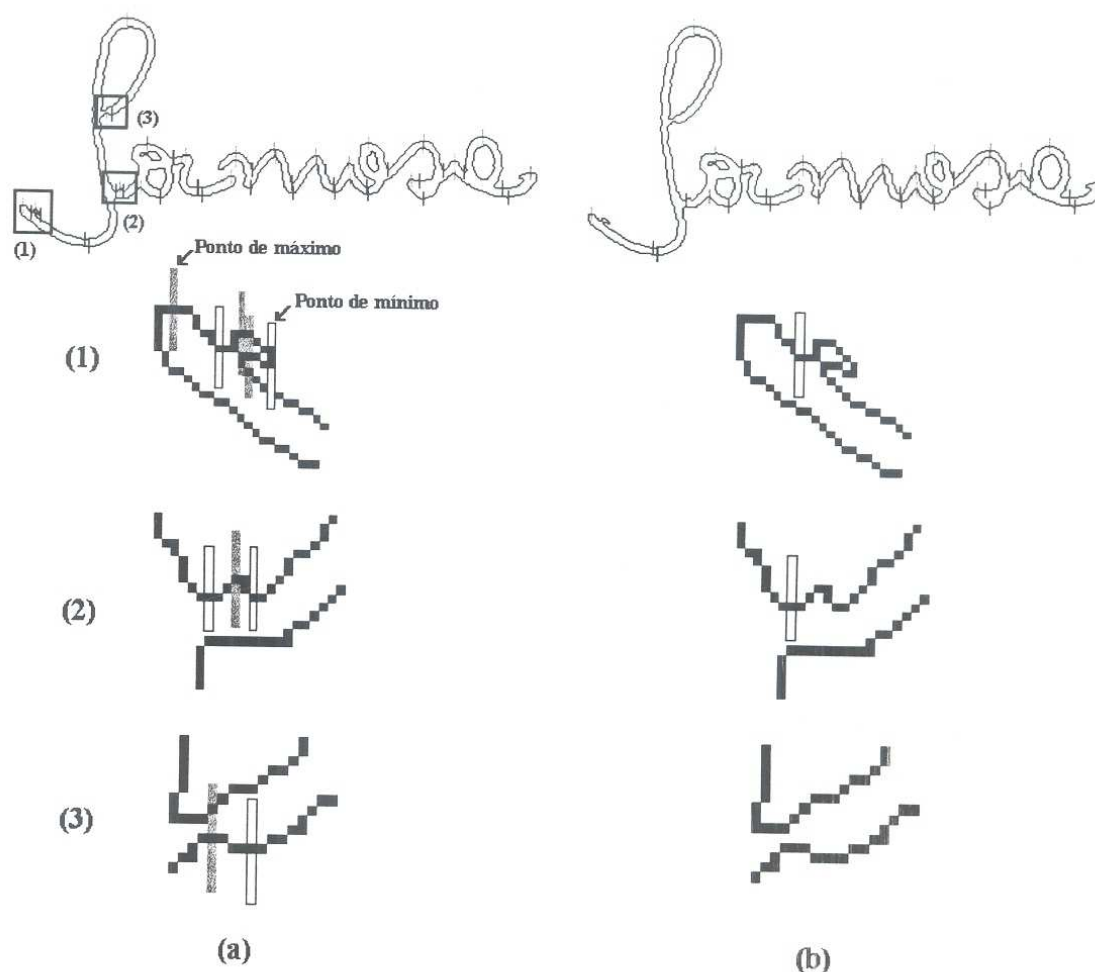


Figura 49 : Exclusão dos pontos de mínimo e máximo próximos. (a) Máximos e mínimos detectados. (b) Máximos e mínimos filtrados.

Distância Entre as Bordas Superior e Inferior

Nesta etapa, o sistema percorre por todos os pontos de mínimo da borda superior do contorno externo calculando a distância entre cada ponto de mínimo da borda superior com seus pontos correspondentes na mesma coordenada horizontal da borda inferior do contorno externo da palavra.

Esta etapa é importante por evitar as quebras que possivelmente ocorram em partes do traço da escrita cuja distância entre as bordas superior e inferior seja muito grande.

O valor da *distância limiar* é adaptado à espessura média do traço da palavra, ou seja, a distância limiar deste processo de busca é igual a duas vezes o valor da espessura média.

Neste processo, podemos encontrar três situações:

1. A distância entre as bordas superior e inferior para o ponto de mínimo atual ser menor que a distância limiar estipulada. Neste caso, o processo encerra a análise deste ponto de mínimo e começa a analisar o próximo ponto de mínimo (ver Figura 50a); ou
2. A distância entre as bordas superior e inferior para o ponto de mínimo atual ser maior ou igual à distância limiar estipulada (ver Figura 50b). Neste caso, o processo tenta satisfazer a distância limiar calculando as distâncias dos pontos anteriores e posteriores ao ponto de mínimo. O número de pontos verificados é o valor resultante da metade da espessura média do traço adicionada a três pontos. Este valor foi estimado desta maneira por se adaptável ao traço da palavra e porque mesmo quando o valor da espessura média da palavra é muito baixa ou muito alta o valor estimado para a busca é satisfatório.
3. Nenhum dos pontos analisados na situação “2” satisfaz a distância limiar. Neste caso, se a distância entre a borda superior e a borda inferior for maior que a altura do corpo palavra e o ponto de mínimo estiver abaixo da linha inferior da palavra (ver Figura 50c), este ponto de mínimo é excluído.

$$\text{AlturaCP} = (\text{Linha de Base Superior} - \text{Linha de Base Inferior})$$

$$\text{PontoSup} = (X_{\text{Cont_Sup}}, Y_{\text{Cont_Sup}})$$

$$\text{PontoInf} = (X_{\text{Cont_Inf}}, Y_{\text{Cont_Inf}})$$

$$\text{DistContSI} = (Y_{\text{Cont_Sup}} - Y_{\text{Cont_Inf}})$$

$$\text{DistânciaLimiar} = (2 \times \text{Espessura_Média})$$

$$(a) \text{ DistContSI} < \text{DistânciaLimiar}$$

$$(b) \text{ DistContSI} \geq \text{DistânciaLimiar}$$

$$(c) (\text{DistContSI} > \text{AlturaCP}) \wedge (Y_{\text{Cont_Sup}} < \text{Linha de Base Inferior})$$

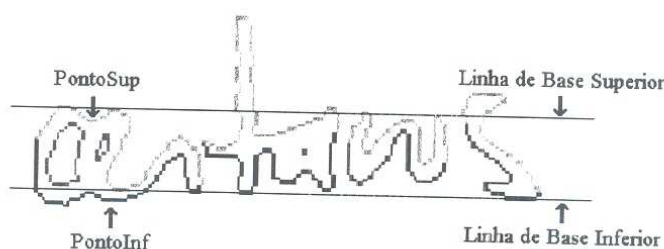


Figura 50 : Situações que podemos encontrar no filtro da distância. (a) Distância entre as bordas superior e inferior é menor que a distância limiar. (b) Distância entre as bordas superior e inferior é maior ou igual à distância limiar. (c) Distância entre as bordas superior e inferior é maior que a altura do corpo da palavra e a coordenada vertical da borda superior é menor que a linha de base inferior da palavra.

Na situação “2”, se for encontrada uma distância que satisfaça o limiar, o ponto de mínimo atual é alterado para ser o ponto da borda superior cuja distância com a borda inferior satisfaz o limiar, como ilustra a Figura 51. Caso o limiar não seja alcançado por nenhuma das verificações, o ponto de mínimo

não sofre alterações e o processo passa a analisar o próximo ponto de mínimo, como ilustra a Figura 52.

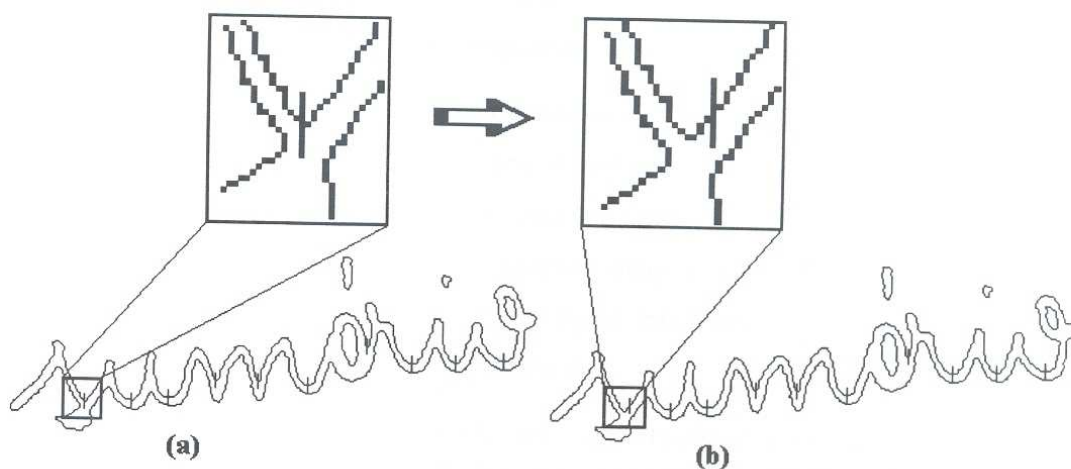


Figura 51 : Alteração do ponto de mínimo pelo filtro da distância entre as bordas superior e inferior. (a) Ponto de mínimo original. (b) Ponto de mínimo alterado.

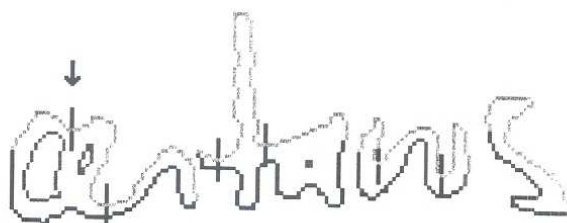


Figura 52 : Exemplo de distância muito grande entre as bordas superior e inferior.

Ponto de Quebra Correspondente Pertencente à Borda Superior

Neste filtro, o sistema realiza uma busca nos pontos de mínimo da borda superior do contorno externo verificando a existência ou não de pontos de mínimo que devam ser invalidados por não serem caracterizados como ponto de corte.

Como citado anteriormente, um ponto de mínimo só será um ponto de corte se o próximo ponto encontrado abaixo deste mínimo pertencer à borda inferior do contorno do mesmo componente a que o ponto de mínimo analisado pertence.

Deste modo, ao analisarmos os pontos de mínimo da borda superior, poderemos encontrar as seguintes situações:

1. O ponto de mínimo analisado não ser um ponto de corte e não pertencer ao corpo da palavra. Neste caso, este ponto de mínimo é excluído. Na ilustração da Figura 53, existe um apontamento para um ponto de mínimo que se enquadra nesta situação;
2. Caso o ponto de mínimo analisado não seja um ponto de corte e pertencer ao corpo da palavra, é realizada uma busca por um ponto de corte válido nos pontos anteriores e posteriores a este ponto de mínimo. Esta busca é limitada aos pontos cuja distância do ponto de mínimo seja menor ou igual a 1,5 vezes a espessura média. Neste processo podemos encontrar três situações:
 - Um ponto de corte válido ser encontrado e a distância entre a borda superior e borda inferior ser menor que duas vezes a espessura média. Neste caso, o ponto de mínimo atual é alterado para ser o ponto da borda superior do ponto de corte válido;
 - A busca percorrer por todos os pontos até o ponto limite e encontrar um ponto de corte válido cuja distância das bordas é menor que duas vezes a espessura média. Neste caso, se a distância entre a borda superior e borda inferior de algum dos pontos verificados for menor que a altura do corpo da palavra, o ponto de mínimo atual é alterado para ser o ponto da borda superior deste ponto de corte válido; ou
 - A busca percorrer por todos os pontos até o ponto limite e não encontrar nenhuma das situações acima. Neste caso, este ponto de mínimo é excluído. Na ilustração da Figura 53, existe um apontamento para um ponto de mínimo que se enquadra nesta situação.

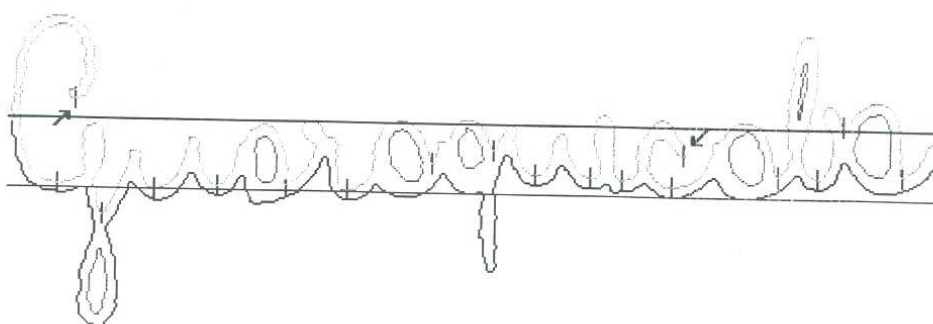


Figura 53 : Exclusão de pontos de mínimo pelo ponto correspondente ser da borda superior.

Ponto de Quebra Correspondente Pertencente ao Contorno Interno

Neste filtro, é verificado se entre o ponto de mínimo da borda superior e o ponto correspondente (mesma horizontal) deste ponto na borda inferior, para

o mesmo componente, existe pontos que não pertençam ao contorno externo da palavra. Esta verificação é necessária para que possamos prevenir uma possível quebra em um ponto de mínimo que não é um ponto de corte.

Como citado anteriormente, um ponto de mínimo só será um ponto de corte se o próximo ponto encontrado abaixo deste mínimo pertencer à borda inferior do contorno do mesmo componente a que o ponto de mínimo analisado pertence.

Deste modo, ao analisarmos os pontos de mínimo da borda superior, poderemos encontrar as seguintes situações:

1. Existe um ponto intermediário entre o ponto de mínimo analisado e seu ponto correspondente na borda inferior. Neste caso, o processo realiza uma busca aos pontos anteriores e posteriores ao ponto de mínimo, a uma distância de duas vezes o valor da espessura média, tentando encontrar um ponto da borda superior que não tenha pontos intermediários entre este ponto e seu correspondente da borda inferior. Se encontrar um ponto desta maneira, e a distância deste ponto ao seu correspondente na borda inferior for menor que a altura do corpo da palavra, então o ponto de mínimo analisado é substituído pelo ponto corrente, como ilustra a Figura 54. Caso não encontre um ponto que satisfaça a regra, o processo passa a analisar outro ponto de mínimo;

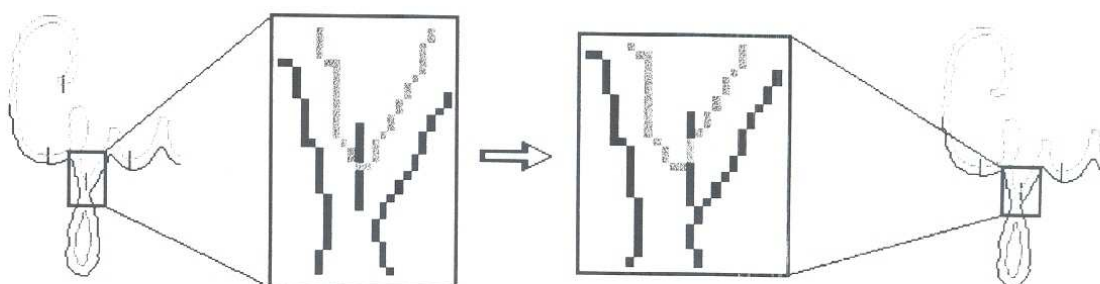


Figura 54 : Alteração no ponto de mínimo após verificação de ponto intermediário. (a) Ponto de mínimo original. (b) Ponto de mínimo alterado

5.6.4 Reagrupamento dos Graphemes

Neste passo, os graphemes são analisados para serem reagrupados. Este reagrupamento pode ser feito entre o grapheme atual e as entidades anterior, posterior ou ambas.

Este processo tenta evitar que, após a quebra por ligações e por pontos de mínimo, os segmentos extremos dos componentes do traço da escrita

fiquem quebrados se estes pertencem a um traço com características de ligação, como ilustra a Figura 56.

A análise realizada sob os graphemes neste processo, é baseada no *tamanho*, na *altura*, e na *localização* do grapheme com relação ao componente a que esta entidade pertence.

O tamanho de um grapheme é obtido através da contagem dos pontos que formam aquele grapheme, portanto, quanto mais pontos o grapheme tiver maior será o tamanho deste grapheme. Esta contagem engloba tanto a borda superior como a borda inferior do contorno do grapheme e é comparada com a espessura média do traço da escrita. Como visto na Figura 55, o valor obtido desta contagem é dividido por dois para que tenhamos uma comparação mais justa com a espessura média.

A altura do grapheme é o valor obtido da seguinte operação: a menor coordenada vertical subtraída da maior coordenada vertical do grapheme analisado.

A Figura 55 ilustra a análise realizada para a se obter o tamanho e a altura do grapheme.

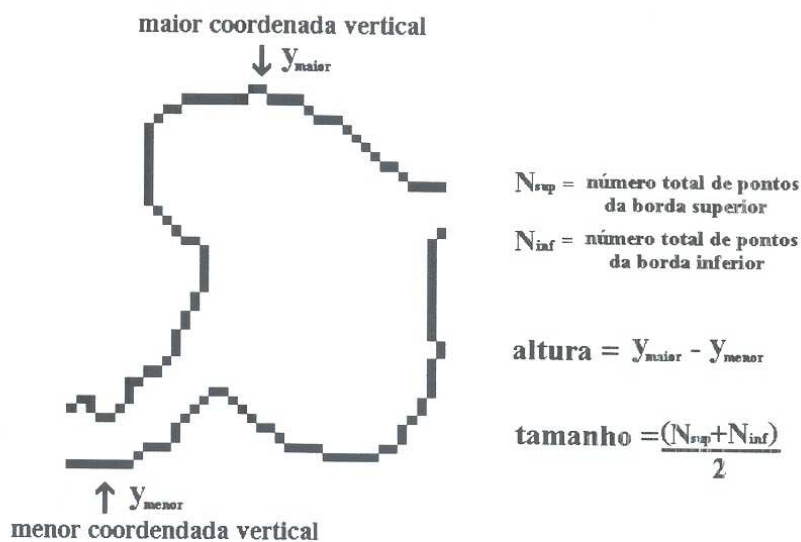


Figura 55 : Obtenção do tamanho e da altura do grapheme.

Após o cálculo do tamanho e da altura do grapheme atual, o sistema verifica se as entidades anterior e posterior ao grapheme atual são ligações ou graphemes. Com estas informações, a análise da equação (5.4) é realizada:

$$\begin{aligned}
 & (Tamanho(EntAtual) \leq (1.5 \times EspMédia)) \text{ ou} \\
 & (((Posição(EntAtual) = primeira) \text{ ou } (Posição(EntAtual) = última)) \text{ e} \\
 & (Altura(EntAtual) \leq \frac{2 \times Altura(CorpoPal)}{5})),
 \end{aligned} \tag{5.4}$$

onde:

- $Tamanho(EntAtual)$ é o valor do tamanho da entidade atual (quantidade de pontos que formam a entidade),
- $EspMédia$ é o valor da espessura média do traço da escrita,
- $Posição(EntAtual)$ é a posição que a entidade atual ocupa no componente,
- $Altura(EntAtual)$ é a altura da entidade atual, e
- $Altura(CorpoPal)$ é a altura do corpo da palavra.

Caso a equação (5.4) seja satisfeita, o grapheme atual é reagrupado. Deste modo, o próximo passo é descobrir com qual entidade do *mesmo componente* este grapheme será reagrupado. Para solucionar esta questão, as seguintes situações são analisadas:

1. Reagrupa à entidade posterior:

$$\begin{aligned}
 & (Posição(EntAtual) = primeira) \text{ ou} \\
 & ((Tamanho(EntAnterior) > Tamanho(EntPosterior)) \text{ e} \\
 & (EntPosterior = Grapheme)) \text{ ou} \\
 & ((EntAnterior = Ligação) \text{ e } (EntPosterior = Grapheme)),
 \end{aligned} \tag{5.5}$$

onde $Tamanho(EntAnterior)$ é o valor do tamanho da entidade anterior, $Tamanho(EntPosterior)$ é o valor do tamanho da entidade posterior, $EntAnterior$ é uma referência à entidade anterior e $EntPosterior$ é uma referência à entidade posterior. Caso a equação (5.5) seja satisfeita, o grapheme atual é reagrupado com a próxima entidade.

2. Reagrupa às duas entidades vizinhas: se a equação (5.5) na situação “1” não for satisfeita, então é analisada a equação (5.6) como segue:

$$\begin{aligned}
 & (EntAnterior = Ligação) \text{ e} \\
 & (EntPosterior = Ligação)
 \end{aligned} \tag{5.6}$$

Caso a equação (5.6) seja satisfeita, o grapheme atual é reagrupado à entidade anterior e à entidade posterior.

3. Reagrupa à entidade anterior: se as equações (5.5) e (5.6) nas situações “1” e “2” não forem satisfeitas, então se a entidade anterior for ligação, o grapheme atual é reagrupado com a entidade anterior.

A ilustração da Figura 56a exibe um reagrupamento como a entidade anterior do primeiro componente do texto e um reagrupamento com a entidade posterior do segundo componente do texto. A ilustração da Figura 56b exibe um reagrupamento como a entidade anterior do segundo componente.

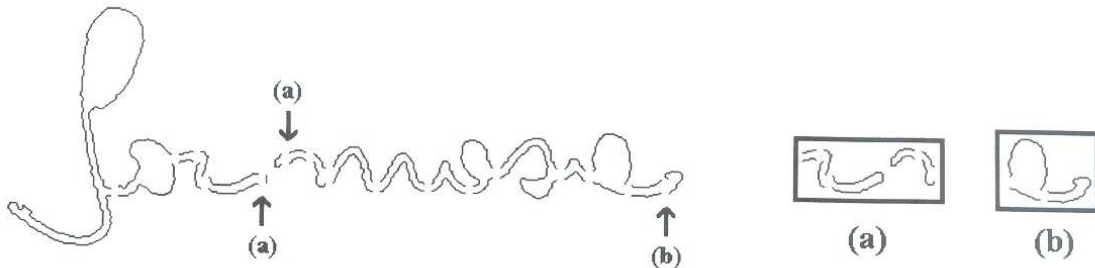


Figura 56 : Reagrupamento de graphemes.

5.7 Pós-processamento

5.7.1 Conceitos

O passo de pós-processamento é realizado para validar as entidades resultantes do processo de segmentação. Esta validação é realizada sob as hipóteses de ligações detectadas no traço da palavra. A meta deste processo é tentar evitar que parte de uma letra seja detectada como ligação. Isto é realizado analisando-se o comportamento das entidades vizinhas às ligações. Os passos que compõem o processo de pós-processamento da segmentação são:

1. Quebra das ligações que possivelmente contenham parte de graphemes;
2. Reagrupamento de ligações a graphemes baseado no contexto local;
3. Reagrupamento de ligações com ligações ou graphemes baseado no contexto global do componente.

Cada um dos passos listados acima tem uma meta específica no processo de validação das ligações geradas pelo processo de segmentação. Estes passos são explorados em sua respectiva seqüência nos tópicos abaixo.

5.7.2 Quebra das Ligações por Interseção

Nesta fase do processo de segmentação, a análise é concentrada às ligações detectadas na palavra. Este processo verifica se parte da ligação possivelmente pertença a um grapheme. Deste modo, se for detectado que

uma ou mais partes da ligação possivelmente pertença a um ou mais graphemes, então, dependendo da situação, é realizada uma quebra dividindo a ligação em duas ou três partes.

Uma ligação poderá ser dividida em no máximo três partes, onde, neste caso, uma parte possivelmente pertença à entidade anterior, outra parte seja realmente uma ligação, e a outra parte pertença à entidade posterior.

A detecção destas partes da ligação é feita pela interseção que existe entre as entidades vizinhas e partes da ligação. A interseção entre as entidades é obtida através das coordenadas horizontais das entidades envolvidas e é comparada à espessura média do traço da escrita. Desta maneira, as seguintes situações podem ocorrer:

1. Interseção à esquerda: uma quebra da ligação é realizada caso a interseção entre a entidade atual (neste caso uma ligação) e a entidade anterior satisfizer a equação (5.7),

$$\begin{aligned} & (Interseção(EntAtual, EntAnterior) > EspMédia) e \\ & ((Compr(EntAtual) - Interseção(EntAtual, EntAnterior) > \frac{EspMédia}{2}) e \\ & (Interseção(EntAtual, EntAnterior) > 3)) e \\ & (EntAtual \neq primeira) \end{aligned} \quad (5.7)$$

onde $Interseção(EntAtual, EntAnterior)$ é o comprimento em pixels da interseção entre as entidades atual e anterior, $Compr(EntAtual)$ é o comprimento em pixels da entidade atual, e $EntAtual$ é uma referência à entidade atual (neste caso uma ligação). A ilustração da Figura 57a exibe em '1' a interseção e em '2' o resto. Neste caso ocorre uma quebra vertical na entidade atual (na ligação analisada) no ponto da entidade atual cuja coordenada horizontal é a mesma que a maior coordenada horizontal da entidade anterior. A Figura 58 exibe uma ilustração deste processo;

2. Interseção à direita: uma quebra da ligação é realizada caso a interseção entre a entidade atual (ligação) e a entidade anterior satisfizer a equação (5.7),

$$\begin{aligned}
 & (Interseção(EntAtual, EntPosterior) > EspMédia) e \\
 & ((Compr(EntAtual) - Interseção(EntAtual, EntPosterior) > \frac{EspMédia}{2}) e \\
 & (Interseção(EntAtual, EntPosterior) > 3)) e \\
 & (EntAtual \neq primeira)
 \end{aligned}
 \tag{5.8}$$

onde $Interseção(EntAtual, EntPosterior)$ é o valor em pixels da interseção entre as entidades atual e posterior. A ilustração da Figura 57b exibe em '1' a interseção e em '2' o resto. Neste caso ocorre uma quebra vertical na entidade atual (na ligação analisada) no ponto da entidade atual cuja coordenada horizontal é a mesma que a menor coordenada horizontal da entidade posterior. A Figura 58 exibe uma ilustração deste processo;



Figura 57 : Interseção entre ligação e graphemes. (a) Interseção à esquerda. (b) Interseção à direita.



Figura 58 : Quebra da ligação por interseção. (a) Detecção das ligações. (b) Quebra das ligações com interseção.

5.7.3 Reagrupamento das Ligações por Interseção

Neste passo, o algoritmo de segmentação realiza um reagrupamento das ligações com graphemes. Este reagrupamento é dependente da interseção existente entre as entidades envolvidas.

A meta deste passo é de detectar ligações que possivelmente pertençam ao grapheme vizinho, realizando a união destas duas entidades.

Vimos no passo anterior, que ligações que tem interseção com graphemes em determinadas situações são quebradas. Estas partes da ligação com interseção são fortes candidatas ao reagrupamento por interseção. Assim como no passo anterior, este processo deve capturar as coordenadas máxima e mínima horizontais para realizar o cálculo da interseção entre as entidades envolvidas, como visto na Figura 59a. A seguinte seqüência de análises devem ocorrer:

1. Se a relação entre entidade atual (neste caso uma ligação) e a próxima entidade satisfizer a equação (5.9)

$$(EntAtual = Ligação) e \quad (Interseção(EntAtual, EntPosterior) \geq \frac{Compr(EntAtual)}{2}), \quad (5.9)$$

então haverá um reagrupamento. Neste momento a ligação (entidade atual) passa a ser um grapheme. Agora precisamos saber se o reagrupamento será feito com a entidade anterior , posterior ou ambas. Deste modo, se a entidade posterior for um grapheme, a entidade atual (a ligação) será reagrupada com a entidade posterior, e se a entidade anterior for um grapheme, a entidade atual (a ligação) será reagrupada com a entidade anterior, como visto nos exemplos da Figura 60. Se for confirmado qualquer reagrupamento, é verificado se existe ponto de mínimo na entidade atual. Caso exista ponto de mínimo, e este ponto não estiver nos 3 pontos das extremidades da ligação, haverá uma quebra neste ponto de mínimo, como ilustra a Figura 60c4;

2. Caso a entidade atual não seja uma ligação e a entidade posterior for ligação, então se a relação entre estas entidades satisfizer a equação (5.10)

$$\begin{aligned}
 & (EntAtual = Grapheme) e \\
 & ((EntPosterior = Ligação) e (Posição(EntPosterior) \neq última)) e \quad (5.10) \\
 & (Interseção(EntAtual, EntPosterior) \geq \frac{Compr(EntPosterior)}{2}),
 \end{aligned}$$

então estes componentes são reagrupados;

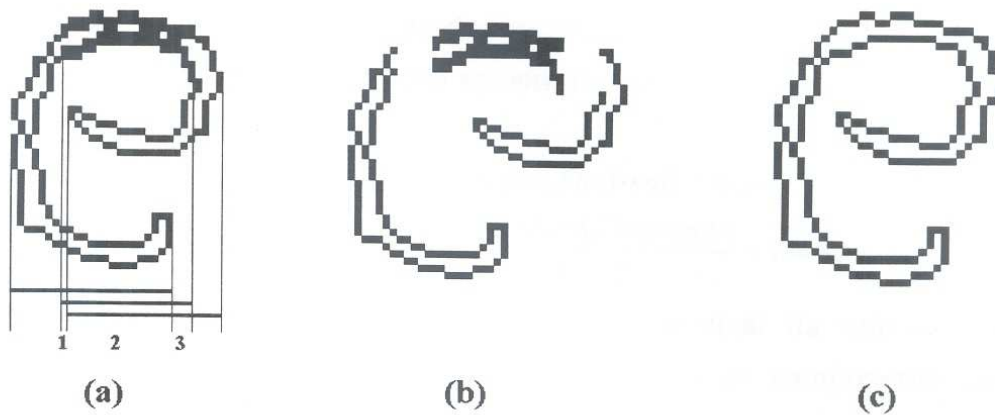


Figura 59 : Reagrupamento das ligações por interseção. (a) Interseção entre as entidades. (b) Quebra da letra. (c) Reagrupamento por Interseção.

O caso da análise 2, onde a entidade atual é grapheme e a entidade posterior é ligação e ultima entidade do componente, pode ser visualizada na letra “c” da palavra “setecentos” na transição da Figura 61c para a Figura 61d.

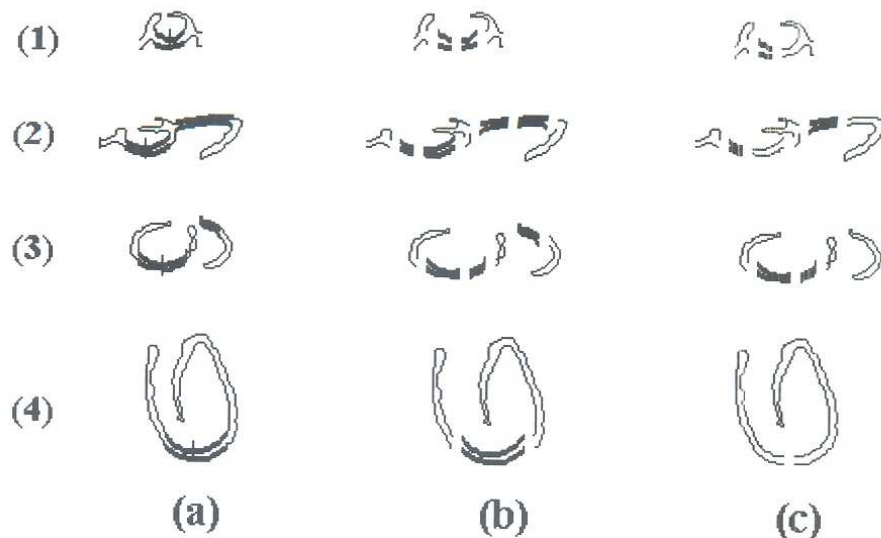


Figura 60 : Exemplos de Reagrupamento por interseção. (a) Detecção das ligações. (b) Quebra das ligações por interseção. (c) Reagrupamento das ligações por interseção.

5.7.4 Reagrupamento das Ligações Altas

Neste passo, o algoritmo de segmentação realiza o reagrupamento das ligações que forem consideradas *altas* com relação ao componente a que pertencem e forem entidades das extremidades do componente. Este processo foi incluído para reagrupar as ligações que possivelmente pertençam a um grapheme comparando as alturas da ligação e do componente a que a ligação pertence. Caso a análise da ligação satisfizer a equação (5.11)

$$\begin{aligned} & (EntAtual = Ligação) e \\ & ((Posição(EntAtual) = primeira) ou (Posição(EntAtual) = última)) e \\ & ((Ymenor(EntAtual) + \frac{Altura(EntAtual)}{2}) > (\frac{Altura(CompAtual)}{2} + Ymenor(CompAtual))), \end{aligned} \quad (5.11)$$

onde $Ymenor(EntAtual)$ é a menor coordenada vertical da entidade atual, $Ymenor(CompAtual)$ é a menor coordenada vertical do componente atual e $Altura(CompAtual)$ é a altura do componente atual, a ligação será reagrupada. A Figura 61 ilustra este conceito na letra “t” da palavra “setecentos”.

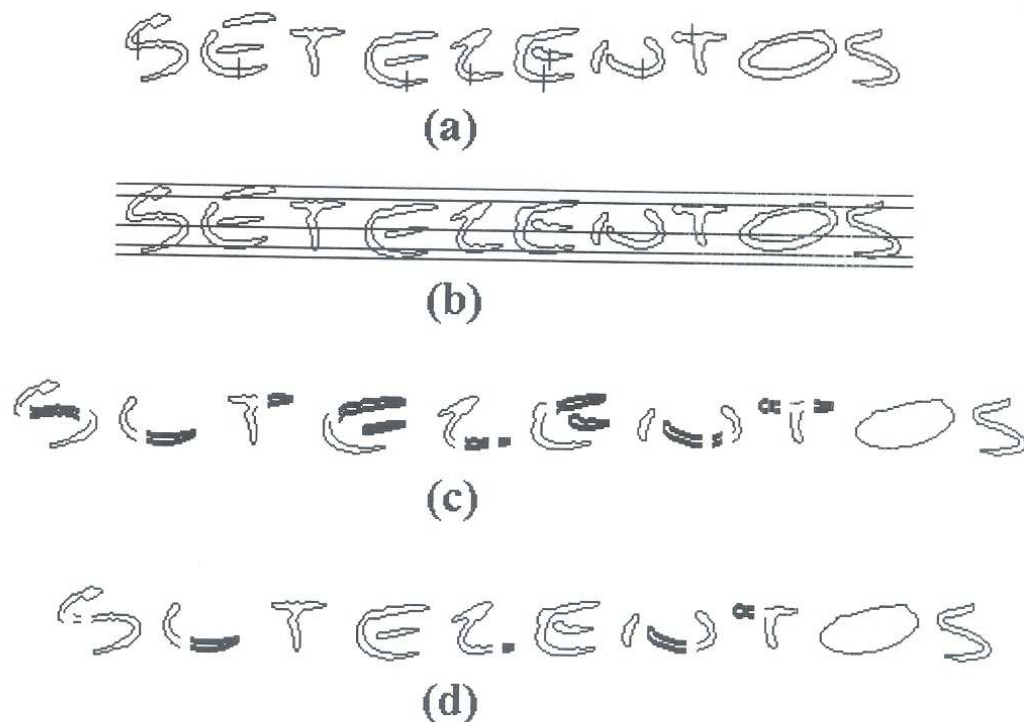


Figura 61 : Reagrupamento por ligações altas e por interseção. (a) Pontos de mínimo. (b) Zonas da palavra e linha média. (c) Quebra das ligações por interseção. (d) Reagrupamento por interseção e por ligações altas.

Na letra “S” da palavra ilustrada na Figura 61d, podemos observar um fragmento de traço da escrita gerado por um ponto de mínimo que pode ser

observado na Figura 61a. Neste caso, o reagrupamento foi realizado por interseção. Já na última letra “t” da mesma ilustração, podemos observar um reagrupamento por altura não realizado por causa de um ponto de mínimo. Se compararmos a Figura 61a e a Figura 61c, notaremos que este ponto de mínimo pertence à ligação e está no limite entre a ligação e o grapheme a que a ligação seria reagrupada, como ilustra a Figura 61d

Note que os traços da primeira letra “e” da palavra ilustrada na Figura 61 que não estão conectados ao corpo da imagem são descartados por não terem pontos que interceptem a linha média do corpo da palavra, como visto na Figura 61b. Este conceito foi abordado na seção 5.4.4.

5.8 Conclusão

Neste capítulo, foram abordados as técnicas e os métodos usados para realizar a pesquisa proposta, ou seja, o processamento necessário para segmentar palavras manuscritas em entidades que se constituem de graphemes e ligações. Grapheme é uma entidade que podem ser parte de uma letra, uma letra inteira e mais de uma letra. Ligações são os traços da escrita que ligam uma letra a outra e são consideradas como ruídos, pois aumentam os detalhes a serem analisados nas letras.

Neste trabalho, evitou-se a quebra de palavras em entidades constituídas por mais de uma letra. Deste modo, o sistema quebra a palavra em várias partes, tentando gerar, desta maneira, entidades que podem ser parte de uma letra ou, no máximo, uma letra inteira. Esta escolha de quebrar o traço da palavra em porções que contenham no máximo uma letra inteira faz sentido ao passarmos estas entidades a um sistema de reconhecimento que seja hábil para determinar qual a melhor segmentação a ser feita para aquela seqüência de entidades. Desta maneira, a segmentação final da palavra fica por conta do sistema que fará o reconhecimento desta palavra.

Este modelo de segmentação não necessita de qualquer tipo de reconhecimento para realizar sua tarefa. O processamento é todo baseado nas circunstâncias em que pontos de mínimo e máximo, e ligações se encontram. A segmentação é adaptada à palavra através da espessura média do traço da escrita analisada. Este é o grande diferencial deste modelo de segmentação, ou

seja, sua capacidade de adaptação ao tipo de escrita da palavra que esta sendo analisada tornando a análise e segmentação da palavra mais confiáveis.

A segmentação é realizada explicitamente, ou seja, a imagem da palavra é dividida em porções de imagem que, como visto acima, são constituídas de graphemes ou ligações.

A detecção das ligações existentes entre as letras da palavra também recebeu filtros que a tornou um processo mais confiável que os métodos de detecção de ligações reportados pela comunidade científica visto nos estudos da seção 2.6. Esta detecção é importante por se tratar de porções da palavra que não pertencem às letras. Desta maneira, podemos prevenir que uma letra torne-se ruidosa por estar conduzindo uma ligação que não a pertence.

Capítulo 6

Análise dos Resultados

6.1 Introdução

Como visto no Capítulo 5, a principal tarefa do processo de segmentação realizado nesta pesquisa é o de particionar um texto manuscrito em dois tipos de entidade de alto nível: *graphemes* e *ligações*. As definições destes elementos são:

- Entidade – é um segmento do traço da palavra resultado do processo de segmentação;
- Grapheme - é uma entidade que pode ser constituído de parte de uma letra, uma letra inteira, ou mais de uma letra; e
- Ligação – é uma entidade que liga as letras de uma palavra mas não faz parte destas letras.

Nesta etapa, testamos o sistema de segmentação através das imagens de palavras manuscritas que compõem a base de teste descrita na seção 5.3. Como descrito na seção 5.5.5, o algoritmo de segmentação pré-classifica as entidades geradas como sendo entidades que são do tipo *ligação* e entidades que são do tipo *grapheme*. Os resultados que são desejados pelo sistema de segmentação estão descritos a seguir:

- Separar as ligações das letras que formam a palavra, e
- Segmentar uma determinada letra em uma quantidade única de entidades.

Como descrito na seção 5.3, a análise dos resultados do processo de segmentação das palavras da base de teste foi feita avaliando-se visualmente cada palavra segmentada. A etapa de avaliação das palavras, resultou em dois

tipos de análise realizadas: *análise por letra* e *análise geral*. Ambas as análises são discutidas nas seções 6.2 e 6.3 respectivamente.

Um outro tipo de análise realizada para testar a eficiência do algoritmo de segmentação pesquisado neste trabalho, foi uma comparação entre os resultados do processo de segmentação realizado neste trabalho e o processo de segmentação tradicional. Os dois processos estão descritos a seguir:

- Processo de segmentação abordado neste trabalho - realiza as quebras pelos pontos de mínimo da borda superior do contorno externo da palavra e pelas ligações detectadas através da espessura média do traço da escrita, e
- Processo de segmentação tradicional - realiza as quebras somente pelos pontos de mínimo da borda superior do contorno externo.

Os resultados obtidos através da segmentação da palavra pelo processo tradicional são discutidos na seção 6.4.

6.2 Análise por Letra

Como descrito na seção 6.1, a análise dos resultados do processo de segmentação das palavras da base de teste foi feita avaliando-se visualmente cada palavra segmentada. Após esta avaliação, foram formulados alguns dados estatísticos de cada letra que compõe a palavra segmentada. Estes dados estatísticos são exibidos na seção 6.2.1.

Os comentários e exemplos dos dados obtidos através das análises estatísticas, expostas na seção 6.2.1, estão descritos na seção 6.2.2. Estes comentários foram abordados pela semelhança existente entre as características da letra e o número de entidades em que esta letra foi segmentada.

6.2.1 Estatísticas

Estes dados estatísticos ajudam a visualizar e compreender o comportamento do algoritmo de segmentação quando conseguimos realizar comparações entre letras com comportamento semelhante quando processados por este algoritmo.

A Tabela 2 exhibe as estatísticas da análise dos resultados obtidos pelo sistema de segmentação após processar as palavras da base de teste, onde é

exposta a quantidade total de amostras de uma determinada letra e a porcentagem da quantidade de entidades em que esta letra foi particionada.

Tabela 2 : Estatísticas do particionamento das letras pelo sistema proposto.

Categ.	Letra	Nº Seg. (%)								Total de Letras
		1	2	3	4	5	6	7	8	
1	a	70	7	20	3	-	-	-	-	68
6	A	67	-	23	10	-	-	-	-	9
3	c	13	87	-	-	-	-	-	-	216
1	d	31	48	16	5	-	-	-	-	69
1	D	60	35	5	-	-	-	-	-	23
2	e	93	7	-	-	-	-	-	-	288
3	E	80	10	10	-	-	-	-	-	21
2	i	100	-	-	-	-	-	-	-	220
2	l	100	-	-	-	-	-	-	-	56
3	L	34	64	-	-	-	-	-	-	12
4	m	-	-	20	15	40	10	8	7	58
5	M	-	41	59	-	-	-	-	-	12
4	n	-	23	30	36	9	2	-	-	250
5	N	-	28	72	-	-	-	-	-	32
1	o	74	14	9	3	-	-	-	-	320
1	q	75	7	14	3	1	-	-	-	84
1	Q	40	22	28	5	5	-	-	-	18
4	r	37	28	35	-	-	-	-	-	46
6	R	100	-	-	-	-	-	-	-	6
4	s	78	14	8	-	-	-	-	-	86
6	S	72	19	7	2	-	-	-	-	42
3	t	81	13	6	-	-	-	-	-	220
5	u	-	9	86	5	-	-	-	-	113
5	v	-	25	75	-	-	-	-	-	46
5	V	-	65	35	-	-	-	-	-	20
6	z	80	12	8	-	-	-	-	-	135
6	Z	68	27	5	-	-	-	-	-	37
	Média (%)	50,7	21,8	20,8	3,6	2,2	0,4	0,3	0,2	2507 10

6.2.2 Observações

Como visto no Capítulo 5, a segmentação das palavras em entidades de alto nível (graphemes) foi baseada nas ligações e pontos de mínimo da borda superior do contorno externo detectados. Portanto, através da Tabela 2,

podemos observar que o número de quebras realizado sob a letra é semelhante entre as letras com as mesmas características de formato, ou seja, letras com *ciclo* interno ao corpo da palavra, letras com *concauidade para a direita*, letras com *concauidade para baixo*, e letras com *concauidade para cima*. As observações realizadas sob estas características serão abordadas nas subseções abaixo, considerando a semelhança entre estas.

Segmentação das Letras com Ciclo no Corpo da Palavra

Como visto na Tabela 2, as letras “a”, “d”, “o”, “q” e “Q” tiveram um comportamento semelhante ao passarem pelo processo de segmentação. Uma das formas de se observar esta semelhança é através do número de segmentos em que estas letras foram particionadas. As características destas letras que delinearam o comportamento do algoritmo estão ilustradas na Figura 62.

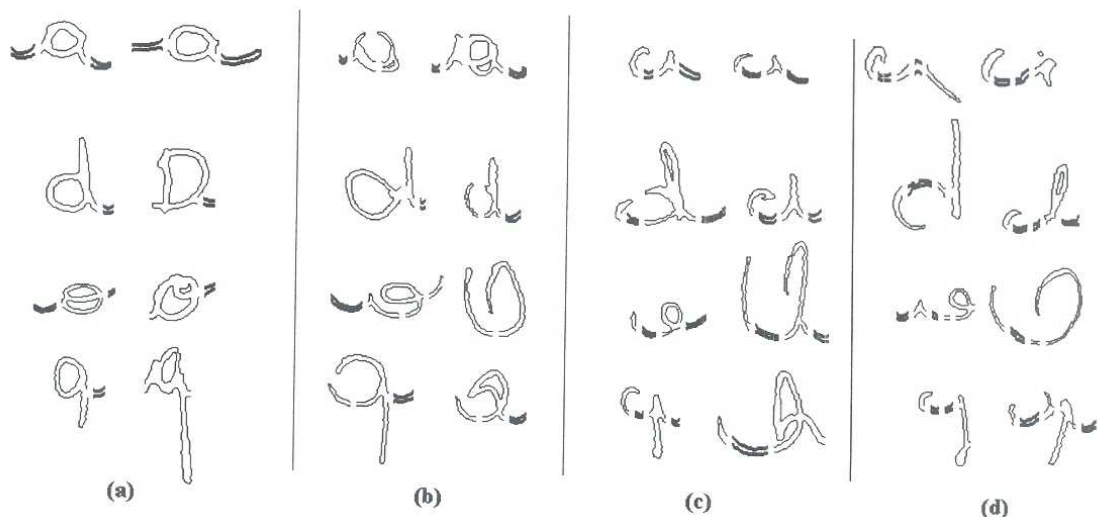


Figura 62 : Exemplos de quebra de letras com ciclo no corpo da palavra. (a) Letra segmentada inteira. (b) Letra segmentada em duas entidades. (c) Letra segmentada em três entidades. (d) Letras segmentadas em quatro entidades.

Observando a Figura 62a, notaremos que as letras “a”, “d”, “o”, “q” e “Q”, foram segmentadas em uma única entidade nas letras fechadas e sem ponto de mínimo na borda superior do contorno externo da entidade.

Na Figura 62b, podemos notar que as letras foram segmentadas em duas entidades por dois motivos:

- A letras estar aberta e, portanto, com um ponto de mínimo no centro da concauidade da abertura, ou
- Terem um ponto de mínimo na borda superior do contorno externo da letra fechada.

Na Figura 62c, podemos notar que as letras foram segmentadas em três entidades nas letras abertas e com ligação detectada na concavidade formada onde os ciclos não estão fechados.

Na Figura 62d, podemos notar que as letras foram segmentadas em quatro entidades nas letras com as seguintes características:

- Letra aberta, com ligação detectada na concavidade formada a partir da abertura, e com interseção entre a ligação e a entidade posterior e/ou anterior.

Devemos notar na Figura 62d que os casos em que a interseção está entre a ligação e a entidade posterior e estas entidades foram reagrupadas, se havia um ponto de mínimo no contorno superior da ligação reagrupada é realizada uma quebra neste ponto.

Deste modo, a quantidade de entidades geradas na segmentação realizada em letras que tenham ciclo no corpo da palavra é dependente do comportamento do ciclo destas palavras, da detecção de ligações e suas interseções entre a entidade anterior e posterior, e a detecção de pontos de mínimo na borda superior do contorno externo da letra, como ilustrado na Figura 62.

Segmentação das Letras sem Ponto de Mínimo

Como visto na Tabela 2, as letras “e”, “i” e “l” tiveram um comportamento semelhante ao passarem pelo processo de segmentação. Podemos perceber isto pela quantidade de quebras que foram realizadas nestas letras. Esta semelhança no processo de segmentação pode ser atribuída às semelhanças encontradas nas características destas letras. As características que delinearam o comportamento do algoritmo estão ilustradas na Figura 63.

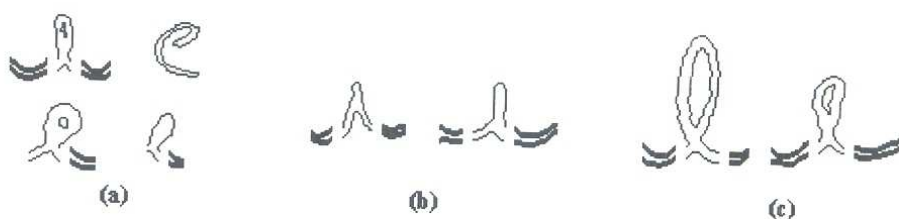


Figura 63 : Exemplos de quebra de letras que não tem ponto de mínimo na borda superior. (a) Letra “e” segmentada em uma e em duas entidades. (b) Letra “i” segmentada em uma entidade. (c) Letra “l” segmentada em uma entidade.

Observando a Figura 63a, notaremos que a letra “e” foi segmentada em uma única entidade nas letras escritas cursivamente e nas letras escritas em letra de forma que constituem um componente unitário, ou seja, em letras de forma que não tem ligação com outros componentes. Já as letras escritas em letras de forma e que tem ligações com outras entidades, são particionadas em duas entidades por terem uma interseção entre a ligação detectada e a entidade anterior, como ilustra a Figura 63a.

Nas letras ilustradas na Figura 63, podemos notar que é incomum a detecção de pontos de mínimo, por isto estas letras são particionadas em uma única entidade.

Segmentação das Letras com Concauidade para a Direita

Como visto na Tabela 2, as letras “c”, “E”, “t” e “L” tiveram um comportamento semelhante ao passarem pelo processo de segmentação. Podemos perceber isto pela quantidade de quebras que foram realizadas nestas letras. Esta semelhança no processo de segmentação pode ser atribuída às semelhanças encontradas nas características destas letras. As características que delinearam o comportamento do algoritmo estão ilustradas na Figura 64.

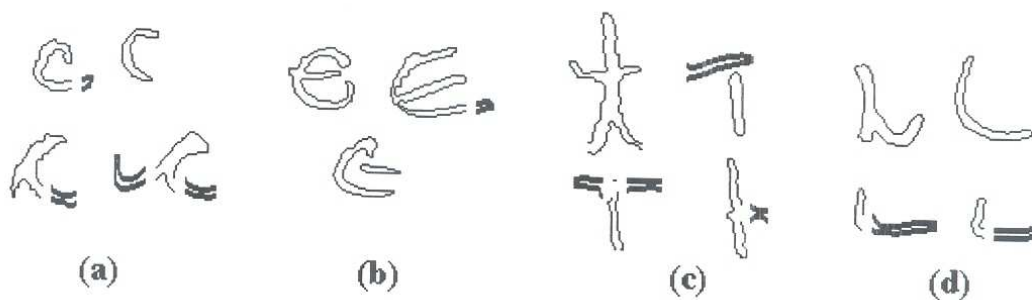


Figura 64 : Exemplos de quebra das letras com concauidade para a direita. (a) Letra “c” segmentada em uma e duas entidades. (b) Letra “E” segmentada inteira. (c) Letra “t” segmentada em uma, duas e três entidades. (d) Letra “L” segmentada em uma e duas entidades

Na Figura 64a, na Figura 64b e na Figura 64c, podemos perceber que as letras “c”, “E” e “t” foram segmentadas em uma única entidade nas letras que constituem um componente unitário, ou seja, letras que não tem ligação com outros componentes, e nas letras que finalizam o componente a que pertencem. Isto ocorre porque as possíveis ligações que são detectadas nestas letras são reagrupadas pela interseção entre estas ligações e o grapheme anterior, como visto na seção 5.7.3. Já as letras que tem ligação com outra

letra à direita, ou seja, letras que não finalizam um componente, são particionadas em duas ou três entidades porque parte destas letras são detectadas como ligação e, neste caso, não há reagrupamento por interseção, como descrito nas seções 5.7.2 e 5.7.3.

Na Figura 64d, podemos notar que os dois exemplos da letra “L” superiores foram segmentados em uma única entidade porque nas partes côncavas do traço a ligação foi filtrada, como visto na seção 5.5.5, e a quebra pelo ponto de mínimo detectado na parte côncava não foi efetuada porque formaria entidades pequenas com relação à espessura média da escrita, como visto na seção 5.6.3. Já nos exemplos da letra “L” inferiores da mesma figura, foram detectadas ligações e, por isso, haverá uma quebra particionando a letra em duas entidades.

Segmentação das Letras com Concavidade para Baixo

As letras “m”, “n”, “r” e “s” tem um comportamento semelhante ao passarem pelo processo de segmentação. Esta semelhança no processo de segmentação pode ser atribuída às semelhanças encontradas nas características destas letras. Este fato pode ser compreendido após percebermos que a estrutura destas letras forma uma concavidade orientada para baixo. As ilustrações da Figura 65 exibem alguns exemplos da segmentação destas letras.

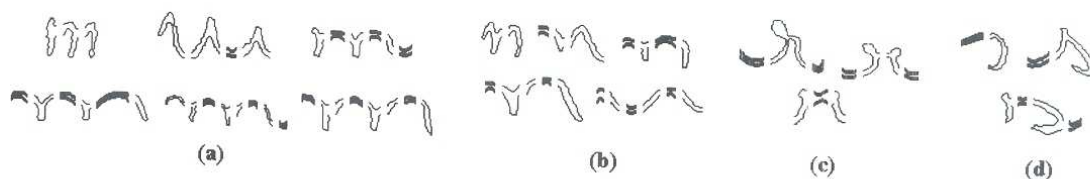


Figura 65 : Exemplos de quebra de letras com concavidade para baixo. (a) Letra “m” segmentada em três, quatro, cinco, seis, sete e oito entidades. (b) Letra “n” segmentada em duas, três, quatro, cinco e seis entidades. (c) Letra “r” segmentada em uma, duas e três entidades. (d) Letra “s” segmentada em uma, duas e três entidades.

Nos exemplos da Figura 65 podemos perceber que cada segmento de letra que forma uma concavidade para baixo pode ser segmentado de duas a quatro entidades. A letra “m”, ilustrada na Figura 65a, é formada por duas concavidades para baixo onde cada segmento côncavo pode ser fragmentado em:

- Duas entidades - quando a quebra é realizada somente pelo ponto de mínimo da borda superior, com ilustra a quebra das duas concavidades da primeira letra na primeira linha da Figura 65a;
- Três entidades - quando o fundo da concavidade é detectado como ligação, com ilustra a quebra das duas concavidades da terceira letra na primeira linha da Figura 65a; e
- Quatro entidades - quando o fundo da concavidade é detectado como ligação e o ponto de mínimo da borda superior estiver a uma distância limiar (ver seção 5.6.3) da ligação, com ilustra a quebra das duas concavidades da terceira letra na segunda linha da Figura 65a.

Deste modo, o número de concavidades e o formato do mesmo guiarão o processo de segmentação nestas letras. O mesmo processo aplicado à letra “m” é aplicado às letras “n”, “r” e “s”, como ilustrado na Figura 65.

Na letra “s” ilustrada na Figura 65d, podem ocorrer agrupamentos de ligações com graphemes pelo processo de *reagrupamento por interseção*, como descrito na seção 5.7.3.

Segmentação das Letras com Concavidade para Cima

As letras “M”, “N”, “u”, “v” e “V” tem um comportamento semelhante ao passarem pelo processo de segmentação. Esta semelhança no processo de segmentação pode ser atribuída às semelhanças encontradas nas características destas letras. Este fato pode ser compreendido após percebermos que a estrutura destas letras forma uma concavidade orientada para cima. As ilustrações da Figura 66 exibem alguns exemplos da segmentação destas letras.

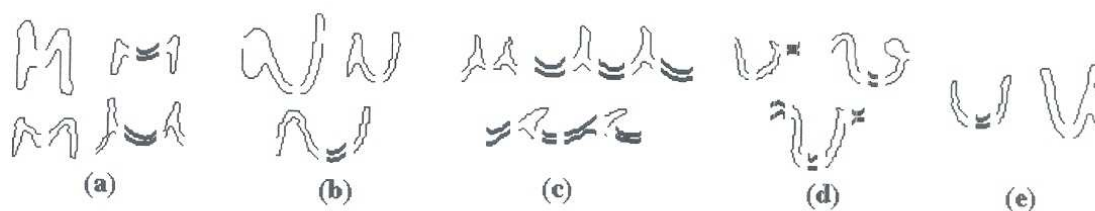


Figura 66 : Exemplos de quebra de letras com concavidade para cima. (a) Letra “M” segmentada em duas e três entidades. (b) Letra “N” segmentada em duas, três e quatro entidades. (c) Letra “u” segmentada em duas, três e quatro entidades. (d) Letra “v” segmentada em duas, três e quatro entidades. (e) Letra “V” segmentada em duas e três entidades.

Nos exemplos ilustrados na Figura 66 podemos perceber que cada segmento de letra que forma uma concavidade para cima pode ser segmentado, no geral, de duas a quatro entidades. A letra “u”, ilustrada na

Figura 66c, é formada por uma concavidade para cima onde o segmento côncavo pode ser fragmentado em:

- Duas entidades - quando a quebra é realizada somente pelo ponto de mínimo da borda superior, com ilustra a quebra da concavidade da primeira letra na primeira linha da Figura 66c;
- Três entidades - quando o fundo da concavidade é detectado como ligação, com ilustra a quebra da concavidade da segunda letra na primeira linha da Figura 66c; e
- Quatro entidades - quando o fundo da concavidade for detectada como ligação e houver interseção (ver seção 5.7.2) entre a ligação detectada e uma das entidades vizinhas, com ilustra a quebra da concavidade da letra na segunda linha da Figura 66c.

Deste modo, o formato da concavidade guiará o processo de segmentação nestas letras. O mesmo processo aplicado à letra “u” é aplicado às letras “M”, “N”, “v” e “V”, como ilustrado na Figura 66.

Note que a letra “u” segmentada em quatro entidades ilustrada na segunda linha da Figura 66c, está inclinada com relação ao eixo vertical. Esta inclinação ocasionou as seguintes características no processo de segmentação desta letra (ver seção 5.7.2):

- Parte da letra que deveria ser detectada como grapheme foi detectada como ligação; e
- A quebra da ligação detectada em duas entidades.

Segmentação de Outras Letras

As letras “A”, “R”, “S”, “Z” e “z” não têm uma estrutura semelhante mas, no entanto, comportam-se de maneira semelhante ao passarem pelo processo de segmentação gerando um número de entidades parecidos. As ilustrações da Figura 67 exibem alguns exemplos da segmentação destas letras.

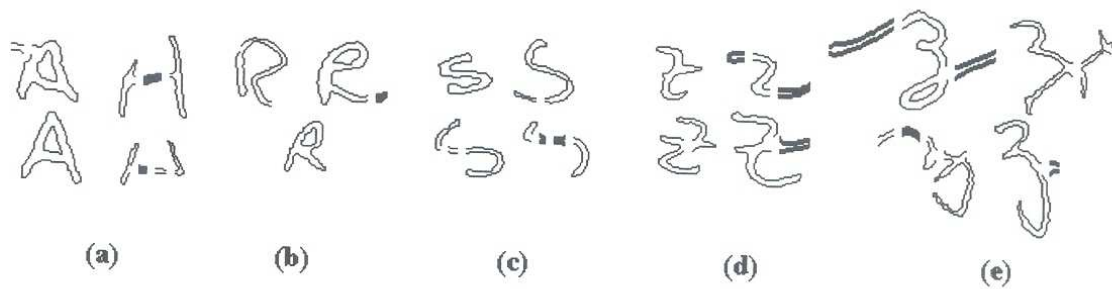


Figura 67 : Exemplo de quebra de outras letras. (a) Letra “A” segmentada em uma, três e quatro entidades. (b) Letra “R” segmentada em uma entidade. (c) Letra “S” segmentada em uma, duas, três e quatro entidades. (d) Letra “Z” segmentada em uma, duas e três entidades. (e) Letra “z” segmentada em uma e três entidades.

Notem que apesar de na Tabela 2 estar especificado que a letra “R” foi quebrada todas as vezes em uma entidade, a estrutura desta letra é muito parecida, em alguns casos, com a estrutura da letra “A” que foi segmentada em até quatro entidades. Não fica difícil de visualizar que se caso a letra “R” for escrita de forma a deixar uma abertura no ciclo desta letra, ela passará a ter o comportamento da letra “A” no processo de segmentação.

Devemos observar que apesar da relação implícita existente entre a letra “A” e a letra “R” no processo de segmentação a Tabela 2 mostra que a quantidade de amostras destas letras não é significativa para obter-se resultados comparativos mais profundos.

Como visto na Figura 67c, a segmentação da letra “S” é dependente da inclinação do traço central desta letra e dependente da interseção entre as entidades que formam esta letra. Por exemplo, se o traço inferior da letra “S” segmentada em quatro entidades tivesse interseção com as ligações, a quebra deste exemplo seria determinada somente pela existência ou não de um ponto de mínimo.

A segmentação da letra “Z”, ilustrada na Figura 67d, é semelhante à segmentação da letra “S” quando não existe o traço que corta a letra ao meio. A existência deste traço pode gerar pontos de mínimo ocasionando um quebra desta letra em duas entidades.

A segmentação da letra “z”, ilustrada na Figura 67e, pode gerar uma, duas ou três entidades dependendo da inclinação da parte não cíclica desta letra. Outro fator que determina a quantidade de entidades geradas desta letra é se a parte cíclica esta fechada ou aberta. Caso a parte cíclica este aberta, um ponto de mínimo é detectado no cruzamento do traço da escrita nesta letra e uma quebra é realizada neste ponto.

6.3 Análise Geral

6.3.1 Tempo de Processamento

O tempo médio que o processo de segmentação leva para segmentar uma palavra foi estimado em 0,5 segundo. Este tempo foi obtido após calcularmos o tempo total que o algoritmo levou para processar 1022 imagens aleatórias continuamente. O resultado da divisão entre o tempo total, em segundos, pela quantidade de imagens processadas é o valor do tempo médio. Para esta estimativa, foram selecionadas imagens de diversas palavras diferentes.

O software utilizado para o desenvolvimento do sistema foi o Visual C++ 5.0. O hardware utilizado para o desenvolvimento da pesquisa teve as seguintes características:

- Processador Pentium II Intel - 266 MHz;
- 64 MB de memória RAM;
- HD de 13 GB;

6.3.2 Problemas que Causam Erros

Em alguns casos isolados, ocorreram sobreposições de parte das letras de uma palavra formando grandes entidades e geralmente com mais de um ciclo interno, como ilustra a seqüência “ta” na Figura 68a.

Nestes casos, a segmentação é dependente do ponto de mínimo que possivelmente seja detectado na borda superior do contorno externo das letras que se tocam. Podemos observar que em um dos exemplos das ilustrações da Figura 68a, foi detectado um ponto de mínimo na letra “t” e, neste caso, ocorreu uma quebra neste ponto. A Figura 68b ilustra a segmentação da seqüência de letras exibida na Figura 68a.

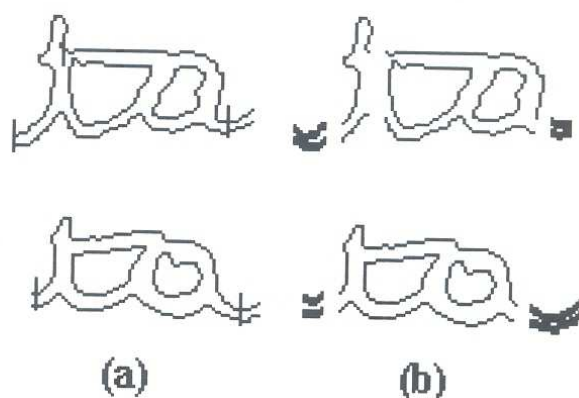


Figura 68 : Exemplo de letras que se tocam. (a) Parte da imagem original. (b) Parte da imagem segmentada.

Outros tipos de sobreposições, como ilustrado na Figura 69a, também são frequentes nos diversos estilos de manuscritos, prejudicando a quebra das letras sobrepostas. A Figura 69b, ilustra a segmentação das respectivas palavras da Figura 69a.

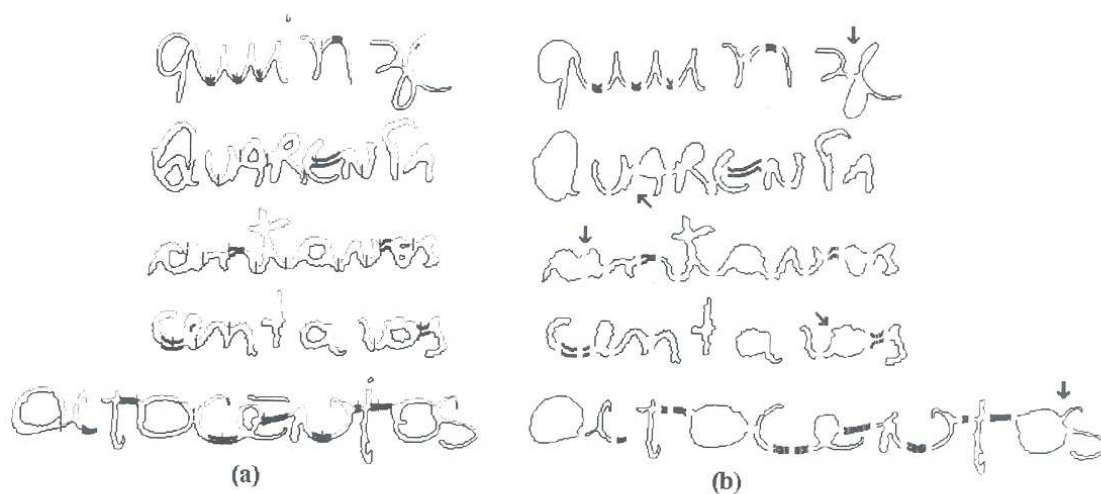


Figura 69 : Exemplos de problemas encontrados na segmentação. (a) Detecção dos pontos de mínimo e das ligações. (b) Quebra da palavra.

Estes problemas são compreendidos após percebermos que o algoritmo de segmentação realiza as quebras em uma linha vertical, envolvendo a borda superior e a borda inferior do contorno externo do componente, e que nas sobreposições perde-se a informação das bordas envolvidas nestes casos.

Como visto na seção 5.6.2 e na seção 5.6.3, o algoritmo realiza a quebra da palavra baseado nas ligações detectadas e nos pontos de mínimo da borda superior do contorno externo da palavra. Em alguns casos, a regra utilizada para realizar a detecção dos pontos de quebra não consegue detectar pontos

de mínimo e nem ligação, como no caso da palavra “cinquenta” ilustrado na Figura 70.

Como visto na seção 5.6.3, após a detecção dos pontos de mínimo é realizada uma filtragem nestes pontos para garantir que os mínimos detectados são pontos de quebra. Se observarmos a imagem da palavra “oitocentos” ilustrada na Figura 70a, notaremos que na sobreposição entre as letras “oi” existia um ponto de mínimo muito próximo a um ponto de máximo e, por isso, foram eliminados.

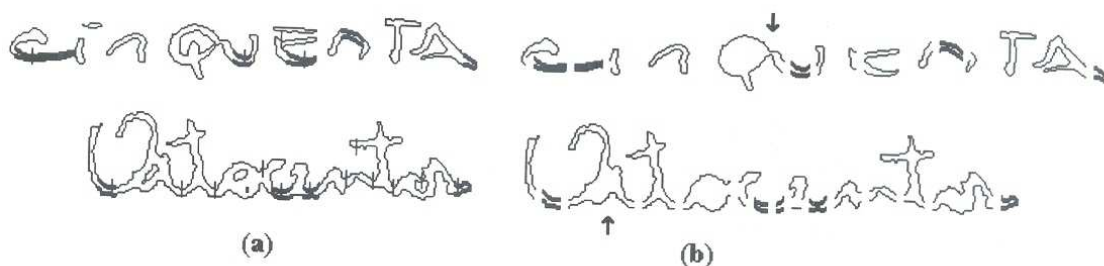


Figura 70 : Exemplo de problemas encontrados na segmentação. (a) Detecção de pontos de mínimo e das ligações. (b) Apontamento para mínimo não detectado e mínimo excluído.

Como exposto na seção 5.5.5, neste trabalho é realizada a detecção de possíveis ligações entre as letras de uma palavra. Um dos objetivos desta detecção é de separar a letra das ligações que neste trabalho são consideradas como ruídos das letras segmentadas. A Figura 71a ilustra a detecção das ligações sem o processo de filtragem por inclinação, descrito na seção 5.5.5. A Figura 71b ilustra a o processo de segmentação após aplicado todos os filtros da detecção das ligações.

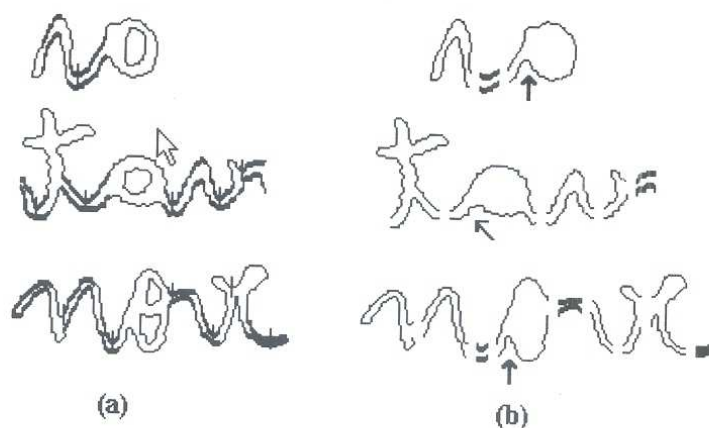


Figura 71 : Exemplos de problema não tratado na segmentação. (a) Detecção de ligações sem o filtro por inclinação. (b) Segmentação após aplicado todos os filtros da detecção das ligações.

Um problema não tratado neste trabalho é a detecção da ligação no segmento cuja inclinação ultrapassa a inclinação limiar, como discutido na seção 5.5.5. A Figura 71b aponta para alguns destes casos.

6.4 Segmentação pelo Processo Tradicional

A segmentação pelo processo tradicional baseia-se somente nos pontos de mínimo da borda superior do contorno externo da palavra para realizar a quebra da palavra. Portanto, a eficiência deste modelo de segmentação é totalmente dependente da detecção dos pontos de mínimo e da validação destes pontos como sendo pontos de quebra.

O objetivo de se realizar uma segmentação pelo processo tradicional é gerar resultados que possam ser comparados aos resultados do modelo usado neste trabalho. Deste modo, poderemos descrever com mais precisão a eficiência de um modelo de segmentação que utiliza as ligações existentes entre as letras de uma palavra como informação relevante para o processo de quebra.

No modelo de segmentação proposto para esta pesquisa, a detecção das ligações de uma palavra tenta diminuir a variabilidade de um letra segmentada, como discutido na seção 5.5.5. Em uma segmentação tradicional, a única informação que se tem para realizar a quebra da palavra são os pontos de mínimo (ou vales) detectados. Deste modo, no processo tradicional de segmentação as porções do traço da escrita que na realidade são ligações, e que portanto não fazem parte da letra segmentada, ficam anexadas nas letras ou em partes das letras, tornando-se ruídos para o processo de reconhecimento. Esta é a característica que torna o processo de detecção das ligações um processo bastante útil no processo de segmentação e reconhecimento da palavra.

Para ilustrar a segmentação pelo processo tradicional e pelo processo proposto por esta pesquisa, a Figura 72 exibe algumas palavras segmentadas pelo processo tradicional em (a) juntamente com suas respectivas palavras segmentadas pelo processo proposto por esta pesquisa em (b).

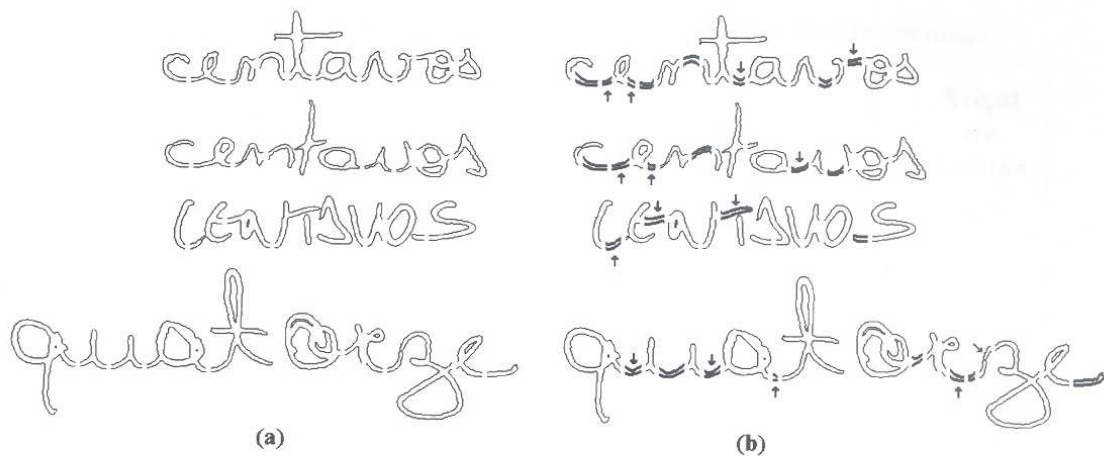


Figura 72 : Segmentação pelo processo tradicional e pelo processo proposto nesta pesquisa. (a) Segmentação por pontos de mínimo. (b) Segmentação por pontos de mínimo e por ligações.

Notem que no caso da palavra “centavos” ilustrada na terceira linha de cima para baixo da Figura 72, na segmentação pelo processo tradicional as letras “e” e “n” foram segmentadas com partes das letras vizinhas que se tornam ruídos, e na segmentação pelo processo que detecta as ligações da palavras, estes mesmos ruídos foram segmentados como ligações deixando, deste modo, a decisão final para o melhor reagrupamento das entidades para o algoritmo de reconhecimento.

A Tabela 3 exhibe as estatísticas da análise dos resultados obtidos pelo sistema de *segmentação tradicional* após processar as palavras da base de teste, onde é exposta a quantidade total de amostras de uma determinada letra e a porcentagem da quantidade de entidades em que esta letra foi particionada.

Tabela 3 : Estatísticas do particionamento das letras pelo sistema tradicional.

Nº Seg. (%) Letra	1	2	3	4	5	6	7	8	Total de Letras
a	76	14	-	-	-	-	-	-	60
A	85	15	-	-	-	-	-	-	7
c	98	2	-	-	-	-	-	-	52
d	25	60	15	-	-	-	-	-	59
D	48	40	12	-	-	-	-	-	23
e	96	4	-	-	-	-	-	-	167
E	90	10	-	-	-	-	-	-	9
i	100	-	-	-	-	-	-	-	65
l	100	-	-	-	-	-	-	-	56
L	33	67	-	-	-	-	-	-	12
m	-	2	96	2	-	-	-	-	58
M	-	100	-	-	-	-	-	-	10
n	3	97	-	-	-	-	-	-	71
N	10	90	-	-	-	-	-	-	11
o	83	17	-	-	-	-	-	-	109
q	68	27	5	-	-	-	-	-	22
Q	12	63	25	-	-	-	-	-	8
r	83	17	-	-	-	-	-	-	29
R	100	-	-	-	-	-	-	-	2
s	100	-	-	-	-	-	-	-	52
S	60	40	-	-	-	-	-	-	23
t	100	-	-	-	-	-	-	-	125
u	-	98	2	-	-	-	-	-	66
v	-	100	-	-	-	-	-	-	43
z	70	29	1	-	-	-	-	-	94
Z	75	18	7	-	-	-	-	-	16
Média (%)	58,2	35	6,2	0,6	-	-	-	-	124 10

As estatísticas da Tabela 3 foram obtidas da análise visual de 285 palavras da base de dados disponível, discutida na seção 5.3, selecionadas de forma a obter todas as letras que compõem o contexto da base.

Para que possamos obter uma comparação real entre modelos de segmentação de palavras manuscritas, devemos dispor de um módulo de reconhecimento que aceite como dados de entrada a saída dos modelos de segmentação a serem comparados.

6.5 Conclusões

No geral, o algoritmo de segmentação proposto para esta pesquisa se comportou como esperado.

O processo de segmentação desta pesquisa foi validado pela base de dados descrita na seção 5.3. Esta base é formada por palavras manuscritas de extensos de valores de cheques bancários brasileiros. Desta forma, as letras “b”, “f”, “g”, “h”, “j”, “k”, “y” e “x” não foram avaliadas pelo sistema proposto. Mas poderíamos prever o comportamento da segmentação destas letras designando cada letra para pertencer a uma das categorias descritas na seção 6.2.2. Por exemplo, poderíamos dizer que as letras “b” e “g” teriam o mesmo comportamento das letras comentadas na subseção “*Segmentação das Letras com Ciclo no Corpo da Palavra*”, que as letras “f” e “j” teriam o mesmo comportamento das letras comentadas na subseção “*Segmentação das Letras sem Ponto de Mínimo*”, que as letras “h”, “k”, “y” e “x” teriam o mesmo comportamento das letras comentadas na subseção “*Segmentação das Letras com Concauidade para Cima*”.

Como visto na seção 6.3, foram encontrados problemas na segmentação de palavras cujas letras tinham as seguintes características:

- Interceptam-se em pontos que não são ligações ou letras que se sobrepunham a letras vizinhas;
- Ponto de quebra não detectado por não haver detecção de ponto de mínimo (sem concauidade) e nem detecção de ligação; e
- Ligações não detectadas por terem o comportamento de um grafeme.

Os casos comentados na seção 6.3 tornaram-se assunto para trabalhos futuros em forma de um passo de pós-processamento que consiga:

- Detectar e quebrar entidades grandes demais para serem uma única letra; e
- Detectar ligações cuja inclinação seja superior à inclinação limiar.

Capítulo 7

Conclusões Gerais

Como visto nos capítulos anteriores, a principal tarefa do processo de segmentação realizado nesta pesquisa é o de particionar um texto manuscrito em dois tipos de entidade de alto nível: *graphemes* e *ligações*. Os *graphemes* são entidades com grande probabilidade de pertencerem a uma letra da palavra analisada. As *ligações* são entidades com grande probabilidade de pertencerem ao traço da escrita que une duas letras e, neste caso, são entidades que não fazem parte das letras.

A detecção dos pontos de mínimo da borda superior do contorno externo e a detecção das ligações existentes entre as letras de uma palavra são os processos mais importantes para o algoritmo de segmentação. É através dos pontos de mínimo e das ligações que o algoritmo de segmentação se baseia para realizar as quebra em uma palavra.

Para que fosse possível a detecção das ligações da palavra e para que o algoritmo de segmentação se adaptasse ao tipo de escrita da palavra de entrada do sistema, uma característica global do traço da escrita foi usada como parâmetro para todo o processo de segmentação. Esta característica é a espessura média do traço da escrita, discutida na seção 5.4.5. Através desta informação o algoritmo de segmentação pôde tomar decisões compatíveis ao tipo de escrita que se está analisando. Esta informação é usada desde os processos iniciais, como a correção da inclinação, até os processos que realizam a segmentação da palavra.

Como visto na seção 6.4, a principal vantagem deste modelo de segmentação é a de deixar a decisão final da segmentação para o algoritmo de reconhecimento. Isto é possível porque os principais objetivos deste modelo são:

- Particionar uma palavra em entidades que sejam no máximo uma letra inteira, e
- Detectar e quebrar as ligações da palavra para que estas entidades se tornem partes do traço da escrita que possam ser reagrupadas pelo algoritmo de reconhecimento.

Deste modo, um algoritmo de reconhecimento poderá reordenar as entidades geradas pelo processo de segmentação de maneira a obter o melhor encaixe entre estas entidades.

Este modelo de segmentação poderá causar erros na quebra da palavra nos casos discutidos na seção 6.3.2 como:

- Sobreposições entre as letras da palavra,
- A não detecção ou filtragem de pontos de mínimo e de ligações em pontos de quebra, e
- A não detecção de ligações que pertencem a traços muito inclinados da escrita.

Estes problemas poderiam ser tratados através das seguintes melhorias:

- Estimar o tamanho médio das entidades geradas no processo de segmentação é realizar uma análise mais profunda nas entidades que forem maior que um limiar baseado neste tamanho médio (ver “under-segmentation” na seção 4.2);
- No processo de filtragem das ligações pela inclinação do traço da escrita, discutido na seção 5.5.5, tornar a porção excluída da ligação filtrada por inclinação uma entidade individual.

Alguns resultados foram obtidos através de um processo tradicional de segmentação, discutido na seção 6.4. A segmentação de um mesmo conjunto de palavras foi realizada pelo processo tradicional e pelo processo proposto neste trabalho. Em uma análise realizada sob os dois conjuntos de resultados conseguimos demonstrar que a detecção dos traços da palavra com espessura menor ou igual a um determinado limiar, neste trabalho chamado de ligações, podem ser informações importantes para o processo de segmentação da palavra.

Através dos resultados obtidos pelo processo de validação do algoritmo, discutidos no capítulo “Análise dos Resultados”, podemos concluir que é muito difícil automatizar processos que envolvam palavras manuscritas. Mesmo utilizando a detecção das ligações como tentativa para diminuir a variabilidade

do texto manuscrito esta variabilidade foi o grande fator que impediu esta pesquisa de prever regras ótimas para a realização de um processo que seja automatizado e que tenha resultados ainda melhores.

Bibliografia

- [Balestri88] BALESTRI, M. e MASERA, L.: **"A System for Isolating Characters in Cursive Scripts"**. Signal Processing IV: Theories and Applications. Lacoume et al. Eds, pp. 845-846. Elsevier Science Publishers. B. V. North-Holland, 1988.
- [Bosinovic89] BOSINOVIC, R. M. e SRIHARI, S. N.: **"Off-line Cursive Script Word Recognition"**. IEEE Trans. PAMI, 11, pp. 68-83, 1989.
- [Breuel94] BREUEL, Thomas M.: **"A System for the Off-Line Recognition of Handwritten Text"**. IEEE, 1994.
- [Casey92] CASEY, R. G. e HORNE, J. Van: **"Segmentation of Touching Characters in Postal Addresses"**. U. S. Postal Service 5th Adv. Technol. Conf. Vol. 3, pp. 743-754, Washington DC, November of 1992.
- [Casey96] CASEY, R. G. e LECOLINET, E.: **"A Survey of Methods and Strategies in Character Segmentation"**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, pp. 690-706, July of 1996.
- [Chen92] CHEN, M. Y., KUNDU, A., ZHOU, J. e SRIHARI, S. N.: **"Off-Line Handwritten Word Recognition Using Hidden Markov Model"**. U. S. Postal Service 5th Adv. Technol. Conf., pp. 563-577, Washington DC, November of 1992.
- [Facon98] FACON, Jacques: **"Processamento e Análise de Imagens"**. PUC-PR, PPGIA, Fevereiro de 1998.
- [Fenrich91] FENRICH, Richard: **"Segmentation of Automatically Located Handwritten Words"**. Proc. Int. Workshop Handwritten Recognition, 33-44. Bonas, France, 1991.
- [Forney73] FORNEY Jr, G. D.: **"The Viterbi Algorithm"**. IEEE Proc. 61, pp. 268-278, March de 1973.
- [Freeman78] FREEMAN, H.: **"Shape Description Via de Use of Critical Points"**. Pattern Recognition, Vol. 10, pp. 159-166, 1978.
- [Gader92] GADER, P., MOHAMED, M. e CHIANG, J. -H.: **"Segmentation-Based Handwritten Word Recognition"**. U. S. Postal Service 5th Adv. Technol. Conf. Vol. 3, pp. 215-236, Washington DC, November of 1992.
- [Guillevic94] GUILLEVIC, D. e SUEN, C. Y.: **"Cursive Script Recognition: A Sentence Level Recognition Scheme"**. Proc. Of 4th IWFHR, pp. 216-223, 1994.
- [Holt92] HOLT, M., BEGLOU, M. e DATTA, S.: **"Slant-Independent Letter Segmentation for Off-Line Cursive Script Recognition"**. From Pixels to Features III, S. Impedovo and J. C. Simon, eds., p. 41, El-sevier, 1992.
- [Kim96] KIM, Gyeonghwan e GOVINDARAJU, Venu: **"Efficient Chain Code Based Image Manipulation for Handwritten Word Recognition"**. In Proc. of the SPIE symposium on electronic imaging science and technology (*Document Recognition III*), San Jose, CA, volume 2660, pp. 262-272, February of 1996.
- [Kim97] KIM, Gyeonghwan e GOVINDARAJU, Venu: **"A Lexicon Driven Approach to Handwritten Word Recognition for Real-Time Applications"**. IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 19, No. 4, April 1997.

- [Kim98] KIM, Gyeonghwan e GOVINDARAJU, Venu: "**Handwritten Phrase Recognition as Applied to Street Name Images**". J. Pattern Recognition, vol. 31(1), pp 41-51, 1998.
- [Kimura93] KIMURA, F., SHRIDHAR, M. e NARASIMHAMURTHI, N.: "**Lexicon Directed Segmentation-Recognition Procedure for Unconstrained Handwritten Words**". Proc. Third Int. Workshop Frontiers Handwriting Recognition, Buffalo, 1993.
- [Lecolinet91] LECOLINET, Eric e CRETTEZ, Jean-Pierre: "**A Grapheme-Based Segmentation Technique for Cursive Script Recognition**". Proc. Int'l Conf. Document Analysis and Recognition, Saint Malo, France, p. 740, Sept. 1991.
- [Lee93] LEE, D. e SRIHARI, S. N.: "**Handprinted Digit Recognition: A Comparison of Algorithms**". Proc. of 3rd IWFHR, pp. 153-162, 1993.
- [Lu95] LU, Yi: "**Machine-Printed Character Segmentation**". J. Pattern Recognition, 28(1), pp 67-80, 1995.
- [Lu96] LU, Yi e SHRIDHAR, M.: "**Character Segmentation in Handwritten Words – An Overview**". Pattern Recognition, vol. 29, N° 1, pp. 77-96, 1996
- [Nohl92] NOHL, C. R., BURGESS, C. J. C. e BEM, J. I.: "**Character-Based Handwritten Address Word Recognition with Lexicon**". U. S. Postal Service 5th Adv. Technol. Conf. Vol 3, pp. 167-182, Washington DC, November of 1992.
- [Sayre73] SAYRE, K. M.: "**Machine Recognition of Handwritten Words: A Project Report**". Pattern Recognition, vol. 5, pp. 213-228, 1973.
- [Shridhar97] SHRIDHAR, M. e KIMURA, F.: "**Segmentation-Based Cursive Handwriting Recognition**". Handbook of Character Recognition and Document Image Analysis. Eds. H. Bunke and P. S. P. Wang, pp. 123-156, 1997.
- [Simon92] SIMON, J. -C.: "**Off-Line Cursive Word Recognition**". Proc. IEEE, vol. 80, no. 7, pp. 1150-1161, 1992.
- [Srihari] SRIHARI, Sagur N. e LAM, Stephen W.: "**Character Recognition**". Center of Excellence for Document Analysis and Recognition. State University of New York at Buffalo. <http://mirach.cs.buffalo.edu/Publications/ThecReps/OCR/ocr.htm>.
- [Stratry93] STRATHY, M. H.: "**A Method for Segmentation of Touching Handwritten Numerals**". MSc Dissertation, Concórdia University, 1993.