

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações

**Reconhecimento de Palavras Manuscritas Baseado em HMM e
no Emprego de Características Topológicas e Geométricas**

Autor

Natanael Rodrigues Gomes

Orientador

Prof. Dr. Lee Luan Ling (FEEC/UNICAMP/CAMPINAS)

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.

Banca Examinadora

Prof. Dr. Max Gerken (USP/SÃO PAULO)

Prof.^a Dr.^a Edna Lúcia Flôres (FEEL/UFU/UBERLÂNDIA)

Prof. Dr. Fábio Violaro (FEEC/UNICAMP/CAMPINAS)

Prof. Dr. João Marcos T. Romano (FEEC/UNICAMP/CAMPINAS)

Prof. Dr. Yuzo Iano (FEEC/UNICAMP/CAMPINAS)

CAMPINAS

2000

Este exemplar corresponde a redação final da tese defendida por <u>Natanael Rodrigues</u> <u>Gomes</u> e aprovada pela Comissão Julgada em <u>14 / 08 / 2000</u> <u>Lee Luan Ling</u> Orientador

200618854

UNIDADE	BC
N.º CHAMADA:	T/Unicomp
	G585r
V.	Ex.
TOMBO BC/	43190
PROC.	16-278/00
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREC.º	R\$ 11,00
DATA	23/12/00
N.º CPD	

CM-00154348-0

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

G585r

Gomes, Natanael Rodrigues

Reconhecimento de palavras manuscritas baseado em HMM e no emprego de características topológicas e geométricas / Natanael Rodrigues Gomes.--Campinas, SP: [s.n.], 2000.

Orientador: Lee Luan Ling.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Reconhecimento de padrões. 2. Escrita. 3. Markov, Processos de. 4. Conjuntos difusos. 5. Processamento de imagens. I. Lee, Luan Ling. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

RESUMO

O desenvolvimento de métodos para reconhecer palavras manuscritas de vários estilos de escrita, em razão das variações envolvidas, não é trivial. Nesta tese é desenvolvido um método para extração de características topológicas e geométricas de palavras manuscritas baseado na teoria de conjuntos *fuzzy*. Durante o processo de reconhecimento a imagem de uma palavra manuscrita é representada por uma seqüência ordenada de segmentos de linha. Esses segmentos são extraídos de cada fração resultante do processo de segmentação da palavra em caracteres. Para cada segmento de linha é atribuído, através de conjuntos *fuzzy*, valores representando sua similaridade a padrões específicos de linhas retas e curvas. O processo de segmentação desenvolvido é explícito e emprega morfologia matemática para determinar possíveis pontos de segmentação da palavra. A classificação de palavras baseia-se nos modelos ocultos de Markov. Também é utilizado no processo de classificação uma generalização dos modelos ocultos de Markov empregando medidas e integrais *fuzzy*. Para treinamento e testes do sistema de reconhecimento utiliza-se uma base de dados composta por palavras empregadas na escrita da quantia numérica em cheques brasileiros.

Palavras-chave: Reconhecimento de padrões, palavras manuscritas, modelos ocultos de Markov, conjuntos e integrais *fuzzy*.

ABSTRACT

Due to the large variability of human handwriting, the development of unconstrained methods for recognizing handwritten words is not trivial. This thesis presents a fuzzy set based method for extracting topologic and geometric characteristics from handwritten words. During the recognition process a handwritten word image is represented by an ordered sequence of line segments. The line segments are extracted from each word fraction produced by the segmentation algorithm. Fuzzy set theory is applied to attribute to each line segment a set of values representing degrees of similarity to straight line and curve line patterns. The developed word segmentation algorithm is explicit and employs mathematical morphology to assign segmentation points on a word image. The classification of the handwritten words is a Hidden Markov Model based process. A generalization of the Hidden Markov Models employing fuzzy measures and fuzzy integrals is also used as a handwritten word classifier. The recognition system is trained and tested with a database compound by Brazilian bank check words.

Keywords: Pattern recognition, handwritten words, Hidden Markov Models, fuzzy sets and fuzzy integrals.

À minha família...

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

I refuse to accept the idea that man is mere flotsam and jetsam in the river of life unable to influence the unfolding events, which surround him. I refuse to accept the view that mankind is so tragically bound to the starless midnight of racism and war that the bright daybreak of peace and brotherhood can never become a reality.

Martin Luther King, Nobel Prize Acceptance Speech.

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

AGRADECIMENTOS

Aos colegas do Departamento de Comunicações da Faculdade de Engenharia Elétrica e de Computação da UNICAMP, especialmente as pessoas que fazem parte do Laboratório de Reconhecimento de Padrões e Redes de Comunicação.

Aos colegas Evandro, José Raimundo e Miguel pela ótima convivência e contribuições para que este trabalho fosse concluído.

Ao meu orientador, Professor Lee Luan Ling, por sua ajuda e atenção durante a realização deste trabalho.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) por financiar totalmente o desenvolvimento deste trabalho.

SUMÁRIO

<i>Resumo</i>	<i>iii</i>
<i>Abstract</i>	<i>iii</i>
<i>Sumário</i>	<i>vii</i>
<i>Índice de Figuras</i>	<i>xii</i>
<i>Índice de Tabelas</i>	<i>xvii</i>
<i>CAPÍTULO I - Introdução</i>	<i>1</i>
1.1 <i>Propostas Desta Tese</i>	<i>2</i>
1.2 <i>Contribuições Desta Tese</i>	<i>3</i>
1.3 <i>Organização da Tese</i>	<i>4</i>
<i>Capítulo II - Uma Visão Genérica do Reconhecimento de Palavras Manuscritas</i>	<i>5</i>
2.1 <i>Influência do Número de Escreventes e Tamanho do Vocabulário</i>	<i>6</i>
2.2 <i>Reconhecendo a Palavra Inteira ou Dividindo-a em Caracteres</i>	<i>7</i>
2.3 <i>Sistemas Para Reconhecer Palavras Manuscritas</i>	<i>9</i>
2.4 <i>Estratégias de Classificação</i>	<i>10</i>
2.4.1 <i>Estratégias Analíticas</i>	<i>10</i>
2.4.2. <i>Métodos Globais ou Baseados na Forma da Palavra</i>	<i>11</i>

2.5 O Sistema de Reconhecimento Desta Tese	12
Capítulo III - Modelos Ocultos de Markov.....	15
3.1 Definição.....	16
3.2 Problemas Básicos do Modelamento Via HMMs.....	17
3.3. Soluções Para os Três Problemas Básicos.....	18
3.3.1 Problema de Avaliação	18
3.3.2 Problema de Determinação da Seqüência de Estados Ótima.....	20
3.3.3 Problema de Treinamento	22
3.4 Alguns Aspectos da Implementação de HMMs.....	25
3.4.1 Escalonamento.....	25
3.4.2 Múltiplas Seqüências de Observação.....	26
3.4.3 Inicialização de Parâmetros.....	28
3.5 Síntese Sobre Formas de Utilização dos HMMs em Reconhecimento de Palavras Manuscritas	28
Capítulo IV - Medidas Fuzzy e Integral Fuzzy.....	31
4.1 Medidas.....	32
4.2 Medidas Fuzzy	33
4.3 Uma Interpretação da Não-Aditividade de Uma Medida Fuzzy	34
4.4 Medidas λ -Fuzzy.....	36
4.5 Integral Fuzzy.....	37
4.6 Medidas Fuzzy Condicionais	39
4.7 Integral de Choquet	39
4.8 Comentários Referentes ao Capítulo.....	40

Capítulo V - Generalização dos Modelos Ocultos de Markov.....	41
5.1 Apresentação da Nomenclatura Utilizada.....	42
5.2 Uma Interpretação Para as Medidas Fuzzy de Estado Inicial e de Transição.....	43
5.3 Formulação Fuzzy Para as Variáveis Forward.....	44
5.4 Formulação Fuzzy Para as Variáveis Backward.....	45
5.5 Generalização dos HMMs Utilizando a Integral de Choquet.....	45
5.6 Formulação Fuzzy do Processo de Reestimação - Modificação do Método de Baum-Welch.....	47
5.7 Algoritmo de Viterbi Modificado.....	49
5.8 Alguns Aspectos da Implementação Para HMMs Fuzzy.....	49
5.8 Comentários.....	51
Capítulo VI - Pré-Processamento de Palavras Manuscritas.....	52
6.1 Processo de Binarização.....	53
6.2 Remoção de Ruídos.....	55
6.3 Correção de Rotação de Palavras Manuscritas.....	57
6.4 Correção de Inclinação de Palavras Manuscritas.....	62
6.5 Comentários Finais do Capítulo.....	66
Capítulo VII - Segmentação de Palavras Manuscritas.....	67
7.1 Algumas Estratégias de Segmentação.....	68
7.2 Descrição do Método de Segmentação.....	70
7.2.1 Determinação das Prováveis Ligações de Uma Palavra.....	72
7.2.2 Validação das Ligações Entre Caracteres.....	75
7.2.3 Segmentação da Palavra nos Pontos Potenciais de Segmentação.....	78
7.3 Processamento Final da Palavra Segmentada.....	79

7.4 Comentários Sobre o Processo de Segmentação.....	80
Capítulo VIII - Extração de Características	82
8.1 Noções a Respeito de Conjuntos Fuzzy	83
8.2 Características Para Uma Palavra Manuscrita.....	85
8.3 Ramos e Pontos Singulares de Um Segmento.....	87
8.4 Posições Relativas de Um Ramo.....	89
8.5 Determinação dos Valores de Pertinência de Um Ramo	92
8.5.1 Similaridade a Uma Linha Reta e a Uma Linha Curva	92
8.5.2 Avaliação da Orientação de Um Ramo.....	95
8.6 Construção do Vetor de Característica de Um Ramo	101
Capítulo IX - Classificador de Palavras Manuscritas e Formação da Base de Dados...110	
9.1 Modelos Para Palavras Manuscritas	111
9.2 Seqüência de Observações Para Uma Palavra Manuscrita.....	115
9.3 Parâmetros Iniciais Para HMMs e FHMMs	115
9.4 Estrutura do Classificador de Palavras	116
9.5 Pós-Processamento.....	118
9.6 Formação da Base de Dados.....	120
Capítulo X - Treinamento do Sistema e Resultados Experimentais	124
10.1 Estratégia Para Treinamento do Sistema.....	125
10.2 Amostras de Caracteres Para Treinamento	128
10.3 Resultados Experimentais.....	131
10.4 Análise dos Resultados	137
10.5 Análise do Trabalho Desenvolvido.....	140
10.6 Direções Futuras Para o Trabalho Desenvolvido.....	142

<i>Referências Bibliográficas</i>	144
<i>Anexo A - Método dos Mínimos Quadrados</i>	150
<i>Anexo B - Rotacionamento de Imagens de Palavras Manuscritas</i>	152
<i>Anexo C - Histogramas Considerando Vários Ângulos de Inclinação de Uma Palavra Manuscrita</i>	154
<i>Anexo D - Diagrama Horizontal de Grupos de Pixels</i>	156
<i>Anexo E - Morfologia Matemática</i>	157
<i>E.1 Algumas Operações Morfológicas Básicas Para Imagens Binárias</i>	158
<i>E.2 Dilatação Condicionada</i>	159
<i>E.3 Operação de Espessamento de Uma Imagem</i>	160
<i>E.4 Dilatação e Erosão em Níveis de Cinza</i>	162
<i>E.5 Distância Geodésica</i>	164
<i>Anexo F - Procedimento de Afinamento</i>	166
<i>F.1 Afinamento e Esqueleto de Um Objeto em Uma Imagem</i>	166
<i>F.2 Transformação de Eixo Médio</i>	167
<i>Anexo G - Amostras de Palavras Manuscritas</i>	169

ÍNDICE DE FIGURAS

<i>Figura 2.1 - Processo de reconhecimento de palavras manuscritas.</i>	12
<i>Figura 6.1 - Imagem de cheque apresentando fundo com desenhos em tons de cinza.</i>	55
<i>Figura 6.2 - Imagem de cheque após processo de binarização.</i>	55
<i>Figura 6.3 - Máscaras para remoção de ruídos.</i>	56
<i>Figura 6.4 - Máscaras para remoção de ruídos em imagens após correção de rotação e inclinação.</i>	57
<i>Figura 6.5 - Contorno ruidoso.</i>	57
<i>Figura 6.6 - Palavra manuscrita rotacionada.</i>	58
<i>Figura 6.7 - Histograma horizontal de blocos e pontos de mínimos locais em uma imagem da palavra "onze".</i>	60
<i>Figura 6.8 - Determinação da linha de base em uma imagem da palavra "dez".</i>	61
<i>Figura 6.9 - Imagem da palavra "dez" após correção de rotação.</i>	62
<i>Figura 6.10 - Palavra manuscrita inclinada para a direita.</i>	63
<i>Figura 6.11 - Eliminação de grupos de pixels de uma palavra "dez".</i>	65
<i>Figura 6.12 - Palavra "dez" com a inclinação corrigida.</i>	66

<i>Figura 7.1 - Possível ambigüidade entre a letra "a" e a seqüência de letras conectadas "c" e "i".</i>	68
<i>Figura 7.2 - Pontos potenciais de segmentação da palavra "dígitos".</i>	69
<i>Figura 7.3 - Divisão de uma palavra em retângulos.</i>	70
<i>Figura 7.4 - Ligações da palavra "dígitos" assinaladas em cinza.</i>	71
<i>Figura 7.5 - Operação de abertura binária.</i>	72
<i>Figura 7.6 - Resultado do preenchimento de loops existentes na palavra "sete".</i>	73
<i>Figura 7.7 - Imagem de uma Letra "F" com colunas numeradas na sua parte superior.</i>	74
<i>Figura 7.8 - Prováveis ligações da palavra "sete" assinaladas em negrito.</i>	75
<i>Figura 7.9 - Ligações válidas de uma palavra "sete".</i>	76
<i>Figura 7.10 - Imagem de uma palavra "doze".</i>	76
<i>Figura 7.11 - Avaliação das ligações de uma palavra "doze".</i>	77
<i>Figura 7.12 - Imagem de uma palavra "sete" após serem eliminadas as ligações entre caracteres.</i>	78
<i>Figura 7.13 - Obtendo a palavra segmentada.</i>	79
<i>Figura 7.14 - Possível região de segmentação determinada por uma Concavidade Superior (CS) e uma Concavidade Inferior (CI).</i>	79
<i>Figura 7.15 - Seqüência de caracteres "s" e "e".</i>	80
<i>Figura 7.16 - Imagem de uma palavra "oito" em que o algoritmo de segmentação não é capaz de identificar uma ligação entre os caracteres "t" e "o".</i>	80
<i>Figura 7.17 - Amostra de uma palavra "um" onde ocorre a segmentação dos caracteres "u" e "m".</i>	81
<i>Figura 7.18 - Segmentação da palavra "for" do vocabulário inglês.</i>	81
<i>Figura 8.1 - Segmentos de linha compondo um caractere manuscrito "A".</i>	83

<i>Figura 8.2 - Segmentos de linha para caracterizar caracteres.</i>	86
<i>Figura 8.3 – Segmentação da palavra manuscrita "cento" apresentando uma fração de um caractere "n" e os caracteres "t" e "o" não separados.</i>	87
<i>Figura 8.4 - Pontos singulares em uma letra manuscrita "O".</i>	88
<i>Figura 8.5 - Posições relativas para ramos de uma letra manuscrita "d".</i>	89
<i>Figura 8.6 - Duas amostras de uma letra manuscrita "t".</i>	90
<i>Figura 8.7 - Função de pertinência para avaliar a similaridade entre ramos da fração de uma palavra e uma linha reta.</i>	94
<i>Figura 8.8 - Funções de pertinência para o conjunto {H, P, V, N}.</i>	96
<i>Figura 8.9 - Funções de pertinência para determinar a orientação vertical (V) ou horizontal (H) de uma curva.</i>	97
<i>Figura 8.10 - Classificação das curvas tipo C, D, A, U, S e Z.</i>	100
<i>Figura 8.11 - Fração de uma palavra segmentada cujos ramos serão classificados.</i>	101
<i>Figura 8.12 - Uma representação das pertinências de um ramo às posições relativas à fração considerada de uma palavra.</i>	104
<i>Figura 8.13 - Diagrama de blocos resumindo o processo de extração de características.</i>	109
<i>Figura 9.1 - Representação da letra "a" por um conjunto de ramos.</i>	112
<i>Figura 9.2 - HMM para modelamento da letra "a".</i>	112
<i>Figura 9.3 - Modelamento das letras "u" e "m".</i>	113
<i>Figura 9.4- Modelo para a palavra "um".</i>	113
<i>Figura 9.5 - Modelo para uma palavra manuscrita.</i>	113
<i>Figura 9.6 - Diagrama representativo da classificação de uma palavra manuscrita.</i>	117

UNICAMP

BIBLIOTECA CENTRAL
CAMPUS CÍRCULO DAS ARTES

<i>Figura 9.7 - Representação dos segmentos ascendentes e descendentes da palavra "dígitos".</i>	118
<i>Figura 9.8 - Operações realizadas para determinar se uma palavra é escrita somente com letras maiúsculas.</i>	119
<i>Figura 9.9 - Quantidade total de amostras por classe de palavras da base de dados.</i>	123
<i>Figura 10.1 - Representação gráfica da quantidade média de amostras por classe de palavras em cada um dos 10 grupos para treinamento e teste do sistema de reconhecimento.</i>	127
<i>Figura 10.2 - Estratégia de treinamento e teste do sistema de reconhecimento.</i>	128
<i>Figura 10.3 - Total de letras minúsculas para treinamento e testes do sistema de reconhecimento.</i>	130
<i>Figura 10.4 - Total de letras maiúsculas para treinamento e testes do sistema de reconhecimento.</i>	131
<i>Figura B.1 - Eixos de coordenadas rotacionados.</i>	152
<i>Figura C.1 - Sistema de coordenadas adotado para uma imagem.</i>	155
<i>Figura D.1 - Histograma horizontal de grupo de pixels pretos de uma imagem da palavra "treze".</i>	156
<i>Figura E.1 - Operações morfológicas binárias de erosão, dilatação e abertura.</i>	159
<i>Figura E.2 - Exemplo de dilatação condicionada.</i>	159
<i>Figura E.3 - Par de elementos estruturantes.</i>	160
<i>Figura E.4 - Representação de um par de elementos estruturantes para espessamento. .</i>	161
<i>Figura E.5 - Exemplo de espessamento de uma imagem.</i>	162
<i>Figura E.6 - Família de elementos estruturantes utilizada na operação de espessamento.</i>	162

<i>Figura E.7 - Exemplo de dilatação em níveis de cinza de uma função $f(x)$ por um elemento estruturante $g(x)$.....</i>	<i>163</i>
<i>Figura E.8 - Distância geodésica.</i>	<i>164</i>
<i>Figura F.1 - Esqueleto da imagem X representado por linhas tracejadas.....</i>	<i>167</i>
<i>Figura F.2 - Esqueleto da palavra quarenta.</i>	<i>168</i>

ÍNDICE DE TABELAS

<i>Tabela 8.1 - Valores de pertinência relacionados a segmentos de linhas retas e linhas curvas (Seção 8.51).</i>	<i>102</i>
<i>Tabela 8.2 - Valores de pertinência relativos a orientação dos ramos do caractere.</i>	<i>102</i>
<i>Tabela 8.3 - Valores de pertinência relativo as posições do ramo em um segmento.....</i>	<i>102</i>
<i>Tabela 8.4 - Valores de pertinência para o ramo 1-2 considerando a reta com inclinação positiva (P)</i>	<i>107</i>
<i>Tabela 9.1 - Dados para treinamento e teste do sistema de reconhecimento.</i>	<i>122</i>
<i>Tabela 10.1- Total de palavras em cada um dos grupos em que foi dividida a base de dados.....</i>	<i>125</i>
<i>Tabela 10.2 - Número médio de amostras por classe de palavras em cada um dos grupos em que foi dividida a base de dados.....</i>	<i>126</i>
<i>Tabela 10.3 – Quantidade média de caracteres por grupo de treinamento.....</i>	<i>129</i>
<i>Tabela 10.4 – Total de caracteres empregados no processo de treinamento e teste.</i>	<i>130</i>
<i>Tabela 10.5 - Desempenho do sistema observada cada classe de palavras (Classificação via HMMs).</i>	<i>132</i>

Tabela 10.6 - Desempenho do sistema observado cada grupo de treinamento (Classificação via HMMs).....	133
Tabela 10.7 - Desempenho do sistema considerando cada grupo de treinamento (Classificação via FHMMs)	133
Tabela 10.8 - Desempenho do sistema considerando cada classe de palavras (Classificação via FHMMs)	134
Tabela 10.9 - Desempenho do sistema produzido pela combinação de classificadores observada cada classe de palavras (Classificação via HMMs e FHMMs) .	136
Tabela 10.10 – Resultado de classificação gerado pela combinação de HMMs e FHMMs considerando cada grupo de treinamento	137

CAPÍTULO I

INTRODUÇÃO

O reconhecimento automático de palavras e caracteres manuscritos é uma das principais fontes de erros em máquinas destinadas a leitura óptica de documentos. Esforços têm sido feitos no sentido de desenvolver técnicas capazes de resolver problemas de reconhecimento relacionados a variação de forma dos caracteres manuscritos. Esses esforços se justificam devido às inúmeras aplicações práticas do reconhecimento de manuscritos, como o reconhecimento automático de endereços em cartas e o processamento automático de cheques. No entanto, o reconhecimento de palavras manuscritas permanece um problema a ser resolvido, apesar de nos últimos anos ter se tornado objeto de intensas pesquisas e as soluções encontradas terem possibilitado o desenvolvimento de produtos úteis ao mercado. A imensa variabilidade de estilos e materiais de escrita têm imposto restrições aos sistemas desenvolvidos como, por exemplo, reconhecimento de um número limitado de palavras e reconhecimento da escrita de um número limitado de pessoas. Prover soluções factíveis para eliminação dessas restrições tem sido a motivação dos trabalhos mais recentes em reconhecimento de palavras manuscritas.

1.1 PROPOSTAS DESTA TESE

Esta tese investiga o reconhecimento automático de palavras manuscritas a partir da consideração de que um caractere pode ser visto como uma coleção ordenada de segmentos de linha, tais como, retas, curvas e *loops*. O termo *caractere* nesta tese refere-se as letras compondo uma palavra manuscrita. Tal abordagem se mostra adequada quando é observada a posição e os tipos de segmentos de linha em um caractere, pois torna-se possível uma análise seqüencial dos segmentos e, por conseguinte, o desenvolvimento de uma metodologia de classificação de caracteres. Porém, a utilização de segmentos de linha para classificar um caractere pode produzir resultados não satisfatórios. Em razão do alto grau de variabilidade de forma apresentado pelos caracteres manuscritos, pode não ser possível a avaliação precisa de um segmento de linha como sendo uma reta ou uma curva. A teoria de conjuntos *fuzzy* fornece o formalismo matemático adequado para tratar essa imprecisão e, desse modo, minimizar os erros de classificação.

Neste trabalho os segmentos de linha de um caractere são avaliados segundo conceitos da teoria de conjuntos *fuzzy*. Essa avaliação permite propor um conjunto de vetores de características para representar o caractere. Dessa forma, o processo de reconhecimento é inicialmente baseado na classificação dos caracteres da palavra sendo processada.

Dois métodos de classificação são empregados visando a utilização de alguma informação contextual e o desenvolvimento de um sistema de reconhecimento de desempenho satisfatório. Um método usa os modelos ocultos de Markov como um meio de modelar estocasticamente as ambigüidades de forma de caracteres em palavras manuscritas. O outro método de classificação é uma generalização dos modelos ocultos de Markov utilizando diretamente as características *fuzzy* extraídas de caracteres. Essa generalização permite uma

agregação não linear da informação originada da interdependência entre segmentos de um caractere e entre caracteres de uma palavra.

1.2 CONTRIBUIÇÕES DESTA TESE

Uma contribuição desta tese é, certamente, a descrição de um sistema original e completo para reconhecimento de palavras manuscritas. Além disso, outras contribuições consideradas principais são:

- Proposta de um algoritmo para correção da inclinação média de palavras manuscritas visando reduzir a influência de traços inerentes ao estilo de escrita de cada indivíduo;
- Tratamento do problema da segmentação de palavras manuscritas utilizando uma síntese original de técnicas conhecidas e proposição de um método para determinação dos possíveis pontos de segmentação de uma palavra;
- Utilização da teoria de conjuntos *fuzzy* para avaliar a similaridade entre segmentos de linha formando um caractere e os símbolos de observações empregados em modelos ocultos de Markov generalizados;
- Utilização de características *fuzzy* para gerar as probabilidades de símbolo empregadas no processo de classificação usando modelos ocultos de Markov.

1.3 ORGANIZAÇÃO DA TESE

Após ter sido realizado neste capítulo, uma breve introdução sobre o problema de reconhecimento de palavras manuscritas, descrição dos objetivos e contribuições deste trabalho, esta tese passa a ser organizada da seguinte maneira: o capítulo II apresenta uma visão genérica do reconhecimento de palavras manuscritas e do sistema de reconhecimento proposto. No capítulo III é apresentada uma noção sobre os modelos ocultos de Markov, onde são introduzidos conceitos essenciais ao entendimento do processo de classificação utilizado. O capítulo IV trata de medidas e integrais *fuzzy*, as quais são necessárias à compreensão da generalização dos modelos ocultos de Markov descrita em detalhes no capítulo V. A partir do capítulo VI começam a ser descritas as fases do sistema de reconhecimento desenvolvido, sendo abordado primeiramente as técnicas de pré-processamento aplicadas às imagens de palavras manuscritas. Uma análise da segmentação de palavras manuscritas é realizada no capítulo VII, bem como a apresentação do método de segmentação utilizado. O capítulo VIII mostra as características utilizadas pelo sistema de reconhecimento e a construção dos vetores de características empregados no processo de classificação. O capítulo IX é dedicado aos métodos de classificação utilizados no sistema de reconhecimento desenvolvido. Os resultados obtidos, assim como as conclusões do trabalho, são assuntos do capítulo X.

CAPÍTULO II

UMA VISÃO GENÉRICA DO RECONHECIMENTO DE PALAVRAS MANUSCRITAS

O reconhecimento de palavras manuscritas é realizado com tal facilidade pela maioria dos seres humanos, que as pessoas em geral não têm percepção da complexidade em simular essa habilidade em uma máquina. Uma das maiores dificuldades no desenvolvimento de um sistema para reconhecer palavras manuscritas reside na variabilidade de formas apresentadas pelos caracteres. Essa variabilidade pode ser verificada quando uma palavra é escrita várias vezes por uma pessoa ou quando diferentes indivíduos a escrevem. Conseqüentemente, torna-se difícil encontrar uma descrição capaz de levar em conta todas as possíveis variações de um caractere. Porém, não é apenas a variabilidade de forma dos caracteres o único problema a afetar o reconhecimento de palavras manuscritas. Existem outras questões que contribuem para a complexidade dessa tarefa de reconhecimento. As conexões e a sobreposição de caracteres impedem a segmentação correta de uma palavra, acarretando erros de reconhecimento em sistemas baseados na segmentação de palavras em caracteres. Além disso, a observação isolada de um caractere ou de uma palavra pode resultar em uma ambigüidade

entre classes¹. Portanto, é necessária alguma informação contextual para auxiliar na decisão por uma determinada classe. A informação contextual é estritamente necessária em sistemas destinados ao reconhecimento de palavras em grandes vocabulários e, também, nos casos em que não são oferecidas restrições (tamanho do vocabulário e número de escreventes) ao processo de reconhecimento. O uso de dicionários para verificar a existência de uma palavra é o meio mais comum para prover informação contextual neste tipo de sistema.

Este capítulo é destinado à descrição sintética de alguns problemas, métodos e estratégias em reconhecimento de palavras manuscritas. O objetivo dessa descrição é prover uma visão genérica do problema e da estrutura dos sistemas de reconhecimento de palavras manuscritas.

2.1 INFLUÊNCIA DO NÚMERO DE ESCRIVENTES E TAMANHO DO VOCABULÁRIO

Devido a diversidade de estilos de escrita há maiores dificuldades em desenvolver um sistema para reconhecer a escrita de muitas pessoas do que a escrita de apenas uma pessoa. De uma maneira similar, existe menor dificuldade em reconhecer palavras de um vocabulário reduzido do que palavras pertencentes a um vocabulário mais extenso. A probabilidade de erros de classificação devido às semelhanças entre palavras aumenta à medida que aumenta a quantidade de palavras do vocabulário. Ambas as questões, número de escreventes e tamanho do vocabulário, têm sido analisadas em trabalhos da literatura. Em [LB94] é proposto a determinação de um ponto de equilíbrio entre número de escreventes e tamanho do

¹ O termo *classe* nesta tese refere-se a cada uma das diferentes palavras de um vocabulário utilizado para classificar uma palavra manuscrita sendo reconhecida.

vocabulário, levando-se em conta o tipo de aplicação. Por exemplo, alguns sistemas poderiam ser desenvolvidos considerando a escrita dos usuários mais freqüentes, em conseqüência, poderiam ser reconhecidas palavras de um vocabulário maior. Por outro lado, em sistemas destinados ao reconhecimento de escritas de um grande número de pessoas, o tamanho do vocabulário deveria ser reduzido, como no processamento automático de cheques, em que o vocabulário é de aproximadamente quarenta palavras.

A tarefa de reconhecimento torna-se mais complexa quando não há restrições, ou seja, é exigido reconhecer palavras em um vocabulário extenso e escritas por inúmeras pessoas (vários estilos de escrita), como no reconhecimento de endereços postais. Nesse caso, tenta-se auxiliar o processo de reconhecimento utilizando informações relacionadas à tarefa sendo desenvolvida. No reconhecimento de endereços em cartas, por exemplo, pode-se validar o resultado de classificação de uma palavra analisando o resultado do reconhecimento em outras partes do endereço como: código de endereçamento postal (CEP), nome do estado, nome da cidade, nome da rua, etc.. A utilização dessas informações pode proporcionar uma solução viável para tratar o problema de reconhecer palavras em um vocabulário extenso e escritas por inúmeras pessoas.

2.2 RECONHECENDO A PALAVRA INTEIRA OU DIVIDINDO-A EM CARACTERES

Pesquisadores têm discutido a necessidade de segmentar uma palavra em caracteres para efetuar sua classificação. Pesquisas vêm sendo feitas no sentido de mostrar qual o melhor procedimento a ser adotado. Alguns trabalhos têm abordado a psicologia da leitura para entender como ocorre a transferência de informação por meio da escrita para o ser humano

[Gui95], [Sen94]. O entendimento da informação que uma pessoa utiliza para reconhecer uma palavra pode dar uma noção das características mais adequadas ao desenvolvimento de um sistema de reconhecimento. Uma análise da psicologia da leitura tem demonstrado que ótimos leitores não identificam individualmente cada caractere em uma palavra, mas se concentram em algumas características na forma da palavra e o contexto em que ela está inserida. Apesar de não se saber com certeza quais são essas características, testes demonstram que pessoas reconheceriam letras observando características correspondendo a *loops*, curvas e segmentos de linha reta [Sen94]. Tais características, juntamente com uma análise sintática e semântica de acordo com o texto sendo lido, permitiriam o reconhecimento de uma palavra sem a necessidade de segmentá-la em caracteres. Essa metodologia vem sendo aplicada em sistemas utilizando modelos ocultos de Markov para classificar palavras manuscritas em vocabulários pequenos, onde é possível construir um modelo para cada classe de palavra [GS97]. Além disso, a não segmentação de uma palavra evitaria tratar o problema de segmentos representando frações de um caractere ou caracteres ainda conectados, causado por uma segmentação imperfeita. No entanto, para vocabulários maiores torna-se impossível construir um modelo para cada palavra, em razão da variedade de estilos de escrita. Nesse caso, o reconhecimento individual de letras pode representar a melhor solução [YSS99].

De qualquer modo, independente da estratégia de classificação, se *loops*, curvas e retas representam informações utilizadas por um leitor no reconhecimento de uma palavra, talvez elas possam representar boas características para desenvolvimento de um sistema computacional de reconhecimento. Uma outra constatação advinda da psicologia da leitura é de que um ótimo leitor utiliza, além de algumas características selecionadas, o contexto para auxiliar no reconhecimento de uma palavra. Isto sugere que um sistema de reconhecimento deverá obter melhor desempenho se alguma informação contextual for utilizada.

2.3 SISTEMAS PARA RECONHECER PALAVRAS MANUSCRITAS

O processo computacional de reconhecimento de palavras visa recuperar da imagem de uma palavra seu significado lingüístico, transformando-a em símbolos que possam ser manipulados via computador. De modo genérico, um sistema de reconhecimento de palavras é considerado ter duas entradas: a imagem digitalizada de uma palavra de classe desconhecida e uma lista de palavras de classes conhecidas. O objetivo final é ordenar as palavras da lista segundo algum critério relacionando a imagem de entrada do sistema, permitindo desse modo, a identificação da palavra de classe desconhecida.

O passo inicial no processo de reconhecimento em questão é a aquisição da imagem digitalizada da palavra. Os dispositivos comumente utilizados para efetuar a aquisição da imagem são os *scanners* e câmaras *CCDs* (*Charge-Coupled Devices*). Durante a operação de digitalização devem ser observadas a resolução espacial e a resolução em níveis de cinza da imagem [GW92]. Esses fatores afetam a extração de características e, em conseqüência, o desempenho do processo de reconhecimento. Após a imagem de uma palavra ter sido adquirida, são realizados pré-processamentos na imagem para eliminar ruídos e melhorá-la para extração de características. Alguns dos pré-processamentos mais empregados em imagens de palavras manuscritas são tratados no capítulo VI desta tese. Ao ser concluído o pré-processamento da imagem, é iniciada a extração de características da palavra de classe desconhecida. Pesquisas em reconhecimento de caracteres têm procurado determinar características que sejam discriminantes. Uma revisão sobre esse assunto pode ser encontrada em [SBM80]. Nesta tese as características utilizadas descrevem a geometria e/ou topologia da imagem de uma palavra manuscrita, sendo por isso classificadas como características topológicas e geométricas. Podem ser citados como exemplos dessas características, tipos de segmentos de linhas curvas, pontos de interseção de segmentos de linha e *loops* [WW94],

[GL97a], [Sen94], [GL99]. As características obtidas para uma palavra de classe desconhecida são analisadas por um classificador, sendo atribuída uma classe a essa palavra. O método adotado para classificação constitui um ponto essencial no reconhecimento de manuscritos. A seguir é dada uma noção sobre algumas estratégias de classificação.

2.4 ESTRATÉGIAS DE CLASSIFICAÇÃO

Em geral, dois tipos principais de estratégias [CL96] vêm sendo aplicadas ao reconhecimento de palavras manuscritas, nominalmente: (1) estratégias analíticas ou baseadas em segmentação e (2) estratégias globais ou baseadas na forma da palavra manuscrita.

2.4.1 ESTRATÉGIAS ANALÍTICAS

Nas estratégias analíticas a classificação de uma palavra é efetuada segmentando-a em caracteres. Essa classificação pode ser feita empregando uma segmentação explícita ou uma segmentação implícita da palavra, como explicado abaixo.

- **Classificação Empregando Segmentação Explícita:** A classificação com segmentação explícita pode ser realizada em três passos distintos: (1) Segmentação da palavra em caracteres; (2) Reconhecimento individual de cada caractere segmentado; (3) Verificação e correção da palavra formada pelos caracteres reconhecidos. A vantagem desse tipo de abordagem é o reconhecimento de caracteres isolados através de técnicas bem conhecidas em reconhecimento de padrões. Dentre essas técnicas estão aquelas empregando o classificador de Bayes, a regra do vizinho mais próximo e redes neurais artificiais [DH73],

[Bis95]. A principal desvantagem dessa estratégia é a ocorrência de erros de classificação gerados por segmentação errônea da palavra. A verificação da palavra formada pelos caracteres reconhecidos é efetuada através de uma comparação com as palavras existentes no vocabulário em uso.

- **Classificação Empregando Segmentação Implícita:** Nesse caso, a palavra também é segmentada, porém a segmentação baseia-se no reconhecimento do caractere. Isto significa que um segmento é considerado somente quando for reconhecido como caractere. O principal interesse nesse tipo de classificação está na utilização de um método de segmentação simples e na ausência de erros devido a segmentações incorretas. Em geral, uma janela móvel e de tamanho variável é usada para prover segmentações da palavra, as quais são confirmadas pelo reconhecedor de caracteres. Alguma informação contextual pode ser obtida durante o processo de reconhecimento e segmentação através de métodos estatísticos. Por exemplo, a utilização dos modelos ocultos de Markov permitem a representação dessa informação contextual através de uma probabilidade de transição entre letras [Chen95], [Sen94].

2.4.2. MÉTODOS GLOBAIS OU BASEADOS NA FORMA DA PALAVRA

Nestes métodos são extraídas características da palavra sem que se realize um processo de segmentação. Essas características permitem associar uma classe à palavra sendo reconhecida de acordo com as palavras do vocabulário utilizado. A palavra é tratada como uma entidade única, não sendo considerada como uma conjunção de caracteres. As características são, em geral, extraídas de uma análise da forma da palavra. Características como *loops*, retas e curvas são comuns nesse método de classificação [GS97].

Em geral os métodos globais são utilizados em vocabulários pequenos ou pré-definidos. Em vocabulários maiores, as características extraídas da forma das palavras podem não ser suficientes para obter-se uma boa discriminação entre as várias classes de palavras. Nos métodos globais uma fase de treinamento empregando amostras das palavras do vocabulário considerado é sempre necessário. No caso dos métodos analíticos é necessário a utilização de um dicionário para cada novo vocabulário a ser reconhecido, no entanto, não é necessário novo treinamento do sistema.

2.5 O SISTEMA DE RECONHECIMENTO DESTA TESE

O sistema de reconhecimento descrito nesta tese pode ser visto segundo duas etapas, denominadas etapa de treinamento e etapa de reconhecimento. A Figura 2.1 ilustra essas etapas e os procedimentos que as compõem.

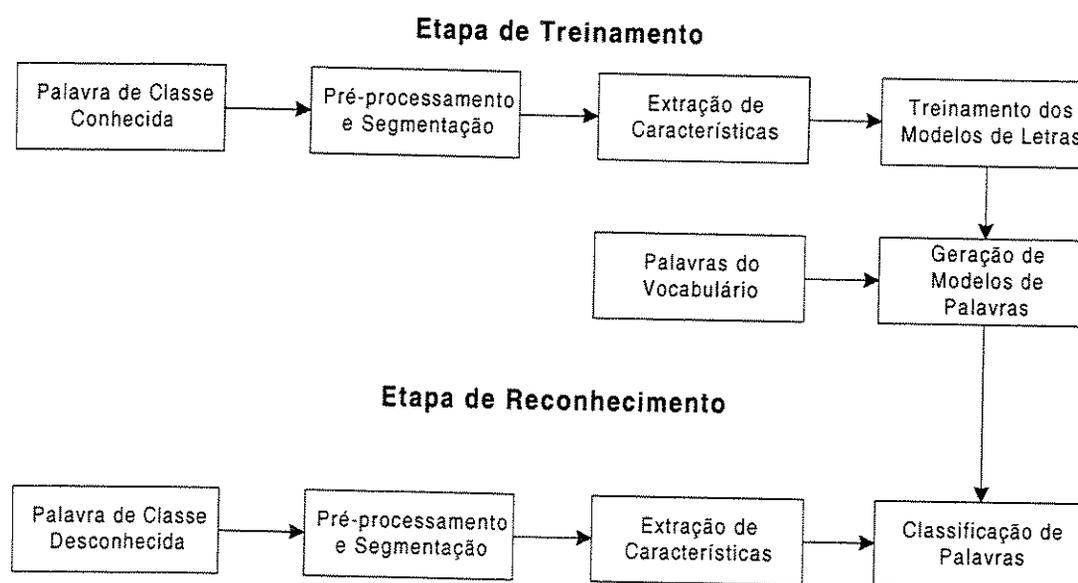


Figura 2.1 - Processo de reconhecimento de palavras manuscritas.

O processo de reconhecimento de uma palavra manuscrita é iniciado com o pré-processamento da imagem digitalizada da palavra. Nesse procedimento a imagem é transformada, passando a ser representada em dois níveis de cinza (preto e branco). São eliminados ruídos, corrigida possível rotação da palavra em relação ao eixo horizontal e inclinação em relação ao eixo vertical. Após o pré-processamento é realizada a segmentação explícita da palavra. Os segmentos resultantes dessa segmentação são afinados a fim de que apresentem uma espessura de 1 pixel [Ser82], [Par97]. Neste trabalho, a segmentação explícita de palavras facilita o processo de extração de características e também, produz amostras de caracteres para treinamento de modelos ocultos de Markov. Em seguida, na fase de extração de características, os segmentos resultantes da segmentação da palavra são decompostos e avaliados segundo tipos de retas, curvas e *loops* utilizando a teoria de conjuntos *fuzzy*. Uma seqüência de vetores de características é construída para representar a imagem em processamento. Os procedimentos de pré-processamento, segmentação e extração de características são comuns às etapas de treinamento e reconhecimento conforme ilustra a Figura 2.1.

Na etapa de treinamento a seqüência de vetores de características representa um caractere de classe conhecida. Essa seqüência é empregada no treinamento de modelos ocultos de Markov e de modelos ocultos de Markov generalizados construídos para cada classe de caractere. Os modelos para as palavras do vocabulário considerado são formados pela concatenação dos modelos de caracteres.

Na etapa de reconhecimento a seqüência de vetores representando uma palavra de classe desconhecida é processada pelos modelos de palavras geradas na etapa de treinamento. O processo de classificação se dá pela análise do modelo de palavra que apresentar o maior valor numérico obtido via algoritmo de Viterbi [Rab89].

Nos capítulos seguintes são descritos detalhadamente cada um dos procedimentos representados na Figura 2.1, os quais compõem o sistema de reconhecimento desenvolvido. Entretanto, antes da descrição desses procedimentos, são apresentados alguns conceitos necessários para compreendê-los. Dessa forma, o próximo capítulo é dedicado à teoria dos modelos ocultos de Markov.

CAPÍTULO III

MODELOS OCULTOS DE MARKOV

Apesar de ter sido introduzida por Baum no final da década de 60 [Rab89], é recente a aplicação da teoria dos modelos ocultos de Markov (*HMMs - Hidden Markov Models*) em reconhecimento de palavras manuscritas. Foi no início da década de 80 que os HMMs foram empregados no processamento de textos, sendo utilizados para analisar caracteres impressos por máquinas [Chen95]. A aplicação para processamento de palavras manuscritas considerando grandes vocabulários foi realizada por Kundu em 1988 [KHB88]. Atualmente, muitos trabalhos em reconhecimento de palavras manuscritas têm empregado com sucesso os modelos ocultos de Markov, [Sen94], [Chen95], [Moh95], [GS97], [YSS99], [LPP00].

Este capítulo é dedicado à teoria dos modelos ocultos de Markov, sendo abordado, primeiramente, a definição dos HMMs e os três problemas básicos associados ao desenvolvimento de sistemas modelados por HMMs. Em seguida, são apresentados alguns aspectos da implementação dos HMMs e comentadas formas de aplicação dos modelos ocultos de Markov no reconhecimento de palavras manuscritas. A simbologia e conceitos deste capítulo seguem aqueles usados em [Rab89].

3.1 DEFINIÇÃO

Um modelo oculto de Markov pode ser visto como um conjunto de estados ligados entre si através de transições. A cada intervalo de tempo ocorre uma transição de estado e um elemento do conjunto de símbolos sendo considerado é emitido. Cada transição de estado possui uma probabilidade de ocorrência e os símbolos são emitidos obedecendo uma dada distribuição de probabilidade. Verifica-se nessa abordagem que o modelo resultante é consequência de dois processos estocásticos interrelacionados, ou seja, um processo que gera uma seqüência finita de estados e outro processo (funções aleatórias) que produz a seqüência de símbolos observados. Cada estado está associado a uma função aleatória responsável pela emissão de um símbolo em cada mudança de estado. O modelo é dito oculto porque a seqüência de estados que gerou a seqüência de símbolos emitidos não é conhecida, isto é, permanece oculta, não podendo ser observada diretamente. Somente pode ser observada diretamente a seqüência de símbolos emitida.

Assumindo uma seqüência de símbolos observados de comprimento T , $O = \{o_1, \dots, o_T\}$, produzida por uma seqüência de estados não observáveis, $Q = \{q_1, \dots, q_T\}$, onde cada observação o_i pertence a um conjunto de M símbolos de observação, $V = \{v_k : 1 \leq k \leq M\}$, e cada estado q_i faz parte de um conjunto de N estados, $S = \{s_i : 1 \leq i \leq N\}$, pode-se formalmente caracterizar um HMM com densidades de observações discretas por, $\lambda = (\Pi, A, B)$, onde:

$\Pi = \{\pi_i \mid \pi_i = P(q_1 = s_i)\}$ é o conjunto representando a distribuição de probabilidade de estado inicial. O parâmetro π_i representa a probabilidade de estar no estado s_i no tempo $t = 1, 1 \leq i \leq N$;

$A = \{a_{ij} \mid a_{ij} = P(q_{t+1} = s_j / q_t = s_i)\}$ é uma matriz de probabilidades de transição entre estados, sendo a_{ij} a probabilidade de transição do estado s_i no tempo t para o estado s_j no tempo $t+1$, $1 \leq i \leq N$ e $1 \leq j \leq N$;

$B = \{b_j(k) \mid b_j(k) = P(o_t = v_k / q_t = s_j)\}$ denota um conjunto de distribuições de probabilidade de emissão de símbolo, onde $b_j(k)$ refere-se a probabilidade de emissão de um símbolo v_k estando no estado s_j no tempo t , $1 \leq j \leq N$ e $1 \leq k \leq M$.

3.2 PROBLEMAS BÁSICOS DO MODELAMENTO VIA HMMs

Três problemas básicos, enumerados a seguir, devem ser solucionados ao modelar sistemas utilizando HMMs:

1. Avaliação;
2. Determinação da seqüência de estados ótima;
3. Treinamento.

Na avaliação, o problema reside em como calcular de modo eficiente $P(O/\lambda)$, isto é, a probabilidade da seqüência de símbolos observados dado o modelo. Em outras palavras, fornecida uma seqüência de observações $O = o_1 o_2 \cdots o_T$ e o modelo $\lambda = (\Pi, A, B)$, deseja-se determinar a probabilidade de que a seqüência observada foi gerada pelo modelo. O problema de avaliação é comum quando, dentre vários HMMs, deseja-se selecionar aquele de maior probabilidade em ter gerado a seqüência de símbolos observados.

No problema referido como determinação da seqüência de estados ótima tenta-se, a partir de uma dada seqüência de observações $O = o_1 o_2 \cdots o_T$ e um modelo $\lambda = (\Pi, A, B)$,

obter a seqüência de estados que melhor explica, segundo algum critério, a dada seqüência de observações.

O problema de treinamento visa otimizar os parâmetros do modelo a fim de melhor descrever um dado fenômeno físico. Formalmente, o objetivo é ajustar os parâmetros do modelo $\lambda = (\Pi, A, B)$ de modo a maximizar $P(O/\lambda)$. O treinamento permite otimizar os parâmetros do modelo utilizando observações do fenômeno físico em questão.

3.3. SOLUÇÕES PARA OS TRÊS PROBLEMAS BÁSICOS

A seguir, são discutidas soluções para os problemas básicos do modelamento de sistemas empregando HMMs conforme apresentado em [Rab89].

3.3.1 PROBLEMA DE AVALIAÇÃO

No problema de avaliação deseja-se obter o valor de $P(O/\lambda)$, ou seja, deseja-se calcular a probabilidade da seqüência de símbolos observados $O = o_1 o_2 \dots o_T$ dado o modelo λ . Um modo para calcular tal probabilidade é enumerar todas as possíveis seqüências de estado de comprimento T , $Q = q_1 q_2 \dots q_T$, capazes de ter gerado O . Como a cada seqüência Q está associada uma probabilidade de gerar a seqüência de observações O , $P(O/\lambda)$ pode ser calculado da seguinte forma:

$$P(O/\lambda) = \sum_{\text{Para todo } Q} P(O, Q/\lambda) = \sum_{\text{Para todo } Q} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T), \quad (3.1)$$

onde $P(O, Q/\lambda)$ é a probabilidade conjunta da seqüência de observações $O = o_1 o_2 \dots o_T$ e da seqüência de estados $Q = q_1 q_2 \dots q_T$ dado o modelo λ .

O cálculo direto de $P(O/\lambda)$, conforme a expressão acima, envolve da ordem de $2T \cdot N^T$ cálculos [Rab89], sendo T o comprimento da seqüência de observações e N o número de estados para gerar tal seqüência de observações. Pode-se observar que o número de cálculos aumenta exponencialmente com o tamanho da seqüência de observações. Uma forma mais eficiente de calcular $P(O/\lambda)$ é utilizando o algoritmo *forward*, onde é definida a variável:

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = s_i / \lambda), \quad (3.2)$$

denotando a probabilidade de ocorrência da seqüência de observações $o_1 o_2 \dots o_t$ até o instante de tempo t e o estado s_i no tempo t dado o modelo λ . A variável $\alpha_t(i)$ pode ser calculada recursivamente de acordo com os seguintes passos:

1. **Inicialização:** A variável $\alpha_t(i)$ é inicializada como sendo a probabilidade conjunta do estado s_i e da observação inicial o_1 , ou seja:

$$\alpha_1(i) = \pi_i b_i(o_1), \quad t = 1, \quad 1 \leq i \leq N. \quad (3.3)$$

2. **Recursão:** Neste passo o produto $\alpha_t(i) \cdot a_{ij}$ denota a probabilidade da seqüência $o_1 o_2 \dots o_t$ ter sido observada no tempo t e o estado s_j ser alcançado no tempo $t+1$. É

observado que o estado s_j no tempo $t + 1$ pode ser alcançado através de qualquer um dos outros N estados no tempo t , o somatório na equação abaixo indica este fato.

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right] b_j(o_{t+1}) \quad 1 \leq t \leq T, 1 \leq j \leq N \quad (3.4)$$

3. **Término:** $P(O/\lambda)$ é obtida considerando todas as possíveis trajetórias de estados capazes de gerar a seqüência de observação em consideração, isto é:

$$P(O/\lambda) = \sum_{i=1}^N \alpha_T(i) \quad 1 \leq i \leq N \quad (3.5)$$

3.3.2 PROBLEMA DE DETERMINAÇÃO DA SEQÜÊNCIA DE ESTADOS ÓTIMA

Existem diferentes métodos para encontrar a seqüência de estados ótima para uma dada seqüência de observações. No entanto, é necessário especificar o critério que define uma seqüência como sendo ótima. Um critério é formar a seqüência de estados escolhendo em cada intervalo de tempo o estado individualmente mais provável. O problema com esse critério é que a seqüência obtida pode envolver transições entre estados que não são permitidas (por exemplo, $a_{ij} = 0$), uma vez que as probabilidades de ocorrência dessas transições não são observadas. Um outro critério, evitando esse tipo de problema, é achar a trajetória de estados de maior valor de probabilidade. Uma técnica formal para determinar esta trajetória é utilizando o algoritmo de Viterbi. Nesse algoritmo é definida a quantidade:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \dots q_{t-1}, q_t = s_i, o_1 o_2 \dots o_t / \lambda], \quad 1 \leq i \leq N \quad (3.6)$$

que expressa o maior valor de probabilidade ao longo de uma trajetória de estados no tempo t , levando em conta os t primeiros símbolos da seqüência de observações e terminando no estado s_i . O procedimento completo para calcular $\delta_t(i)$ recursivamente é apresentado abaixo:

1. Inicialização - Para tempo $t = 1$ têm-se:

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N, \quad (3.7a)$$

$$\varphi_1(i) = 0. \quad (3.7b)$$

2. Recursão - Para tempo $1 < t \leq T$ $\delta_t(i)$ é dado por:

$$\delta_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}] b_j(o_t), \quad 1 \leq i \leq N, \quad (3.8a)$$

$$\varphi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}], \quad 1 \leq j \leq N. \quad (3.8b)$$

3. Término - A seqüência de estados que maximiza $P(O/\lambda)$ tem probabilidade e último estado expressos, respectivamente, por:

$$P^* = \max_{1 \leq j \leq N} [\delta_T(j)] \quad 1 \leq j \leq N, \quad (3.9a)$$

$$q_T^* = \arg \max_{1 \leq j \leq N} [\delta_T(j)]. \quad (3.9b)$$

4. Recuperação da seqüência de estados maximizando $P(O/\lambda)$:

$$q_t^* = \varphi_{t+1}(q_{t+1}^*), \quad T > t \geq 1. \quad (3.10)$$

3.3.3 PROBLEMA DE TREINAMENTO

No problema de treinamento visa-se ajustar os parâmetros (A, B, Π) de um modelo oculto de Markov de modo a maximizar a probabilidade de uma dada seqüência de observações. Como não existe um método analítico para realizar tal operação, pode-se escolher os parâmetros do modelo de forma a maximizar localmente a função de verossimilhança $P(O/\lambda)$. Para este propósito é geralmente empregado o método de reestimação de Baum-Welch. O referido método utiliza o algoritmo *forward-backward* [Rab89] para solucionar o problema de treinamento através da reestimação dos parâmetros (A, B, Π) .

Para definir as fórmulas que compõem o procedimento de reestimação de Baum-Welch é necessário o estabelecimento de algumas definições. Inicialmente, de modo similar ao algoritmo *forward*, é definido o algoritmo *backward* que utiliza a variável $\beta_t(i)$ em contrapartida a $\alpha_t(i)$. A variável $\beta_t(i)$ expressa a probabilidade de ser gerada a seqüência $o_{t+1}o_{t+2}\dots o_T$ a partir do tempo $t+1$ até o tempo T , dado o estado s_i no tempo t e o modelo λ . Pode-se calcular $\beta_t(i)$ da seguinte maneira:

1. **Inicialização:** No tempo $t = T$, $\beta_t(i)$ é definida como sendo igual a 1 para todos os estados s_i do modelo, $1 \leq i \leq N$, ou seja:

$$\beta_T(i) = 1. \quad (3.11)$$

2. **Recursão:** Para determinar a probabilidade de observar a seqüência $o_{t+1}o_{t+2}\dots o_T$ estando no estado s_i , $1 \leq i \leq N$, no tempo $1 \leq t < T$, são considerados: todos os possíveis estados s_j do modelo, $1 \leq j \leq N$, no tempo $t+1$; a ocorrência de transição do estado s_i para o estado s_j ; o símbolo observado o_{t+1} no estado s_j , bem como, $B_{t+1}(j)$. Formalmente $\beta_t(i)$ é dado por:

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j). \quad (3.12)$$

3. **Término:** Este passo mostra uma outra alternativa para obtenção de $P(O/\lambda)$ utilizando o algoritmo *backward*:

$$P(O/\lambda) = \sum_{i=1}^N \pi_i b_i(o_1) \beta_1(i), \quad 1 \leq i \leq N. \quad (3.13)$$

Para obter as fórmulas de Baum-Welch também é definido o parâmetro, $\gamma_t(i) = P(q_t = i/O, \lambda)$, isto é, a probabilidade de estar no estado s_i no tempo t dada a seqüência de observações $O = o_1 o_2 \dots o_T$ e o modelo λ . Pode-se expressar $\gamma_t(i)$ utilizando as variáveis $\alpha_t(i)$ e $\beta_t(i)$, do seguinte modo:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)} \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq j \leq N \end{array}, \quad (3.14)$$

O algoritmo de reestimação de Baum-Welch utiliza ainda, o parâmetro $\varepsilon_t(i, j) = P(q_t = i, q_{t+1} = j | O, \lambda)$, definido como sendo a probabilidade de estar no estado s_i no tempo t e no estado s_j no tempo $t+1$, dada a seqüência de observações $O = o_1 o_2 \dots o_T$ e o modelo λ . Matematicamente, $\varepsilon_t(i, j)$ é dado pela seguinte expressão:

$$\varepsilon_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}, \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq j \leq N \end{array} \quad (3.15)$$

Utilizando os parâmetros $\varepsilon_t(i, j)$ e $\gamma_t(i)$ obtém-se:

- O número esperado de transições realizadas do estado s_i para o estado s_j para uma dada seqüência de observações, o qual é expresso por $\sum_{t=1}^{T-1} \varepsilon_t(i, j)$;
- O número esperado de transições realizadas a partir do estado s_i dada uma seqüência de observações, o qual é obtido por $\sum_{t=1}^{T-1} \gamma_t(i)$.

Usando as fórmulas acima é estabelecido um conjunto de operações para reestimar os parâmetros (A, B, Π) do modelo oculto de Markov com densidades de observações discretas,

como segue:

$$\bar{\pi}_j = \gamma_1(i), \quad (3.16)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \varepsilon_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad (3.17)$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)} \quad (3.18)$$

3.4 ALGUNS ASPECTOS DA IMPLEMENTAÇÃO DE HMMs

3.4.1 ESCALONAMENTO

Pode ser demonstrado [RJ93] que o parâmetro $\alpha_i(i)$ na equação (3.2) consiste da soma de um grande número de termos, cada termo da forma,

$$\left(\prod_{l=1}^{i-1} a_{q_l q_{l+1}} \prod_{l=1}^T b_{q_l}(o_l) \right),$$

e cada um desses termos tem valor menor do que a unidade. Então, para uma seqüência de observações suficientemente longa, $\alpha_t(i)$ terá um valor próximo de zero, podendo exceder a precisão de computadores digitais. Por conseguinte, pode ser necessário realizar um processo de escalonamento. Um forma bastante utilizada para realizar tal escalonamento é dividir cada $\alpha_t(i)$ pela soma de todo α no tempo t , ou seja:

$$\alpha'_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^N \alpha_t(j)}, \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq j \leq N \end{array} \quad (3.19)$$

Na equação (3.19), $\alpha'_t(i)$ representa o resultado do escalonamento de $\alpha_t(i)$ e o denominador é, desse modo, o fator de escala desejado. Os fatores de escala obtidos para o algoritmo *forward*, em cada intervalo de tempo t , devem ser utilizados ao longo do algoritmo *backward* para escalonamento da variável $\beta_t(i)$. Tais fatores de escala são cancelados no processo de reestimação em razão da operação de divisão, não afetando portanto as fórmulas de reestimação já apresentadas.

3.4.2 MÚLTIPLAS SEQÜÊNCIAS DE OBSERVAÇÃO

Em aplicações práticas são necessárias várias seqüências de observações para treinar um modelo oculto de Markov. As fórmulas de reestimação devem ser, portanto, modificadas para tratar essas múltiplas seqüências. Seja um conjunto com K seqüências de observações destinado ao treinamento de um HMM e representado por:

$$O = [O^1, O^2, \dots, O^K], \quad (3.20)$$

onde, $O^k = o_1^k o_2^k \dots o_T^k$, é a k -ésima seqüência de observações de comprimento T . Considerando que as K seqüências de observações sejam independentes e que o objetivo é ajustar os parâmetros do modelo λ de modo a maximizar $P(O/\lambda)$, tem-se:

$$P(O/\lambda) = \prod_{k=1}^K P(O^k/\lambda) \quad (3.21)$$

Para modificar as fórmulas de reestimação para tratar as múltiplas seqüências de observação basta, uma vez que tais fórmulas baseiam-se na freqüência de ocorrência de vários eventos, adicionar as freqüências de ocorrência individual de cada seqüência. Desse modo, as probabilidades de transição entre estados são reestimadas de acordo com a seguinte expressão:

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \frac{1}{P(O^k/\lambda)} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(o_{t+1}^k) \beta_{t+1}^k(j)}{\sum_{k=1}^K \frac{1}{P(O^k/\lambda)} \sum_{t=1}^{T_k-1} \alpha_t^k \beta_t^k(i)}, \quad (3.22)$$

De modo similar, utilizando as variáveis *forward* e *backward* escalonadas, são determinadas as fórmulas de reestimação para os demais parâmetros de um modelo oculto de Markov [R]93].

3.4.3 INICIALIZAÇÃO DE PARÂMETROS

As equações de reestimação garantem que os parâmetros estimados para um HMM correspondem a um máximo local da função de verossimilhança $P(O/\lambda)$, porém não garantem que um máximo global seja atingido. Procura-se então, encontrar valores iniciais para os parâmetros de um HMM de modo que o máximo local seja igual ou esteja próximo do máximo global da função de verossimilhança. Segundo Rabiner [Rab89], na reestimação dos parâmetros Π e A , tanto valores iniciais uniformes ou aleatórios proporcionam bons resultados. No entanto, para o parâmetro B , valores iniciais adequados ajudam no processo de convergência das fórmulas de reestimação, sendo essenciais quando a distribuição de probabilidade de emissão de símbolo é contínua. Em [RJ93] são apresentados alguns procedimentos para prover estimativas iniciais mais adequadas para os parâmetros de um modelo, possibilitando melhores resultados de convergência para as fórmulas de reestimação de Baum-Welch.

3.5 SÍNTESE SOBRE FORMAS DE UTILIZAÇÃO DOS HMMs EM RECONHECIMENTO DE PALAVRAS MANUSCRITAS

Uma das razões da utilização de um processo Markoviano, em reconhecimento de palavras, relaciona-se ao fato de que uma palavra pode ser analisada como uma seqüência de observações ou de símbolos de observações. Desse modo, pode-se modelar matematicamente uma palavra empregando uma cadeia de Markov em que os estados, por exemplo, podem representar caracteres individuais. No entanto, em se tratando de palavras manuscritas, é necessário um modelamento mais robusto da palavra, permitindo tratar ambigüidades de

forma dos caracteres causadas, principalmente, pelos estilos de escrita. Os modelos ocultos de Markov, em razão de apresentarem em sua composição um duplo processo estocástico, permitem um melhor modelamento de palavras manuscritas. A distribuição de probabilidade da emissão de símbolos em cada estado, proporciona um tratamento mais adequado de ambigüidades em uma palavra manuscrita, tornando o modelamento via HMM mais robusto.

Basicamente, em reconhecimento de palavras manuscritas, os HMMs podem ser definidos para modelar uma letra, uma palavra ou mesmo um conjunto de palavras. No procedimento em que cada palavra de um vocabulário é modelada por um HMM, ou seja, é definido um HMM distinto para cada palavra, a operação de classificação pode ser realizada ordenando as classes de palavras de acordo com a probabilidade da seqüência de observações dado o modelo, ou de acordo com as seqüências de estados mais prováveis segundo o algoritmo de Viterbi. Um exemplo desta abordagem foi utilizada por Guillevic [GS97] para processamento de cheques. Guillevic criou um HMM específico para cada classe de palavra utilizando uma topologia esquerda para direita, sendo o número de estados uma função do comprimento médio das seqüências de símbolos representando a classe de palavra em questão. O método empregando um HMM específico para modelar cada classe de palavra é factível se o número de palavras do vocabulário em uso é pequeno, caso contrário, há a necessidade de construir um grande número de modelos, tendo-se em vista variações na escrita e a quantidade de palavras do vocabulário.

Quando se deseja reconhecer palavras de vários estilos de escrita e/ou o vocabulário possui um grande número de palavras, uma abordagem mais adequada é a utilização de um HMM modelando cada caractere do alfabeto considerado. O modelo para uma palavra é obtido conectando adequadamente os HMMs modelando caracteres. Nesse tipo de abordagem, há a necessidade de segmentar a palavra manuscrita em caracteres. Como o

processo de segmentação não é exato, isto é, nem sempre são obtidos segmentos que são caracteres perfeitos, tem-se procurado tratar esse problema usando os HMMs. No trabalho desenvolvido por Chen [Chen95] o emprego dos HMMs visa, além de classificar a palavra de classe desconhecida, reduzir os efeitos da inexatidão do processo de segmentação. Chen utilizou um HMM onde os estados podiam representar um caractere, fração de um caractere ou caracteres conectados.

Um exemplo em que um HMM é empregado para classificar um conjunto de palavras é encontrado em [KHB88]. No referido trabalho os caracteres de uma palavra manuscrita eram considerados estar segmentados perfeitamente, possibilitando atribuir a cada estado do HMM uma letra do alfabeto. O modelo apresentava uma topologia de estados completamente conectada e o reconhecimento era realizado encontrando as $K = 1, 2, 3, \dots$ melhores seqüências de estado segundo o algoritmo de Viterbi. As seqüências eram avaliadas e, se lingüisticamente incorretas, eram eliminadas.

Nesta tese os HMMs são utilizados para modelar caracteres, sendo que cada estado pode representar um ou mais segmentos de linha em que é decomposto um caractere (capítulo VIII). Os HMMs modelando caracteres são conectados de modo a formar modelos para cada palavra do vocabulário considerado. A palavra de classe desconhecida é classificada de acordo com o modelo que apresentar a trajetória de estados de maior probabilidade segundo o algoritmo de Viterbi.

No próximo capítulo são abordados alguns dos conceitos da teoria de medidas e integrais *Fuzzy*, os quais são empregados em uma generalização dos Modelos Ocultos de Markov. A generalização desses modelos é utilizada na implementação de um dos classificadores empregados pelo sistema de reconhecimento tratado nesta tese.

CAPÍTULO IV

MEDIDAS *FUZZY* E INTEGRAL *FUZZY*

Este capítulo é dedicado à apresentação de alguns conceitos da teoria de medidas *fuzzy* e integrais *fuzzy* necessários ao método de classificação descrito nesta tese. As medidas *fuzzy* podem ser consideradas como uma escala subjetiva para graus de *fuzzyness* [Ger88], uma forma mais adequada para expressar valores que dependem quase que exclusivamente da subjetividade humana. A integral *fuzzy* pode ser vista como um operador de agregação definido com respeito às medidas *fuzzy*. Nos próximos parágrafos, buscando apresentar conceitos de medidas *fuzzy* e integral *fuzzy*, são introduzidas noções sucintas sobre σ -álgebra, espaços mensuráveis e medidas. Também é abordada a propriedade de não-aditividade das medidas *fuzzy* e apresentadas as definições formais da integral de Sugeno e da integral de Choquet.

4.1 Medidas

Antes de dar a definição formal do que seja uma medida, é interessante determinar os elementos que poderão ser medidos. Para isso é importante o conceito de σ -álgebra. Seja X um conjunto não vazio e ζ uma família de subconjuntos de X . Diz-se que ζ é uma σ -álgebra dos subconjuntos de X se:

1. $X \in \zeta$;
2. $A \in \zeta$ então, $A^c \in \zeta$, onde, $A^c = (X - A)$ é definido como sendo o complemento de A ;
3. A_n é uma seqüência enumerável de conjuntos em ζ , então, $\bigcup_{n=1}^{\infty} A_n \in \zeta$.

O par (X, ζ) consistindo do conjunto X e da σ -álgebra ζ dos subconjunto de X é denominado um espaço mensurável. Um conjunto em ζ será chamado σ -mensurável.

Uma medida μ é uma função de conjunto [KT66] sobre um espaço mensurável (X, ζ) que assume valores não negativos satisfazendo as seguintes condições:

1. $\mu(\emptyset) = 0$, onde \emptyset representa um conjunto vazio;
2. $\mu(A) \geq 0$ para todo $A \in \zeta$;
3. Para uma seqüência enumerável de conjuntos disjuntos A_n em ζ , tem-se

$$\mu\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mu(A_n), \text{ que é definida como propriedade de } \sigma\text{-aditividade;}$$

4. Para uma seqüência de conjuntos A_n em ζ , $n = 1, 2, \dots$, tal que $A_n \subset A_{n+1}$

$$\text{(seqüência crescente), tem-se } \lim_{n \rightarrow \infty} \mu(A_n) = \mu\left(\lim_{n \rightarrow \infty} A_n\right).$$

A probabilidade é um exemplo de medida largamente empregada do ponto de vista matemático. A propriedade de σ -aditividade para a medida de probabilidade é expressa por,

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n), \text{ onde } A_n, n = 1, 2, \dots, \text{ são conjuntos disjuntos entre si.}$$

Uma propriedade também verificada em uma medida μ é a aditividade. Uma medida é dita aditiva quando seu valor para a união finita de conjuntos disjuntos entre si é igual a soma dos valores individuais associados a cada conjunto. Em termos matemáticos a propriedade de

aditividade é expressa por $\mu\left(\bigcup_{n=1}^N A_n\right) = \sum_{n=1}^N \mu(A_n)$, sendo $A_n, n = 1, 2, \dots, N$, conjuntos

disjuntos entre si.

4.2 MEDIDAS FUZZY

Em determinados casos, a propriedade de aditividade não parece ser adequada [WK92], por exemplo, em casos envolvendo um julgamento subjetivo como identificar uma pessoa através da observação de seus olhos e boca. Nesse caso, uma avaliação parcial dos conjuntos denominados $A_1 = \{\text{olhos}\}$, $A_2 = \{\text{boca}\}$ e $A_3 = \{\text{olhos, boca}\}$ reflete diferentes graus de importância na identificação de uma pessoa, no entanto, a importância de A_3 não é necessariamente a soma dos graus de importância de A_1 e A_2 .

Em 1974, Sugeno [Sug77] introduziu o conceito de medidas *fuzzy* como uma generalização do conceito de medidas descrito na seção anterior. A propriedade de aditividade foi substituída pela condição de monotonicidade como é definido a seguir.

A função de conjunto $g : \zeta \mapsto [0,1]$ é uma medida *fuzzy* se satisfaz as condições :

UNICAMP

BIBLIOTECA CENTRAL

CRÓICA CIRCULANTE

1. $g(\emptyset) = 0$ e $g(X) = 1$;
2. Se $A \in \zeta$ e $B \in \zeta$, $A \subseteq B$, então $g(A) \leq g(B)$, o que caracteriza a propriedade de monotonicidade;
3. Se a seqüência de conjuntos $\{A_n\} \in \zeta$ é tal que, $A_1 \subseteq A_2 \subseteq \dots \subseteq A_n$ ou $A_1 \supseteq A_2 \supseteq \dots \supseteq A_n$, tem-se $\lim_{n \rightarrow \infty} g(A_n) = g\left(\lim_{n \rightarrow \infty} A_n\right)$, que representa a propriedade de continuidade.

Uma medida g dos conjuntos $A_1 = \{\text{olhos}\}$ e $A_3 = \{\text{olhos, boca}\}$, comentados anteriormente, pode ser interpretada como o grau de importância desses conjuntos na identificação da face de uma determinada pessoa. A condição de monotonicidade para esta medida seria dada por $g(A_1) \leq g(A_3)$, ou seja, o grau de importância de A_3 é maior ou igual ao grau de importância de A_1 , uma vez que $A_1 \subseteq A_3$.

4.3 UMA INTERPRETAÇÃO DA NÃO-ADITIVIDADE DE UMA MEDIDA FUZZY

O objetivo desta seção é dar uma idéia a respeito da não-aditividade de uma medida *fuzzy* em um contexto real. Para isto, um exemplo utilizado por Murofushi e Sugeno [MS89] é usado para dar uma interpretação à propriedade de não-aditividade de uma medida *fuzzy*. Na literatura podem ser encontradas outras interpretações [Sug77].

Seja uma medida *fuzzy* μ definida sobre uma σ -álgebra ζ dos subconjuntos do universo de discurso X e utilizada para medir um certo atributo em ζ . Tomando a adição

como sendo uma operação válida no processo de medição de um atributo, considera-se o seguinte exemplo:

Seja X o conjunto de todos os trabalhadores em uma oficina e A um grupo qualquer de trabalhadores pertencente a ζ . Uma vez que existem várias maneiras de um grupo trabalhar, ou seja, vários modos de divisão do trabalho entre seus integrantes, supõe-se que determinado grupo A consiga trabalhar do modo mais eficiente possível. Se $\mu(A)$ é proporcional ao número de produtos produzidos por A em uma hora, então, μ pode ser vista como uma medida expressando a produtividade do grupo. Devido a sua definição, a medida μ possui as seguintes propriedades:

1. $\mu(\emptyset) = 0$;
2. Se $A \subseteq B \Rightarrow \mu(A) \leq \mu(B)$.

De acordo com as propriedades acima, μ é uma medida *fuzzy*, no entanto, não é necessariamente aditiva. Por exemplo, se A e B são subconjuntos disjuntos de X , uma análise da produtividade do par $A \cup B$ pode mostrar que, se A e B trabalham separados, então $\mu(A \cup B) = \mu(A) + \mu(B)$. Mas, se existe uma interação entre A e B , então $\mu(A \cup B)$ pode ser diferente de $\mu(A) + \mu(B)$. Em outras palavras, se houver cooperação entre os integrantes de A e B , pode ocorrer que $\mu(A \cup B) > \mu(A) + \mu(B)$. Por outro lado, se houver incompatibilidade então pode-se obter $\mu(A \cup B) < \mu(A) + \mu(B)$. Como conclusão, verifica-se que a propriedade de não-aditividade de μ , neste caso, reflete uma maior ou menor interação entre dois grupos de trabalhadores quaisquer, mostrando que a aditividade expressa por $\mu(A \cup B) = \mu(A) + \mu(B)$ pode não ser verdade em todas as situações.

4.4 MEDIDAS λ -FUZZY

Em razão da definição da medida *fuzzy* g , a medida resultante da união de dois conjuntos disjuntos não é diretamente calculada das medidas individuais desses conjuntos. Buscando obter um método para o cálculo de tal medida, Sugeno introduziu as medidas λ -*fuzzy*, satisfazendo a seguinte propriedade:

$$g_\lambda(A \cup B) = g(A) + g(B) + \lambda \cdot g(A) \cdot g(B), \quad \lambda > -1, \quad (4.1)$$

onde g_λ é uma medida λ -*fuzzy* ou medida de Sugeno e A e B dois subconjuntos disjuntos pertencentes à σ -álgebra ζ dos subconjuntos do universo de discurso X . O parâmetro λ é empregado para descrever uma interação entre os conjuntos que estão sendo combinados através da operação de união. Quando $\lambda = 0$ não ocorre interação entre os conjuntos e a medida g_λ é aditiva. Se $\lambda > 0$ obtém-se a denominada *super-aditividade*, significando que a evidência associada à união de A e B é maior que as evidências individuais relacionadas a estes conjuntos, ou seja, $g_\lambda(A \cup B) \geq g(A) + g(B)$. Por outro lado, se $\lambda < 0$ tem-se uma *sub-aditividade*, podendo significar alguma incompatibilidade ou redundância entre os conjuntos A e B de tal forma que o efeito resultante é $g_\lambda(A \cup B) < g(A) + g(B)$.

O valor de λ pode ser obtido utilizando a condição $g_\lambda(X) = 1$. Para o caso de X apresentando dois elementos x_1 e x_2 , $x_1 \cap x_2 = \emptyset$, tem-se .

$$g_\lambda(X) = g_\lambda(x_1 \cup x_2); \quad (4.2a)$$

$$g_\lambda(X) = g(x_1) + g(x_2) + \lambda g(x_1)g(x_2); \quad (4.2b)$$

$$1 = g(x_1) + g(x_2) + \lambda g(x_1)g(x_2); \quad (4.2c)$$

$$1 = \frac{(1 + \lambda g(x_1))(1 + \lambda g(x_2)) - 1}{\lambda}; \quad (4.2d)$$

$$1 = \frac{\prod_{i=1}^2 (1 + \lambda g^i) - 1}{\lambda}, \quad g^i = g(x_i). \quad (4.2e)$$

Considerando o caso de n elementos, $n = 1, 2, 3, \dots$, o parâmetro λ é obtido resolvendo a seguinte equação:

$$1 + \lambda = \prod_{i=1}^n (1 + \lambda g^i), \quad g^i = g(x_i). \quad (4.3)$$

4.5 INTEGRAL FUZZY

A integral *fuzzy* é uma operação não linear definida sobre todos os conjuntos de um espaço mensurável. Esta integral também é conhecida como integral de Sugeno [Sug77], sendo formalmente definida do seguinte modo:

Seja (X, ζ) um espaço mensurável e $h: X \rightarrow [0,1]$ uma função de conjunto σ -mensurável. A integral *fuzzy* sobre um conjunto $A \subseteq X$ da função h com respeito à medida *fuzzy* g é expressa por:

$$\int_A h(x) \circ g(\cdot) = \sup_{\alpha \in [0,1]} \min[\alpha, g(A \cap H_\alpha)], \quad (4.4)$$

sendo:

$$H_\alpha = \{x \mid h(x) \geq \alpha\};$$

sup, refere-se ao valor supremo dos parâmetros envolvidos na operação da equação (4.4);

min, refere-se ao valor mínimo dos parâmetros considerados na equação (4.4).

Se o conjunto X é finito, $X = \{x_1, x_2, \dots, x_n\}$, o cálculo da integral *fuzzy* torna-se menos complexo [PG98]. Para isso a função $h : X \rightarrow [0,1]$ deve obedecer a seguinte condição, $h(x_1) \geq h(x_2) \geq \dots \geq h(x_n)$, caso isso não aconteça deve-se reordenar $h(x_i)$, $i = 1, \dots, n$, para que esta condição seja obedecida. A integral de Sugeno pode ser então definida da seguinte forma:

$$\int_A h(x) \circ g(\cdot) = \max_{i=1}^n [\min(h(x_i), g(A_i))], \quad (4.5)$$

onde a função \max diz respeito ao valor máximo dos parâmetros envolvidos na equação (4.5)

e $A_i = \{x_1, x_2, \dots, x_i\}$ define uma seqüência crescente de conjuntos, isto é, $A_1 \subset A_2 \subset \dots \subset A_i$.

Se g for uma medida *lambda-fuzzy*, $g(A_i)$ pode ser calculada recursivamente da seguinte maneira [TK90]:

$$g(A_i) = g^i + g(A_{i+1}) + \lambda g^i g(A_{i+1}), \quad (4.6)$$

sendo $g^i = g(\{x_i\})$ e $g(X) = 1$.

4.6 MEDIDAS FUZZY CONDICIONAIS

Sejam X e Y dois universos de discurso, uma medida *fuzzy* condicional sobre Y com respeito a X , denotada:

$$\sigma_Y(\cdot/x),$$

é uma medida *fuzzy* sobre Y para x pertencente a X . Uma medida g_Y sobre Y também pode ser induzida de $\sigma_Y(\cdot/x)$ e de uma medida *fuzzy* g_X , do seguinte modo:

$$g_Y(B) = \int_Y \sigma_Y(B/x) \circ g_X(\cdot) \quad \text{com } B \subseteq Y, \quad (4.7)$$

$g_X(\cdot)$ é denominada medida *fuzzy* a priori e $\sigma_Y(\cdot/x)$ uma medida *fuzzy* condicional; $\sigma_Y(B/x)$ mede o grau de *fuzziness* da declaração de que um elemento de B é devido a $x \in X$.

4.7 INTEGRAL DE CHOQUET

Enquanto a medida *fuzzy* é uma generalização do conceito de medida descrito na seção 4.1, a integral de Sugeno não pode ser considerada uma extensão da integral de Lebesgue. Em outras palavras, a integral de Sugeno não se reduz à integral de Lebesgue quando a medida é aditiva. Visando obter uma integral com respeito a uma medida *fuzzy* que fosse uma extensão da integral de Lebesgue, Murofushi e Sugeno [MS89] propuseram a integral de Choquet. Essa integral foi mais tarde utilizada por Mohamed [Moh95] para generalizar os modelos ocultos de Markov. Tal generalização utilizou o fato de que, para uma medida aditiva, a integral de Choquet se reduz à integral de Lebesgue. Desse modo, utilizando a integral de Choquet como

integral *fuzzy*, a multiplicação como operador de interseção *fuzzy* e a medida de probabilidade como medida *fuzzy*, Mohamed mostrou que sua generalização se reduz ao referido modelo de Markov.

A integral de Choquet pode ser definida da seguinte maneira:

$$\int_X h(x) \circ g(\cdot) = \sum_{i=1}^n [h(x_i) - h(x_{i+1})] g(A_i), \quad h(x_{n+1}) = 0, \quad (4.8)$$

do mesmo modo que na integral de Sugeno para um conjunto finito $X = \{x_1, x_2, \dots, x_n\}$, a função $h(x_i)$ deve obedecer a condição $h(x_1) \geq h(x_2) \geq \dots \geq h(x_n)$, e $g(A_i) = g(A_1, A_2, \dots, A_i)$.

4.8 COMENTÁRIOS REFERENTES AO CAPÍTULO

Neste capítulo foram expostos os conceitos necessários para entender uma generalização dos modelos ocultos de Markov empregando integrais *fuzzy*. No próximo capítulo o desenvolvimento de tal generalização é apresentado juntamente com os conceitos de variável *fuzzy forward* e variável *fuzzy backward* necessários para a formulação *fuzzy* do procedimento de reestimação de parâmetros desenvolvido por Baum-Welch [Rab89].

CAPÍTULO V

GENERALIZAÇÃO DOS MODELOS OCULTOS DE MARKOV

Neste capítulo é apresentada uma generalização dos modelos ocultos de Markov (HMMs) através do emprego de integrais *fuzzy* [Moh95]. Essa generalização tem como resultado o HMM *fuzzy* (*Fuzzy Hidden Markov Models - FHMMs*) e utiliza a integral de Choquet como integral *fuzzy*. Enquanto na definição de um HMM são empregadas esperanças condicionais, a generalização para um HMM *fuzzy* é baseada no uso de medidas *fuzzy* condicionais. O HMM *fuzzy* é um dos métodos do processo de classificação de palavras manuscritas utilizado nesta tese. A integração *fuzzy*, em razão do relaxamento da propriedade de aditividade, permite utilizar alguma informação resultante da interdependência entre letras formando uma palavra ou entre segmentos formando uma letra para treinamento do HMM *fuzzy*. A utilização dessa informação produz uma maior ou menor interação entre os estados do HMM *fuzzy*, podendo melhorar o desempenho do classificador de palavras manuscritas.

5.1 APRESENTAÇÃO DA NOMENCLATURA UTILIZADA

Nesta tese, para facilitar a apresentação do processo de generalização, os modelos ocultos de Markov serão denominados HMMs clássicos. A notação listada abaixo é aquela proposta por Mohamed [Moh95] na definição de um HMM *fuzzy*.

T - representa o comprimento da seqüência de observações;

$O = \{O_1 O_2 \cdots O_T\}$ - denota uma seqüência de observações;

Ω_{t_1, t_2} - refere-se ao conjunto de todas as subseqüências de observação $O_{t_1} O_{t_1+1} \cdots O_{t_2}$;

N - especifica o número de estados do modelo;

q_t - representa um estado visitado no tempo $1 \leq t \leq T$;

$Q = \{q_1 q_2 \cdots q_T\}$ - refere-se a uma seqüência de estados de comprimento T ;

$S = \{S_1 S_2 \cdots S_N\}$ - diz respeito aos estados em um modelo;

$X = \{x_1 x_2 \cdots x_t\}$ - denota os estados do HMM *fuzzy* visitados no intervalo de tempo t ;

$Y = \{y_1 y_2 \cdots y_{t+1}\}$ - representa os estados do HMM *fuzzy* no intervalo de tempo $(t+1)$;

$\hat{\pi}_s(\cdot)$ - denota uma medida *fuzzy* sobre S , é referida como medida *fuzzy* de estado inicial;

$\hat{\pi}_i = \hat{\pi}_s^i = \hat{\pi}_s(\{S_i\})$ - é uma densidade *fuzzy* de estado inicial, $1 \leq i \leq N$;

$\hat{\pi} = [\hat{\pi}_i]$ - representa um vetor de densidades *fuzzy* de estado inicial, $1 \leq i \leq N$;

$\hat{b}_j(O_t)$ - denota a pertinência da observação O_t , $1 \leq t \leq T$, ao estado S_j , $1 \leq j \leq N$;

$\hat{B} = [\hat{b}_i(O_t)]$ - refere-se a uma matriz de valores de pertinência;

$\hat{a}_y(\cdot/x)$ - é uma medida *fuzzy* condicional em Y com respeito a $x \in X$, é denominada medida *fuzzy* de transição;

$\hat{a}_{ij} = \hat{a}_y(\{y_j\}/x_i)$ - denota uma densidade *fuzzy* de transição entre os estados S_i e S_j , $1 \leq i \leq N$ e $1 \leq j \leq N$;

$\hat{A} = [\hat{a}_{ij}]$ - representa uma matriz de densidades *fuzzy* de transição.

5.2 UMA INTERPRETAÇÃO PARA AS MEDIDAS *FUZZY* DE ESTADO INICIAL E DE TRANSIÇÃO

A densidade *fuzzy* de estado inicial $\hat{\pi}_i$ pode ser interpretada como uma medida do grau de certeza da declaração de que o estado inicial q_1 é igual a S_i , isto é, um dos estados do modelo. Estendendo essa interpretação para uma medida *fuzzy* de estado inicial $\hat{\pi}_s(G)$, $G \subset S$, ter-se-ia que tal medida reflete o grau de certeza da declaração de que o estado inicial está contido no subconjunto G , ou seja, $q_1 \in G$.

A interpretação dada para densidade *fuzzy* de transição \hat{a}_{ij} é de que essa densidade mede o grau de certeza em se declarar que estar em y_j (estado S_j no intervalo de tempo $t+1$) é resultado de se ter visitado x_i (estado S_i no intervalo de tempo t). Uma extensão dessa interpretação para uma medida *fuzzy* de transição referida como $\hat{a}_y(F/x)$, $F \subset Y$ e $x \in X$, reflete o grau de certeza da declaração de que visitar um dos elementos do subconjunto F é resultado de se ter visitado um dos estados x do HMM *fuzzy* no tempo t .

A seguir, são apresentadas as formulações *fuzzy* para os algoritmos *forward*, *backward*, Viterbi e Baum-Welch, todos já comentados no terceiro capítulo desta tese.

5.3 FORMULAÇÃO FUZZY PARA AS VARIÁVEIS FORWARD

Seja o produto cartesiano de $\Omega_{1,t}$ e X dado por $\Omega_x = \Omega_{1,t} \times X$, e seja $\hat{\alpha}_{\Omega_x} : 2^{\Omega_x} \rightarrow [0,1]$, uma medida *fuzzy* no espaço $(\Omega_x, 2^{\Omega_x})$. Para qualquer conjunto $E \subset X$, $\hat{\alpha}_{\Omega_x}(\{O_1 \cdots O_t\} \times E)$ mede o grau de certeza da declaração de que a seqüência $O_1 O_2 \cdots O_t$ foi observada e se está visitando um estado contido em E . Para essa medida, considerando um determinado estado x (estado S_i no intervalo de tempo t) e uma dada seqüência $O_1 O_2 \cdots O_t$, $\hat{\alpha}_{\Omega_x}^i = \hat{\alpha}_{\Omega_x}(\{O_1 \cdots O_t\} \times \{x_i\})$ denota uma densidade *fuzzy* referida como variável *fuzzy forward*, podendo ser também denotada por $\alpha_i(i)$. A variável *fuzzy forward* mede o grau de certeza da declaração de que foi observada a seqüência $O_1 O_2 \cdots O_t$ e se está visitando x_i (estado S_i no intervalo de tempo t). Para o tempo $t = 1$ a variável *fuzzy forward* é calculada como segue:

$$\hat{\alpha}_1(i) = \hat{\pi}_i \wedge \hat{b}_i(O_1), \quad 1 \leq i \leq N, \quad (5.1)$$

onde “ \wedge ” denota uma operação de interseção *fuzzy* [PG98]. Para $1 < t \leq T$ essa variável é obtida através da expressão:

$$\hat{\alpha}_{t+1}(j) = \int_X \hat{a}_y(\{y_j\}/x) \circ \hat{\alpha}_{\Omega_x}(\{O_1 \cdots O_t\}, \cdot) \wedge \hat{b}_j(O_{t+1}), \quad 1 \leq j \leq N. \quad (5.2)$$

5.4 FORMULAÇÃO FUZZY PARA AS VARIÁVEIS *BACKWARD*

Seja $\hat{\beta}_{\Omega_{t+1,T}}(\cdot/x)$ uma medida *fuzzy* condicional sobre $\Omega_{t+1,T}$ com respeito a x . Para qualquer subsequência $O_{t+1}O_{t+2}\cdots O_T$, $\hat{\beta}_{\Omega_{t+1,T}}(\cdot/x_i)$ é uma medida do grau de certeza existente na declaração de que observar a subsequência $O_{t+1}O_{t+2}\cdots O_T$ é resultado de se ter visitado x_i (estado S_i no intervalo de tempo t). A medida *fuzzy* condicional $\hat{\beta}_{\Omega_{t+1,T}}(\cdot/x)$ pode ser calculada a partir da medida *fuzzy* $\hat{\beta}_{\Omega_{t+2,T}}(\cdot/y)$ e da medida *fuzzy* de transição $\hat{a}_y(\cdot/x)$ como segue:

$$\hat{\beta}_{\Omega_{t+1,T}}(\{O_{t+1}\cdots O_T\}/x) = \int_Y [\hat{\beta}_{\Omega_{t+2,T}}(\{O_{t+2}\cdots O_T\}/y) \wedge \hat{b}_j(O_{t+1})] \circ \hat{a}_y(\cdot/x), \quad 1 \leq t < T \quad (5.3)$$

$\hat{\beta}_{\Omega_{t+1,T}}(\cdot/x_i)$ é denominada variável *fuzzy backward* e pode ser denotada por $\hat{\beta}_t(i)$. Para o tempo $t = 1$, $\hat{\beta}_t(i)$ é igual a unidade para todo $1 \leq i \leq N$.

5.5 GENERALIZAÇÃO DOS HMMs UTILIZANDO A INTEGRAL DE CHOQUET

Para realizar a generalização de um modelo oculto de Markov, conforme proposto em [Moh95], são definidas algumas variáveis a partir da integral de Choquet. A expressão para a referida integral, como apresentada no capítulo IV, é dada por:

$$e = \sum_{i=1}^n h(x_i) [g_i^n - g_{i+1}^n], \quad (5.4)$$

Da equação (5.4) são definidas as variáveis δ_i e ρ_i , da seguinte maneira:

$$\delta_i = [g_i^n - g_{i+1}^n], \quad (5.5a)$$

$$\rho_i = \frac{\delta_i}{g_i^n}, \quad 1 \leq i \leq N. \quad (5.5b)$$

Utilizando a equação (5.3) e a definição de ρ_i obtida da equação (5.5b), pode-se redefinir a variável *fuzzy backward* da seguinte forma:

$$\hat{\beta}_i(i) = \sum_{j=1}^N [\hat{\beta}_{i+1}(j) \hat{b}_j(o_{i+1})] \hat{a}_{ij} \rho_i(i, j), \quad (5.6)$$

onde:

$$\rho_i(i, j) = \frac{\hat{a}_y(\{y_j, y_{j+1}, \dots, y_N\} / x_i) - \hat{a}_y(\{y_{j+1}, y_{j+2}, \dots, y_N\} / x_i)}{\hat{a}_{ij}}, \quad \begin{array}{l} 1 \leq i \leq N, \\ 1 \leq j \leq N. \end{array} \quad (5.7)$$

A expressão na equação (5.6) é similar a equação (3.12) utilizada para o modelo clássico, exceto pelo termo representado por $\rho_i(i, j)$. Esse termo representa uma interação entre os estados do modelo devido a agregação não linear de informação resultante do uso da integral *fuzzy*. Se não ocorrer essa interação (medida aditiva), como exemplificado na seção 4.3,

a medida *fuzzy* de transição expressa por $\hat{a}_y(\cdot/x_i)$ na equação (5.7) será igual a zero e $\hat{a}_y(\cdot/x_i)$ será igual a \hat{a}_{ij} . Assim, a equação (5.6) torna-se igual a equação (3.12) estabelecida para um modelo oculto de Markov. Dessa forma, se utilizada a integral de Choquet como integral *fuzzy*, a operação de multiplicação como operador de interseção *fuzzy* e a medida de probabilidade como medida *fuzzy*, pode-se recuperar o HMM clássico a partir de um HMM *fuzzy*. Resultado similar a equação (5.6) é obtido para a equação em (5.2), utilizando a integral de Choquet e a variável $\rho_t(i, j)$:

$$\hat{\alpha}_{t+1}(j) = \left[\sum_{i=1}^N \hat{a}_{ij} \rho_t(i, j) \hat{\alpha}_t(i) \right] \hat{b}_j(O_{t+1}), \quad 1 \leq j \leq N. \quad (5.8)$$

5.6 FORMULAÇÃO FUZZY DO PROCESSO DE REESTIMAÇÃO - MODIFICAÇÃO DO MÉTODO DE BAUM-WELCH

Seja $\hat{P}(O/\hat{\lambda})$, como denominada em [Moh95], a possibilidade de observação da seqüência O dado um modelo $\hat{\lambda}$, isto é, uma medida do grau de pertinência [SS99] de O ao conjunto de seqüências de observações geradas pelo modelo $\hat{\lambda}$, tem-se que:

$$\hat{P}(O/\hat{\lambda}) = \sum_{i=1}^N \hat{\alpha}_t(i) \hat{\beta}_t(i); \quad (5.9a)$$

$$\hat{P}(O/\hat{\lambda}) = \sum_{i=1}^N \sum_{j=1}^N \hat{\alpha}_t(i) [\hat{\beta}_{t+1}(j) \hat{b}_j(O_{t+1})] \hat{a}_{ij} \rho_t(i, j); \quad (5.9b)$$

$$\hat{P}(O/\hat{\lambda}) = \sum_{\text{Para todo } Q} \hat{\pi}_{q_1} \hat{b}_{q_1}(O_1) \prod_{t=2}^T [\hat{a}_{q_{t-1}, q_t} \rho_t(q_{t-1}, q_t)] \hat{b}_{q_t}(O_t). \quad (5.9c)$$

Maximizando $\hat{P}(O/\hat{\lambda})$ e derivando a equação (5.9c) em relação a \hat{a}_{ij} é obtida a fórmula de reestimação para a medida *fuzzy* de transição:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^T \hat{\alpha}_t(i) \rho_t(i, j) \hat{a}_{ij} \hat{\beta}_{t+1}(j) \hat{b}_j(O_{t+1})}{\sum_{k=1}^N \sum_{t=1}^T \hat{\alpha}_t(i) \rho_t(i, k) \hat{a}_{ij} \hat{\beta}_{t+1}(k) \hat{b}_k(O_{t+1})}. \quad (5.10)$$

Do mesmo modo, maximizando $\hat{P}(O/\hat{\lambda})$ e derivando a equação (5.9c) em relação a $\hat{\pi}$ e $b_j(O_t)$, obtêm-se as fórmulas de reestimação para:

1 - Densidade *fuzzy* de estado inicial:

$$\bar{\pi}_i = \frac{\hat{\alpha}_1(i) \hat{\beta}_1(i)}{\sum_{j=1}^N \hat{\alpha}_1(j) \hat{\beta}_1(j)}; \quad (5.11)$$

2 - Pertinência da observação $O_t = v_k$ no estado S_j , $1 \leq j \leq N$:

$$\bar{b}_{jk} = \frac{\sum_{t=1, O_t=v_k}^T \hat{\alpha}_t(i) \hat{\beta}_t(i)}{\sum_{t=1}^T \hat{\alpha}_t(i) \hat{\beta}_t(i)}, \quad (5.12)$$

onde, v_k , representa um símbolo do conjunto de M símbolos de observação sendo utilizado,

$1 \leq k \leq M$.

5.7 ALGORITMO DE VITERBI MODIFICADO

A formulação do algoritmo de Viterbi leva em conta a variável $\rho_t(i, j)$, introduzida no processo de generalização do HMM clássico pela integral de Choquet. Basicamente, a modificação desse algoritmo [Moh95] consiste na substituição de a_{ij} por $\hat{a}_{ij}\rho_t(i, j)$.

A variável $\rho_t(i, j)$ varia com o tempo e pode ser interpretada como uma informação relacionada a seqüência de símbolos sendo observada. Essa informação caracteriza uma maior ou menor interação entre os estados do modelo *fuzzy* em razão da seqüência de símbolos observada.

5.8 ALGUNS ASPECTOS DA IMPLEMENTAÇÃO PARA HMMs FUZZY

Similarmente aos HMMs clássicos, é necessário realizar um processo de escalonamento no modelamento de sistemas empregando HMMs *fuzzy*. Para uma seqüência de observações suficientemente longa, o cálculo de $\hat{\alpha}_t(i)$ e $\hat{\beta}_t(i)$ pode exceder a precisão de computadores digitais. Isto pode acontecer em razão do emprego da integral de Choquet como integral *fuzzy* e da operação de multiplicação como operador de interseção *fuzzy*, uma vez que os elementos envolvidos nessas operações possuem valores menores do que a unidade. O mesmo procedimento de escalonamento utilizado para os HMMs clássicos pode ser utilizado nos HMM *fuzzy*, observada a medida *fuzzy* empregada com a integral de Choquet.

Outro aspecto da implementação dos HMMs *fuzzy*, semelhante aos HMMs clássicos, diz respeito a utilização de múltiplas seqüências de observações no procedimento de reestimação. Seja um conjunto com K seqüências de observações representado por:

$$O = [O^1, O^2, \dots, O^K], \quad (5.13)$$

onde, $O^k = o_1^k o_2^k \dots o_T^k$, é a k -ésima seqüência de observações de comprimento T .

Considerando o ajuste dos parâmetros do modelo $\hat{\lambda}$ de modo a maximizar $\hat{P}(O/\hat{\lambda})$, tem-se:

$$\hat{P}(O/\hat{\lambda}) = \prod_{k=1}^K \hat{P}(O^k/\hat{\lambda}). \quad (5.14)$$

Usando as equações (5.9c) e (5.14), a modificação da fórmula de reestimação para medida *fuzzy* de transição entre estados é dada por:

$$\bar{\hat{a}}_{ij} = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \hat{\alpha}_i^k(i) \hat{a}_{ij} \rho_i^k(i, j) \hat{b}_j(o_{t+1}^k) \hat{\beta}_i^k(j)}{\sum_{j=1}^N \sum_{k=1}^K \sum_{t=1}^{T_k-1} \hat{\alpha}_i^k(i) \hat{a}_{ij} \rho_i^k(i, j) \hat{b}_j(o_{t+1}^k) \hat{\beta}_i^k(j)}, \quad (5.15)$$

do mesmo modo, usando as variáveis *fuzzy forward* e *fuzzy backward* escalonadas, são obtidas as fórmulas de reestimação para os parâmetros $\bar{\hat{\pi}}$ e $\bar{\hat{b}}_{jk}$.

5.8 COMENTÁRIOS

Foi apresentada neste capítulo uma generalização dos modelos ocultos de Markov utilizando integrais *fuzzy*. Essa generalização foi denominada HMM *fuzzy* (*Fuzzy Hidden Markov Models* - FHMM). Os FHMMs são utilizados na implementação de um dos classificadores de palavras manuscritas utilizados pelo sistema de reconhecimento abordado por esta tese. No capítulo IX é descrita a forma como os FHMMs são empregados para classificar palavras manuscritas. A partir do próximo capítulo passa-se a descrever cada uma das fases que compõem o sistema de reconhecimento de palavras manuscritas. Especificamente, o próximo capítulo trata do melhoramento de imagens de palavras manuscritas para obtenção de características de boa qualidade.

CAPÍTULO VI

PRÉ-PROCESSAMENTO DE PALAVRAS MANUSCRITAS

A imagem de uma palavra manuscrita está sujeita a deformações introduzidas pelo processo de escrita e de digitalização. Em geral, as deformações causadas pelo processo de escrita estão relacionadas ao material usado para escrever e ao estilo de escrita de cada indivíduo. As deformações geradas pelo processo de digitalização podem ser relacionadas à escolha de parâmetros como níveis de cinza e resolução da imagem digitalizada. O emprego de técnicas de pré-processamento visa eliminar ou reduzir essas deformações, com a finalidade de facilitar a extração de características da imagem e minimizar as variações entre amostras de uma mesma palavra. As próximas seções deste capítulo são dedicadas às técnicas de pré-processamento empregadas no sistema de reconhecimento sendo abordado.

6.1 PROCESSO DE BINARIZAÇÃO

O processo de reconhecimento descrito nesta tese é aplicado a imagens binárias, ou seja, imagens empregando dois níveis de cinza: um nível de cinza representando os objetos considerados principais na imagem e um outro nível de cinza representando o fundo da imagem (*background*). Uma parte das imagens utilizadas para treinamento e teste do sistema de reconhecimento são provenientes de cheques de bancos brasileiros. Em geral, estas imagens apresentam padrões de fundo que identificam a instituição bancária, mas prejudicam a transformação em imagens binárias. Como o sistema desenvolvido deve reconhecer palavras manuscritas em diversos tipos de documentos, é necessário um processo de binarização capaz de tratar imagens com diferentes padrões de fundo.

Em imagens de cheques, a pequena variação de tons de cinza entre a região considerada principal e a região do fundo da imagem, bem como o grande número de tons de cinza, dificultam a determinação de um limiar para separar estas regiões. Neste trabalho, para obtenção de uma separação satisfatória entre a região principal e o fundo da imagem, foi empregado o método desenvolvido por Otsu [Par97], que utiliza informações da distribuição de níveis de cinza da imagem.

O método de Otsu emprega o princípio do procedimento estatístico de análise de variância para determinar um valor de limiar que divide uma imagem de L níveis de cinza em duas classes de pixels: c_0 (pixels pertencentes ao fundo da imagem) e c_1 (pixels pertencentes ao objeto ou região principal). Para cada valor de limiar, $0 \leq t < L$, é calculada a variância entre classes, σ_{ec}^2 , a partir das diferenças entre o tom de cinza médio da imagem, μ_{im} , e os tons de cinza médios, μ_0 e μ_1 , das respectivas classes c_0 e c_1 . O valor de limiar t a ser utilizado no

processo de binarização é aquele que produzir a menor razão, η_t , entre σ_{ec}^2 e a variância da imagem σ_{im}^2 :

$$\eta_t = \frac{\sigma_{ec}^2}{\sigma_{im}^2}. \quad (6.1)$$

O valor de σ_{ec}^2 , representando a variância entre classes, pode ser calculado de acordo com a expressão:

$$\sigma_{ec}^2 = \omega_0 \omega_1 (\mu_0 \mu_1)^2, \quad (6.2)$$

onde, $\omega_0 = \sum_{i=0}^t P_i$, representa a probabilidade de ocorrência da classe c_0 e $\omega_1 = 1 - \omega_0$, representa a probabilidade de ocorrência da classe c_1 . P_i é a probabilidade do nível de cinza i obtido do histograma da imagem. Os valores de níveis de cinza médio μ_0 e μ_1 , das respectivas classes, c_0 e c_1 , são dados por:

$$\mu_0 = \frac{\sum_{i=0}^t i \cdot P_i}{\omega_0}, \quad (6.3)$$

$$\mu_1 = \frac{\sum_{i=t+1}^{L-1} i \cdot P_i}{1 - \omega_0}. \quad (6.4)$$

Na Figura 6.2 é mostrado o resultado do processo de binarização da Figura 6.1 empregando o método de Otsu.

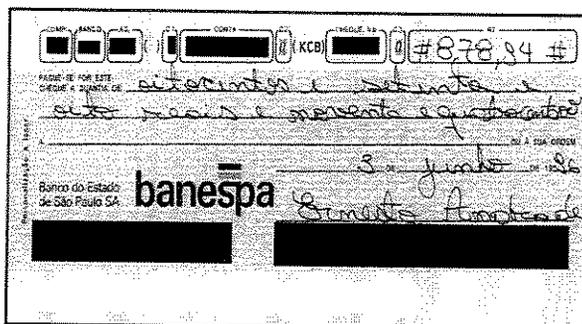


Figura 6.1 - Imagem de cheque apresentando fundo com desenhos em tons de cinza.

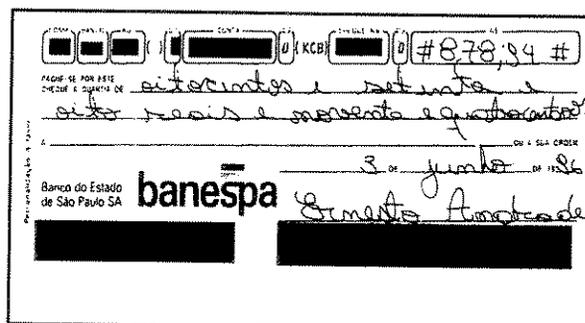


Figura 6.2 - Imagem de cheque após processo de binarização.

6.2 REMOÇÃO DE RUÍDOS

O processo de remoção visa eliminar ruídos provenientes tanto do documento original, do processo de digitalização, quanto de transformações como, correção de rotação e correção de inclinação da imagem. Na remoção de ruídos é empregado o conjunto de máscaras de dimensão 3x3 ilustradas nas Figuras 6.3 e 6.4 e as configurações geradas da rotação das referidas máscaras por ângulos de 90°, 180° e 270° graus. Estas máscaras têm sido empregadas

em vários trabalhos, [WW94], [GL97b], para melhorar a qualidade de imagens, e suas configurações de pixels permitem suavizar contornos e eliminar pontos isolados na imagem. Sempre que uma configuração de pixels na imagem coincidir com a configuração apresentada por uma das máscaras, um pixel preto é adicionado ou eliminado da posição da imagem correspondente ao elemento central da máscara. Um pixel preto é eliminado quando ocorrer a configuração apresentada pelas máscaras (a), (b), (c), (d) e (e) da Figura 6.3. Um pixel preto é adicionado quando ocorrer as configurações das máscaras (f), (g) e (h) apresentadas na Figura 6.3. Nessas máscaras os pixels brancos são representados pelo número “0” e os pixels pretos pelo número “1”, a situação onde a cor do pixel não é importante é simbolizada por “X”.

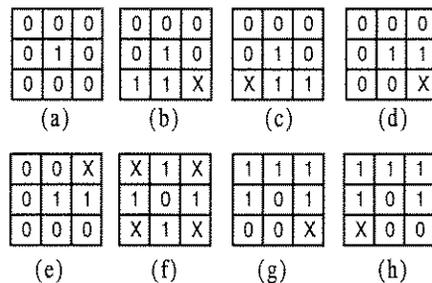


Figura 6.3 - Máscaras para remoção de ruídos.

No caso da eliminação de ruídos gerados pelos processos de correção de rotação e correção de inclinação de palavras manuscritas, as máscaras ilustradas na Figura 6.3 são usadas conjuntamente com as máscaras da Figura 6.4. Tanto a correção de rotação como a correção de inclinação provocam uma mudança na posição de vários pixels da imagem, produzindo um contorno ruidoso, como é exemplificado na Figura 6.5.

X	1	X
0	1	0
0	0	0

(a)

0	0	1
0	1	0
0	0	0

(b)

Figura 6.4 - Máscaras para remoção de ruídos em imagens após correção de rotação e inclinação.

As máscaras da Figura 6.4 reduzem ruídos através da eliminação de pixels pretos causadores de elevações abruptas no contorno da imagem.

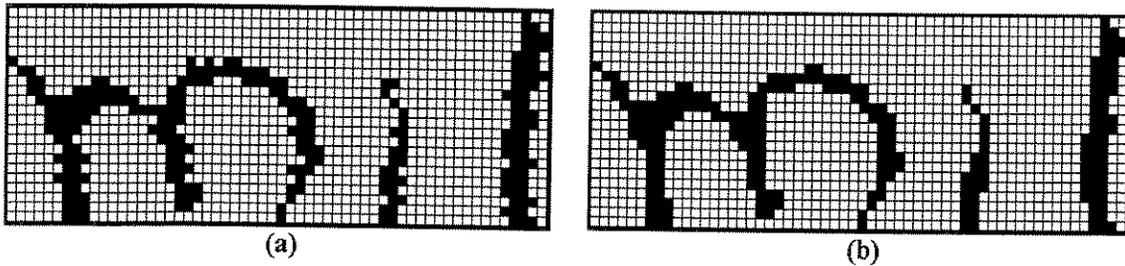


Figura 6.5 - Contorno ruidoso. (a) Imagem apresentando contorno ruidoso. (b) Imagem após processo de remoção de ruídos.

6.3 CORREÇÃO DE ROTAÇÃO DE PALAVRAS MANUSCRITAS

Palavras, figuras e outros elementos gráficos que compõem a imagem digitalizada de um documento, podem estar rotacionados por um ângulo φ em relação ao eixo horizontal, como ilustra a Figura 6.6. Imagens digitais rotacionadas podem ser causadas, por exemplo, pelo posicionamento inadequado de um documento em um dispositivo digitalizador. Para prevenir problemas com o rotacionamento de uma palavra, optou-se por determinar o segmento de linha passando na base da palavra e corrigir sua inclinação, fazendo-o paralelo ao eixo horizontal. Tal segmento foi denominado *linha de base* da palavra manuscrita.



Figura 6.6 - Palavra manuscrita rotacionada.

Podem ser encontrados na literatura diferentes métodos para determinar a linha de base de uma palavra e sua inclinação [Gui95], [Sen94], [BG83]. O rotacionamento de uma palavra não constitui um problema crítico neste trabalho, uma vez que os cheques brasileiros possuem linhas horizontais servindo de referência à escrita. Por esta razão, não houve a necessidade de se avaliar todos os métodos citados na literatura, tendo sido desenvolvido um procedimento de resultados bastantes satisfatórios. O princípio básico deste procedimento é a utilização de uma regressão linear para determinar a linha de base e o ângulo φ de inclinação desta linha. O procedimento é descrito pelo algoritmo abaixo, e pode ser considerado uma combinação dos métodos utilizados em [Gui95] e [Sen94]. Todos os passos do algoritmo são realizados varrendo-se a imagem de uma palavra manuscrita da esquerda para a direita e de cima para baixo.

Algoritmo:

1. Obter o histograma horizontal de blocos da imagem conforme mostrado no Anexo D;
2. Determinar a linha da imagem onde ocorre o ponto de máximo do histograma;
3. A partir da linha obtida no item 2 determinar a linha da imagem onde ocorre o ponto de mínimo do histograma;

4. Assinalar o último pixel de cada coluna da imagem entre a linha onde ocorre o ponto de máximo e a linha onde ocorre o ponto de mínimo do histograma;
5. Determinar os mínimos locais [Ser96] dos pixels assinalados no item anterior;
6. Empregar o método dos mínimos quadrados (Anexo A) para estimar o segmento de reta passando entre os pontos de mínimo locais, ou seja, estimar a linha de base da palavra;
7. Corrigir a imagem rotacionada utilizando o ângulo de inclinação do segmento de reta estimado.

O algoritmo acima não emprega todos os pixels formando a palavra para estimar a linha de base, apenas os mínimos locais obtidos no passo 5 são utilizados. O emprego dos pontos de mínimos locais tem por objetivo reduzir a influência de pixels afastados da base da palavra na estimação da linha de base. A posição do máximo e a posição do mínimo do histograma horizontal de blocos também visam eliminar pixels distantes da base da palavra. Assim, pixels localizados acima da linha da imagem onde ocorre o ponto de máximo do histograma são eliminados, bem como os pixels localizados abaixo da linha onde ocorre o ponto de mínimo do histograma. Na Figura 6.7, à direita da imagem da palavra *onze*, pode ser visto graficamente os pontos de máximo e mínimo do histograma de blocos horizontais. Os mínimos locais, conforme o passo 5 do algoritmo, estão assinalados em negrito na base da palavra *onze*.

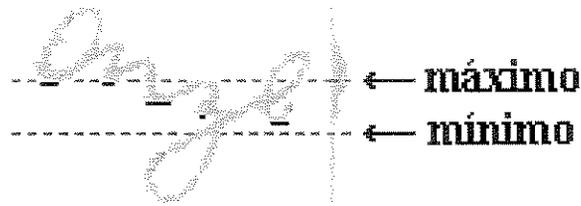


Figura 6.7 - Histograma horizontal de blocos e pontos de mínimos locais em uma imagem da palavra *onze*.

Uma outra medida visando melhorar a estimação da linha de base foi a ponderação da ordenada y_i , $i = 1, 2, \dots, N$, dos pontos de mínimo locais observando a distância em relação a ordenada média desses pontos dada por,

$$\bar{y} = \frac{1}{N} \cdot \sum_{i=1}^N y_i, \quad (6.4)$$

onde N é o número total de pontos. Esta ponderação representada por p_i , $i = 1, 2, \dots, N$, foi realizada do seguinte modo:

$$p_i = \begin{cases} \frac{\sigma}{|y_i - \bar{y}|} & \text{se } |y_i - \bar{y}| > \sigma \\ 1 & \text{se } |y_i - \bar{y}| \leq \sigma \end{cases}, \quad (6.5)$$

onde σ é o desvio padrão dos valores de ordenada dos pontos de mínimo locais. Os valores de ordenada, y_i , a serem utilizados para estimação da linha de base são dados por:

$$y'_i = p_i \cdot y_i. \quad (6.6)$$

O efeito na estimação da linha de base devido a ponderação da ordenada y_i dos mínimos locais é a redução da influência daqueles pontos mais afastados de \bar{y} , como o ponto ilustrado na parte inferior da letra “z” na palavra *dez* da Figura 6.8. Os pontos de mínimo locais da Figura 6.8 estão assinalados em negrito na base da palavra, onde passa o segmento de linha estimado para a linha de base.

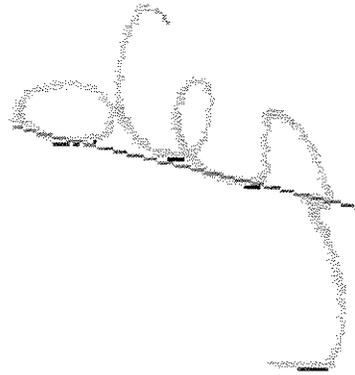


Figura 6.8 - Determinação da linha de base em uma imagem da palavra *dez*.

A rotação da imagem em relação ao plano horizontal é corrigida utilizando o ângulo de inclinação da linha de base. Melhores resultados de correção são obtidos utilizando como origem do eixo de coordenadas da imagem seu centro de gravidade. Os seguintes passos são observados na correção de rotação da imagem de uma palavra manuscrita:

1. Calcular as coordenadas do centro de gravidade da imagem (x_g, y_g) ;
2. Tomar como centro do eixo de coordenadas da imagem o ponto (x_g, y_g) ;

3. Calcular as novas coordenadas (u_i, v_i) para cada pixel preto i da imagem (Anexo B);
4. Proceder a remoção de ruídos da imagem causados pelo processo de correção de rotação.

A Figura 6.9 abaixo, apresenta o resultado da correção de rotação da imagem da palavra *dez* apresentada na Figura 6.8.



Figura 6.9 - Imagem da palavra dez após correção de rotação.

6.4 CORREÇÃO DE INCLINAÇÃO DE PALAVRAS MANUSCRITAS

A inclinação apresentada pelos traços de uma palavra é característica do estilo de escrita de cada pessoa. Quando escrevendo, uma pessoa pode produzir palavras com pouca ou acentuada inclinação para a direita ou para a esquerda. A Figura 6.10 apresenta um exemplo de uma palavra com um ângulo inclinação α para a direita.



Figura 6.10 - Palavra manuscrita inclinada para a direita.

Tanto a inclinação, como a rotação de uma palavra dificultam o processo de reconhecimento, pois geram variações de um mesmo padrão. A correção de inclinação visa normalizar uma palavra de modo que sua inclinação em relação ao eixo vertical seja zero. Alguns métodos para determinação do ângulo de inclinação de uma palavra com relação ao eixo vertical, têm sido desenvolvidos. O método descrito nesta tese é resultado da análise dos métodos propostos em [Gui95] e [BS89]. Os resultados alcançados, considerando a base de dados utilizada, são semelhantes ou algumas vezes melhores do que os obtidos quando utilizando os dois métodos referidos acima. Abaixo são descritos os procedimentos que compõem o algoritmo desenvolvido para obtenção do ângulo de inclinação de uma palavra manuscrita.

Algoritmo:

1. Em cada linha da imagem eliminar todo grupo de pixels pretos consecutivos possuindo comprimento maior que um limiar C_{\max} ;
2. Em cada coluna da imagem eliminar todo grupo de pixels pretos com altura menor do que um limiar A_{\min} ;
3. Para cada um dos segmentos remanescentes na imagem construir histogramas observando o segmento em ângulos de inclinação variando entre -70° a 70° conforme Anexo C;

4. Tornar como ângulo de inclinação de um segmento, o ângulo do histograma apresentando maior número de pixels em uma posição;
5. Calcular a média dos ângulos da inclinação dos segmentos e usar este valor como uma estimativa da inclinação média da palavra;
6. Corrigir a palavra inclinada em relação ao eixo vertical;
7. Remover ruídos da imagem gerados pelo processo de correção de inclinação.

A Figura 6.11(b) mostra em cinza os grupos de pixels eliminados da imagem da palavra *dez* na Figura 6.11(a) durante a execução do passo número 1 do algoritmo acima. Na Figura 6.11(c) são mostrados os grupos de pixels eliminados da imagem segundo o passo número 2. Os valores das variáveis C_{\max} e A_{\min} são obtidos de uma análise da imagem sendo processada. A quantidade média de pixels pretos consecutivos existentes, respectivamente, em linhas e colunas da imagem podem servir de referência para calcular C_{\max} e A_{\min} , do seguinte modo:

$$C_{\max} = Q_1 + 1; \quad (6.7)$$

$$A_{\min} = Q_2 + 1, \quad (6.8)$$

onde:

Q_1 é a quantidade média de pixels pretos consecutivos em linhas de uma imagem;

Q_2 é a quantidade média de pixels pretos consecutivos em colunas de uma imagem.

O resultado da execução dos dois primeiros passos do algoritmo é a permanência na imagem dos segmentos mais longos e aproximadamente verticais da palavra como mostra a Figura 6.11(d). A média das inclinações desses segmentos é usada como uma estimativa da inclinação média da palavra.

A utilização dos histogramas no passo número 3 permite determinar a inclinação de um segmento de acordo com a maior quantidade de pontos do segmento em uma determinada direção. Em outras palavras, o ângulo do histograma apresentando maior número de pontos em uma determinada posição é o que melhor representa a inclinação do segmento. Caso dois ou mais histogramas apresentem o maior número de pontos, o ângulo médio desses histogramas é considerado como sendo o ângulo do segmento. O Anexo C apresenta um método para calcular esses histogramas.

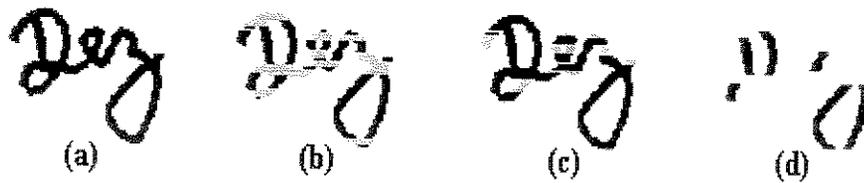


Figura 6.11 - Eliminação de grupos de pixels de uma palavra *dez*: (a) Palavra Original; (b) Eliminação de grupos de pixels em linhas da imagem; (c) Eliminação de grupo de pixels em colunas da imagem; (d) Segmentos remanescentes.

A nova posição (u, v) para um pixel da imagem sendo corrigida é dada por:

$$v = i, \quad (6.9a)$$

$$u = j - (A_{im} - i) \cdot \tan(\phi). \quad (6.9b)$$

Onde, ϕ representa a inclinação média da palavra, (i, j) as coordenadas da posição de um pixel da imagem antes do processo de correção de inclinação e A_m a altura da imagem. A Figura 6.12 mostra o resultado da correção de inclinação da palavra da Figura 6.11(a).

Figura 6.12 - Palavra dez com a inclinação corrigida.

6.5 COMENTÁRIOS FINAIS DO CAPÍTULO

No presente capítulo foram discutidas as técnicas de pré-processamento de imagens utilizadas no trabalho descrito nesta tese de doutorado. Tais técnicas visam reduzir deformações em imagens de palavras manuscritas introduzidas pelo processo de escrita e de digitalização. A redução ou eliminação dessas deformações tem por finalidade facilitar a extração de características da imagem e minimizar as variações entre amostras de uma mesma palavra. O próximo capítulo desta tese é dedicado a descrição do processo de segmentação de palavras manuscritas em caracteres. Essa segmentação tem por objetivo facilitar a extração de características dos caracteres em uma palavra, e também permitir a construção de modelos de letras para posterior formação dos modelos de palavras.

CAPÍTULO VII

SEGMENTAÇÃO DE PALAVRAS MANUSCRITAS

Considerando uma palavra manuscrita como sendo uma seqüência de caracteres, poder-se-ia concluir que segmentar uma palavra em caracteres e após classificá-los, seria a melhor estratégia ao se desenvolver um sistema para reconhecimento de palavras. No entanto, as variações na forma dos caracteres e a sobreposição de caracteres impedem a segmentação de uma palavra em caracteres perfeitos. Em conseqüência, segmentos resultantes do processo de segmentação podem representar caracteres completos, parte de caracteres ou caracteres ainda conectados. Como a segmentação de uma palavra manuscrita em caracteres não é um processo determinístico, há dificuldades em se obter segmentos que correspondam ao conjunto de símbolos que o sistema está apto a reconhecer. Algumas vezes, mesmo uma ótima segmentação e uma correta classificação dos segmentos segundo o conjunto de símbolos utilizados, não são suficientes para proporcionar um reconhecimento confiável. A ocorrência de ambigüidades entre caracteres pode causar erros no processo de classificação de uma palavra. Uma amostra de ambigüidade entre caracteres é ilustrado na Figura 7.1, onde o segmento apresentado pode ser identificado tanto como sendo a letra “a” ou como sendo a seqüência de caracteres conectados “i” e “z”. É necessário, portanto, uma análise de contexto

para validar o resultado do processo de classificação. Deve-se, por exemplo, verificar a existência da palavra no vocabulário utilizado.



Figura 7.1 - Possível ambigüidade entre a letra “a” e a seqüência de letras conectadas “i” e “j”.

Em resumo, pode-se dizer, analisando as operações realizadas ao segmentar-se uma palavra, que dois tipos de decisões devem ser tomadas: (1) uma decisão local, ao se observar a forma dos segmentos e os tipos de símbolos utilizados pelo sistema, e (2) uma decisão global ao validar-se o resultado de acordo com uma análise do contexto.

Nas seções seguintes são comentadas algumas estratégias de segmentação e descrito o algoritmo desenvolvido para este trabalho para efetuar a segmentação de palavras manuscritas.

7.1 ALGUMAS ESTRATÉGIAS DE SEGMENTAÇÃO

Em um processo de reconhecimento de palavras pode-se decidir por segmentar ou não as palavras a serem processadas. Em geral, essa decisão é tomada tendo-se em vista o tamanho do vocabulário sendo considerado. Em vocabulários pequenos, devido ao número reduzido de palavras, pode-se obter características que proporcionam uma adequada discriminação entre classes de palavras sem a necessidade de segmentar a palavra manuscrita em reconhecimento. Porém, quando o vocabulário é grande, os atributos extraídos da palavra podem não ser

capazes de proporcionar uma distinção satisfatória entre todas as classes consideradas. Nesse caso, a segmentação da palavra e extração de características dos segmentos podem levar a uma classificação com melhores resultados.

Com base nos métodos de segmentação estudados e de acordo com o trabalho desenvolvido em [CL96], foram consideradas duas estratégias para segmentar palavras manuscritas, nominalmente: (1) segmentação explícita e (2) segmentação implícita ou baseada em reconhecimento de caracteres. Os demais métodos podem ser vistos como combinações dessas duas estratégias de segmentação.

A primeira estratégia é baseada em uma análise da imagem da palavra, visando determinar seus pontos potenciais de segmentação. São analisadas propriedades tais como: *loops*, altura de traço e largura de traço da palavra. Exemplos dessa estratégia podem ser encontrados em [Chen95], [BS89], [KG97] e na seção 7.2 desta tese. As regiões assinaladas em cinza na Figura 7.2 indicam os locais mais prováveis de ocorrência de pontos potenciais de segmentação em um exemplo da palavra *dígitos*.



Figura 7.2 - Pontos potenciais de segmentação da palavra *dígitos*.

Na segunda estratégia, denominada segmentação baseada em reconhecimento de caracteres, a palavra a ser segmentada é dividida em vários retângulos, não havendo preocupação em se determinar pontos potenciais de segmentação. Os traços da palavra em

UNICAMP

BIBLIOTECA CURTZA

cada um desses retângulos são associados a uma letra do alfabeto, procurando-se obter um resultado de segmentação ou reconhecimento coerente. O procedimento básico é variar a largura de cada retângulo e avaliar através de um reconhecimento de caracteres os segmentos da palavra nesses retângulos. A Figura 7.3 mostra a palavra *dígitos* dividida em retângulos de diferentes larguras. Estes retângulos tiveram suas larguras ajustadas de forma a maximizar o resultado da avaliação efetuada no reconhecimento de caracteres. Uma variação desta estratégia de segmentação pode ser encontrada em [Sen94], onde a palavra a ser processada é dividida em retângulos de uma mesma largura e uma rede neural é treinada para associar a cada retângulo uma letra do alfabeto.

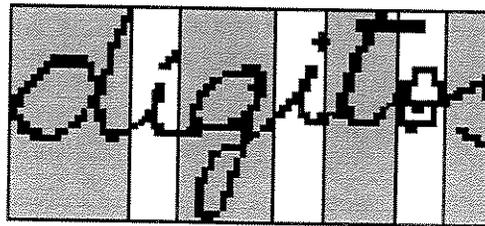


Figura 7.3 - Divisão de uma palavra em retângulos.

No sistema de reconhecimento proposto nesta tese é utilizada a primeira estratégia de segmentação, ou seja, procura-se determinar os pontos potenciais de segmentação da palavra.

7.2 DESCRIÇÃO DO MÉTODO DE SEGMENTAÇÃO

Neste trabalho o procedimento de segmentação foi desenvolvido com base nos métodos propostos em [KG97], [Chen95], [FNK92] e consiste em determinar os pontos potenciais de segmentação de uma palavra. Em um primeiro instante, são obtidas as prováveis

ligações entre caracteres de uma palavra. A Figura 7.4 apresenta um exemplo onde as ligações entre caracteres da palavra *dígitos* estão assinaladas em cinza.

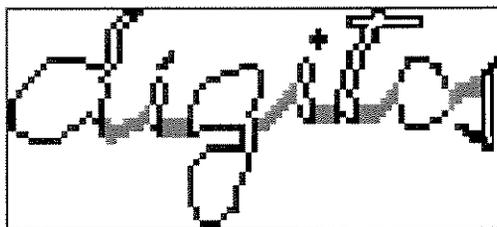


Figura 7.4 - Ligações da palavra *dígitos* assinaladas em cinza.

É assumido que os pontos potenciais de segmentação de uma palavra estão localizados nas prováveis ligações entre caracteres. A segmentação de uma palavra nesses pontos gera segmentos que podem representar um caractere, parte de um caractere ou caracteres ainda conectados. O algoritmo de segmentação desenvolvido neste trabalho pode ser dividido em três procedimentos básicos enumerados:

1. Determinação das prováveis ligações de uma palavra;
2. Validação das ligações;
3. Segmentação da palavra em pontos localizados nas ligações validadas.

Para determinação das prováveis ligações entre caracteres de uma palavra foi empregada a operação da morfologia matemática denominada abertura binária (Anexo E). Para validar as ligações a serem utilizadas na segmentação de uma palavra foram avaliados os traços que formam o contorno da palavra, mais especificamente, o contorno situado na base da palavra manuscrita denominado contorno inferior. Os pontos potenciais de segmentação foram determinados através de uma transformação *hit or miss* da morfologia matemática. A

seguir, são explicados os detalhes dos três procedimentos básicos do algoritmo de segmentação desenvolvido.

7.2.1 DETERMINAÇÃO DAS PROVÁVEIS LIGAÇÕES DE UMA PALAVRA

Tomando como exemplo a Figura 7.4, pode-se observar que as prováveis ligações entre caracteres de uma palavra correspondem, em geral, aquelas regiões da palavra contendo traços de menor altura (regiões sombreadas em cinza da Figura 7.4). Um meio para determinar tais regiões é o emprego da operação de abertura binária apresentada no Anexo E desta tese. Essa operação permite eliminar da imagem de uma palavra os traços cuja a altura é menor do que aquela apresentada pelo elemento estruturante (Anexo E) utilizado. A Figura 7.5(b) apresenta o resultado de uma abertura binária aplicada a Figura 7.5(a).

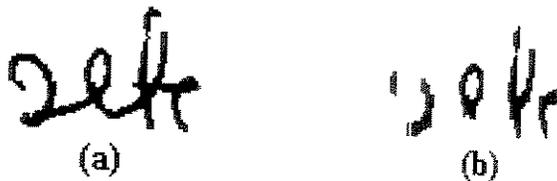


Figura 7.5 - Operação de abertura binária (a) Imagem Original;
(b) Imagem após efetuada a abertura binária.

É importante observar que, antes de efetuar a abertura da imagem de uma palavra, devem ser preenchidos os *loops* apresentados em caracteres tais como: “a”, “b”, “d”, “e”, “g”, “o”, e “p”. Este procedimento evita a segmentação destes caracteres em seus *loops*. A Figura 7.6 mostra o preenchimento do *loop* ocorrendo na letra “e” da palavra *sete* apresentada na Figura 7.5(a).



Figura 7.6 - Resultado do preenchimento de *loops* existentes na palavra sete.

O elemento estruturante utilizado na operação de abertura é uma linha vertical com altura A_{ee} estimada a partir de A_{mt} que representa a altura média de traço da palavra em processamento. As alturas A_{ee} e A_{mt} são determinadas pelo seguinte algoritmo [KG97]:

Algoritmo

1. $A_{mt} = p_{f \max}$;

2. Para cada valor de p e $f(p)$:

$$\text{Se } (p < p_{\max}) \text{ e } \left(f(p) > \frac{f(p_{f \max})}{2} \right) \text{ então, } A_{mt} = p;$$

3. $A_{ee} = A_{mt} + 1$.

Onde:

p representa o número de pixels existentes em um grupo de pixels pretos consecutivos de uma coluna da imagem;

p_{\max} é o maior valor assumido por p na imagem sendo processada;

$f(p)$ é a frequência com que cada valor de p aparece na imagem;

$p_{f \max}$ representa o valor de p de maior frequência.

O presente algoritmo determina A_m examinando valores de p situados entre p_{\max} e p com frequência igual a metade da maior frequência encontrada [KG97]. A Figura 7.7, apresentando a imagem de uma letra "F", é utilizada para exemplificar como os valores de p são obtidos. Na parte superior da referida Figura estão numeradas as 9 colunas da imagem, sendo que cada uma das colunas 2, 3, 6, 7 e 8 apresentam apenas um grupo de pixels pretos consecutivos, enquanto as colunas 4 e 5 possuem cada uma 2 grupos. Nas colunas 2 e 3 o valor assumido por p é 8. Em cada uma das colunas 4 e 5, p assume o valor 2 e também o valor 1. Nas colunas 6, 7 e 8 p apresenta um valor igual a 2. Os valores obtidos para p_{\max} e $p_{f\max}$ considerando a imagem na Figura 7.7 são, respectivamente, 8 e 2. Neste caso, aplicando o algoritmo descrito acima, a altura média de traço A_m para a letra da Figura 7.7 será 2 e A_{ee} , denominada a altura do elemento estruturante, terá valor 3.

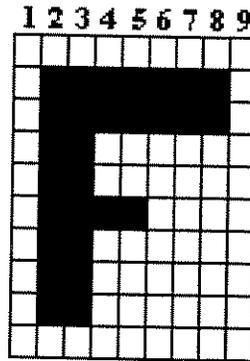


Figura 7.7 - Imagem de uma Letra "F" com colunas numeradas na sua parte superior.

As prováveis ligações entre caracteres são obtidas subtraindo a imagem original da imagem resultante do processo de abertura. A Figura 7.8 ilustra em negrito as ligações resultantes da subtração das Figuras 7.5(a) e 7.5(b).

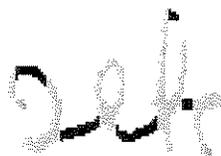


Figura 7.8 - Prováveis ligações da palavra *sete* assinaladas em negrito.

7.2.2 VALIDAÇÃO DAS LIGAÇÕES ENTRE CARACTERES

Nem todas as prováveis ligações obtidas de acordo com o procedimento descrito na seção 7.2.1 são utilizadas para segmentar uma palavra manuscrita. Duas medidas são adotadas para determinar as ligações válidas ao processo de segmentação. A primeira medida considera como sendo válidas somente aquelas ligações que estabelecem conexões entre segmentos da palavra segmentada. Isto é, prováveis ligações que não conectam um segmento a outro da palavra não são consideradas. A segunda medida é heurística, e estabelece que somente são usadas as ligações possuindo pixels pertencentes ao contorno inferior da palavra a ser segmentada. Essa heurística foi obtida de uma análise de segmentações realizadas com várias palavras manuscritas. A Figura 7.9(a) mostra em um tom de cinza mais escuro as prováveis ligações da palavra *sete* e em negrito seu contorno inferior. A Figura 7.9(b) mostra em negrito as ligações da palavra da Figura 7.9(a) que possuem pixels pertencentes ao contorno inferior da palavra. Note, por exemplo, que a ligação existente na parte superior do primeiro caractere na Figura 7.9(a) não foi considerada, pois não possui pixels pertencentes ao contorno inferior da palavra.

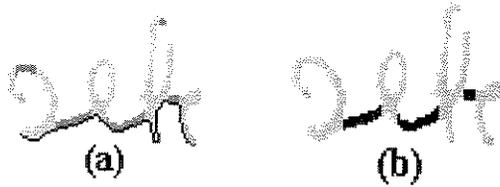


Figura 7.9 - Ligações válidas de uma palavra *sete*. (a) Prováveis ligações e contorno inferior; (b) Ligações com pixels no contorno inferior.

Na execução da primeira medida de validação de ligações a imagem resultante do processo de abertura binária sofre uma dilatação condicionada a sua imagem original. O processo de dilatação condicionada é apresentado no Anexo E desta tese. Os seguintes passos são utilizados na execução dessa medida:

1. Dilatação condicionada da imagem para a esquerda;
2. Dilatação condicionada da imagem para a direita;
3. Interseção das imagens resultantes dos passos 1 e 2.

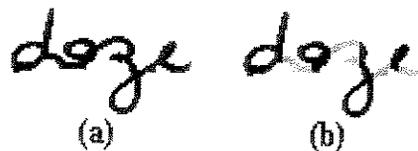


Figura 7.10 - Imagem de uma palavra *doze*: (a) Imagem original; (b) Representação da abertura binária em negrito e ligações entre caracteres em cinza.

O resultado da operação descrita pelos passos 1, 2 e 3, é a preservação das ligações conectando segmentos da palavra. Prováveis ligações conectadas a apenas um segmento da imagem desaparecem quando efetuado o passo 3 da operação. A Figura 7.10(b) ilustra em negrito o resultado da abertura binária da imagem da palavra *doze* na Figura 7.10(a) e em cinza

as prováveis ligações entre caracteres da palavra. A Figura 7.11(a) e 7.11(b) mostram, respectivamente, o resultado do passo 1 e 2 da operação descrita acima quando aplicados a imagem em 7.10(b). A imagem da Figura 7.11(c) é resultante da interseção das imagens da Figura 7.11(a) e 7.11(b). Pode ser observado que a provável ligação inferior no caractere “z” e a provável ligação no final do caractere “e” desapareceram em razão de não conectarem dois segmentos da palavra.

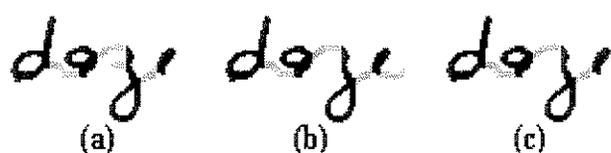


Figura 7.11 - Avaliação das ligações de uma palavra *doze*: (a) Resultado da dilatação condicionada para a esquerda; (b) Resultado da dilatação condicionada para a direita; (c) Interseção das imagens em (a) e (b).

Para determinação do contorno inferior da imagem de uma palavra manuscrita, foi desenvolvido o seguinte algoritmo.

Algoritmo

1. Assinalar o último pixel preto de cada coluna da imagem da palavra a ser processada;
2. Determinar os mínimos locais [Ser82] da configuração resultante dos pixels assinalados no item anterior;
3. Utilizar uma distância geodésica (Anexo E) para estabelecer o caminho de menor distância entre dois mínimos locais consecutivos;
4. O resultado final do item 3 determina o contorno inferior da palavra, o qual é constituído pelos caminhos conectando mínimos locais consecutivos.

7.2.3 SEGMENTAÇÃO DA PALAVRA NOS PONTOS POTENCIAIS DE SEGMENTAÇÃO

A palavra segmentada é encontrada eliminando-se da palavra original as ligações válidas obtidas entre caracteres. Por exemplo, eliminando da Figura 7.9(b) as ligações assinaladas em negrito, obtém-se a palavra segmentada ilustrada pela Figura 7.12. Porém, a simples eliminação dessas ligações, como ilustra a Figura 7.12, pode provocar o desaparecimento de traços que contêm informações úteis para o reconhecimento da palavra. Assim, é preciso determinar em cada ligação válida um ponto potencial de segmentação da palavra.

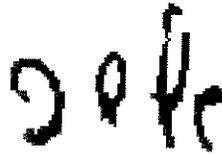


Figura 7.12 - Imagem de uma palavra *sete* após serem eliminadas as ligações entre caracteres.

Neste trabalho os pontos potenciais de segmentação da palavra não foram determinados diretamente nas ligações válidas, mas através de um espessamento (Anexo E) da imagem após eliminadas suas ligações. A imagem é espessada até se ter um espaçamento de um pixel entre os segmentos da palavra. Após o espessamento, é realizada uma interseção da imagem espessada com a imagem original da palavra sendo processada, obtendo o resultado final do processo de segmentação. Para realizar o espessamento foi utilizada uma transformação *bit or miss* da morfologia matemática, apresentada no Anexo E desta tese. A Figura 7.13(a) apresenta o espessamento da imagem da Figura 7.12 e a Figura 7.13(b) apresenta o resultado final da interseção da imagem espessada com a imagem original (Figura 7.5(a)).



Figura 7.13 - Obtendo a palavra segmentada. (a) Uma palavra *sete* espessada; (b) Palavra segmentada.

7.3 PROCESSAMENTO FINAL DA PALAVRA SEGMENTADA

Após o processo de segmentação de uma palavra podem ocorrer segmentos apresentando caracteres ainda conectados que podem ser segmentados. Visando desconectar tais caracteres foi aplicada a metodologia desenvolvida em [Chen95] e [KG97], a qual consiste em encontrar concavidades em segmentos da palavra segmentada. Quando uma concavidade superior for localizada diretamente acima de uma concavidade inferior, o segmento é separado naquela região. A Figura 7.14 ilustra tal procedimento.

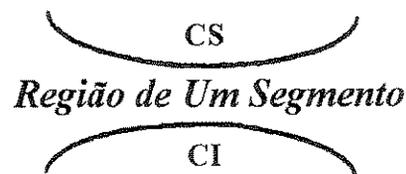


Figura 7.14 - Possível região de segmentação determinada por uma Concavidade Superior (CS) e uma Concavidade Inferior (CI) [Chen95].

A Figura 7.15(a) apresenta um exemplo de segmentação de caracteres conectados utilizando concavidades superiores e inferiores.



Figura 7.15 - Seqüência de caracteres “s” e “e”: (a) concavidades assinaladas em cinza; (b) seqüência de caracteres segmentados.

7.4 COMENTÁRIOS SOBRE O PROCESSO DE SEGMENTAÇÃO

O método de segmentação apresentado neste capítulo foi testado em várias imagens de palavras manuscritas, tendo apresentado resultados satisfatórios. O algoritmo não efetua a segmentação de caracteres sobrepostos ou de caracteres apresentando conexões como a da seqüência de letras “t” e “o” na imagem da palavra *oito* da Figura 7.16, pois não é possível a identificação de uma ligação entre caracteres.



Figura 7.16 - Imagem de uma palavra *oito* em que o algoritmo de segmentação não é capaz de identificar uma ligação entre os caracteres “t” e “o”.

Letras como “u”, “m” ou “n” podem ser segmentadas em virtude da utilização da abertura binária para determinação das ligações entre caracteres de uma palavra, como ilustrado pela imagem da palavra *um* na Figura 7.17. No entanto, esse problema de segmentação pode ser tratado durante o processo de classificação com a utilização de alguma informação contextual.



Figura 7.17 - Amostra de uma palavra um onde ocorre a segmentação dos caracteres “u” e “m”.

Outro comentário sobre o método de segmentação desenvolvido diz respeito a utilização da operação de espessamento para determinar os pontos potenciais de segmentação de uma palavra manuscrita. O espessamento produz segmentos com contornos mais adequados a classificação de uma palavra manuscrita. Isso pode ser observado na Figura 7.18 mostrando a segmentação da palavra *for* do vocabulário inglês. A Figura 7.18(b) mostra o processo de segmentação da palavra no ponto médio das ligações entre caracteres representadas em cinza na Figura 7.18(a). Na Figura 7.18(c) a operação de espessamento é utilizada para efetuar a segmentação da palavra.

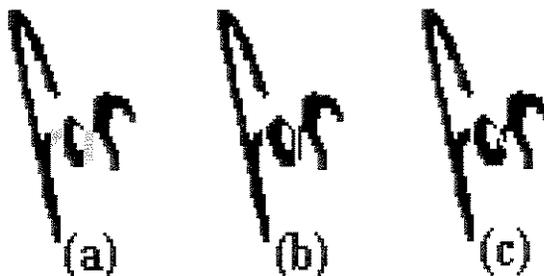


Figura 7.18 - Segmentação da palavra *for* do vocabulário inglês: (a) Ligações da palavra assinaladas em cinza. (b) Segmentação no ponto médio das ligações entre caracteres. (c) Segmentação utilizando a operação morfológica de espessamento.

A seguir, no Capítulo VIII, é demonstrada a forma como são extraídas as características de cada segmento resultante do processo de segmentação de uma palavra manuscrita. Também é explicada a formulação dos vetores de características utilizados pelo sistema para representar uma palavra manuscrita

CAPÍTULO VIII

EXTRAÇÃO DE CARACTERÍSTICAS

As características utilizadas neste trabalho têm como base pesquisas tratando da natureza de processos de reconhecimento realizados por seres humanos. Em alguns de seus trabalhos, Senior [Sen94] tem sugerido linhas retas, curvas e *loops* como, possivelmente, as principais características observadas por um leitor quando reconhecendo uma palavra. Pedrycz [Ped90] descreveu o reconhecimento realizado por um ser humano como um processo em que a maioria da informação utilizada é de caráter não numérico. Desse modo, uma pessoa ao realizar um reconhecimento, em geral, não teria como parâmetros valores numéricos, mas relações entre classes e rótulos atribuídos ao objeto sob investigação. Com vista nesses aspectos, é razoável pensar o reconhecimento de um caractere como resultado de uma relação entre classes de caracteres e rótulos atribuídos aos segmentos de linha formando o caractere. Dessa forma, um caractere seria associado a uma dada classe do alfabeto se existir alguma relação entre os rótulos de seus segmentos e a referida classe. Por exemplo, poder-se-ia classificar um caractere desconhecido como sendo a letra “A”, como ilustra a Figura 8.1, se este fosse composto por: (1) um segmento de linha reta inclinada à direita; (2) um segmento de linha reta horizontal; (3) um segmento de linha reta inclinada à esquerda.



Figura 8.1 - Segmentos de linha compondo um caractere manuscrito "A".

Apesar de parecerem adequadas à classificação de um caractere, as características acima são bastantes significativas ao reconhecimento realizado por um ser humano, mas muito vagas em se tratando de reconhecimento de manuscritos através de um computador. Na implementação de um algoritmo computacional, em razão das variações inerentes aos caracteres manuscritos, é difícil decidir se um segmento de linha é uma reta ou uma curva, se o segmento é vertical ou está inclinado para a direita, etc.. A falta de precisão em realizar tal decisão pode ser tratada através da teoria de conjuntos *fuzzy* [Zad65], [Ger88]. Nas próximas seções deste capítulo é mostrado como segmentos de linha reta, curvas e *loops* são utilizados para caracterizar um caractere manuscrito e como conjuntos *fuzzy* são utilizados para classificar estes segmentos.

8.1 NOÇÕES A RESPEITO DE CONJUNTOS *FUZZY*

A noção de conjunto *fuzzy* foi introduzida por Zadeh [Zad65] como um meio de tratar conjuntos cujas fronteiras não são bem definidas. Na teoria clássica a relação de pertinência entre o conjunto e seus elementos é clara, ou seja, um elemento está completamente dentro do conjunto ou ele não está no conjunto. Em termos matemáticos, se C é um conjunto em determinado espaço e se x é um elemento deste espaço, é dito que $x \in C$ ou $x \notin C$. Um

exemplo é o conjunto dos números inteiros, onde sabe-se precisamente que o número 2 pertence ao referido conjunto e o número 0,5 não pertence. No entanto, há situações em que a relação de pertinência entre conjunto e elemento não é precisa, isto é, não se sabe efetivamente se um elemento pertence ou não a determinado conjunto. Por exemplo, um conjunto A definido como o conjunto cujos elementos são alturas de pessoas em torno de 1,70 metros. Este conjunto pode ser especificado segundo a expressão:

$$A = \{x \mid x \text{ esteja em torno de } 1,70 \text{ m}\}. \quad (8.1)$$

Observando a definição do conjunto A pode-se questionar se alturas como 1,68m e 1,85m pertencem ao referido conjunto. A resposta a esta pergunta é incerta, pois não se sabe até que ponto uma determinada altura é considerada estar em torno de 1,70m. Porém, pode-se associar valores numéricos às alturas 1,68m e 1,85m de forma a demonstrar uma compatibilidade com o conceito expresso pelo conjunto A . Assim, ao invés de se dizer que $1,68\text{m} \in A$ e $1,85\text{m} \notin A$ atribui-se em relação ao conjunto A , por exemplo, um grau de pertinência de 0,8 para altura de 1,68m e 0,4 para a altura 1,85m. Em conjuntos *fuzzy* o grau para qual um elemento é dito ser um membro de um conjunto é representado por um número no intervalo $[0, 1]$. Em conjuntos tradicionais pode-se dizer que existe apenas o grau de pertinência 1 quando os elementos pertencem ao conjunto e o grau 0 para elementos não pertencentes ao conjunto.

Formalmente, um conjunto *fuzzy* pode ser definido [PG98] como um conjunto caracterizado por uma função de pertinência mapeando elementos de um domínio, espaço ou universo de discurso X para o intervalo unitário $[0,1]$. Considerando um conjunto *fuzzy* A tem-se:

$$A: X \rightarrow [0,1]. \quad (8.2)$$

Assim, um conjunto *fuzzy* A em X pode ser representado [PG98] através de um conjunto de pares ordenados de um elemento genérico $x \in X$ e seu grau de pertinência, ou seja, $A = \{A(x)/x \mid x \in X\}$. O valor de $A(x)$ descreve o grau de pertinência de x em A .

8.2 CARACTERÍSTICAS PARA UMA PALAVRA MANUSCRITA

As características utilizadas neste trabalho estão relacionadas à forma e posição dos traços em caracteres de uma palavra manuscrita. Especificamente, essas características representam uma avaliação da similaridade dos traços de um caractere a tipos de segmentos de linha retas e curvas, observada a posição segundo uma região do caractere. Em razão de variações na escrita de uma palavra manuscrita, os traços de um caractere não podem ser caracterizados precisamente como uma reta ou uma curva, sendo portanto utilizada a teoria de conjuntos *fuzzy* para tratar tal falta de precisão. Para cada traço do caractere são atribuídos graus de pertinência a conjuntos *fuzzy* representando tipos de retas e curvas. São empregados dez (10) conjuntos *fuzzy* cada um representando, respectivamente:

- (1) Retas horizontais (H);
- (2) Retas verticais (V);
- (3) Retas com inclinação positiva (P);
- (4) Retas com inclinação negativa (N);
- (5) Curvas tipo C;

- (6) Curvas tipo D;
- (7) Curvas tipo A;
- (8) Curvas tipo U;
- (9) Curvas tipo S;
- (10) Curvas tipo Z.

Os *loops* encontrados em caracteres de uma palavra manuscrita, são também utilizados como característica, sendo simbolizados nesta tese pela consoante L. A característica de *loops* não é representada por um conjunto *fuzzy*. Um traço somente é considerado como sendo um *loop* se formar uma circunferência fechada, nesse caso recebe o valor 1, caso contrário recebe o valor 0. Dessa maneira, incluindo a característica de *loops*, pode-se representar os traços de um caractere de acordo com 11 tipos de segmentos, sendo dez desses segmentos representados por conjuntos *fuzzy*, como enumerado no parágrafo anterior. A Figura 8.2 ilustra os 11 tipos de segmentos de linha empregados para caracterizar um caractere.

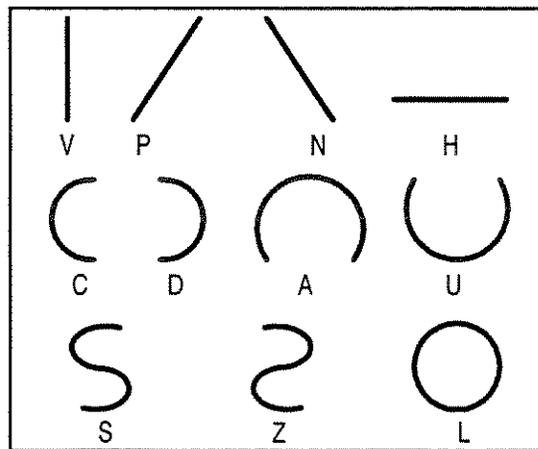


Figura 8.2 - Segmentos de linha para caracterizar caracteres.

8.3 RAMOS E PONTOS SINGULARES DE UM SEGMENTO

Tendo-se em vista facilitar a descrição do processo de extração de características, são estabelecidas nesta seção algumas nomenclaturas. Mencionou-se nos capítulos anteriores que a segmentação de uma palavra em caracteres pode não ser exata, implicando na ocorrência de segmentos representando caracteres que não estão completos (frações de caracteres) ou segmentos representando caracteres ainda conectados como indicado por setas na Figura 8.3. Como consequência, partes da palavra resultantes do processo de segmentação podem não ser caracteres e, por esse motivo, são genericamente denominadas de frações da palavra.

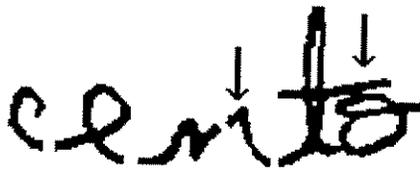


Figura 8.3 – Segmentação da palavra manuscrita *cento* apresentando uma fração de um caractere “n” e os caracteres “c” e “o” não separados.

Para efetuar a extração de características, cada fração de uma palavra é afinada e decomposta em segmentos de linha chamados ramos. O procedimento de afinamento (Anexo F), transforma a imagem da fração de uma palavra em uma imagem com espessura de traço de aproximadamente um pixel, permitindo a extração dos ramos. A cada ramo são atribuídos valores numéricos expressando pertinências a conjuntos *fuzzy* representativos de cada elemento da Figura 8.2. Tais valores de pertinência representam graus de similaridade do ramo analisado a cada um dos elementos da Figura 8.2.

Para proceder a extração dos ramos são assinalados pontos singulares em cada fração afinada da palavra segmentada. Esses pontos apresentam uma configuração especial de vizinhança na imagem sendo analisada, constituindo os nós da referida fração. Os pontos

singulares utilizados neste trabalho, considerando uma imagem em preto e branco, podem ser enumerados da seguinte maneira:

- 1) **Ponto final:** Pixel preto com apenas um outro pixel preto na sua vizinhança;
- 2) **Ponto de interseção:** Pixel preto com mais de 2 pixels pretos na sua vizinhança;
- 3) **Ponto de inflexão:** Pixel preto com apenas 2 pixels pretos na sua vizinhança e onde ocorre uma mudança abrupta da direção dos pixels do ramo [BK88].

Na Figura 8.4. é ilustrada uma letra manuscrita “O” onde são apresentados pontos finais, pontos de interseção e pontos de inflexão indicados, respectivamente, pelos números 1, 2 e 3. Esses pontos são analisados considerando uma vizinhança 8 conectada [GW92].

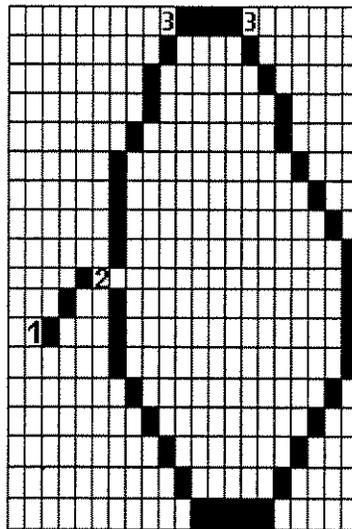


Figura 8.4 - Pontos singulares em uma letra manuscrita “O”.

Um ramo pode ser então definido como o segmento de linha situado entre dois pontos singulares (dois nós) de uma fração da palavra. Os números na Figura 8.4 caracterizam pontos singulares em uma letra “O” resultante do processo de segmentação de uma palavra.

8.4 POSIÇÕES RELATIVAS DE UM RAMO

Cada uma das letras do alfabeto pode ser representada segundo o modo como são conectados um ou mais dos segmentos de linha apresentados na Figura 8.2. Desta forma, é importante que cada ramo seja avaliado no que diz respeito a sua posição relativa na fração da palavra em que se encontra. Para uma imagem sendo analisada de cima para baixo, um ramo pode estar localizado nas seguintes regiões em relação à fração da palavra: (1) no topo; (2) no meio e (3) na base. Varrendo a imagem da esquerda para a direita pode-se localizar um ramo em relação a fração da palavra analisada: (4) à esquerda; (5) no centro e (6) à direita. Na Figura 8.5, analisada de cima para baixo e da esquerda para a direita, são ilustradas algumas das posições relativas dos ramos da fração de uma palavra, nesse caso, a letra “d”. O ramo entre os nós 1 e 2, por exemplo, situa-se aproximadamente no topo e à direita da letra em questão; O ramo representado pela curva cujos pontos finais são os nós 2 e 3 situa-se aproximadamente no meio e à esquerda da fração da palavra analisada; O segmento de reta entre os nós 2 e 3 está localizado aproximadamente no meio e no centro da referida fração da palavra; Finalizando, o ramo localizado entre os nós 3 e 4 pode ser localizado na base e à direita da letra ilustrada na Figura 8.5.

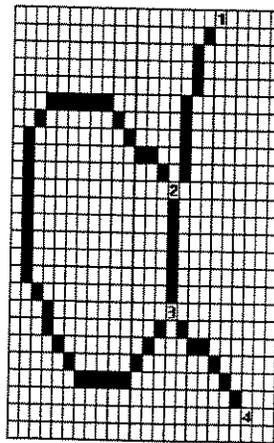


Figura 8.5 - Posições relativas para ramos de uma letra manuscrita “d”.

A posição relativa de um ramo em amostras de uma mesma classe varia segundo o estilo de escrita de cada pessoa. Pode-se verificar tal fato nas duas amostras da letra “t” na Figura 8.6. Analisando a Figura 8.6(a) de cima para baixo verifica-se que o traço horizontal superior está localizado aproximadamente no meio da letra “t”, enquanto na Figura 8.6(b) esse traço se encontra no topo da letra.



Figura 8.6 - Duas amostras de uma letra manuscrita “t”.

Os conjuntos *fuzzy* podem ser empregados para considerar variações na posição dos traços de caracteres de uma palavra manuscrita. Pode-se construir funções para representar a pertinência de traços de um caractere a cada uma das posições relativas já enumeradas. Para definir essas funções de pertinência a seguinte nomenclatura é utilizada:

- prd* = Pertinência do ramo com relação ao lado direito da fração da palavra;
- pre* = Pertinência do ramo com relação ao lado esquerdo da fração da palavra;
- prc* = Pertinência do ramo com relação ao centro da fração da palavra;
- prt* = Pertinência do ramo com relação ao topo da fração da palavra;
- prb* = Pertinência do ramo com relação a base da fração da palavra;
- prm* = Pertinência do ramo com relação ao meio da fração da palavra;
- npse* = Número de pixels pretos da fração da palavra à esquerda do ramo;

$npsd$ = Número de pixels pretos da fração da palavra à direita do ramo;

$npsa_c$ = Número de pixels pretos da fração da palavra acima do ramo;

$npsa_b$ = Número de pixels pretos da fração da palavra abaixo do ramo.

Os valores de pertinência de um ramo referentes a sua posição na fração da palavra são dados pelas seguintes equações:

$$prd = \frac{npse}{npse + npsd}; \quad (8.3)$$

$$pre = \frac{npsd}{npse + npsd}; \quad (8.4)$$

$$prc = 1 - \frac{|npse - npsd|}{npse + npsd}; \quad (8.5)$$

$$prt = \frac{npsa_b}{npsa_c + npsa_b}; \quad (8.6)$$

$$prb = \frac{npsa_c}{npsa_c + npsa_b}; \quad (8.7)$$

$$prm = 1 - \frac{|npsa_c - npsa_b|}{npsa_c + npsa_b}; \quad (8.8)$$

Os valores obtidos pelas equações acima expressam a pertinência de um ramo a cada uma das posições relativas a fração da palavra sendo analisada. Essas pertinências são determinadas da relação de posição existente entre os pixels pretos (imagens em preto e branco) formando a fração da palavra e os pixels pretos formando o ramo. Por exemplo, um ramo apresenta um alto valor de pertinência em relação a posição à esquerda da fração da palavra se a maioria dos pixels pretos da referida fração estiverem localizados à direita do ramo. Ocorrerá um alto valor de pertinência com relação a posição central se a quantidade de pixels pretos da fração da palavra situados à esquerda e à direita do ramo forem aproximadamente iguais. O mesmo raciocínio é empregado para as demais posições relativas a fração da palavra segmentada.

8.5 DETERMINAÇÃO DOS VALORES DE PERTINÊNCIA DE UM RAMO

Para cada ramo de um segmento é calculado um valor de pertinência representando sua similaridade a cada um dos elementos ilustrados na Figura 8.2. Essas pertinências assumem valores no intervalo $[0,1]$ e são calculadas empregando a teoria de conjuntos *fuzzy* e alguns dos métodos utilizados em [SC74].

8.5.1 SIMILARIDADE A UMA LINHA RETA E A UMA LINHA CURVA

Para determinar o grau de similaridade entre os símbolos ilustrados na Figura 8.2 e cada um dos ramos constituintes de uma fração da palavra segmentada é necessário, primeiramente, determinar a similaridade desses ramos em relação a uma linha reta e a uma linha curva. Tal similaridade é expressa numericamente a partir do segmento de reta que mais se aproxima do

ramo sendo processado, empregando o método dos mínimos quadrados [BK88]. São estimadas duas equações para o segmento de reta aproximando o ramo em questão. Considerando coordenadas cartesianas, a primeira equação utiliza como valor para a variável independente a abscissa das coordenadas de cada ponto do ramo, enquanto que a segunda equação utiliza para a variável independente a ordenada. A proximidade entre os pontos do ramo e a reta estimada é usada como uma avaliação da similaridade entre ambos. Quanto mais próximos os pontos do ramo estiverem da reta estimada, maior a semelhança entre ramo e reta. Uma vez que esta avaliação é subjetiva e baseada na experiência obtida pela análise de vários ramos, optou-se pela função de pertinência ilustrada na Figura 8.7 para estabelecer um valor de avaliação. O grau de similaridade (Gr) entre reta e ramo é então dado por:

$$Gr = \min \left\{ \frac{1}{N} \sum_{i=1}^N f(d_y(i)), \frac{1}{N} \sum_{i=1}^N f(d_x(i)) \right\}, \quad (8.9)$$

onde:

- $f(\cdot)$ é a função de pertinência ilustrada na Figura 8.7;
- $d_y(i)$ é a diferença entre as ordenadas do i -ésimo ponto do ramo e do i -ésimo ponto da reta estimada pela equação que utiliza a abscissa de cada ponto do ramo como valor para a variável independente;
- $d_x(i)$ é a diferença entre a abscissa do i -ésimo ponto do ramo e a abscissa do i -ésimo ponto da reta estimada pela equação que utiliza como valor para a variável independente a ordenada de cada ponto do ramo;
- N denota o total de pontos (pixels pretos) pertencentes ao ramo sendo processado;

– \min é a operação para determinar o mínimo entre dois ou mais valores.

O grau de similaridade (G_c) entre um ramo e uma linha curva pode ser obtido a partir de G_r fazendo-se:

$$G_c = 1 - G_r. \quad (8.10)$$

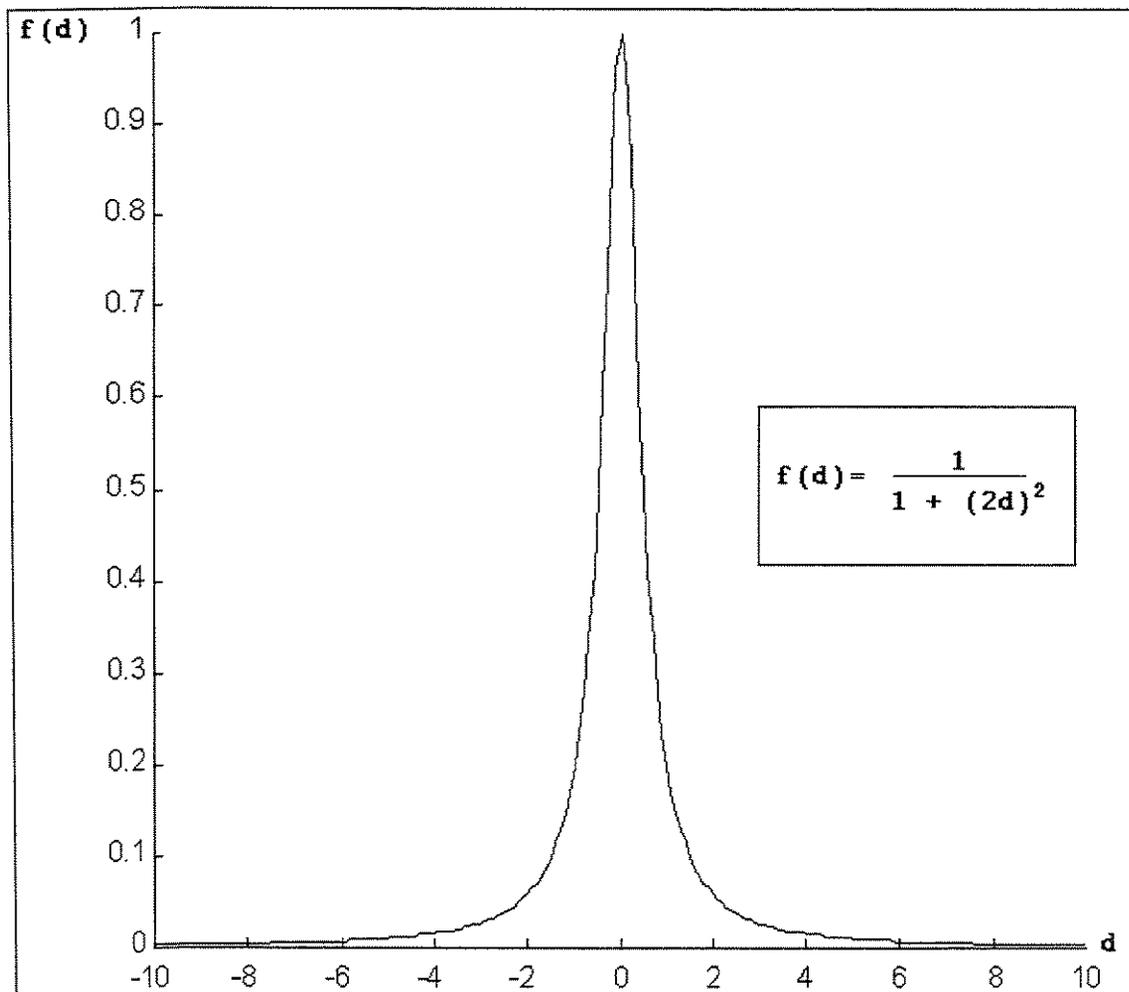


Figura 8.7 - Função de pertinência para avaliar a similaridade entre ramos da fração de uma palavra e uma linha reta.

8.5.2 AVALIAÇÃO DA ORIENTAÇÃO DE UM RAMO

Tendo sido determinado o grau de pertinência a uma linha reta e a uma linha curva, é então avaliada a semelhança do ramo em relação aos tipos de linhas retas e curvas ilustradas na Figura 8.2. Por se tratar de um processo de reconhecimento computacional, a similaridade de um ramo a , por exemplo, uma linha reta vertical, horizontal ou inclinada é expressa numericamente. É necessário portanto, determinar um meio para quantificar esta similaridade. Uma maneira é avaliar a orientação do ramo em questão significando, neste caso, determinar quão perto tal orientação está das inclinações de 0° , 45° , 90° , 135° , 180° , 225° , 270° , 315° e 360° . Os seguintes procedimentos foram introduzidos para avaliar a orientação de um ramo previamente analisado como linha reta, linha curva e *loop*.

Valores de Orientação de Um Ramo Considerando Linhas Retas

Neste caso, somente são considerados conjuntos *fuzzy* representando os segmentos de retas do grupo $\{H, P, V, N\}$, cujos elementos são ilustrados na Figura 8.2. O valor de pertinência do ramo analisado a cada elemento do grupo de segmentos de reta é determinado pelo ângulo $\theta = \tan^{-1}(a)$, onde “ a ” é o coeficiente angular da reta estimada na subseção 8.5.1. São definidas as seguintes funções de pertinência a cada elemento do grupo $\{H, P, V, N\}$

[SC74]:

$$H(\theta) = 1 - \min\{\min\{|\theta|, |180 - \theta|, |360 - \theta|\} / 45, 1\}, \quad (8.11)$$

$$V(\theta) = 1 - \min\{\min\{|90 - \theta|, |270 - \theta|\} / 45, 1\}, \quad (8.12)$$

$$P(\theta) = 1 - \min\{\min[|45 - \theta|, |225 - \theta|] / 45, 1\}; \quad (8.13)$$

$$N(\theta) = 1 - \min\{\min[|135 - \theta|, |315 - \theta|] / 45, 1\} \quad (8.14)$$

Na Figura 8.8 é apresentada graficamente a forma das funções de pertinência para cada elemento do grupo $\{H, P, V, N\}$.

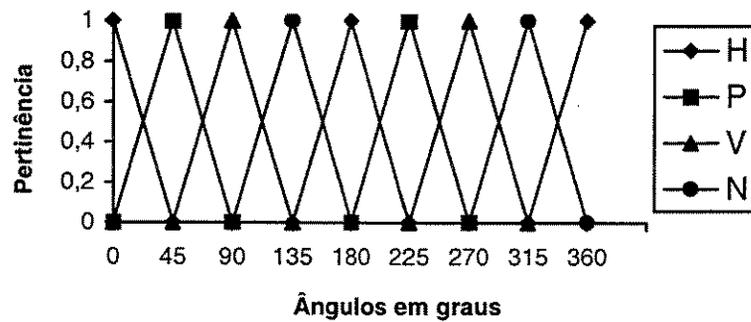


Figura 8.8 - Funções de pertinência para o conjunto $\{H, P, V, N\}$.

Valores de Orientação de Um Ramo Considerando Linhas Curvas

Para avaliar a orientação de um ramo segundo linhas curvas são empregados conjuntos *fuzzy* relacionados aos segmentos de linha do grupo $\{C, D, A, U, S, Z\}$ ilustrados na Figura 8.2. Os elementos desse grupo são divididos em 3 subgrupos de acordo com a inclinação do segmento de reta unindo suas extremidades (pontos finais). Esses subgrupos são denominados: subgrupo de curvas verticais $\{C, D\}$, subgrupo de curvas horizontais $\{A, U\}$ e subgrupo de curvas inclinadas $\{S, Z\}$. Para obter os valores de pertinência de um ramo analisado, a cada

elemento desses subgrupos, é avaliado o ângulo $\theta = \tan^{-1}((y_2 - y_1)/(x_2 - x_1))$. As coordenadas (x_1, y_1) e (x_2, y_2) são, respectivamente, as coordenadas dos dois pontos finais do ramo em análise e θ é o ângulo de inclinação do segmento de reta unindo estes pontos finais. As seguintes funções de pertinência são calculadas para caracterizar uma curva como pertencente ao subgrupo de curvas horizontais ou subgrupo de curvas verticais:

$$HC(\theta) = 1 - \min\{\min[|\theta|, |180 - \theta|, |360 - \theta|]/90, 1\}; \quad (8.15)$$

$$VC(\theta) = 1 - \min\{\min[|90 - \theta|, |270 - \theta|]/90, 1\} \quad (8.16)$$

É ilustrado graficamente na Figura 8.9 as funções de pertinência para as curvas verticais e curvas horizontais.

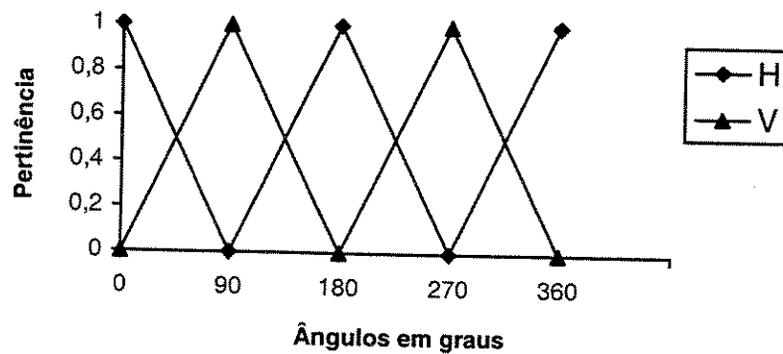


Figura 8.9 - Funções de pertinência para determinar a orientação vertical (V) ou horizontal (H) de uma curva.

O subgrupo $\{S, Z\}$ pode ser caracterizado como curva horizontal $\{S_H, Z_H\}$ ou curva vertical $\{S_V, Z_V\}$, dependendo do ângulo de inclinação θ . Para determinar as pertinências de um ramo a cada um dos elementos do subgrupo $\{C, D\}$, $\{A, U\}$ ou $\{S, Z\}$ avalia-se, primeiramente, a posição do ramo em relação ao segmento de reta LN apresentado na Figura 8.10. Esta avaliação pode ser feita de forma semelhante aquela utilizada para as equações (8.1) a (8.6) observada as seguintes constatações:

- (1) Maior parte dos pontos do ramo situado à direita de LN estabelece uma maior pertinência (prd_{LN}) do ramo à região direita de LN e, portanto, uma maior similaridade do ramo a curva tipo C . Uma medida desta similaridade é expressa por:

$$C(\theta) = VC(\theta) \cdot prd_{LN}; \quad (8.17)$$

- (2) Estando a maioria dos pontos do ramo à esquerda de LN ocorre uma maior pertinência (pre_{LN}) do ramo à região esquerda de LN , proporcionando uma maior similaridade do ramo a curva tipo D , assim:

$$D(\theta) = VC(\theta) \cdot pre_{LN}; \quad (8.18)$$

- (3) Caso o número de pontos do ramo situados à esquerda e à direita de LN sejam aproximadamente iguais, é dito ocorrer maior pertinência (prc_{LN}) em relação a região central do segmento LN , implicando maior similaridade do ramo a curva tipo S_V ou

Z_V , desse modo tem-se:

$$S_V(\theta) = VC(\theta) \cdot prc_{LN}; \quad (8.19a)$$

$$Z_V(\theta) = VC(\theta) \cdot prc_{LN}; \quad (8.19b)$$

- (4) Maior parte dos pontos do ramo situada na região acima de LN produz uma maior pertinência (prt_{LN}) do ramo a referida região, acarretando maior similaridade a curva tipo A . Essa similaridade pode ser quantificada da seguinte maneira:

$$A(\theta) = HC(\theta) \cdot prt_{LN}; \quad (8.20)$$

- (5) Se a maioria dos pontos do ramo estão localizados na região abaixo de LN é produzida uma maior pertinência (prb_{LN}) aquela região, permitindo inferir uma maior similaridade a curva tipo U , dessa forma:

$$U(\theta) = HC(\theta) \cdot prb_{LN}; \quad (8.21)$$

- (6) Sendo o número de pontos do ramo aproximadamente iguais nas regiões superior e inferior ao segmento LN é dito ocorrer uma maior pertinência do ramo a região onde se encontra LN , denominada região mediana. Uma maior pertinência (prm_{LN}) a tal região contribui para uma similaridade do ramo a curva tipo S_H ou Z_V , assim:

$$S_H(\theta) = HC(\theta) \cdot prm_{LN}; \quad (8.22a)$$

$$Z_H(\theta) = VC(\theta) \cdot prm_{LN}. \quad (8.22b)$$

Para decidir por uma curva S ou Z adota-se o método desenvolvido por [SC74] onde é avaliada a posição das interseções entre segmentos de retas paralelos a linha LN e o ramo analisado, conforme ilustrado na Figura 8.10. Um ramo é dito ser:

$$S_V \text{ se } \min(y_1, y_2) < \max(y_3, y_4) \quad (8.23)$$

$$S_H \text{ se } \max(x_1, x_2) > \max(x_3, x_4) \quad (8.24)$$

$$Z_V \text{ se } \max(y_1, y_2) > \min(y_3, y_4) \quad (8.25)$$

$$Z_H \text{ se } \min(x_1, x_2) < \max(x_3, x_4) \quad (8.26)$$

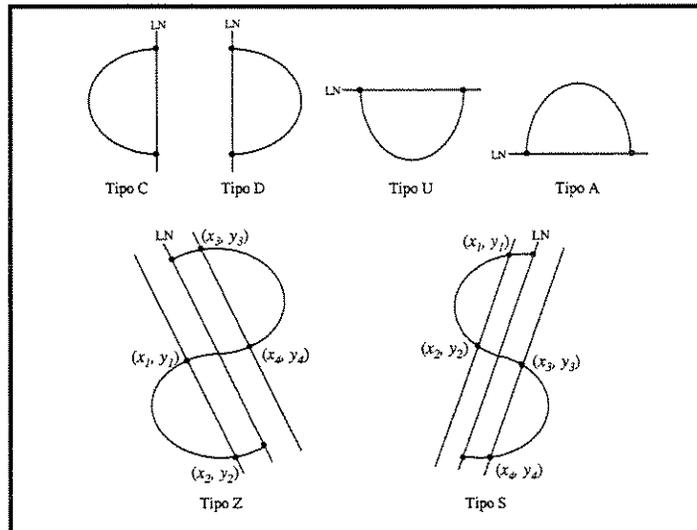


Figura 8.10 - Classificação das curvas tipo C, D, A, U, S e Z.

8.6 CONSTRUÇÃO DO VETOR DE CARACTERÍSTICA DE UM RAMO

Os elementos do vetor de características construído para um ramo representam graus de similaridade a cada um dos tipos de segmentos de linha da Figura 8.2. É construído um vetor de características para cada ramo extraído de uma fração da palavra sendo processada. Isto significa que a fração em questão é representada por tantos vetores de características quanto forem o número de ramos que a compõem. Em conseqüência, uma palavra passa a ter sua representação dada por uma seqüência de vetores de características de ramos, uma vez que é composta pelas várias frações resultantes do processo de segmentação. Supondo a situação em que a fração de uma palavra representa um caractere completo, é utilizado o caractere manuscrito "E" da Figura 8.11 para demonstrar o processo de construção de vetores de características. O referido caractere possui 3 ramos localizados entre pontos singulares numerados de 1 a 4. A Tabela 8.1 apresenta os valores de pertinência referentes ao grau de similaridade desses ramos a uma linha reta e a uma linha curva. A Tabela 8.2 mostra os valores resultantes da avaliação da orientação dos ramos do caractere segundo a subseção 8.5.2. A Tabela 8.3 apresenta os valores de pertinências referentes à posição relativa dos ramos do caractere na Figura 8.11.

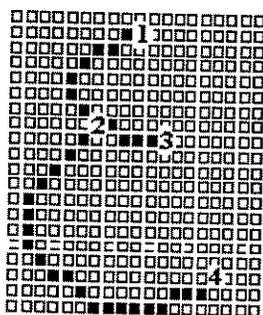


Figura 8.11 - Fração de uma palavra segmentada cujos ramos serão classificados.

Tabela 8.1 - Valores de pertinência relacionados a segmentos de linhas retas e linhas curvas (Seção 8.51).

Valores de Pertinência		
Pares de Nós	Linha Reta (Gr)	Linha Curva (Gc)
1-2	0,3341	0,6659
2-4	0,1650	0,8350
2-3	0,8130	0,1870

Tabela 8.2 - Valores de pertinência relativos a orientação dos ramos do caractere.

Ramo	Nós			Ramo	Nós		
	1-2	2-4	2-3		1-2	2-4	2-3
H(θ)	0,2	0,45	0,71	A(θ)	0,37	0	0
V(θ)	0	0	0	U(θ)	0	0,47	0
P(θ)	0,82	0	0	S(θ)	0	0	0
N(θ)	0	0,55	0,29	Z(θ)	0	0	0
C(θ)	0,62	0,53	0	L	0	0	0
D(θ)	0	0	0		-	-	-

Tabela 8.3 - Valores de pertinência relativo as posições do ramo em um segmento.

Pertinências a Posição	Nós		
	1-2	2-4	2-3
Esquerda (pre)	0,48	0,84	0,58
Centro (prc)	0,96	0,32	0
Direita (prd)	0,52	0,16	0,23
Topo (prt)	1,0	0,69	0
Meio (prm)	0	0,63	0
Base (prb)	0	0,31	1,0

Cada uma das características utilizadas para representação de um ramo são expressas por valores de pertinência considerando a posição do ramo na fração da palavra e sua similaridade a cada um dos padrões ilustrados na Figura 8.2. Esses valores são obtidos combinando os valores das pertinências obtidos para as Tabelas 8.1, 8.2 e 8.3 por meio de uma multiplicação. Um exemplo desse processo é dado a seguir, considerando o ramo entre os nós 1-2 na Figura 8.11 e o segmento de reta com inclinação positiva (P) na Figura 8.2. Inicialmente os valores Gr e $P(\theta)$ das Tabelas 8.1 e 8.2 são combinados. O resultado dessa combinação é representada por, $C_{Gr,P}$, na equação (8.27):

$$C_{Gr,P} = Gr \cdot P(\theta). \quad (8.27)$$

Os valores da Tabela 8.3, antes de serem combinados ao resultado obtido na equação (8.27), são combinados entre si originando os seguintes valores de pertinência:

1. Pertinência do ramo em relação a região superior esquerda da fração da palavra

$$prs_E = prt \cdot pre ;$$

2. Pertinência do ramo em relação a região mediana esquerda da fração da palavra

$$prm_E = prm \cdot pre ;$$

3. Pertinência do ramo em relação a região inferior esquerda da fração da palavra

$$pri_E = prb \cdot pre ;$$

4. Pertinência do ramo em relação a região superior central da fração da palavra

$$prs_C = prt \cdot prc ;$$

5. Pertinência do ramo em relação a região mediana central da fração da palavra

$$prm_C = prm \cdot prc ;$$

6. Pertinência do ramo em relação a região inferior central da fração da palavra

$$pri_C = prb \cdot prc ;$$

7. Pertinência do ramo em relação a região superior direita da fração da palavra

$$prs_D = prt \cdot prd ;$$

8. Pertinência do ramo em relação a região mediana direita da fração da palavra

$$prm_D = prm \cdot prd ;$$

9. Pertinência do ramo em relação a região inferior direita da fração da palavra

$$pri_D = prb \cdot prd .$$

A combinação dos valores da Tabela 8.3 visa expressar de forma adequada as variações de um ramo segundo as posições relativas à fração sendo processada da palavra. Pois a posição de um ramo pode variar em caracteres de uma mesma classe devido aos estilos de escrita, conforme foi ilustrado na Figura 8.6. Por essa razão, foi empregada a teoria de conjuntos *fuzzy* para estabelecer uma relação de pertinência do ramo a cada uma das posições relativas à fração da palavra. Na Figura 8.12 são representadas as posições da fração de uma palavra as quais se referem essas pertinências resultantes da combinação dos valores da Tabela 8.3.

prs_E	prs_C	prs_D
prm_E	prm_C	prm_D
pri_E	pri_C	pri_D

Figura 8.12 - Uma representação das pertinências de um ramo as posições relativas à fração considerada de uma palavra.

As pertinências geradas a partir da Tabela 8.3, considerando as nove posições ilustradas na Figura 8.12, são finalmente combinadas com a pertinência resultante da equação (8.27). O resultado dessa combinação é uma avaliação da similaridade existente entre o ramo em questão e o segmento de linha (P) da Figura 8.2, observada a posição do ramo em relação a fração da palavra considerada. As seguintes características são geradas:

1. Característica reta com inclinação positiva na região superior esquerda da fração da palavra

(RP_{SE}) :

$$RP_{SE} = C_{Gr,P} \cdot prs_E; \quad (8.28)$$

2. Característica reta com inclinação positiva na região superior central da fração da palavra

(RP_{SC}) :

$$RP_{SC} = C_{Gr,P} \cdot prs_C; \quad (8.29)$$

3. Característica reta com inclinação positiva na região superior direita da fração da palavra

(RP_{SD}) :

$$RP_{SD} = C_{Gr,P} \cdot prs_D; \quad (8.30)$$

4. Característica reta com inclinação positiva na região mediana esquerda da fração da palavra

(RP_{ME}) :

$$RP_{ME} = C_{Gr,P} \cdot prm_E; \quad (8.31)$$

5. Característica reta com inclinação positiva na região mediana central da fração da palavra

(RP_{MC}) :

$$RP_{MC} = C_{Gr,P} \cdot prm_C; \quad (8.32)$$

6. Característica reta com inclinação positiva na região mediana direita da fração da palavra

(RP_{MD}) :

$$RP_{MD} = C_{Gr,P} \cdot prm_D; \quad (8.33)$$

7. Característica reta com inclinação positiva na região inferior esquerda da fração da palavra

(RP_{IE}) :

$$RP_{IE} = C_{Gr,P} \cdot pri_E; \quad (8.34)$$

8. Característica reta com inclinação positiva na região inferior central da fração da palavra

(RP_{IC}) :

$$RP_{IC} = C_{Gr,P} \cdot pri_C; \quad (8.35)$$

9. Característica reta com inclinação positiva na região inferior direita da fração da palavra

(RP_{ID}) :

$$RP_{ID} = C_{Gr,P} \cdot pri_D. \quad (8.36)$$

No exemplo descrito acima envolvendo as equações (8.27) a (8.36), foi primeiramente obtido um valor de similaridade do ramo 1-2 da Figura 8.11 ao padrão (P) da Figura 8.2. Esse valor foi combinado com as pertinências do referido ramo às possíveis posições relativas (Figura 8.12) à fração da palavra ilustrada na Figura 8.11. Observe que tal combinação gerou 9

novas características cujos valores numéricos foram obtidos através das equações (8.28) a (8.36) e representados na Tabela 8.4.

Tabela 8.4 - Valores de pertinência para o ramo 1-2 considerando a reta com inclinação positiva (P) da Figura 8.2 e as posições relativas à fração da palavra na Figura 8.11.

RP_{SE}	RP_{SC}	RP_{SD}	RP_{ME}	RP_{MC}	RP_{MD}	RP_{IE}	RP_{IC}	RP_{ID}
0,132	0,263	0,142	0	0	0	0	0	0

Portanto, a Tabela 8.4 constitui uma parte do vetor de característica construído para o ramo 1-2 da Figura 8.11. Nessa Tabela estão representados unicamente as características geradas a partir da análise de similaridade do ramo 1-2 à reta com inclinação positiva (P) da Figura 8.2, observando as posições do ramo em relação sua fração da palavra. Por conseguinte, para completar o vetor de características para o ramo 1-2 deve-se ainda analisar a similaridade do referido ramo aos demais padrões da Figura 8.2, observando para cada um desses padrões a pertinência do ramo às posições relativas à fração da palavra considerada. Os valores obtidos nessa análise são calculados de forma análoga àquela utilizada nas equações (8.27) a (8.36) para o padrão reta com inclinação positiva (P). Desse modo, cada padrão Y da Figura 8.2 gera 9 características, como exemplificado na Tabela 8.4, com as seguintes denominações:

1. Característica Y na região superior esquerda da fração da palavra;
2. Característica Y na região superior central da fração da palavra;
3. Característica Y na região superior direita da fração da palavra;
4. Característica Y na região mediana esquerda da fração da palavra;
5. Característica Y na região mediana central da fração da palavra;
6. Característica Y na região mediana direita da fração da palavra;
7. Característica Y na região inferior esquerda da fração da palavra;

8. Característica Y na região inferior central da fração da palavra;
9. Característica Y na região inferior direita da fração da palavra.

Uma vez que a Figura 8.2 apresenta 11 padrões e cada padrão gera 9 características, o vetor final para representar um ramo de uma fração da palavra será composto por 99 (9×11) elementos. Uma palavra manuscrita é então representada por vários vetores de características resultantes da decomposição em ramos das frações geradas pela sua segmentação em caracteres. Esses vetores de características são processados pelos modelos de palavras manuscritas descritos no Capítulo IX a seguir. Como conclusão, é apresentado na Figura 8.13 um diagrama de blocos que resume o processo de extração de características apresentado neste capítulo.

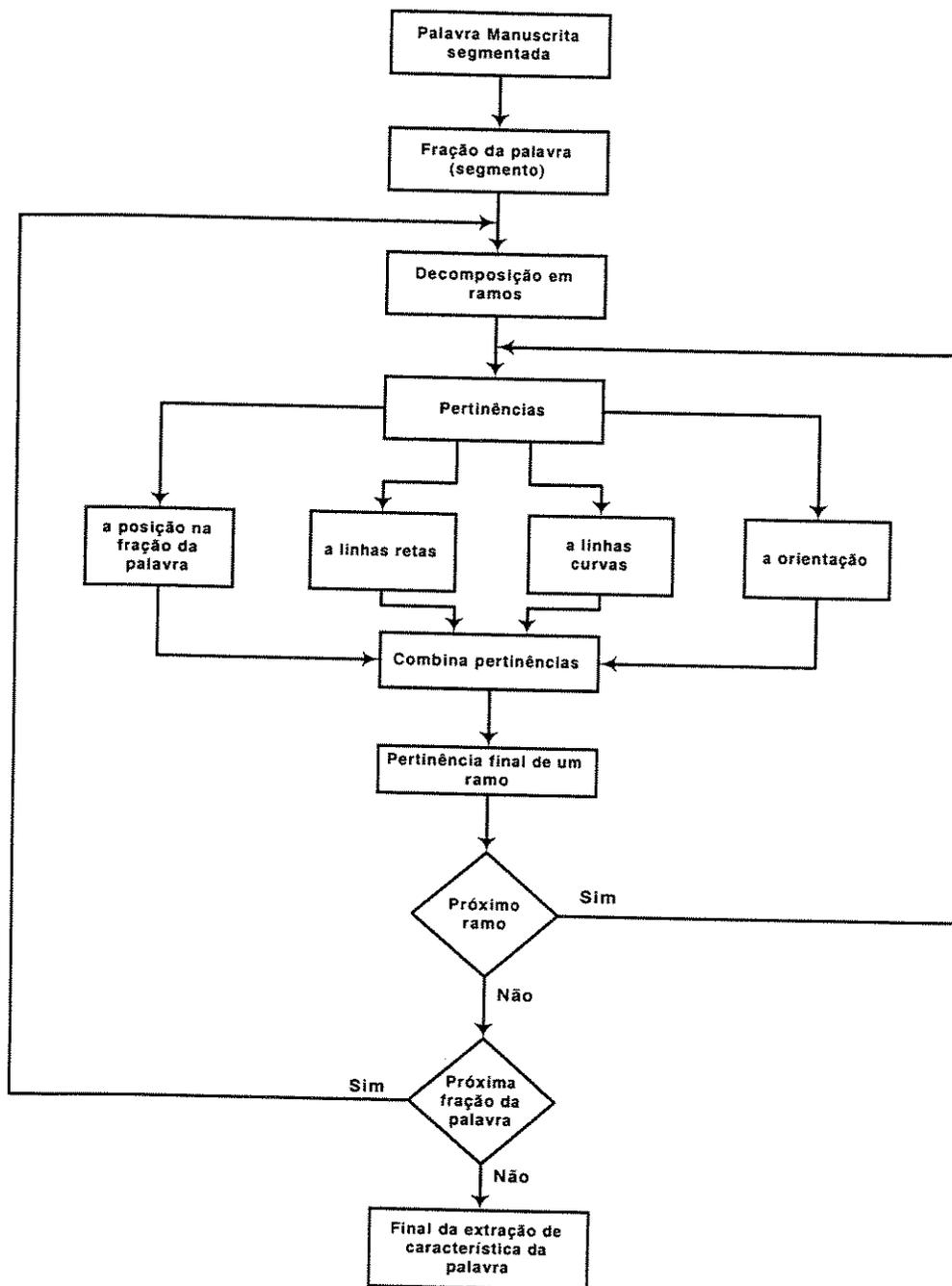


Figura 8.13 - Diagrama de blocos resumindo o processo de extração de características.

CAPÍTULO IX

CLASSIFICADOR DE PALAVRAS MANUSCRITAS E FORMAÇÃO DA BASE DE DADOS

O sistema de reconhecimento abordado nesta tese, conforme discutido nos capítulos anteriores, começa seu processamento após a imagem digitalizada de uma palavra ter sido apresentada à sua entrada. Essa imagem passa por pré-processamento que a transforma em uma imagem binária (preto e branco), eliminando ruídos e corrigindo possíveis rotação e inclinação da palavra. Para possibilitar a extração de características a palavra em processamento é segmentada. Cada fração da palavra resultante da segmentação é decomposta em uma seqüência de segmentos de linha denominados ramos. O resultado final do processo de extração de características é a representação da palavra manuscrita por uma seqüência de vetores de características. Cada vetor da referida seqüência corresponde a características extraídas de cada ramo das frações da palavra.

Este capítulo é voltado à descrição do processo de classificação da seqüência de vetores de características representando uma palavra manuscrita. Tal processo de classificação é realizado através dos modelos ocultos de Markov (HMMs) e também, através de modelos generalizados de Markov (FHMMs - *Fuzzy Hidden Markov Models*). Um HMM (FHMM) é

construído para cada caractere do alfabeto existente no vocabulário considerado. O modelamento de uma palavra do vocabulário se dá pela concatenação adequada dos HMMs (FHMMs) de cada caractere. Durante os próximos parágrafos o modelamento matemático de uma palavra será explicado utilizando os HMMs. O mesmo procedimento de modelamento é adotado quando são empregados os FHMMs. As seções finais deste capítulo são dedicadas à descrição da base de dados utilizada para treinamento e teste do sistema de reconhecimento.

9.1 MODELOS PARA PALAVRAS MANUSCRITAS

Um dos objetivos do trabalho apresentado nesta tese é possibilitar a utilização do sistema desenvolvido em diferentes vocabulários. A construção de um HMM para cada caractere fornece tal característica ao sistema, pois concatenando adequadamente os modelos para caracteres constrói-se modelos para palavras de diferentes vocabulários. Além disso, o modelamento de caracteres é adequado ao método de extração de características empregado neste trabalho, o qual é baseado na segmentação de caracteres.

Para a construção dos HMMs adotou-se a topologia esquerda para direita, que é apropriada para a construção de modelos para palavras a partir de modelos de caracteres, estabelecendo menos transições entre estados. O número de estados de um HMM modelando um caractere foi estabelecido observando o número máximo e mínimo de ramos em amostras de um determinado caractere manuscrito. Decidiu-se por um número de estados que representasse um meio termo entre esses dois limites, pois as variações de estilos podem fazer com que amostras de um mesmo caractere apresentem diferentes números de ramos. Por exemplo, a letra "a" pode apresentar o conjunto de ramos ilustrado na Figura 9.1. No entanto,

este caractere pode sofrer variações em sua forma e apresentar um número de ramos maior ou menor do que aquele da Figura 9.1.



Figura 9.1 - Representação da letra “a” por um conjunto de ramos.

Devido a variações no número de ramos compondo um caractere e para evitar situações em que o número de estados do HMM é maior que o comprimento da seqüência de observações do caractere, optou-se por utilizar HMMs apresentando algumas transições constituindo saltos entre estados. Desse modo, para uma letra “a” construiu-se o modelo exemplificado na Figura 9.2, onde a transição a_{13} constitui um salto entre o primeiro e o último estado do modelo.

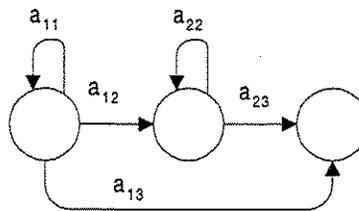


Figura 9.2 - HMM para modelamento da letra “a”.

Os modelos para as palavras são construídos conectando os modelos de caracteres. Para a palavra *um* escrita em letras minúsculas, são conectados os modelos das letras “u” e “m” apresentados na Figura 9.3.

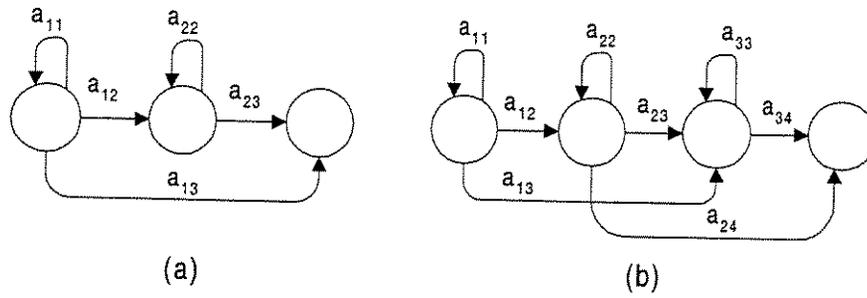


Figura 9.3 - Modelamento das letras "u" e "m": (a) HMM para letra "u"; (b) HMM para letra "m".

O último estado de cada modelo de caracteres é utilizado como ponto de conexão na concatenação dos modelos para formar uma palavra. O resultado da conexão dos dois modelos da Figura 9.3 é apresentado na Figura 9.4.

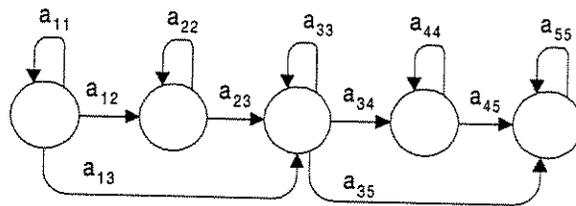


Figura 9.4- Modelo para a palavra um.

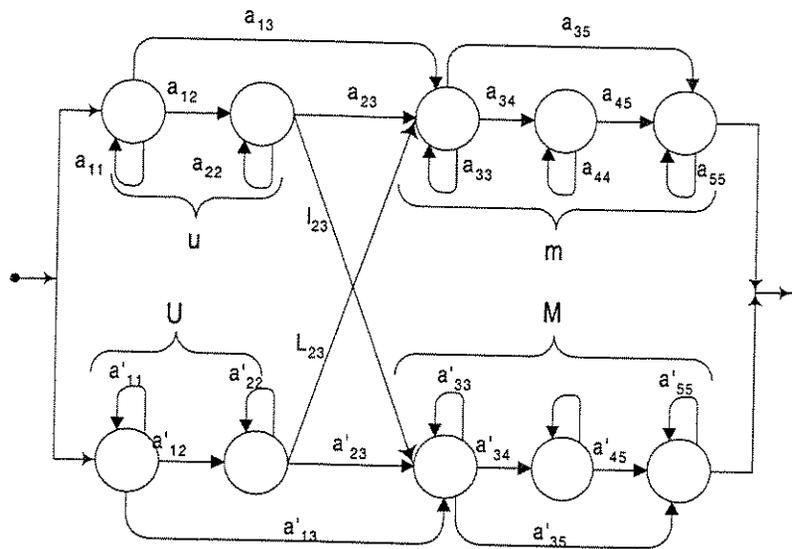


Figura 9.5 – Modelo para uma palavra manuscrita.

Uma palavra manuscrita pode ser escrita com letras minúsculas, maiúsculas ou uma combinação de ambas. Sob esse ponto de vista, existem diferentes formas de escrever uma mesma palavra, sendo difícil construir um modelo para cada uma dessas formas. Por esse motivo, adotou-se uma estratégia semelhante àquela proposta em [YSS99] para modelar palavras manuscritas. Essa estratégia é exemplificada na Figura 9.5 para a palavra *um*, e mostra como os HMMs modelando letras maiúsculas e minúsculas são conectados para modelar uma palavra através das transições l_{23} e L_{23} , sendo:

$$l_{23} = a_{23}, \quad a_{22} + a_{23} + l_{23} = 1; \quad (9.1)$$

$$L_{23} = a'_{23}, \quad a'_{22} + a'_{23} + L_{23} = 1. \quad (9.2)$$

Utilizando a estratégia ilustrada na Figura 9.5, pode-se modelar qualquer forma de escrita de uma palavra no que diz respeito a combinação de letras maiúsculas e minúsculas e, desse modo, reduzir o número de modelos construídos para representar palavras de um vocabulário. Esse fato pode ser verificado no modelamento da palavra *um* observando, respectivamente, sua escrita com letras maiúsculas, minúsculas ou uma combinação destes dois tipos de letras. Nesse caso, seria necessário desenvolver um modelo para cada uma das seguintes situações:

1. palavra *um* escrita somente com letras minúsculas;
2. palavra *um* escrita somente com letras maiúsculas;
3. palavra *um* escrita com a primeira letra maiúscula e a última letra minúscula;
4. palavra *um* escrita com a primeira letra minúscula e a última letra maiúscula.

9.2 SEQÜÊNCIA DE OBSERVAÇÕES PARA UMA PALAVRA MANUSCRITA

Antes de ser processada por um HMM (FHMM) a seqüência de vetores de características representando uma palavra (capítulo VIII) é transformada em um seqüência de símbolos de observação armazenados em um arquivo de códigos (*code book*). O arquivo de códigos é formado a partir de uma quantização vetorial [GO] dos vetores de características extraídos dos dados utilizados para treinamento do sistema. A quantização vetorial divide o espaço de características em 99 regiões correspondentes às 99 características apresentadas na seção 8.6 desta tese. O vetor referente ao centróide de cada região é relacionado a um símbolo de observação para compor o arquivo de códigos. A seqüência de vetores de características de uma palavra passa a ter cada um de seus vetores representado pelo símbolo correspondendo ao centróide da região que lhe é mais próximo segundo a distância Euclidiana.

9.3 PARÂMETROS INICIAIS PARA HMMs e FHMMs

Os valores iniciais para as matrizes A e Π para um modelo λ (ver Capítulo III), \hat{A} e $\hat{\Pi}$ para um modelo $\hat{\lambda}$ (ver Capítulo V), correspondendo respectivamente, a um HMM e um FHMM modelando um caractere, são aleatórios. A matriz B para um HMM e \hat{B} para um FHMM são estimadas empregando o método *segmental K-Means* [Rab89]. Para a estimação de \hat{B} o algoritmo de Viterbi modificado [Moh95] é utilizado no lugar do algoritmo de Viterbi clássico. O valor inicial dos correspondentes elementos de \hat{B} em um estado é obtido da média dos vetores de características cujos símbolos são processados naquele estado. No caso do HMM esse valor é calculado a partir da probabilidade de um vetor de características ser

classificado como sendo um dos símbolos processados pelo estado. Os símbolos a serem processados em um estado são determinados, inicialmente, pela observação dos ramos que compõem um caractere. O seguinte algoritmo de treinamento [Rab89], [Moh95], foi empregado para estimação dos parâmetros dos HMMs e FHMMs:

Algoritmo:

1. Escolher valores iniciais para os parâmetros do modelo;
2. Enquanto todos os parâmetros do modelo não forem otimizados:
 - 2.1. Segmentar as seqüências de observações para treinamento do modelo em N estados usando o algoritmo de Viterbi (Algoritmo de Viterbi modificado);
 - 2.2. Agrupar os vetores relativos as observações em cada estado dentro de M clusters;
 - 2.3. Atualizar os parâmetros do modelo a partir dos estados e clusters resultantes;
 - 2.4. Refinar os parâmetros do modelo usando algoritmo de Baum-Welch (Baum-Welch modificado - seção 5.6).

9.4 ESTRUTURA DO CLASSIFICADOR DE PALAVRAS

Como explicado em seções anteriores, dois classificadores são empregados para reconhecer uma palavra como sendo uma das N palavras do vocabulário considerado. Um classificador modela palavras manuscritas utilizando HMMs, o outro classificador emprega os FHMMs. A estrutura para classificação de uma palavra manuscrita pode ser representada pela Figura 9.6, onde λ_i , $i = 1, 2, \dots, N$, representa um HMM (FHMM) modelando uma palavra do vocabulário. Durante o processo de classificação a seqüência de observações para uma

palavra de classe desconhecida é processada pelos modelos λ_i e os valores obtidos pelo algoritmo de Viterbi são utilizados para organizar uma lista das palavras do vocabulário em ordem decrescente. Algumas operações de pós-processamento são realizadas para determinar o resultado final de classificação.

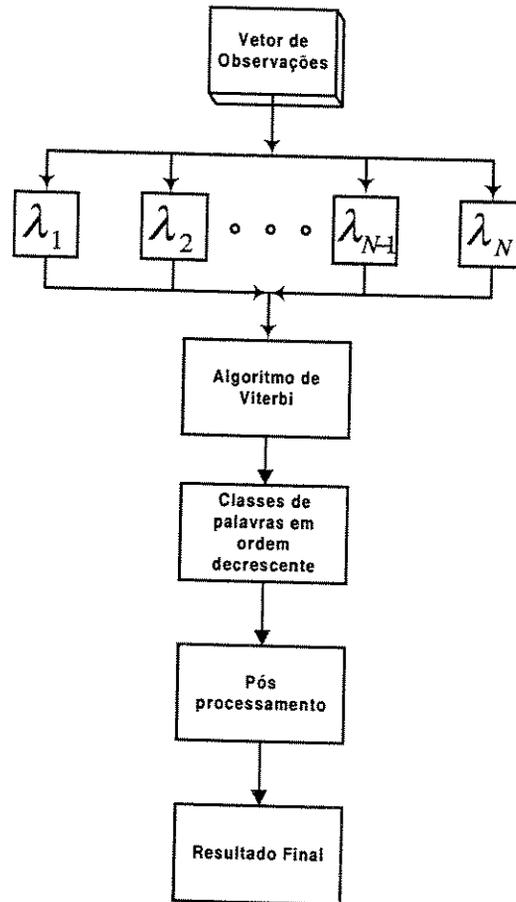


Figura 9.6 - Diagrama representativo da classificação de uma palavra manuscrita.

9.5 PÓS-PROCESSAMENTO

Devido ao processo de segmentação explícita empregado neste trabalho e ao modelamento de uma palavra através de concatenação de modelos de letras, informações globais sobre a palavra são perdidas, acarretando uma redução de desempenho do sistema. A fim de otimizar o desempenho do sistema durante o processo de classificação, é utilizado o comprimento médio das seqüências de símbolos de cada classe de palavras de treinamento no procedimento de pós-processamento. São também analisados no referido procedimento o número e a posição de segmentos de linha situados abaixo da linha de base (segmentos descendentes) ou acima da linha superior da palavra (segmentos ascendentes), conforme é ilustrado na Figura 9.7.

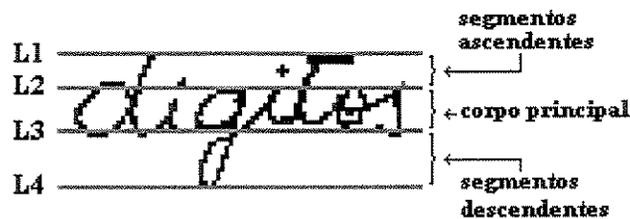


Figura 9.7 – Representação dos segmentos ascendentes e descendentes da palavra dígitos.

Na utilização do comprimento médio da seqüência de símbolos de cada classe, o valor obtido pelo algoritmo de Viterbi é multiplicado por $e^{-|x-y|}$ sempre que o comprimento y da seqüência de símbolos da palavra sendo classificada for menor ou exceder em K símbolos o comprimento médio x da seqüência de símbolos de uma classe. A variável K é um número inteiro obtido a partir da variância máxima verificada na análise do comprimento das seqüências de símbolos em cada classe de palavra.

Os segmentos denominados ascendentes e descendentes, conforme Figura 9.7, não ocorrem quando a palavra for composta somente por letras maiúsculas e, nesse caso não são produzidas informações para o procedimento de pós-processamento. É necessário, portanto, determinar se a palavra sendo processada é escrita com letras maiúsculas. A solução encontrada para esse problema foi a construção de um modelo para palavra concatenando apenas modelos para letras maiúsculas. Este modelo, simbolizado por λ_{MASC} , deve apresentar os mesmos parâmetros dos HMMs (FHMMs) modelando letras maiúsculas no modelo λ ilustrado na Figura 9.5.

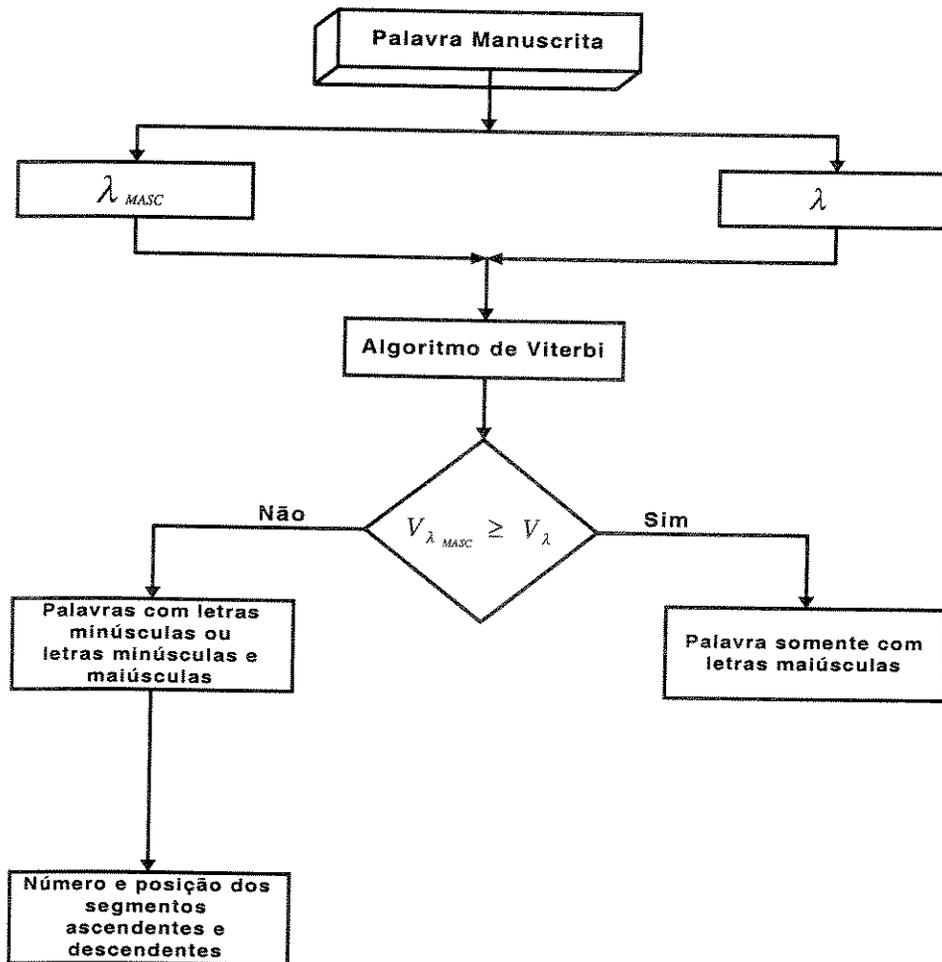


Figura 9.8 – Operações realizadas para determinar se uma palavra é escrita somente com letras maiúsculas.

Uma palavra é então considerada ser escrita com letras maiúsculas quando o valor $V_{\lambda_{MASC}}$ obtido pelo algoritmo de Viterbi for maior ou igual ao valor V_{λ} . A Figura 9.8 apresenta através de um diagrama de blocos as operações realizadas para determinar se uma palavra é escrita somente com letras maiúsculas.

A linha de base de uma palavra manuscrita é estimada de acordo com o algoritmo descrito no capítulo VI desta tese. A linha superior de uma palavra é determinada de acordo com o método desenvolvido em [Gui95].

9.6 FORMAÇÃO DA BASE DE DADOS

A base de dados empregada para treinamento e teste do sistema de reconhecimento é composta por imagens de palavras manuscritas expressando valores em cheques brasileiros (Anexo G). Foram utilizadas 2416 imagens de palavras para compor a base de dados representando o estilo de escrita de cerca de 90 pessoas. Cerca de 1200 dessas imagens provêm de um grupo de 200 cheques de várias instituições bancárias e 1216 imagens são originárias de palavras escritas em folhas de papel sulfite. O processo de digitalização foi realizado em uma resolução de 200 dpi (*dots per inch*) e 256 níveis de cinza, sendo a imagem resultante armazenada no formato bitmap [Lev94].

Um problema enfrentado na formação da base de dados são palavras com traços incomuns muitas vezes escritas para gerar dificuldades ao processo de reconhecimento. Por esse motivo, as imagens de palavras são analisadas e algumas são descartadas, o que dificulta a formação de um banco de dados com um maior número de amostras. Uma consequência da utilização de um grande número de amostras é uma melhor definição das regiões nas quais é dividido o espaço de característica via quantização vetorial [GO]. Uma melhor estimação dos

parâmetros dos HMMs é também obtida com um maior número de amostras. A questão do número de amostras necessárias em banco de dados para reconhecimento de palavras manuscritas é abordada em [GMV98]. No referido trabalho, são realizados cálculos a fim de demonstrar o número de amostras adequado para teste e treinamento do sistema, observando uma desejada taxa de erro em relação ao melhor classificador. Os cálculos realizados demonstram, por exemplo, que para uma taxa de erro de 5% uma base de dados com 3000 amostras seria adequada. Observando uma taxa de erro de 1% um total de 10000 amostras seriam necessárias. No entanto, se levadas em conta variações como número de escreventes, estado da pessoa no momento da escrita, tipo de material, o número de amostras para uma taxa de erro de 1% estaria em torno de 200.000.

As 2416 imagens de palavras manuscritas utilizadas neste trabalho foram organizadas de acordo com as classes de palavras do vocabulário considerado. O total de amostras por classe é apresentado na Tabela 9.1.

Na Tabela 9.1 é atribuído a cada uma das classes de palavras um número inteiro no intervalo de 1 a 1002 para facilitar a apresentação da quantidade de amostras por classe no gráfico da Figura 9.9. A quantidade total de imagens envolve palavras escritas com letras maiúsculas, minúsculas ou com ambos os tipos de letras. A quantidade de palavras escritas com letras minúsculas é maior do que a quantidade de palavras escritas com letras maiúsculas. No próximo capítulo é explicado como a base de dados é organizada para treinamento e teste do sistema de reconhecimento, sendo também apresentados os resultados experimentais decorrentes desse teste.

Tabela 9.1 - Dados para treinamento e teste do sistema de reconhecimento.

CLASSES DE PALAVRAS	Nº DA CLASSE	TOTAL DE AMOSTRAS	CLASSES DE PALAVRAS	Nº DA CLASSE	TOTAL DE AMOSTRAS
Um	0001	67	Trinta	0030	88
Dois	0002	88	Quarenta	0040	77
Três	0003	71	Cinqüenta	0050	85
Quatro	0004	91	Sessenta	0060	71
Cinco	0005	75	Setenta	0070	85
Seis	0006	85	Oitenta	0080	82
Sete	0007	93	Noventa	0090	95
Oito	0008	74	Cem	0100	16
Nove	0009	66	Cento	0101	106
Dez	0010	19	Duzentos	0200	104
Onze	0011	19	Trezentos	0300	46
Doze	0012	33	Quatrocentos	0400	29
Treze	0013	30	Quinhentos	0500	27
Quatorze	0014	22	Seiscentos	0600	21
Quinze	0015	31	Setecentos	0700	29
Dezesseis	0016	36	Oitocentos	0800	22
Dezessete	0017	34	Novocentos	0900	27
Dezoito	0018	27	Mil	1000	110
Dezenove	0019	36	Centavos	1001	112
Vinte	0020	82	Reais	1002	105

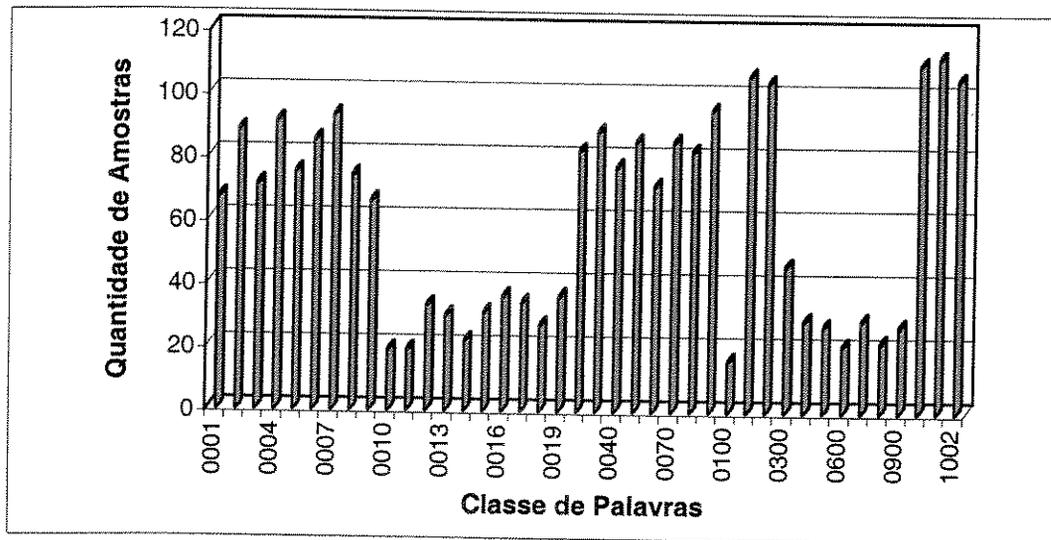


Figura 9.9 - Quantidade total de amostras por classe de palavras da base de dados.

CAPÍTULO X

TREINAMENTO DO SISTEMA E RESULTADOS EXPERIMENTAIS

A estratégia para treinamento dos modelos para caracteres apresentados no capítulo IX e os testes experimentais do sistema de reconhecimento estão entre os assuntos tratados neste capítulo. O processo de treinamento e o desempenho de reconhecimento do sistema são influenciados pela variedade de estilos de escrita e tamanho da base de dados utilizada. Por conseguinte, é necessário a utilização de alguma estratégia de treinamento e teste para obtenção de uma avaliação mais realística do desempenho do sistema. A estratégia de utilização da base de dados para treinamento do sistema e para obtenção de resultados experimentais é descrita nas primeiras seções deste capítulo. As demais seções são dedicadas, respectivamente, à análise dos resultados e avaliação do trabalho desenvolvido.

10.1 ESTRATÉGIA PARA TREINAMENTO DO SISTEMA

Para treinamento e teste do sistema de reconhecimento dividiu-se a base de dados utilizada em 10 grupos. A quantidade total de palavras em cada um desses grupos é mostrada na Tabela 10.1.

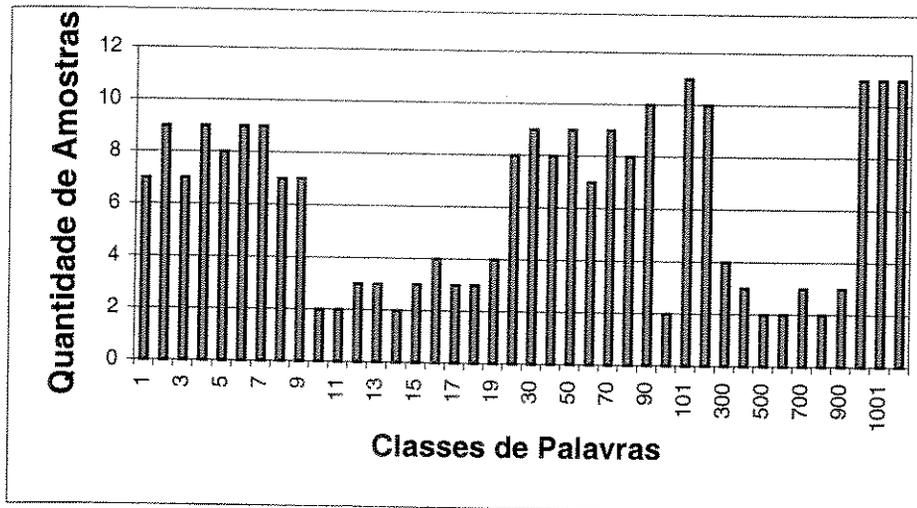
Tabela 10.1- Total de palavras em cada um dos grupos em que foi dividida a base de dados.

Grupos	Total de Palavras
Grupo 1	255
Grupo 2	253
Grupo 3	251
Grupo 4	245
Grupo 5	247
Grupo 6	242
Grupo 7	234
Grupo 8	237
Grupo 9	226
Grupo 10	226

Cada um dos 10 grupos da Tabela 10.1 é subdividido em classes de forma similar a Tabela 9.1. A quantidade de amostras em cada uma dessas classes é aproximadamente igual a 10% daquela apresentada pela respectiva classe na Tabela 9.1. A quantidade média de amostras por classe considerando cada um dos 10 grupos é apresentada na Tabela 10.2. Uma representação gráfica desta Tabela é ilustrada pela Figura 10.1.

Tabela 10.2 - Número médio de amostras por classe de palavras em cada um dos grupos em que foi dividida a base de dados.

CLASSES DE PALAVRAS	Nº DA CLASSE	TOTAL DE AMOSTRAS	CLASSES DE PALAVRAS	Nº DA CLASSE	TOTAL DE AMOSTRAS
Um	0001	7	Trinta	0030	9
Dois	0002	9	Quarenta	0040	8
Três	0003	7	Cinqüenta	0050	9
Quatro	0004	9	Sessenta	0060	7
Cinco	0005	8	Setenta	0070	9
Seis	0006	9	Oitenta	0080	8
Sete	0007	9	Noventa	0090	10
Oito	0008	7	Cem	0100	2
Nove	0009	7	Cento	0101	11
Dez	0010	2	Duzentos	0200	10
Onze	0011	2	Trezentos	0300	4
Doze	0012	3	Quatrocentos	0400	3
Treze	0013	3	Quinhentos	0500	2
Quatorze	0014	2	Seiscentos	0600	2
Quinze	0015	3	Setecentos	0700	3
Dezesseis	0016	4	Oitocentos	0800	2
Dezessete	0017	3	Novocentos	0900	3
Dezoito	0018	3	Mil	1000	11
Dezenove	0019	4	Centavos	1001	11
Vinte	0020	8	Reais	1002	11



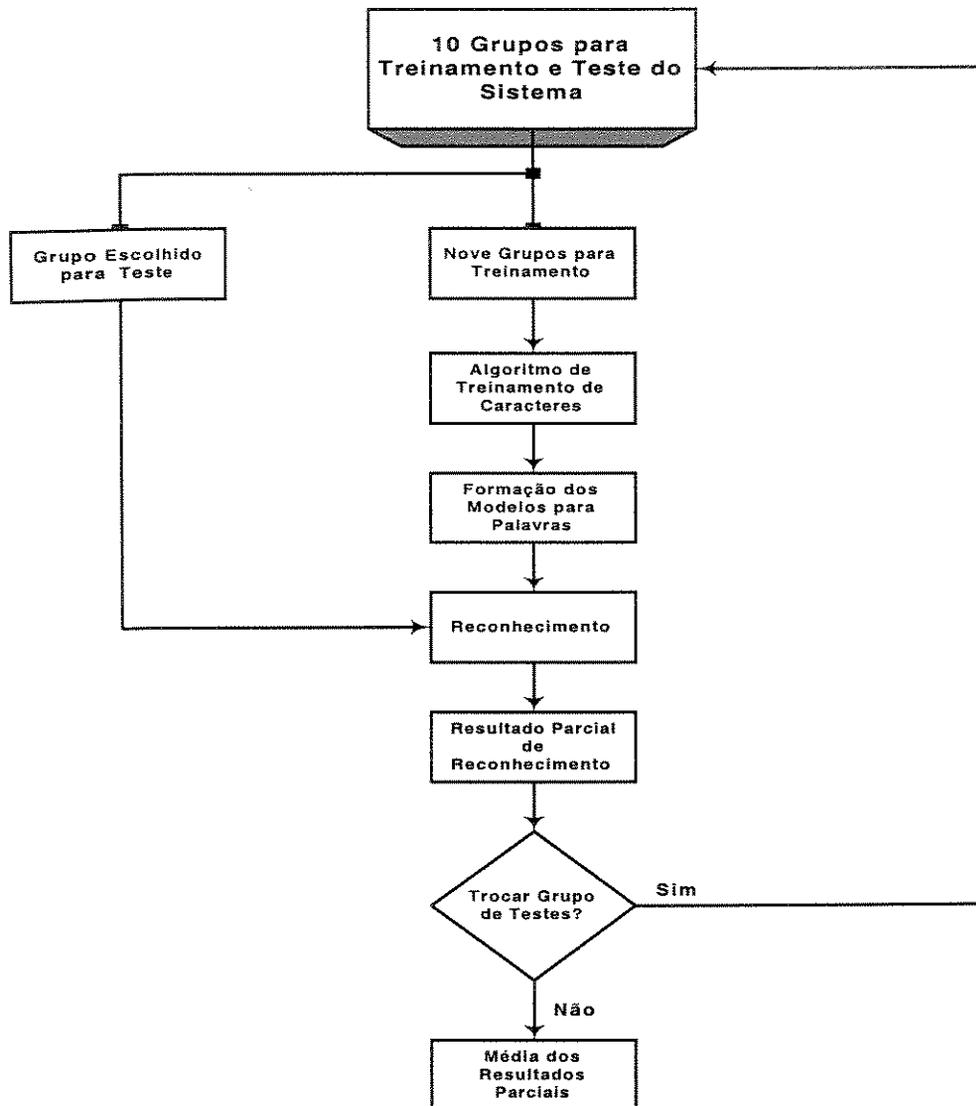


Figura 10.2 - Estratégia de treinamento e teste do sistema de reconhecimento.

10.2 AMOSTRAS DE CARACTERES PARA TREINAMENTO

Os modelos construídos para caracteres são treinados utilizando os caracteres resultantes da segmentação de cada palavra dos nove grupos de treinamento. Como explicado no Capítulo VII, essa segmentação nem sempre é perfeita, pois os segmentos obtidos podem representar partes de um caractere ou caracteres ainda conectados. Portanto, o procedimento

de treinamento deve observar uma possível segmentação dos caracteres de uma palavra em uma ou mais partes. O processo de treinamento empregado neste trabalho primeiramente segmenta as palavras manuscritas utilizando o algoritmo descrito no Capítulo VII. As frações da palavra resultantes do processo de segmentação são então, visualmente inspecionadas. A fração ou frações representando cada caractere de uma palavra são armazenadas com amostras de sua respectiva classe de caractere. Tal armazenamento não é automático, mas realizado manualmente com o objetivo de assegurar que a fração ou frações armazenadas representem um caractere completo. Os vetores de características das amostras armazenadas são construídos e, após uma quantização vetorial, são empregados para treinamento do HMM e FHMM modelando a classe de caractere considerada. Note que esse procedimento de treinamento considera que um caractere pode ser composto por uma ou mais frações da palavra. Desse modo leva-se em conta um possível seccionamento do caractere na fase de reconhecimento devido ao algoritmo de segmentação. O treinamento dos modelos de caracteres emprega o algoritmo descrito na seção 9.3 desta tese. A Tabela 10.3 mostra a quantidade média de caracteres armazenadas em cada um dos 10 grupos para treinamento do sistema.

Tabela 10.3 – Quantidade média de caracteres por grupo de treinamento.

Letras	Total de Letras		Letras	Total de Letras	
	Minúsculas	Maiúsculas		Minúsculas	Maiúsculas
a	91	13	o	148	0
c	58	0	q	28	9
d	29	11	r	53	6
e	225	15	s	102	50
h	3	1	t	189	18
i	107	0	u	56	0
l	11	1	v	13	30
m	19	2	z	35	10
n	153	19			

A Tabela 10.4 mostra o total de caracteres utilizados em todo o processo de treinamento e teste do sistema de reconhecimento de palavras manuscritas. Esse total é também ilustrado nas Figuras 10.3 e 10.4. Na Figura 10.3 é apresentado o número total de letras minúsculas existentes na base de dados, enquanto que na Figura 10.4 é ilustrado o total de letras maiúsculas. As letras do alfabeto que não aparecem na Tabela 10.4 possuem quantidade de amostras igual a zero nos gráficos apresentados nas Figuras 10.3 e 10.4. Tais letras não constam em palavras do vocabulário utilizado.

Tabela 10.4 – Total de caracteres empregados no processo de treinamento e teste.

Letras	Total de Letras		Letras	Total de Letras	
	Minúsculas	Maiúsculas		Minúsculas	Maiúsculas
a	914	129	o	1482	0
c	581	0	q	285	91
d	287	113	r	528	60
e	2248	15	s	1023	503
h	26	10	t	1888	178
i	1066	0	u	561	0
l	107	10	v	130	299
m	185	21	z	347	105
n	1529	191			

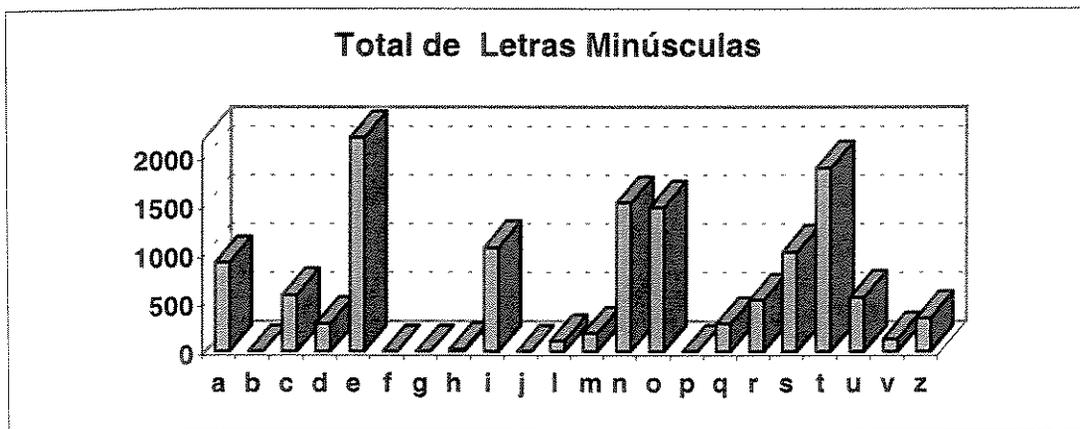


Figura 10.3 - Total de letras minúsculas para treinamento e testes do sistema de reconhecimento.

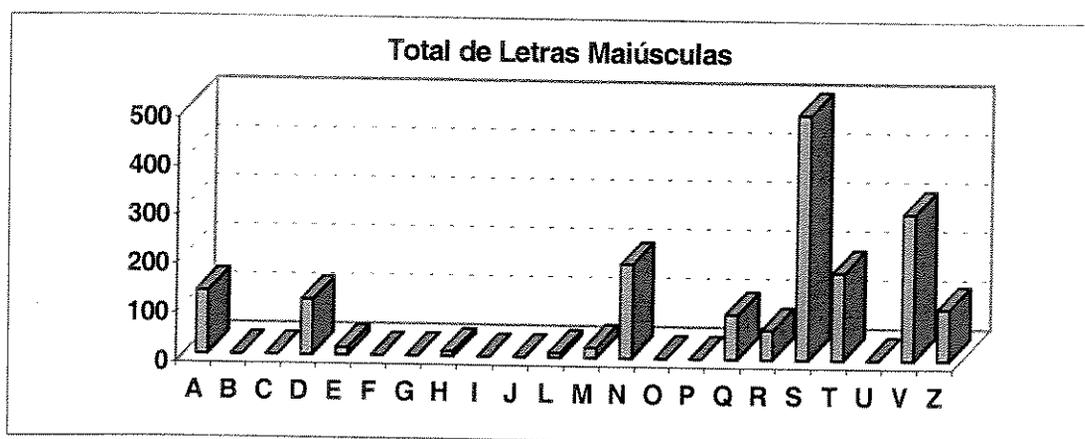


Figura 10.4 - Total de letras maiúsculas para treinamento e testes do sistema de reconhecimento.

10.3 RESULTADOS EXPERIMENTAIS

O resultado do processo de classificação é uma lista das possíveis classes de palavras a serem atribuídas a palavra de classe desconhecida. Essa lista está organizada em ordem decrescente de acordo com os valores obtidos pelo algoritmo de classificação. Na Tabela 10.5 os resultados de reconhecimento são apresentados considerando que a classe correta para uma palavra sendo reconhecida esteja, na primeira posição da referida lista ($N = 1$), entre as 5 primeiras posições da lista ($N = 5$), entre as 10 primeiras posições da lista ($N = 10$).

A percentagem de acertos apresentada na Tabela 10.5 para cada palavra, reflete a média dos resultados de reconhecimento considerando os 10 grupos de treinamento segundo os quais foi dividida a base de dados. Estes resultados são referentes a utilização dos HMMs como classificador do sistema de reconhecimento. Na Tabela 10.6 é apresentada a percentagem total de acerto em cada um destes 10 grupos.

Tabela 10.5 - Desempenho do sistema observada cada classe de palavras (Classificação via HMMs).

PALAVRAS	PERCENTAGEM DE ACERTO (%)			PALAVRAS	PERCENTAGEM DE ACERTO (%)		
	N=1	N=5	N=10		N=1	N=5	N=10
Um	64	87	97	Trinta	57	90	98
Dois	36	74	97	Quarenta	61	91	97
Três	49	80	90	Cinqüenta	49	82	92
Quatro	38	91	95	Sessenta	51	76	86
Cinco	23	63	79	Setenta	52	75	95
Seis	39	73	88	Oitenta	27	71	94
Sete	46	81	96	Noventa	54	78	89
Oito	41	85	93	Cem	56	100	100
Nove	61	89	95	Cento	41	82	93
Dez	37	84	100	Duzentos	54	93	98
Onze	84	100	100	Trezentos	70	98	100
Doze	42	91	100	Quatrocentos	83	100	100
Treze	70	87	100	Quinhentos	52	89	89
Quatorze	77	100	100	Seiscentos	24	57	76
Quinze	71	94	94	Setecentos	52	90	93
Dezesseis	33	81	92	Oitocentos	09	50	86
Dezessete	53	94	97	Novocentos	26	52	70
Dezoito	44	85	89	Mil	49	76	87
Dezenove	47	81	97	Centavos	47	90	99
Vinte	49	88	98	Reais	45	82	90

Tabela 10.6 - Desempenho do sistema observado cada grupo de treinamento (Classificação via HMMs).

GRUPOS	PERCENTAGEM TOTAL DE ACERTOS POR GRUPO (%)		
	N=1	N=5	N=10
1	52	83	94
2	49	83	94
3	48	84	94
4	52	86	94
5	45	82	91
6	50	80	90
7	47	84	97
8	42	80	93
9	46	83	93
10	48	81	92
Total Geral	48	83	93

Na Tabela 10.7 e 10.8 são apresentados os resultados de reconhecimento utilizando os FHMMs na classificação de palavras manuscritas. Na Tabela 10.7 esses resultados dizem respeito a percentagem total de acertos em cada um dos 10 grupos de treinamento considerados. A Tabela 10.8 apresenta a percentagem média de acertos obtida a partir das percentagens parciais para cada classe de palavra em cada um dos referidos grupos.

Tabela 10.7 - Desempenho do sistema considerando cada grupo de treinamento (Classificação via FHMMs).

GRUPOS	PERCENTAGEM TOTAL DE ACERTOS POR GRUPO (%)		
	N=1	N=5	N=10
1	51	82	95
2	50	85	93
3	47	83	95
4	53	85	96
5	51	83	93
6	52	80	92
7	51	83	97
8	44	78	93
9	50	82	92
10	47	81	93
Total Geral	50	82	94

Tabela 10.8 - Desempenho do sistema considerando cada classe de palavras
(Classificação via FHMMs).

PALAVRAS	PERCENTAGEM DE ACERTO (%)			PALAVRAS	PERCENTAGEM DE ACERTO (%)		
	N=1	N=5	N=10		N=1	N=5	N=10
Um	49	85	96	Trinta	56	89	100
Dois	40	75	95	Quarenta	60	92	99
Três	46	77	87	Cinqüenta	54	79	93
Quatro	45	89	93	Sessenta	38	73	89
Cinco	36	64	81	Setenta	53	75	93
Seis	42	75	89	Oitenta	41	73	98
Sete	34	80	95	Noventa	51	78	87
Oito	45	84	95	Cem	62	100	100
Nove	61	91	97	Cento	53	85	96
Dez	47	79	95	Duzentos	51	91	97
Onze	95	100	100	Trezentos	65	93	98
Doze	48	91	100	Quatrocentos	79	97	100
Treze	60	87	100	Quinhentos	44	89	89
Quatorze	86	100	100	Seiscentos	29	52	86
Quinze	65	94	94	Setecentos	52	90	93
Dezesseis	56	83	94	Oitocentos	14	77	86
Dezessete	53	88	97	Novencentos	37	44	78
Dezoito	41	85	93	Mil	50	75	90
Dezenove	64	92	94	Centavos	53	92	98
Vinte	56	82	98	Reais	45	82	90

A Tabela 10.9 apresenta uma combinação dos resultados apresentados nas Tabelas 10.5 e 10.8 devido a utilização dos HMMs e FHMMs, respectivamente. A Tabela 10.10 mostra as percentagens de acerto produzidas pela combinação dos resultados de classificação das Tabelas 10.6 e 10.7. A combinação dos classificadores é efetuada através do método denominado *borda count* [BCM], [Moh95], o qual baseia-se em uma atribuição de pesos às posições de classificação obtidas. Neste trabalho, a atribuição dos pesos para os dois processos de classificação, HMMs e FHMMs, se dá de forma semelhante ao exemplo a seguir, onde associa-se a 1ª posição o maior peso conforme o total de palavras do vocabulário.

Considerando como exemplo um vocabulário de 5 palavras e a combinação de dois classificadores, poderiam ser empregados os seguintes pesos: peso 5 para a 1ª posição; peso 4 para a 2ª posição; peso 3 para a 3ª posição; peso 2 para a 4ª posição e peso 1 para 5ª posição. Resumidamente, a combinação dos dois classificadores poderia ser expressa por:

$$V_{C_1, C_2} = w_1 \cdot P_{C_1} + w_2 \cdot P_{C_2}, \quad (10.1)$$

onde:

V_{C_1, C_2} representa o valor resultante da combinação dos classificadores C_1 e C_2 a ser analisado para obter uma nova posição de classificação para a classe atribuída a uma palavra;

w_1 peso relacionado a posição considerada no classificador C_1 ;

w_2 peso relacionado a posição considerada no classificador C_2 .

P_{C_1} é a posição da classe atribuída a uma palavra segundo o classificador C_1 ;

P_{C_2} é a posição da classe atribuída a uma palavra segundo o classificador C_2 ;

Tabela 10.9 - Desempenho do sistema produzido pela combinação de classificadores observada cada classe de palavras (Classificação via HMMs e FHMMs).

PALAVRAS	PERCENTAGEM DE ACERTO (%)			PALAVRAS	PERCENTAGEM DE ACERTO (%)		
	N=1	N=5	N=10		N=1	N=5	N=10
Um	61	87	97	Trinta	58	90	99
Dois	50	75	95	Quarenta	58	92	99
Três	56	80	87	Cinqüenta	52	82	92
Quatro	51	91	95	Sessenta	48	75	89
Cinco	33	64	77	Setenta	53	75	95
Seis	45	75	89	Oitenta	37	77	95
Sete	46	80	97	Noventa	49	79	91
Oito	46	85	95	Cem	62	100	100
Nove	68	89	95	Cento	49	86	95
Dez	58	79	100	Duzentos	52	92	97
Onze	89	100	100	Trezentos	63	96	100
Doze	48	91	100	Quatrocentos	86	97	100
Treze	60	90	100	Quinhentos	41	89	89
Quatorze	82	100	100	Seiscentos	33	52	76
Quinze	61	94	94	Setecentos	45	90	93
Dezesseis	67	83	94	Oitocentos	09	73	86
Dezessete	59	94	97	Novocentos	37	48	70
Dezoito	44	85	89	Mil	46	75	89
Dezenove	53	92	97	Centavos	47	92	99
Vinte	50	83	98	Reais	39	82	90

Tabela 10.10 – Resultado de classificação gerado pela combinação de HMMs e FHMMs considerando cada grupo de treinamento

GRUPOS	PERCENTAGEM TOTAL DE ACERTOS POR GRUPO (%)		
	N=1	N=5	N=10
1	55	84	95
2	50	83	94
3	50	84	94
4	56	87	95
5	50	84	92
6	53	80	91
7	51	84	97
8	43	80	94
9	46	83	92
10	52	83	93
Total Geral	51	83	94

10.4 ANÁLISE DOS RESULTADOS

As características utilizadas neste trabalho estão relacionadas ao modo como um caractere pode ser visto quando analisado os segmentos de linha que o compõem. Ao ser processada pelo sistema de reconhecimento, uma palavra manuscrita é transformada em uma seqüência ordenada dos segmentos de linha que compõem seus caracteres, a forma desses segmentos pode ser observada na Figura 8.2. Empregando essa seqüência de segmentos e os HMMs durante a fase de classificação do sistema de reconhecimento, obteve-se uma taxa de acerto de 48% conforme a Tabela 10.6. Quando utilizando os FHMMs no procedimento de classificação, tal taxa de acerto subiu para 50% como mostrado na Tabela 10.7. Com a combinação dos resultados obtidos pelos 2 classificadores, HMMs e FHMMs, foi atingida uma taxa de 51% de palavras corretamente reconhecidas conforme a Tabela 10.10. É importante observar que esse desempenho é devido principalmente as características utilizadas, uma vez que não foram efetuadas modificações nos algoritmos empregados no processo de classificação.

Entre os fatores que podem reduzir o desempenho do sistema está a utilização do processo de segmentação de palavras em caracteres e a concatenação de modelos para caracteres. Estes dois fatores provocam a perda de informações globais e de informações referentes a transição entre caracteres de uma palavra, podendo resultar em um menor desempenho quando comparado a sistemas que efetuam o reconhecimento de uma palavra sem segmentá-la. Note, entretanto, que neste trabalho um dos motivos para utilização de um processo de segmentação e posterior modelamento de palavras via concatenação de modelos para caracteres foi devido ao número de palavras existentes na base de dados. Experimentos anteriormente realizados demonstraram que o número de amostras de palavras para treinamento não é suficiente para uma estimação adequada dos parâmetros de HMMs modelando palavras diretamente, isto é, sem segmentar palavras. Porém, o processo de segmentação de palavras produz um número maior de amostras de caracteres, permitindo um melhor treinamento de HMMs quando modelando caracteres. Uma vantagem da concatenação de modelos para caracteres está relacionada a possibilidade de reconhecer palavras de outros vocabulários, pois o processo de concatenação possibilita a construção de modelos para diferentes palavras a partir de modelos para caracteres.

Uma comparação direta de resultados com outros trabalhos não é possível, pois essa comparação deveria ser realizada utilizando uma mesma base de dados. De qualquer forma, algumas análises podem ser feitas com base em aspectos de implementação e características utilizadas, a fim de se ter uma idéia dos resultados obtidos pelo sistema.

No sistema de reconhecimento desenvolvido em [Chen95] utilizando um dicionário de 271 palavras, foi empregada a segmentação explícita de palavras e um SCHMM (*Single Contextual* HMM) para modelamento de caracteres manuscritos. O processo de treinamento foi efetuado através da utilização de 1489 imagens e os testes do sistema envolveram 94 imagens

de palavras. Usando quantização vetorial para obter os símbolos a serem utilizados no HMM e uma modificação do algoritmo de Viterbi, foi alcançada uma taxa de acerto de 26%. Após a inclusão de um método para geração de hipóteses de classificação uma taxa de reconhecimento de 64,9% foi alcançada.

Em [Moh95] o sistema desenvolvido não utiliza uma segmentação explícita da palavra em caracteres mas emprega a concatenação de modelos para letras para construir modelos para palavras manuscritas. Em testes experimentais do referido sistema, foram empregadas 3106 imagens para treinamento e 317 imagens para testes, sendo consideradas 10 classes de palavras durante o processo de classificação. Utilizando uma função densidade de probabilidade gaussiana para modelar o conjunto de observações em cada estado de HMMs, foi atingida uma taxa de acerto de 68,8%.

Os resultados dos trabalhos acima, [Chen95], [Moh95], foram comentados em razão das técnicas de classificação utilizadas serem semelhantes as técnicas empregadas no sistema de reconhecimento descrito nesta tese. Esse sistema emprega uma quantização vetorial para obter os símbolos de observações usados nos HMMs (FHMMs) e o algoritmo de Viterbi para classificação de palavras. Não foram efetuadas modificações na estrutura desses algoritmos, sendo os resultados obtidos produtos do conjunto de características empregado.

Um critério de grande relevância ao analisar-se os resultados de reconhecimento obtidos é a não existência de restrições na base de dados quanto aos estilos e implementação da escrita (tipo de caneta, lápis, etc.). As palavras utilizadas para testar o sistema foram escritas em letras maiúsculas, minúsculas ou uma combinação desses dois tipos de letras, não havendo considerações sobre o tipo de material empregado nessa escrita. Em razão da não existência dessas restrições, o percentual de reconhecimento considerando a primeira escolha de classificação ($N = 1$) não foi muito alto. No entanto, é fundamental evidenciar o fato de que a

classe correta para uma palavra se encontra, na grande maioria das vezes, entre as 5 ou 10 primeiras posições da lista fornecida pelo classificador conforme mostrado na seção 10.3. Desse modo, é muito provável que tal classe seja recuperada com o emprego de alguma informação do contexto considerado. Por exemplo, no caso de processamento de cheques o resultado do reconhecimento do valor representado pelos dígitos poderia ser utilizado para validar o resultado do reconhecimento de palavras manuscritas.

É importante a observação de que o vocabulário considerado neste trabalho apresenta palavras com formas bastantes semelhantes, por exemplo, palavras como: *dezesesseis* e *dezenove*, *sessenta* e *seiscentos*, *quatrocentos* e *quinzentos* como mostrado no Anexo G. Isso dificulta a seleção de características discriminantes e uma determinação mais eficiente das fronteiras entre classes no espaço de características, gerando erros de classificação. Em outros idiomas esse vocabulário pode não apresentar essa semelhança de forma entre palavras, havendo uma maior separação entre classes e por conseguinte um menor erro de classificação.

10.5 ANÁLISE DO TRABALHO DESENVOLVIDO

O trabalho desenvolvido representa um sistema completo para reconhecimento de palavras manuscritas implementado em linguagem de programação C++. Além disso, o sistema trata do reconhecimento de palavras manuscritas de um vocabulário da língua portuguesa, podendo servir como parâmetro para discussão e desenvolvimento de novos modelos para palavras desse vocabulário.

No decorrer do trabalho foram discutidos problemas como correção de rotação, correção de inclinação e segmentação de palavras manuscritas. Algoritmos para correção de rotação e inclinação foram desenvolvidos a partir de uma combinação original de algoritmos

propostos na literatura. No processo de segmentação foram avaliados pontos potenciais de segmentação de uma palavra manuscrita sendo sugerido a distância geodésica para obter o contorno inferior da palavra.

Para obter resultados consistentes a respeito da base de dados empregada, foi utilizada uma validação cruzada dos dados de treinamento. Desse modo, foram avaliadas todas as variações de estilos existentes na base de dados, fazendo com que os resultados obtidos estejam mais próximos daqueles considerando situações práticas.

Tendo como base a análise de trabalhos da literatura, foi desenvolvida uma metodologia para extração de características de uma palavra manuscrita baseando-se em conceitos da teoria de conjuntos *fuzzy*. A utilização de conjuntos *fuzzy* permitiu analisar variações na forma e posição de segmentos compondo caracteres manuscritos. A extração destas características possibilitou empregar um método recente de classificação originado de uma generalização dos modelos ocultos de Markov. Desse modo puderam ser empregados dois tipos de classificadores utilizando, respectivamente, os HMMs e os FHMMs.

A probabilidade de símbolos para um HMM foi gerado a partir de verificação da frequência com que um vetor de características é classificado como um dos símbolos utilizados pelo sistema. No FHMM, os valores obtidos para a matriz \hat{B} (ver Capítulo V) refletem a similaridade de segmentos de um caractere em relação a cada um dos símbolos usados pelo sistema. Como o FHMM e HMM cometem erros diferentes devido as medidas utilizadas para obter seus parâmetros, efetuou-se as combinações desses dois classificadores. A combinação de classificadores vem sendo empregada em reconhecimento de manuscritos como uma forma para melhorar os resultados de classificação.

Em resumo, este trabalho propôs um método para extrair características consideradas, sob algum aspecto, relevantes ao reconhecimento de palavras manuscritas efetuado por um ser

humano, baseando-se na idéia de que essas características poderiam ser apropriadas para realização desse reconhecimento via computador. Para extração dessas características foram empregadas funções de pertinência procurando utilizar o conhecimento adquirido na análise de várias imagens de palavras manuscritas. Da análise de resultados pode-se verificar que a informação seqüencial originada dessas características produz resultados consistentes com aqueles gerados por sistemas utilizando técnicas de classificação semelhantes. Esses resultados podem ser melhorados com a utilização de informações relativas ao contexto e/ou com modificações estruturais dos algoritmos de classificação.

10.6 DIREÇÕES FUTURAS PARA O TRABALHO DESENVOLVIDO

A base de dados usada para obtenção dos resultados experimentais é pequena, tendo-se em vista as variações de escrita de uma palavra manuscrita e os estilos de escrita de cada pessoa. Essa questão é discutida em [GMV98], onde é mostrado que para uma taxa de erro de 1%, seria necessário uma base de dados contendo dez mil amostras. Entretanto, se levadas em conta variações como número de escreventes, estado da pessoa no momento da escrita, tipo de material, o número de amostras necessárias para atingir uma taxa de erro de 1% estaria em torno de duzentos mil. Portanto, aumentar a base de dados utilizada seria um primeiro passo para desenvolvimento de um sistema de reconhecimento de manuscritos com melhor desempenho e voltado a aplicações de mercado.

O emprego de quantização vetorial, em razão do número de dados de treinamento, pode não ter produzido um conjunto adequado de vetores para representar as regiões segundo as quais foi dividido o espaço de características considerado. A utilização de técnicas como os K vizinhos mais próximos [DH73] ou o emprego de uma mistura de gaussianas [Moh95]

poderia produzir uma melhor divisão do espaço de características e classificação de um vetor segundo as regiões deste espaço.

Uma redução da dimensão dos vetores de características através de uma análise de componentes principais [Fuk90] ou da utilização de algoritmos genéticos para seleção de características [PW96] pode ser uma alternativa para reduzir possíveis redundâncias entre classes, melhorando o desempenho final de reconhecimento do sistema.

Algumas palavras do vocabulário apresentam formas muito semelhantes, dificultando a extração de características capazes de separar suas classes. Pode-se observar esse problema em palavras como *dezesesseis* e *dezenove*, *quatrocentos* e *quinzentos*, *seiscentos* e *sessenta*. A definição de uma medida capaz de quantificar a dificuldade de classificação desses tipos de palavras permitiria uma melhor avaliação e aperfeiçoamento do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AB81] C. Arcelli and G. S. Di Baja. "A thinning Algorithm Based on Prominence Detection". *Pattern Recognition, Vol. 113, Nº 3, pp. 225-235, 1981.*
- [BCM] Borda Count Method - <http://www.as.ua.edu/CTL/math103/voting/bord.htm>.
- [BG83] M.K. Brown and S. Ganapathy. "Preprocessing Techniques for Cursive Script Word Recognition". *Pattern Recognition Vol. 16, nº 5, pp. 447-458, 1983.*
- [Bis95] Christopher M. Bishop. "Neural Networks for Pattern Recognition". *Oxford University Press, 1995.*
- [BK88] G. Baptista e K. M. Kulkarni "A High Accuracy Algorithm for Recognition of Handwritten Numerals". *Pattern Recognition, Vol. 21, nº 4, pp. 287-291, 1988.*
- [BP92] J. C. Bezdek, S. K. Pal. "Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data". *IEEE Press, 1992.*
- [BS89] Radmilo M. Bozinovic and Sargur N. Srihari. "Off-line Cursive Script Word Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, nº 1, pp. 68-83, Jan. 1992.*
- [Chen95] M. Chen, "Handwritten Word Recognition Using Hidden Markov Models". *Tese de Doutorado, University of New York at Buffalo, USA, 1995.*
- [CL96] Richard G. Casey and Eric Licolinet. "A Survey of Methods and Strategies in Character Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, nº 7, pp. 690-706, Jul. 1996.*

- [DH73] Richard O. Duda, Peter E. Hart "Pattern Classification and Scene Analysis". *John Willey & Sons, 1973.*
- [Fac96] J. Facon. "Morfologia Matemática: Teoria e Exemplos". *Editora Universitária Champagnat da Pontifícia Universidade Católica do Paraná, 1996.*
- [FNK92] H. Fujisawa, Y. Nakano and K. Kurino. "Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis". *Proceedings of IEEE, Vol. 80, nº 7, pp. 1079-1091, Jul., 1992.*
- [Fuk90] K. Fukunaga. "Introduction to Statistical Pattern Recognition". *Academic Press, 1990.*
- [Ger88] J. R. Gerônimo. "Medidas Fuzzy". *Tese de Mestrado, Universidade Estadual de Campinas, Dez., 1988.*
- [GL97a] N. R. Gomes, L. L. Lee. "Disconnected Handwritten Numeral Image Recognition". *4th International Conference on Document Analysis and Recognition, pp. 467-470, Ulm-Alemanha, Ago., 1997.*
- [GL97b] N. R. Gomes, L. L. Lee "Handwritten Numeral Recognition Via Fuzzy Logic and Local Discriminating Features". *First Brazilian Symposium on Document Image Analysis, pp. 225-236, Nov., 1997.*
- [GL99] N. R. Gomes, L. L. Lee. "Automatic Classification of Deformed Handwritten Numeral Characters". *5th International Conference on Document Analysis and Recognition, pp. 269-272, Bangalore-India, Set., 1999.*
- [GLL97] N. R. Gomes, L. L. Lee, M. Lizarraga., A. Koerich. "A Prototype for Brazilian Bankcheck Recognition". *International Journal of Pattern Recognition and Artificial Intelligence, vol.11, nº 4, pp. 549-570, 1997.*
- [GMV98] I. Guyon, J. Makhoeir, R. Schwartz, V. Vapnik. "What Size Test Set Gives Good Error Rate Estimates?" *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, nº 1, pp. 52-64, Jan., 1998.*
- [GO] R. M. Gray, R. A. Olshen. "Vector Quantization and Density Estimation". <http://www~isl.stanford.edu/~gray/compression.html>.

- [GS94] D. Guillevic, C. Y. Suen. "Cursive Script Recognition: A Sentence Level Recognition Scheme". *International Workshop on the Frontiers of Handwriting Recognition, Taipei, Taiwan, pp. 216-223, Dez., 1994.*
- [GS97] D. Guillevic, C. Y. Suen. "HMM Word Recognition Engine". *4th International Conference on Document Analysis and Recognition, pp. 544-547, Ulm-Alemanha, Ago., 1997.*
- [Gui95] D. Guillevic, "Unconstrained Handwritten Recognition Applied to the Processing of Bank Checks". *Tese de Doutorado, Concordia University, Montreal, Quebec, Canada, 1995.*
- [GW92] Rafael C. Gonzalez and Richard E. Woods. "Digital Image Processing". *Addison Wesley Publishing Company, 1992.*
- [Hei92] Henk J. A. M. Heijmans. "Mathematical Morphology as a Tool for Shape Description". *Workshop Shape in Picture, Driebergen, The Netherlands, Set., 1992.*
- [Ho92] T. K. Ho. "A Theorem of Multiple Classifier Systems and its Applications to Visual Word Recognition". *Tese de Doutorado, University of New York at Buffalo, Mai., 1992.*
- [HSZ87] Robert M. Haralick, Stanley R. Sternberg and Xinhua Zhuang. "Image Analysis Using Mathematical Morphology". *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, nº 4, pp. 532-550, 1987.*
- [KBH88] A. Kundu, Y. He, P. Bahl. "Word Recognition and Word Hypothesis Generation for Handwritten Script : A Hidden Markov Model Based Approach". *Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 457-462, 1988.*
- [KBH89] A. Kundu, Y. He, P. Bahl. "Recognition of Handwritten Word: First and Second Order Hidden Markov Model Based Approach". *Pattern Recognition, Vol. 22, nº 3, pp. 283-297, 1989.*
- [KG97] G. Kim e V. Govindaraju, "A Lexicon Drive Approach to Handwritten Word Recognition for Real-Time Applications". *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, nº 4, pp. 366-379, 1997.*
- [KK76] W. M. Kickert, H. Koppelaar. "Application of Fuzzy Set Theory to Syntactic Patterns Recognition of Handwritten Capitals". *IEEE Transactions on Systems, Man and Cybernetics, pp. 148-151, Fev., 1976.*

- [Koh95] R. Kohavi. "A Study of Cross Validation and Bootstrap for Accuracy Estimation and Model Selection". <http://robotics.stanford.edu/users/ronnyck>.
- [KT66] J. F. C. Kingman, S. J. Taylor. "Introduction to Measure and Probability". *Cambridge at the University Press, 1966*.
- [LB94] E. Lecolinet, O. Baret, "Cursive Word Recognition: methods and Strategies". In *Fundamentals in Handwriting Recognition – Ed. S. Impedovo, pp. 235-263, NATO Asi Séries F: Computer and Systems Sciences, Vol. 124, Springer Verlag, 1994*.
- [Lev94] J. Levine. "Programming for Graphics Files in C and C++". *John Willey & Sons, Inc., 1994*.
- [LM00] B. Lazzerini, F. Marcelloni. "A Linguistic Fuzzy Recognizer of Off-Line Handwritten Characters". *Pattern Recognition Letters, 21, pp. 319-327, 2000*.
- [LPP00] X. Li, M. Parizeau, R. Plamondon. "Training Hidden Markov Models with Multiple Observations - A Combinatorial Method". *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, nº 4, pp. 371-377, Abr., 2000*.
- [Moh95] M. A. Mohamed. "Handwritten Word Recognition Using Generalized Hidden Markov Models". *Ph.D. Dissertation presented to the Faculty of the Graduate School University of Missouri-Columbia, Mai., 1995*.
- [MS89] T. Murofushi and M. Sugeno. "An Interpretation of Fuzzy Measures and The Choquet Integral as an Integral with Respect to a Fuzzy Measure". *Fuzzy Sets and Systems, nº 29, pp. 201-227, 1989*.
- [Par97] J. R. Parker. "Algorithms for Image Processing and Computer Vision". *John Willey & Sons, Inc., 1997*.
- [Ped90] W. Pedrycz. "Fuzzy Sets in Pattern Recognition: Methodology and Methods". *Pattern Recognition, Vol. 23, Nº 1/2, pp. 121-146, 1990*.
- [PG98] W. Pedrycz e F. Gomide. "An Introduction to Fuzzy Sets Analysis and Design". *A Bradford Book, The MIT Press, 1998*.
- [PW96] Sankar. K. Pal, Paul P. Wang. "Genetic Algorithms for Pattern Recognition". *CRC Press, 1996*.

- [Rab89] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of IEEE*, vol.77 nº 2, Fev., 1989.
- [RJ93] L. Rabiner, B. Juang. "Fundamentals of Speech Recognition". *Prentice Hall Professional Technical Reference Abr.*, 1993.
- [SBM80] C. Y. Suen, M. Buthod and S. Mori "Automatic Recognition of Handprinted Characters - The State of the Art". *Proceedings of the IEEE*, Vol. nº 68, nº4, pp. 469-487, Abr., 1980.
- [SBS97] J. B. Santos, F. Bortolozzy e R. Sabourin. "A simple Methodology to Bank Cheque Segmentation". *First Brazilian Symposium on Document Image Analysis*, pp. 334-343, Nov., 1997.
- [SC74] P. Siy e C. S. Chen. "Fuzzy Logic for Handwritten Numeral Character Recognition". *IEEE Transactions on Systems, Man, and Cybernetics*, Nov., 1974.
- [Sen94] A. W. Senior, "Off-Line Cursive Handwriting Recognition Using Recurrent Neural Networks". *Tese de Doutorado, Cambridge University, Cambridge, Inglaterra*, 1994.
- [Ser82] J. Serra. "Image Analysis and Mathematical Morphology". *Academic Press Inc., London, England*, 1982.
- [SS99] I. S. Shaw e M. G. Simões. "Controle e Modelagem Fuzzy". *Editora Edgard Blücher Ltda., São Paulo - SP, Brasil*, 1999.
- [Sug77] M. Sugeno. "Fuzzy Measures and Fuzzy Integrals - A Survey". *Fuzzy Automata and Decision Process*, Eds. New York: North Holland, pp. 89-102, 1977.
- [TK90] H. Tahani and J. Keller. "Information Fusion in Computer Vision Using The Fuzzy Integral". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, nº 3, Mai., 1990.
- [WK92] Z. Wang and G. Klir. "Fuzzy Measure Theory". *Plenum Press, New York and London*, 1992.
- [WW94] G. Wang, J. Wang. "A New Hierarchical Approach for Recognition of Unconstrained Numerals". *IEEE Transactions on Consumer Eletronics*, Vol. 40, nº 3, pp. 428-436, Ago., 1994.

- [YCS99] Xiang Y. Ye, Mohamed Cherriet e Ching Y. Suen. "Model-Based Character Extraction from Complex Backgrounds". *International Conference on Document Analysis and Recognition*, pp. 511-514, Set., 1999.
- [YSS99] A. El-Yacoubi, M. Gilloux, R. Sabourin, C. Y. Suen. "An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, nº 8, pp. 752-760, Ago., 1999.
- [Zad65] L. A. Zadeh. "Fuzzy Sets". *Information Control*, Vol. 8, pp. 338-353, 1965.

ANEXO A

MÉTODO DOS MÍNIMOS QUADRADOS

O objetivo do método dos mínimos quadrados é determinar a reta

$$y = ax + b, \quad (\text{A.1})$$

que melhor se ajusta a um determinado conjunto de pontos. Para isso, são encontrados para equação acima os coeficientes a e b que minimizam a soma de quadrados $S(a,b)$ expressa por:

$$S(a,b) = \sum_{i=1}^n [y_i - (ax_i + b)]^2, \quad (\text{A.2})$$

onde (x_i, y_i) representam as coordenadas de cada ponto i pertencente ao conjunto de n pontos considerado.

Calculando as derivadas parciais de $S(a,b)$ em relação aos coeficientes a e b , e igualando a zero, obtêm-se:

$$a = \frac{\sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2}; \quad (\text{A.3})$$

$$b = \sum_{i=1}^n \bar{y} - a\bar{x}, \quad (\text{A.4})$$

sendo:

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n}, \quad (\text{A.5a})$$

$$\bar{y} = \sum_{i=1}^n \frac{y_i}{n}. \quad (\text{A.5b})$$

ANEXO B

ROTACIONAMENTO DE IMAGENS DE PALAVRAS MANUSCRITAS

Para normalizar a imagem de uma palavra manuscrita rotacionada é necessário alinhar a linha de base estimada, como explicado na seção 6.3, com o eixo de coordenadas do sistema [Gui95]. A Figura B.1 ilustra, através do eixo u , a linha de base de uma palavra manuscrita rotacionada por um ângulo φ do eixo horizontal x .

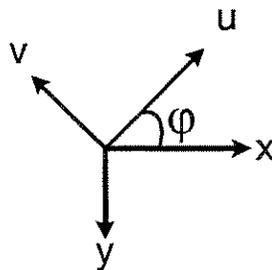


Figura B.1 - Eixos de coordenadas rotacionados.

As seguintes equações possibilitam o alinhamento dos eixos de coordenadas (x, y) e (u, v) , representados na Figura B.1.

$$v = x \cos \varphi + y \sin \varphi ; \quad (\text{B.1})$$

$$u = x \sin \varphi - y \cos \varphi . \quad (\text{B.2})$$

Onde:

(x, y) são as coordenadas originais do sistema;

(u, v) são as novas coordenadas do sistema;

φ é o ângulo entre a linha de base da palavra manuscrita e o eixo horizontal.

ANEXO C

HISTOGRAMAS CONSIDERANDO VÁRIOS ÂNGULOS DE INCLINAÇÃO DE UMA PALAVRA MANUSCRITA

Neste anexo é transcrito o algoritmo empregado em [Gui95] para construção de histogramas de uma palavra considerando seus possíveis ângulos de inclinação. É assumido que uma palavra pode ter sua inclinação média variando entre -70° e 70° graus. O algoritmo abaixo permite construir histogramas da palavra manuscrita em cada um dos ângulos entre -70° a 70° graus. Considera-se um intervalo entre ângulos de 5° graus, sendo portanto obtidos 29 histogramas para uma palavra manuscrita.

Algoritmo

1. Para cada pixel preto i de coordenada (x_i, y_i) na imagem:

1.1 Para cada ângulo de inclinação $(k \cdot \text{delta})$, sendo $k = -14, -13, \dots, 13, 14$:

a) Calcule o novo valor u da abscissa x_i no histograma da imagem:

$$u = x_i - (A_{im} - y_i) \cdot \tan(k \cdot \text{delta}), \text{ e } A_{im} = \text{altura da imagem};$$

b) Incremente o contador na u -ésima coluna do k -ésimo histograma:

$$\text{histograma}_k[u] = \text{histograma}_k[u] + 1.$$

O algoritmo descrito acima, considera as coordenadas do sistema como ilustrado na Figura C.1.

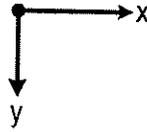


Figura C.1 - Sistema de coordenadas adotado para uma imagem.

ANEXO D

DIAGRAMA HORIZONTAL DE GRUPOS DE PIXELS

Neste anexo mostra-se como é construído o histograma horizontal de grupos de pixels de uma imagem binária. A Figura D.1 abaixo mostra o histograma para a imagem de uma palavra *treze*. Esse histograma representa o número de grupos de pixels pretos consecutivos em cada uma das linha que formam a imagem. Na Figura D.1 esse histograma é ilustrado por pequenos retângulos em negrito. O ponto de máximo do histograma corresponde à linha da imagem onde ocorre o maior número de grupos de pixels pretos da palavra. Esta linha tem seus grupos de pixels pretos assinalados em negrito na Figura D.1. O ponto de mínimo do histograma corresponde a linha da imagem possuindo apenas um grupo de pixels pretos. O grupo de pixels desta linha também está assinalado em negrito na Figura D.1.

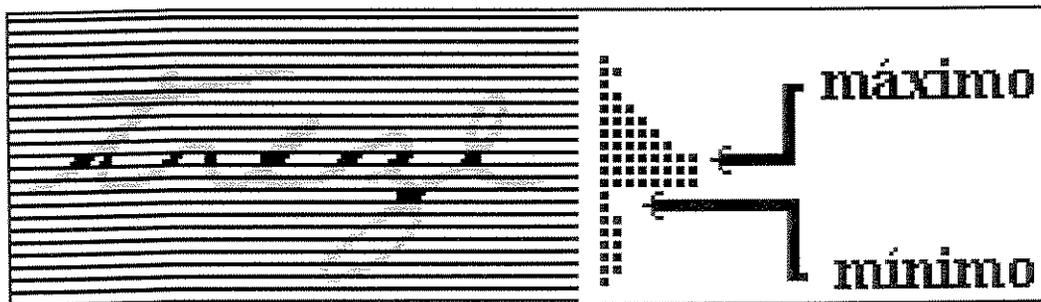


Figura D.1 - Histograma horizontal de grupo de pixels pretos de uma imagem da palavra *treze*.

ANEXO E

MORFOLOGIA MATEMÁTICA

A morfologia matemática, formalizada inicialmente por Georges Matheron e Jean Serra, constitui uma metodologia adequada para examinar a estrutura geométrica de formas de uma imagem. A idéia central é analisar a estrutura geométrica de uma imagem através do emprego de pequenos padrões de comparação [Hei92]. Variando o tamanho e a forma dos padrões de comparação, denominados elementos estruturantes, pode-se extrair informações úteis a respeito da forma da imagem. Neste anexo são descritos, resumidamente, operações da morfologia matemática empregadas no método de segmentação tratado no capítulo VII desta tese. Uma introdução mais detalhada das operações da morfologia matemática pode ser encontrada em [Fac96] e [HSZ87].

E.1 ALGUMAS OPERAÇÕES MORFOLÓGICAS BÁSICAS PARA IMAGENS BINÁRIAS

A dilatação e a erosão constituem duas operações básicas da morfologia matemática. Considerando A e B imagens binárias no espaço Euclidiano E^2 , sendo B um elemento estruturante, pode-se definir matematicamente a erosão de A por B pela seguinte expressão:

$$A \ominus B = \{x \in E^2 \mid (x+b) \in A \text{ para todo } b \in B\}. \quad (\text{E.1})$$

A dilatação de A pelo elemento estruturante B é expressa matematicamente por:

$$A \oplus B = \{x \in E^2 \mid x = a+b \text{ para } a \in A \text{ e } b \in B\}. \quad (\text{E.2})$$

Uma outra operação importante da morfologia matemática é a abertura binária. Em termos simples a abertura binária é definida como sendo uma erosão seguida por uma dilatação. A abertura pode ser usada para suavizar contornos e eliminar regiões da imagem menores que o elemento estruturante. Formalmente a abertura de uma imagem A por um elemento estruturante B é definida:

$$A \circ B = (A \ominus B) \oplus B. \quad (\text{E.3})$$

Exemplos das operações de erosão, dilatação e abertura binária são apresentados nas Figuras E.1(a), E.1(b) e E.1(c), respectivamente.

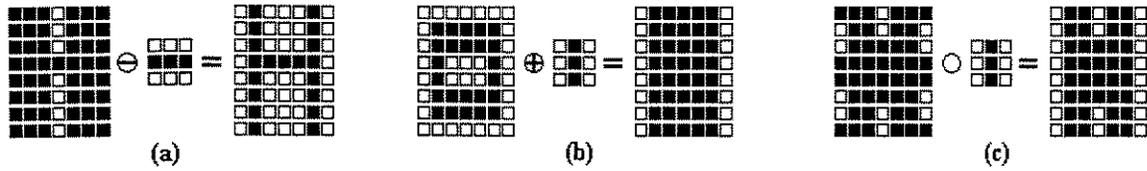


Figura E.1 - Operações morfológicas binárias de: (a) erosão; (b) dilatação; (c) abertura.

E.2 DILATAÇÃO CONDICIONADA

Algumas vezes deseja-se processar uma imagem apenas em determinadas regiões. Uma maneira de realizar tal processamento é a utilização de operadores condicionais. A dilatação de uma imagem A por um elemento estruturante B condicionada a uma imagem C é exemplificada na Figura E.2, onde o símbolo \cap representa a operação de interseção. Matematicamente a dilatação condicionada é definida por:

$$(A \oplus B)_C = (A \oplus B) \cap C. \tag{E.4}$$

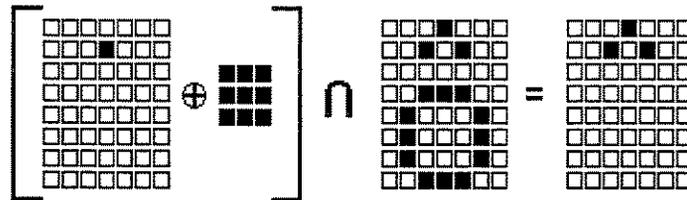


Figura E.2 - Exemplo de dilatação condicionada.

E.3 OPERAÇÃO DE ESPESSAMENTO DE UMA IMAGEM

Espessar uma imagem A é, de uma forma resumida, adicionar uma dada configuração de pontos a imagem. O espessamento de uma imagem binária pode ser realizado através de uma transformação *hit or miss*. A transformação *hit or miss* é uma ferramenta básica para detecção de formas e consiste em testar uma imagem A e seu complemento $A^c = \{x \in E^2 \mid x \notin A\}$ empregando, respectivamente, um elemento estruturante B_1 e um elemento estruturante B_2 . Essa transformação somente é possível se B_1 estiver contido em A e B_2 em A^c . Pode-se expressar a transformação *hit or miss* sobre uma imagem A por:

$$A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2), \quad (\text{E.5})$$

onde $B = (B_1, B_2)$.

Os elementos estruturantes B_1 e B_2 podem ser representados através duas matrizes como ilustra a Figura E.3, onde os pixels pretos são representados por “1” e os brancos por “0”.

0	0	0	0	0	0
1	0	0	0	0	1
0	0	0	0	0	0
B_1			B_2		

Figura E.3 - Par de elementos estruturantes.

Na literatura pode-se encontrar os elementos estruturantes B_1 e B_2 combinados em um único elemento estruturante [Fac96]. Essa combinação é mostrada na Figura E.4, onde

“X” representa os pixels de B_1 e B_2 da Figura E.3 coincidentes em posição e em valor. Os pixels de B_1 com valor 1 na Figura E.3 são representados por 1 na Figura E.4, enquanto que os de B_2 são representados por 0.

X	X	X
1	X	0
X	X	X

B

Figura E.4 - Representação de um par de elementos estruturantes para espessamento.

Utilizando a definição da transformação *bit or miss*, o espessamento de uma imagem pode ser dado pela seguinte expressão:

$$A \text{ esp } B = A \cup (A \otimes B). \quad (\text{E.6})$$

A operação completa de espessamento de um conjunto ou imagem é, geralmente, obtida após várias iterações nas quais são utilizadas um ou vários pares de elementos estruturantes. Em outras palavras, a imagem resultante do espessamento da iteração $i = 1, 2, 3, \dots, n$ é novamente espessada até não ocorrer mais modificações. A expressão matemática abaixo, expressa o espessamento de uma imagem como uma operação seqüencial [GW92], onde são utilizados os elementos estruturantes $\{B\} = B^1, B^2, \dots, B^n$.

$$A \text{ esp } \{B\} = \left(\left(\dots \left((A \text{ esp } B^1) \text{ esp } B^2 \right) \dots \right) \text{ esp } B^n \right). \quad (\text{E.7})$$

A Figura E.5 apresenta o resultado final de um espessamento utilizando apenas um par de elementos estruturantes.

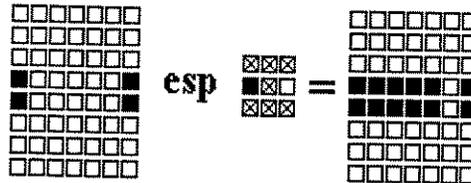


Figura E.5 - Exemplo de espessamento de uma imagem.

Para realizar a operação de espessamento, citada na seção 7.2 desta tese, foi utilizado o conjunto de elementos estruturantes apresentado na Figura E.6.

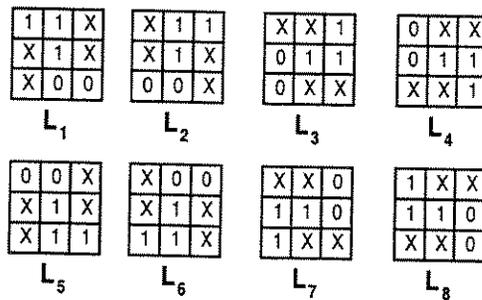


Figura E.6 - Família de elementos estruturantes utilizada na operação de espessamento.

E.4 DILATAÇÃO E EROÇÃO EM NÍVEIS DE CINZA

Muitos operadores morfológicos binários foram estendidos a operações em níveis de cinza. Neste trabalho, são abordadas apenas as definições de erosão e dilatação. A operação de dilatação é utilizada na implementação do algoritmo de distância geodésica descrito no final deste Anexo.

Tratando uma imagem digital e um elemento estruturante como sendo as respectivas funções f e g , que assinalam um nível de cinza a cada par de coordenadas $(x, y) \in Z \times Z$, a dilatação e erosão de $f(x, y)$ por $g(x, y)$ podem ser definidas [HSZ87], respectivamente, por:

$$(f \oplus g)(s, t) = \max\{f(s-x, t-y) + g(x, y) \mid (s-x, t-y) \in D_f; (x, y) \in D_g\}; \quad (\text{E.8})$$

$$(f \ominus g)(s, t) = \min\{f(s+x, t+y) - g(x, y) \mid (s+x, t+y) \in D_f; (x, y) \in D_g\}, \quad (\text{E.9})$$

onde D_f e D_g representam os respectivos domínios de f e g . Na Figura E.7 é ilustrada a operação de dilatação em níveis de cinza para funções de uma variável. A expressão para dilatação se reduz, então, a:

$$(f \oplus g)(s) = \max\{f(s-x) + g(x) \mid (s-x) \in D_f; x \in D_g\}; \quad (\text{E.10})$$

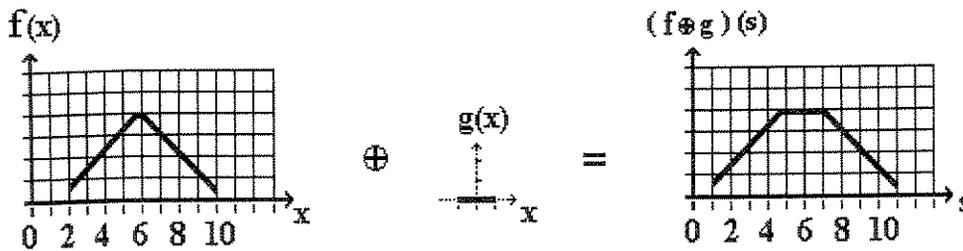


Figura E.7 - Exemplo de dilatação em níveis de cinza de uma função $f(x)$ por um elemento estruturante $g(x)$.

E.5 DISTÂNCIA GEODÉSICA

A distância geodésica entre dois pontos P_1 e P_2 pertencentes a uma imagem S é o menor caminho contido em S ligando estes pontos. Para obter a distância geodésica entre dois pontos P_1 e P_2 pertencentes a uma imagem S , pode-se construir um mapa de distâncias [Chen95] a partir do ponto inicial P_1 até atingir o ponto final P_2 .

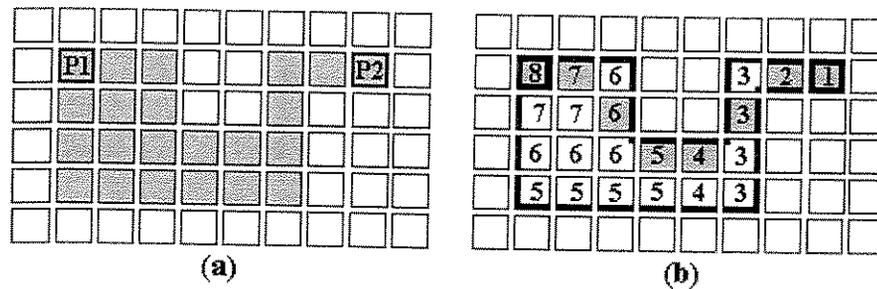


Figura E.8 - Distância geodésica: (a) Imagem S com pontos P_1 e P_2 assinalados; (b) Mapa de distâncias e o traçado do caminho obtido entre os pontos P_1 e P_2 .

A Figura E.8(b) exemplifica um mapa de distâncias entre os pontos P_1 e P_2 da imagem S ilustrada na Figura E.8(a).

A seguir, é apresentado o algoritmo usado em [Chen95] para construção de um mapa de distâncias. Inicialmente, o mapa de distâncias tem todos os valores de seus pontos iguais a zero. O valor no ponto de partida P_1 é, então, iterativamente incrementado de 1, sendo realizada uma dilatação em níveis de cinza do mapa condicionada ao conjunto S . O caminho de menor distância pode ser encontrado a partir de P_2 , selecionando pontos de maior valor do mapa em uma vizinhança 8 conectada.

Algoritmo:

1. Inicializar cada ponto da imagem mapa de distâncias, Map , com zero;
2. Enquanto $Map(P_2) = 0$:
 - 2.1 Incrementar mapa de distância no ponto P_1 : $Map(P_1) = Map(P_1) + 1$;
 - 2.2 Realizar a dilatação em níveis de cinza condicionada ao conjunto S :
$$Map = (Map \oplus B) \cap S$$
, B é o elemento estruturante escolhido para a operação de dilatação.

ANEXO F

PROCEDIMENTO DE AFINAMENTO

O procedimento de afinamento é uma técnica bastante conhecida em processamento de imagens, sendo inúmeros algoritmos encontrados na literatura. Por esse motivo, este anexo é dedicado a dar apenas uma idéia sobre o processo de afinamento e esqueleto de uma imagem aquelas pessoas não familiarizadas com estes conceitos.

F.1 AFINAMENTO E ESQUELETO DE UM OBJETO EM UMA IMAGEM

A obtenção do esqueleto é, geralmente, um dos primeiros passos tomados por um sistema de visão computacional quando tentando extrair características de um objeto em uma imagem. O esqueleto é dito representar a forma de um objeto em um número mínimo de pixels. O procedimento de afinamento produz o esqueleto do objeto sendo analisado e pode ser definido como o ato de identificar aqueles pixels essenciais para representar a forma do objeto. Como existem vários algoritmos de afinamento propostos na literatura [GW92], [Par97], [AB81], este anexo apresenta uma definição de esqueleto de um objeto como uma transformação de eixo médio.

F.2 TRANSFORMAÇÃO DE EIXO MÉDIO

O esqueleto de uma região pode ser definido através de uma transformação de eixo médio [GW92], da seguinte forma:

“Seja X uma região com bordas δX . Denomina-se x um ponto pertencente a X . Para cada ponto x encontra-se seus vizinhos mais próximos em δX . Se x tem mais do que um vizinho, ele faz parte do esqueleto de X (eixo médio).”

A figura F.1 mostra um exemplo do esqueleto (linhas tracejadas) de uma região X com bordas δX (linhas contínuas) quando a distância Euclidiana é usada.

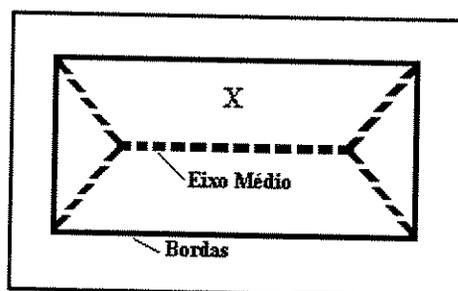


Figura F.1 - Esqueleto da imagem X representado por linhas tracejadas.

A Figura F.2(b) apresenta o esqueleto da palavra *quarenta* resultado do processo de afinamento da Figura F.2(a). O algoritmo empregado nesta tese é uma combinação de 3 algoritmos desenvolvidos respectivamente por, Zhang-Suen, Stentiford e Holey. Esse algoritmo [Par97] conserva a topologia e a conectividade das imagem de palavras manuscritas utilizadas neste trabalho.

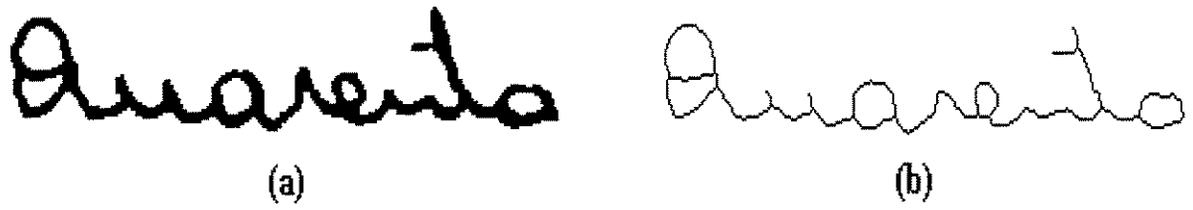


Figura F.2 - Esqueleto da palavra *quarenta*. (a) Palavra original; (b) Palavra após afinamento.

ANEXO G

AMOSTRAS DE PALAVRAS MANUSCRITAS

Neste anexo são mostradas algumas amostras de palavras manuscritas existentes na base de dados empregada neste trabalho. Observe que muitas dessas palavras são escritas utilizando uma combinação de letras maiúsculas e minúsculas. Outras palavras são escritas somente com letras minúsculas ou somente com letras maiúsculas.

UM dois três Quatro
cinco Seis sete oito
Nove dez onze doze
treze quatorze quinze
dezesseis dezessete dezoito
dezenove vinte trinta

um um um um um
 dois dois dois dois dois
 dois dois dois dois dois
 três três três três três
 três três três três três
 quatro quatro quatro quatro
 quatro quatro quatro quatro
 quatro quatro quatro quatro
 cinco cinco cinco cinco
 cinco cinco cinco cinco
 seis seis seis seis seis
 seis seis seis seis seis
 sete sete sete sete sete
 sete sete sete sete sete
 oito oito oito oito oito
 nove nove nove nove nove
 nove nove dez dez dez

onze onze onze onze ONZE
 doze doze doze doze Treze
 treze treze Treze TREZE
 catorze quatorze Quatorze
 Quatorze quinze quinze
 quinze quinze dezesseis
 dezesseis dezesseis dezesseis
 dezesseis dezesseis ~~dezesseis~~
 dezoto dezoto Dezoto
 dezanove dezanove dezanove
 Dezanove VINTE VINTE
 vinte vinte vinte vinte

trinta trinta trinta trinta
 quarenta QUARENTA
 quarenta quarenta quarenta
 cinquenta cinquenta
 CINCUENTA sessenta
 sessenta sessenta sessenta
 setenta SETENTA setenta
 setenta SETENTA oitenta
 oitenta Oitenta oitenta
 Oitenta NOVENTA noventa
 noventa NOVENTA
 NOVENTA cem Cem Cem

cento cento cento cento
cento duzentos duzentos
duzentos duzentos duzentos
duzentos trezentos trezentos
quatrocentos quatrocentos
quinhentos quinhentos quinhentos
seiscentos seiscentos seiscentos
seiscentos seiscentos seiscentos
setecentos setecentos
oitocentos oitocentos oitocentos
oitocentos oitocentos oitocentos

noventa ~~noventa~~ nove centos

Nove centos mil mil Mil

mil mil mil mil

centos ~~centos~~ centos

centos centos CENTAVOS

reais reais REAIS

reais reais reais reais