Sistema de Reconhecimento de Palavras Manuscritas Dependente do Usuário

Luciana Ribeiro Veloso

Tese submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica .

Área de Concentração: Processamento da Informação

João Marques de Carvalho, Ph.D.
Orientador

Campina Grande, Paraíba, Brasil
©Luciana Ribeiro Veloso, Março de 2009

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

V343s
2009 Veloso, Luciana Ribeiro.
Sistema de reconhecimento de palavras manuscritas dependente do usuário / Luciana Ribeiro Veloso. — Campina Grande, 2009.
144 f.: il.

Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.
Referências.
Orientadores: Prof. Dr. João Marques de Carvalho.

1. Processamento Digital de Imagens. 2. Reconhecimento de Padrões. 3. Reconhecimento de Manuscritos. 4. Análise de Documentos. I. Titulo.

CDU – 004.383.5(043)

Sistema de Reconhecimento de Palavras Manuscritas Dependente do Usuário

Luciana Ribeiro Veloso

Tese apresentada em Março de 2009

João Marques de Carvalho, Ph.D.
Orientador

Cinthia Obladen de Almeida Freitas, D. Sc.
Componente da Banca

Luiz Eduardo S. Oliveira, Ph.D.
Componente da Banca

Herman Martins Gomes, Dr.
Componente da Banca

Benedito Guimarães Aguiar Neto, Dr.-Ing.
Componente da Banca

Campina Grande, Paraíba, Brasil, Março de 2009

Dedicatória

Dedico este trabalho a memória de meu pai, Areoaldo e a minha mãe, Dinalva, exemplos de luta e respeito à vida, e aos meus filhos, Heloysa e Pedro Henrique.

Agradecimentos

Gostaria de expressar meus sinceros agradecimentos a algumas pessoas e instituições que contribuíram para a realização deste trabalho.

Agradeço, primeiramente a Deus, pelas graças recebidas que fizeram possível a realização deste trabalho.

Ao prof. João Marques, por haver despertado em mim o interesse por processamento de imagens, sua valiosa orientação e dedicação, imprescindíveis em todas as fases deste trabalho. Por seus diversos e incontáveis conselhos.

À prof. Cinthia Freitas por sua ajuda e incentivo. Seus conselhos e suas palavras de encorajamento, durante este longo trabalho, foram um bálsamo para meu coração aflito.

Ao prof. Herman Gomes por sua orientação, contribuindo para aumentar e solidificar meus conhecimentos, além de entender a minha necessidade de concluir o doutorado, liberando parte do tempo de trabalho no projeto iPhotoBot para a conclusão dos trabalhos referentes ao doutorado.

Aos meus filhos, Heloysa e Pedro Henrique, por entenderem e me apoiarem, mesmo quando tive que trabalhar nos finais de semana, feriados e noites, sem tempo para as nossas brincadeiras.

Ao meu esposo Jean, por sua compreensão, carinho e ajuda nas implementações.

Aos meus grandes amigos, Josemar, Rinaldo e Madeiro, pelas longas conversas, cheias de sugestões e críticas, que possibilitaram o desenvolvimento deste trabalho.

A Thiago, Julio e Danyelle, sem os quais a tarefa de aquisição do banco de imagens ficaria inviável.

À prof. Joseana, pela inspiração na proposta do sistema.

A todos os que fazem a COPELE, em especial à Ângela e Pedro, pelo apoio constante.

Enfim, a todos que de algum modo contribuíram para a realização deste trabalho.

Resumo

Este trabalho apresenta um sistema de reconhecimento de palavras manuscritas isoladas dependente do escritor. Este sistema caracteriza-se por utilizar uma etapa de pré-processamento, que visa corrigir imperfeições e normalizar variações na imagem da palavra manuscrita, uma etapa de segmentação explícita, que visa dividir a palavra em caracteres ou segmentos de caracteres, uma etapa de extração de características, que tem por finalidade representar a imagem por três vetores de características (perceptivas, globais e direcionais) e um módulo de quantização vetorial, que tem o objetivo de realizar o mapeamento de um vetor de características em um vetor de observação (ou vetor de símbolos). Os símbolos correspondem aos índices (dos vetores-código) gerados na representação (quantização vetorial) da sequência de características com o uso dos dicionários. Finalizando, tem-se a etapa de classificação realizada por Modelos Escondidos de Markov, na qual os caracteres são reconhecidos individualmente e combinados para formar a palavra. Testes experimentais foram realizados com uma base de dados construída especificamente para este fim, contendo amostras de manuscritos de 4 escritores distintos. O sistema de reconhecimento de palavras manuscritas isoladas dependente do escritor obteve taxas de reconhecimento que variaram entre 83,31% a 92,96% dependendo do escritor analisado. Os resultados apresentados mostram que o sistema apresenta um ótimo desempenho quando utilizado para reconhecer palavras através dos modelos de caracteres.

Palavras chaves: reconhecimento de padrões, reconhecimento de manuscritos, processamento digital de imagens, análise de documentos.

Abstract

This work presents a writer-dependent system for isolated handwritten cursive word recognition. This system is characterized by the utilization of a pre-processing state, which corrects imperfections and normalizes variations in the word image, an explicit segmentation stage, which splits the word into characters or character segments, a feature extraction stage, which represents the image by three feature vectors (perceptive, global and directional features), and a vector quantization module, which performs the mapping of a feature vector into an observation vector (or symbols vector). The symbols correspond to indices (the code vectors) generated by the representation (vector quantization) of the feature sequences with the use of dictionaries. Finally, there is the classification stage, performed by Hidden Markov Models, where characters are individually recognized and combined to form a valid word. Experimental tests were conducted with a database specifically built for this problem, containing samples of manuscripts from 4 different writers. The writer-dependent system for isolated handwritten cursive word recognition was recognition rate between 83.31% and 92.96% depending writer analyzed. The results show that the system offers optimum performance when used word recognize by the characters models.

Conteúdo

1 Introdução **1**

1.1 Comunicação homem-máquina 1

1.2 Motivação 5

1.3 Objetivos do trabalho 6

1.4 Organização do trabalho 8

2 Reconhecimento de palavras manuscritas - Revisão bibliográfica **10**

2.1 Pré-processamento 12

2.2 Segmentação de palavras em caracteres 14

2.3 Extração de características e reconhecimento 19

2.4 Comentários finais 30

3 Descrição geral do sistema proposto **33**

3.1 Pré-Processamento 33

3.2 Segmentação 35

3.3 Gerador de sub-imagens 35

3.4 Extração de características 35

3.5 Quantização vetorial 35

3.6 Reconhecimento estatístico 37

3.7 Comentários finais 38

4 Descrição dos módulos implementados **39**

4.1 Pré-processamento 39

4.1.1 Normalização da inclinação média dos caracteres da palavra . . 39
4.1.2 Normalização da inclinação da linha de base da palavra 43
4.1.3 Suavização 46
4.1.4 Análise dos resultados 49
4.2 Segmentação de palavras manuscritas 52
4.2.1 Segmentação nos traços de ligação 53
4.2.2 Segmentação de caracteres fortemente conectados ou sobrepostos 68
4.2.3 Resultados experimentais 71
4.3 Extração de características 75
4.3.1 Vetor de características perceptivas 77
4.3.2 Vetor de características globais 80
4.3.3 Vetor de características direcionais 81
4.4 Quantização vetorial 82
4.4.1 Visão geral 83
4.4.2 Algoritmo LBG 83
4.4.3 Projeto do quantizador vetorial e do seu dicionário 84
4.5 Modelos Escondidos de Markov 85
4.5.1 Fundamentos dos Modelos Escondidos de Markov 86
4.5.2 Parâmetros do modelo 87
4.5.3 Projeto dos Modelos Escondidos de Markov 92
4.6 Cometários finais 94

5 Resultados experimentais 96
5.1 Base de imagens 96
5.1.1 Base de palavras manuscritas dependente do escritor 97
5.1.2 Base de dados dos meses do ano 102
5.1.3 Base IRONOFF 102
5.2 Análise dos experimentos realizados com a base de dados dos meses do ano 105
5.3 Análise dos experimentos realizados com a base de caracteres IRONOFF 110

5.4 Análise do resultados com a base de dados dependente do usuário . . . 113
5.4.1 Experimento com a base de palavras manuscritas produzidas pelo escritor 1 114
5.4.2 Experimento com a base de palavras manuscritas produzidas pelos escritores 2, 3 e 4 119
5.5 Comentários finais 124

6 Conclusão **126**

Lista de Figuras

2.1 Diagrama de blocos de um sistema geral de reconhecimento de palavras manuscritas. 12
2.2 Linhas de base. 14
2.3 Forma intuitiva de reconhecimento de palavras manuscritas: (a) palavra manuscrita; (b) separação dos segmentos; (c) reconhecimento do primeiro segmento; (d) análise do segundo segmento; e) sobresegmentação do segundo segmento. 15
2.4 Exemplos de palavras manuscritas. 15

3.1 Diagrama de blocos do sistema de reconhecimento de palavras manuscritas dependente do usuário. 34
3.2 Exemplos de seqüências de sub-imagens (hipóteses de segmentação), partindo de pontos fixos determinados pelo algoritmo de segmentação. . . 36

4.1 Resultados da operação de abertura em amostras de palavras. (a), (b) e (c):imagens originais; (d), (e) e (f): imagens resultantes da operação de abertura. 40
4.2 Correção da inclinação das letras da palavra utilizando o histograma de projeção vertical: (a) imagem original, (b) imagem normalizada, (c) e (d) são os respectivos histogramas de projeção vertical. 42
4.3 Normalização da inclinação vertical dos caracteres das palavras. (a) imagem original, (b) imagem normalizada a partir da imagem original e (c) imagem normalizada a partir dos traços verticais da palavra. 43

4.4 (a) Imagem original, (b) Contorno inferior da imagem original. 44
4.5 Representação gráfica do cálculo da nova coordenada (i', j'). 45
4.6 Normalização da inclinação da linha de base das palavras. (a) imagem original, (b) imagem normalizada. 46
4.7 Exemplo de ruído provocado pelos processos de normalização da inclinação média dos caracteres e da inclinação da linha de base. (a) imagem original, (b)imagem normalizada em inclinação vertical e inclinação da linha de base. 47
4.8 Máscaras utilizadas para remoção de ruído pontual. 47
4.9 Máscaras utilizadas para remoção de ruído de 2 *pixels*. 48
4.10 Exemplo de ruído provocado pelos processos de normalização da inclinação média dos caracteres e da inclinação da linha de base. (a)imagem normalizada em inclinação vertical e inclinação da linha de base, (b) imagem suavizada pelo algoritmo desenvolvido. 48
4.11 Resultado do processo de normalização da inclinação da linha de base da palavra. (a) imagem original, (b) contorno inferior da imagem original e (c) resultado final do processo. 50
4.12 Resultado do processo de normalização da inclinação da linha de base da palavra. (a) imagem original, e (b) resultado final do processo. . . . 50
4.13 Resultado final do pré-processamento. (a) imagem original, (b)imagem normalizada na inclinação vertical e linha de base e suavizada. 51
4.14 Exemplo de ligações entre caracteres e caracteres fortemente conectados. O ponto de segmentação entre caracteres fortemente conectados é ilustrado através dea seta. 53
4.15 Exemplo das ilhas e pontes na palavra *amigo*: (a) palavra original; (b)ilhas e (c)pontes. 54
4.16 Vizinhança de um ponto p. 56
4.17 Processo de rotulação de ilhas: (a) palavra manuscrita; (b)ilhas. 57
4.18 Exemplo de uma treliça. 60

4.19 Histograma de projeção vertical. 61
4.20 Exemplos de segmentação natural nos seguintes ângulos a) -20, b) -10, c) 0, d) 10, e) 20 graus, f) união entre os pontos de segmentação natural g) resultado final do processo de segmentação natural. 63
4.21 Elementos estruturantes e cromossomos. 64
4.22 Imagem de entrada e de referência. 65
4.23 Palavra cursiva e seu histograma de projeção vertical. 67
4.24 (a) união dos pontos de segmentação produzidos pela combinação dos resultados dos algoritmos de segmentação e (b) resultado final do processo de segmentação após a etapa de pós-segmentação. 71
4.25 Exemplos de segmentação (método 1). 72
4.26 Segmentação de palavras manuscritas (método 2). 73
4.27 Exemplos de segmentação (método 3). 74
4.28 Falhas na determinação dos pontos de segmentação entre caracteres fortemente conectados. 74
4.29 Resultado final do processo de segmentação utilizando o método 3 e o método dos mínimos locais. 74
4.30 Exemplos de palavras mal segmentadas. 75
4.31 Exemplo do processo de detecção das zonas da palavra. 78
4.32 Exemplo do processo de rotulação utilizado pelo extrator de características direcionais. 82
4.33 Um modelo escondido de Markov de um caractere com 6 estados. . . . 92
4.34 Um modelo escondido de Markov de um caractere com 11 estados. . . . 92
4.35 Um modelo escondido de Markov de um caractere com 16 estados. . . . 93

5.1 Amostras da base de dados dos meses do ano. 102
5.2 Amostras de imagens de caracteres presente na base de dados IRONOFF. 105
5.3 Zoneamento em 7 partes 109

5.4 Amostras da base de palavras redigidas pelo escritor 4. Estas imagens correspondem as seguintes palavras, no sentido esquerda direita, de cima para baixo: ser, básicas, um, existentes, manuscritos, baseada e bloco. . 120

Lista de Tabelas

2.1 Características utilizadas pelos algoritmos de segmentação 20

2.2 Desempenho de alguns sistemas de reconhecimento de manuscritos . . . 31

2.3 Desempenho de alguns sistemas de reconhecimento de manuscritos (continuação) 32

4.1 Conversão da rotulação dos pixels pelo algoritmo de extração das características direcionais. 81

5.1 Total de amostras na base de dados em formato PCX e PAV. 99

5.2 Exemplos de Palavras Manuscritas. 99

5.3 Informações sobre os escritores. 100

5.4 Maiores(em número de letras), menores, tamanho médio e o desvio padrão (STD) das palavras na base. 100

5.5 Palavras por escritores nos conjuntos de treinamento/validação e teste. 101

5.6 Número de amostras no conjunto treinamento/validação (Trn/Vld) e teste (Tst) para os escritores 1, 2, 3 e 4. (caracteres normais) 103

5.7 Número de amostras no conjunto treinamento/validação (Trn/Vld) e teste (Tst) para os escritores 1, 2, 3 e 4. (caracteres acentuados) 104

5.8 Taxa de reconhecimento obtida pelo classificador HMM utilizando diferentes espaços de características. 106

5.9 Taxa de reconhecimento do classificador HMM utilizando combinação de vetores de características. 108

5.10 Taxa de reconhecimento do classificador NN 109

5.11 Taxas de reconhecimento (%) dos caracteres da base IRONOFF quando utilizado HMM. 110
5.12 Taxas de reconhecimento dos caracteres da base IRONOFF quando utilizado HMM T11 ou HMMT16. 112
5.13 Taxas de reconhecimento dos caracteres da base IRONOFF quando utilizado NN. 112
5.14 Taxas de reconhecimento dos caracteres da base IRONOFF quando utilizados HMM, sem pré-processamento e segmentação. 113
5.15 Taxas de reconhecimento para a base de teste dos caracteres do escritor 1.115
5.16 Matriz de confusão para o escritor 1 (%). 116
5.17 Matriz de confusão para o escritor 2 (%). 120
5.18 Matriz de confusão para o escritor 3 (%). 121
5.19 Matriz de confusão para o escritor 4 (%). 122
5.20 Taxas de reconhecimento para as bases de teste de **caracteres** dos escritores 1, 2, 3 e 4, utilizando o vetor de observações CCPGD e o classificador HMM T16, com a estratégia 2. 123
5.21 Taxas de reconhecimento para as bases de teste de **palavras** dos escritores 1, 2, 3 e 4, utilizando o vetor de observações CCPGD e o classificador HMM T16 com a estratégia 2. 123

Lista de Símbolos e Abreviaturas

AG	Algoritmos Genéticos
AMS	Algoritmo Morfológico de Segmentação
AMSAG	Método Morfológico de Segmentação utilizando Algoritmo Genético
ASHP	Método de Segmentação por Histogramas de Projeções
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CCGD	Concatenação dos vetores de observações referentes às características globais e direcionais
CCPD	Concatenação dos vetores de observações referentes às características perceptivas e direcionais
CCPG	Concatenação dos vetores de observações referentes às características perceptivas e globais
CCPGD	Concatenação dos vetores de observações referentes às características perceptivas, globais e direcionais
CD	Características Direcionais
CDVDHMM	HMM de duração variável com densidade contínua (*Continuous Density Variable Duration Hidden Markov Model*)
CEDAR	*Center of Excellence for Document Analysis and Recognition*
CENPARMI	*Center for Pattern Recognition and Machine Intelligence*
COPELE	Coordenação de Pós-Graduação em Engenharia Elétrica
$CPGD_z$	Concatenação dos vetores de observações referentes às características perceptivas, globais e direcionais com zoneamento nas características direcionais

CRF	Campos randômicos condicionais (*Conditional Random Fields*)
DCT	*Discrete Cosine Transform*
DEE	Departamento de Engenharia Elétrica
DicICCT	Dicionário Inicial Construído a partir do Conjunto de Treino
DPI	Pontos por polegada (*dots per inch*)
FI	*Integral fuzzy*
HMM	Modelos Escondidos de Markov (*Hidden Markov Models*)
HMM ED	HMM esquerda-direita
HMM ERG	HMM ergódigo
HMM T11	Modelo HMM apresentado na Figura 4.34
HMM T16	Modelo HMM apresentado na Figura 4.35
HMM T6	Modelo HMM apresentado na Figura 4.33
K	Dimensão do quantizador vetorial
KNN	K Vizinhos mais Próximos (*K Nearest Neighbors*)
LAPS	Laboratório de Automação e Processamento de Sinais
LBG	Algoritmo para projeto de dicionários conhecido como *Linde-Buzo-Gray*
LI	Linha inferior da palavra
LM	Linha média da palavra
LS	Linha superior da palavra
M	Número de níveis do dicionário
MD-HMM	*Model discriminant* HMM
MLP	*Multi-Layer Perception*
MV	Algoritmo Modificado de Viterbi
N	Número de estados do modelo
NEHMM	HMM não ergódigo
NN	Redes neurais (*Neural Network*)
NV	Número de palavras no Vocabulário
OCR	Reconhecimento ótico de caracteres (*Optical Character Recognition*)

PCX	*Zsoft IBM PC Paintbrush*
PDAs	*Personal Digital Assistants*
PD-HMM	*Path Discriminant* HMM
PPHI	Perfil de Projeção Horizontal Inclinado
PPI	Perfil de projeção inclinado
PNSs	Pontos naturais de segmentação
PROCAD	Programa Nacional de Cooperação Acadêmica
PUCPR	Pontíficia Universidade Católica do Paraná
QV	Quantização vetorial (VQ - *Vector Quantization*)
SOFM	Mapas auto-organizáveis
SPs	Pontos de Segmentação
STD	Desvio padrão
T	Comprimento da seqüência de observação
TR	Taxa de Reconhecimento
Trn/Vld	Conjunto treinamento/validação
Tst	Conjunto de teste
UFCG	Universidade Federal de Campina Grande
VDHMM	*Variable Duration Hidden Markov Models*
$\alpha_t(i)$	Probabilidade de *forward*
$\beta_t(i)$	Probabilidade de *backward*
η_{02} e η_{20}	Momentos centrais normalizados
$\pi = \pi_i$	Vetor de probabilidade do estado inicial
a	Área do traçado da letra
$A = [a_{ij}]$	Matriz transição de estados
$B = [b_{ij}(k)]$	Matriz de função de probabilidade das observações
C	Parâmetro de Circularidade
$HistogramaK(j)$	Quantidade de pixels pretos existentes na linha j inclinada com ângulo k
P	Perímetro do traçado da letra

P_v	Probabilidade de um pixel preto ser encontrado na coluna inclinada v
R	Parâmetro de Retangularidade
X_{cm} e Y_{cm}	Centro de massa

Capítulo 1

Introdução

1.1 Comunicação homem-máquina

O ser humano moderno sempre buscou meios de comunicação que facilitassem a sua interação com as máquinas, em especial com os computadores. Em função disto e do crescente avanço tecnológico de *hardware* e *software*, foram desenvolvidos, por exemplo, o *mouse* e as telas sensíveis ao toque, e muitos trabalhos estão sendo realizados no campo do reconhecimento de voz [1]. Entretanto, ainda há muito que se pesquisar para tornar a comunicação entre os usuários e os computadores a mais natural possível.

O reconhecimento de fala é uma forma muito conveniente de comunicação com a máquina, embora nem sempre prática. Em ambientes ruidosos, ou onde o silêncio é importante, ou até mesmo onde um grande número de pessoas necessita trabalhar com os computadores, fica claro que a comunicação por voz não é a melhor solução. Neste contexto, o reconhecimento de manuscritos surge como uma opção adequada para facilitar a interação homem-máquina, visto que a escrita manual também é uma forma de comunicação entre seres humanos. Partindo desta constatação, muitos pesquisadores e cientistas vêm investigando a utilização dos sistemas de reconhecimento de manuscritos para tornar a interface homem-máquina fácil e segura [2–5] .

Os sistemas de reconhecimento ótico de caracteres (OCR - *Optical Character Recognition*) apareceram em meados de 1940 com o desenvolvimento dos computadores

digitais [6]. Durante os últimos anos, o reconhecimento de manuscritos tem sido um tema de grande interesse na comunidade científica, devido à enorme quantidade de aplicações encontradas, dentre as quais podem ser citadas: os sistemas de leitura para deficientes visuais [7–9] e os sistemas de automatização de processos de leitura de códigos postais, cheques bancários e formulários.

A análise de imagens de documentos é um rico campo de pesquisa, no qual está inserido o problema de reconhecimento de manuscritos. A finalidade da análise de imagens de documentos não é apenas o problema de reconhecimento ótico de caracteres (OCR), mas também a análise das estruturas lógicas e físicas de documentos digitalizados e, mais genericamente, o reconhecimento, a identificação e a interpretação de textos manuscritos e/ou impressos e de assinaturas [10–14].

Através de manuscritos pode-se determinar a informação presente no texto, bem como identificar ou verificar o seu autor. Portanto, o campo de processamento automático de manuscritos se divide nas seguintes sub-áreas principais:

- Verificação;
- Identificação;
- Reconhecimento.

Os sistemas de verificação e identificação realizam a extração de parâmetros que correspondam às variações de escrita peculiares ao escritor, com a finalidade de estabelecer a autoria do mesmo independentemente do conteúdo manuscrito [10, 15]. Um sistema de verificação decide a autoria através de comparação de um-para-um, uma vez que se sabe de antemão a qual classe deverá ou não pertencer o manuscrito [10], ou seja, uma autoria é alegada ao manuscrito e a decisão do sistema é estritamente binária (verdadeira ou falsa), isto é, consiste em aceitar ou rejeitar a autoria alegada. A tarefa de identificação difere da verificação, pois neste caso, o sistema é requisitado a identificar um entre todos os escritores possíveis. Para tanto, é feita a comparação entre atributos específicos do escritor a ser identificado com aqueles de todos os outros

escritores, armazenados em uma base de dados de referência. Ou seja, é realizada uma comparação de um para muitos.

Os sistemas de reconhecimento de manuscritos interpretam o conteúdo da escrita manual, extraindo características que identificam as palavras ou os caracteres que compõem as palavras. Para isto, o texto manuscrito precisa ser localizado, extraído e separado em palavras isoladas. Uma vez segmentado o texto em palavras, se estabelece o problema de determinar a melhor forma de representar essas palavras considerando a variação entre elas [16]. Diferentemente dos sistemas de verificação e identificação, esses sistemas de reconhecimento reduzem ou eliminam as variações de escrita peculiares ao escritor para reduzir a variabilidade da escrita manual.

Tradicionalmente duas abordagens têm sido empregadas no processamento automático de manuscritos: reconhecimento *on-line* e reconhecimento *off-line* [17,18]. Esta classificação está associada à forma com que os dados são adquiridos pelo sistema. No reconhecimento *on-line*, os dados são obtidos dinamicamente à medida que são produzidos através de dispositivos acoplados ao computador, como por exemplo, mesa digitalizadora ou caneta eletrônica. Nos sistemas *off-line*, os manuscritos são escritos em um papel com um instrumento escrevente convencional e depois digitalizados através de dispositivos como *scanner* ótico ou câmera digital.

Em geral, as técnicas de reconhecimento *on-line* apresentam taxas de reconhecimento superiores aos sistemas de reconhecimento *off-line* [2, 19, 20]. Este fato deve-se à natureza dinâmica da aquisição. Informações sobre a velocidade, pressão do instrumento escrevente sobre o dispositivo de aquisição, pontos inicias e finais do traçado, dentre outros, são exemplos de informações coletadas durante o processo de aquisição. Estas informações, quando usadas, facilitam a tarefa de reconhecimento dos caracteres. A localização dos pontos inicial e final do traçado, por exemplo, são importantes para os algoritmos de segmentação dos caracteres, que visam separar as palavras ou conjunto de dígitos em caracteres isolados.

Os sistemas de reconhecimento *off-line* são mais limitados, não apenas por apresentarem perda das informações dinâmicas, mais também por serem susceptíveis ao

ruído introduzido pelos dispositivos de aquisição dos dados. No entanto, o interesse nos sistemas de reconhecimento *off-line* deve-se ao fato desta abordagem preservar ao máximo a naturalidade da escrita e não requerer nenhum tipo de dispositivo especial de aquisição, nem tampouco a presença do autor durante a fase de aquisição, pressupostos estes necessários para fazer o reconhecimento *on-line*.

Apesar de diversos métodos já terem sido propostos, o reconhecimento automático de palavras manuscritas continua sendo um problema difícil de ser resolvido com robustez e precisão, devido às inúmeras possibilidades de representação de um mesmo caractere manuscrito. Normalmente, os caracteres manuscritos são não-uniformes, apresentando diversas formas e tamanhos quando escritos por pessoas diferentes ou até pela mesma pessoa, podendo assumir aparência ambígua ou ilegível. Muitos são os fatores que contribuem para esta diversidade, como o tipo de papel e o instrumento de escrita utilizados, a habilidade, o estilo, o nível de educação, a profissão, a idade e o estado emocional do escritor [12, 21].

Sistemas de reconhecimento de manuscritos são capazes de atingir taxas médias de reconhecimento que variam entre 50 à 99% dependendo das restrições impostas (tamanho do vocabulário, dependência do escritor, estilo de escrita, dentre outras) e também das condições experimentais [22, 23]. Uma das limitações comuns dos sistemas de reconhecimento de manuscritos é o fato de serem capazes de reconhecer palavras que estejam presentes em um vocabulário restrito. O vocabulário restrito, geralmente denominado de léxico, é uma lista de todas as palavras válidas que, espera-se, sejam reconhecidas pelo sistema.

As pesquisas realizadas na última década no campo de reconhecimento de dígitos manuscritos isolados contribuíram para a obtenção de altas taxas de reconhecimento, na faixa de 85-99% [24–30]. Esforços estão sendo dirigidos ao aumento da confiabilidade dos sistemas de reconhecimento de dígitos manuscritos isolados, dígitos conectados e de seqüências de dígitos, visando aplicações comerciais [31–34].

1.2 Motivação

Como já mencionado, a escrita manual ou manuscrita possui como principais características a grande diversidade de formas e tamanhos com que são representados os caracteres, bem como a existência de palavras e letras ambíguas. Por conseguinte, o reconhecimento de manuscritos sem que haja restrições é ainda um problema em aberto, tanto quanto seja do nosso conhecimento. A alternativa adotada pelos pesquisadores tem sido identificar e resolver subclasses de problemas (subproblemas), com a expectativa de que eventualmente a união das soluções destes subproblemas leve à solução do problema maior.

Neste trabalho foi desenvolvido um sistema de reconhecimento de manuscritos que é capaz de se adaptar ao escritor, aprendendo sobre a caligrafia do usuário, de forma a permitir o reconhecimento dos caracteres que compõem as palavras. Este sistema de reconhecimento de manuscritos dependente do escritor foi utilizado no reconhecimento de um vocabulário irrestrito (desde que conhecido) escrito por uma pessoa específica, podendo ser estendido a um grupo de pessoas. Deste modo, ao invés de se restringir o vocabulário a ser reconhecido pelo sistema, alternativa esta já explorada em outros trabalhos com resultados limitados, foi restrito o universo de possíveis escritores.

Para o correto funcionamento desse sistema é necessário realizar um treinamento do mesmo com o objetivo de ensinar a caligrafia do usuário.

Logicamente, esse tipo de sistema não poderá ser utilizado em aplicações de reconhecimento de cartas postais e nem em cheques bancários, por exemplo. Entretanto, outros nichos de aplicações são vislumbrados, tais como o reconhecimento de documentos históricos e a sua utilização em computadores pessoais, para reconhecer os textos escritos pelos usuários.

Diversas técnicas têm sido utilizadas para o reconhecimento de palavras manuscritas, dentre as quais se destacam: Modelos Escondidos de Markov (HMMs - *Hidden Markov Models*) [35–47], redes neurais (NN - *Neural Network*) [15, 48–51], lógica Fuzzy [51], programação dinâmica [49] e quantização vetorial (VQ - *Vector Quantization*) [52].

Dentre todas estas técnicas, os Modelos Escondidos de Markov [53, 54] tem sido bastante utilizados em sistemas de reconhecimento de manuscritos, devido a algumas características vantajosas: (i) os HMMs são ricos em fundamentação matemática e, conseqüentemente, podem formar uma base teórica forte para uso em um conjunto amplo de situações; (ii) quando aplicados apropriadamente trabalham para várias aplicações práticas; (iii) os HMMs são modelos estocásticos que podem acomodar ruído e variações que ocorrem na escrita manual; (iv) os HMMs podem tratar com palavras de entrada que possuem um número variável de parâmetros representativos; (v) adequação para modelos de características que ocorrem em seqüência [1, 40, 54].

Diante do limitado sucesso alcançado pelos sistemas de reconhecimento de palavras manuscritas independente do escritor, torna-se interessante investigar a utilização de HMMs de forma a projetar um sistema de reconhecimento de palavras manuscritas dependente do escritor.

1.3 Objetivos do trabalho

Este trabalho teve como objetivo principal o desenvolvimento de um sistema de reconhecimento de palavras manuscritas dependente do escritor para a língua portuguesa. O sistema desenvolvido caracteriza-se por adota método estatístico para o reconhecimento automático de palavras manuscritas, utilizando a modelagem por Modelos Escondidos de Markov. O reconhecimento das palavras foi realizado baseado na segmentação-reconhecimento, através da concatenação das probabilidades dos caracteres utilizados na ortografia da língua portuguesa.

Os trabalhos realizados resultaram na concepção de um algoritmo de pré-processamento para normalização da inclinação vertical dos caracteres e de uma modificação no algoritmo de normalização da inclinação da linha de base das palavras proposto origalmente por Côté [55]. Estes algoritmos são importantes, pois diminuem a variabilidade intra-classe.

Foi implementado e avaliado o algoritmo de segmentação proposto por Chen [36]. Esta avaliação resultou na concepção de três novas técnicas de segmentação de caracte-

res. A primeira utiliza morfologia matemática e um algoritmo genético na determinação dos traços de ligação entre caracteres. A segunda e terceira técnicas utilizam a análise de perfis de projeção vertical em diversos ângulos para a determinação dos pontos de segmentação natural e dos pontos de segmentação localizados nos traços de ligação. Uma quarta técnica de segmentação foi desenvolvida, utilizando informações sobre os pontos de curvatura, para determinar os possíveis pontos de segmentação.

Foram implementados algoritmos para extração de três tipos de características (perceptivas, globais e direcionais). Esses vetores de características foram desenvolvidos conjuntamente pelos pesquisadores da UFCG (Luciana Veloso, José Josemar de Oliveira Júnior e João Marques de Carvalho) e da PUCPR (Cinthia Obladen de Almeida Freitas). Este trabalho em equipe foi realizado através de um projeto de cooperação entre as duas instituições no âmbito do programa PROCAD/CAPES.

A quantização vetorial foi utilizada para a obtenção dos dicionários referentes aos caracteres, permitindo realizar o mapeamento do vetor de características no vetor de observação. Os caracteres foram modelados por Modelos Escondidos de Markov.

Foram realizados diversos experimentos utilizando diferentes bases de dados (meses do ano, IRONOFF e a base de palavras dependente do escritor) para avaliar o desempenho do sistema para diversas aplicações práticas, bem como avaliar o potencial dos algoritmos implementados.

Experimentos adicionais foram realizados substituindo o classificador HMM por redes neurais. A finalidade desses testes foi avaliar o desempenho dos classificadores HMM e NN no sistema desenvolvido.

Desta forma, os objetivos específicos deste trabalho foram:

- Construção de uma base de palavras dependente do usuário para servir como suporte ao desenvolvimento do sistema desenvolvido neste trabalho.

- Desenvolvimento de algoritmos de pré-processamento para a normalização da inclinação vertical das letras que compõem as palavras. Desenvolvimento de algoritmo de suavização para reduzir os picos e buracos produzidos pelos algoritmos

de normalização. Melhoramento no algoritmo de normalização da linha de base proposto originalmente por Côté [55];

- Desenvolvimento de três algoritmos de segmentação;
- Desenvolvimento de algoritmos de extração de características perceptivas, globais e direcionais;
- Arquitetura dos Modelos Escondidos de Markov para modelagem dos caracteres utilizando os espaços de características perceptivas, globais e direcionais;
- Comparação dos desempenhos dos classificadores de caracteres e palavra, usando HMM e NN, na tarefa de reconhecer caracteres isolados e no reconhecimento de palavras.

1.4 Organização do trabalho

O presente trabalho foi estruturado em seis capítulos. Este capítulo tem por objetivo permitir ao leitor uma visão geral da área de processamento de documentos, bem como a definição da motivação e dos objetivos deste trabalho. Além disto, esta seção contempla uma breve visualização dos demais capítulos deste documento.

No Capítulo 2 é realizada uma revisão bibliográfica dos sistemas *off-line* de reconhecimento de palavras manuscritas existentes na literatura especializada, além de uma descrição dos principais algoritmos de pré-processamento, segmentação e extração de características e reconhecimento existentes na literatura.

No Capítulo 3 é dada uma visão geral do sistema desenvolvido para o reconhecimento de palavras manuscritas dependente do usuário.

No Capítulo 4 são descritos os algoritmos desenvolvidos de normalização, suavização, segmentação, extração de características, quantização vetorial e Modelos Escondidos de Markov.

O Capítulo 5 contém a descrição das bases de dados utilizadas para avaliação dos algoritmos implementados e para avaliação do sistema, além da descrição do procedi-

mento experimental utilizado e os resultados obtidos na avaliação do desempenho do sistema.

Finalizando, tem-se o Capítulo 6, onde são apresentadas as considerações finais e trabalhos futuros são sugeridos.

Capítulo 2

Reconhecimento de palavras manuscritas - Revisão bibliográfica

Grandes avanços da civilização devem-se à escrita, que foi desenvolvida há muito tempo com o propósito de expandir a memória humana e facilitar a comunicação. A escrita tem se modificado enormemente ao longo do tempo, e muitas tecnologias contribuíram para a sua expansão. A imprensa e a máquina de datilografar abriram as portas do mundo para os documentos formatados, aumentando o número de leitores, que por sua vez, aprenderam a escrever e a comunica-se. O computador e as tecnologias da informação e da comunicação, como os processadores de texto, as máquinas de fax, os serviços de *e-mail*, também tiveram um impacto na escrita. Essas e outras tecnologias, como PDAs (*Personal Digital Assistants*) e tecnologias sem fio (*wireless*), têm levado a novas formas de interpretar o papel da escrita e das mensagens e texto manuscritos. Cada vez mais o nicho ocupado pela escrita vem se tornando mais claramente definido e popularizado. Apesar das discussões a respeito dos desafios impostos pelos computadores ao futuro da escrita, há, indubitavelmente, situações numerosas em que a caneta e o papel, ou ainda um pequeno *notepad*, são mais convenientes que um teclado [18]. Considere-se, por exemplo, o caso de estudantes em uma sala de aula, os quais continuam armazenando as informações (texto, equações e gráficos) por meio da caneta e do papel.

Os sistemas de reconhecimento de manuscritos atuam nestas situações em que o papel e o instrumento escrevente são utilizados, permitindo aos computadores transformarem uma linguagem representada em sua forma espacial ou marcas gráficas em uma representação simbólica.

O problema de reconhecimento de manuscritos possui duas abordagens: holística ou analítica. Abordagens holísticas ou globais tratam as palavras como entidades únicas, indivisíveis, e procuram reconhecê-las como um todo [23]. Uma das grandes vantagens de utilizar os modelos de palavra completa é que esses modelos são capazes de capturar os efeitos de co-articulação intra-palavra. Quando os modelos de palavra completa são adequadamente treinados, em geral levam ao melhor desempenho em termos de reconhecimento. Essa abordagem constitui a opção preferida quando a aplicação possui um vocabulário pequeno, como é o caso de aplicações de cheques bancários.

No entanto, muitas aplicações práticas requerem vocabulários maiores. Nesse caso, em vez de serem usadas palavras completas, a opção mais adequada são as abordagens analíticas, que utilizam unidades de sub-palavras, tais como caracteres ou pseudo-caracteres, como unidades básicas para o reconhecimento.

Independente da abordagem (holística ou analítica) utilizada, os sistemas de reconhecimento são complexos e necessitam de várias técnicas. Em geral, sistemas *off-line* de reconhecimento de manuscritos possuem os seguintes módulos básicos: pré-processamento, segmentação de palavras, extração de características e classificação, conforme é ilustrado na Figura 2.1.

As duas primeiras etapas (pré-processamento e segmentação de palavras) são opcionais, podendo ser ou não implementadas, dependendo da complexidade do problema a ser resolvido. Por exemplo, aplicações com vocabulário restrito permitem que cada palavra seja modelada como um todo dispensando a etapa de segmentação explícita, extraindo primitivas globais diretamente da palavra (reconhecimento global). Quando o problema a ser resolvido possui um largo vocabulário, como o reconhecimento automático de endereços postais, a segmentação da palavra em unidades menores, como letras ou segmentos de letras (pseudo-letras), é necessária, com a finalidade de reduzir

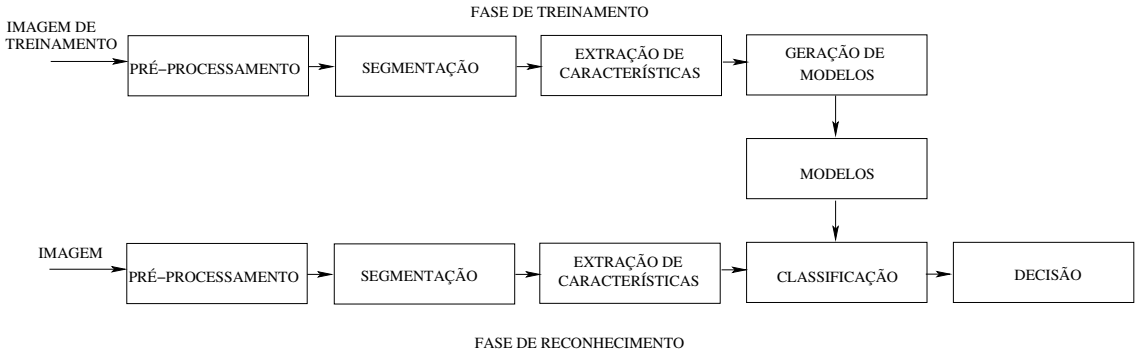


Figura 2.1: Diagrama de blocos de um sistema geral de reconhecimento de palavras manuscritas.

o conjunto de modelos para um tamanho tratável (reconhecimento local). Deste modo, procura-se modelar os caracteres que compõem as palavras.

A utilização dos algoritmos de pré-processamento é função dos dados tratados pelo sistema. Por exemplo, devido à presença de linhas de referência, em cheques bancários a correção da inclinação da linha de base da palavra não é realizada, realizando apenas a normalização da inclinação vertical dos caracteres que compõem as palavras. Entretanto, nos sistemas de reconhecimento automático de endereços postais é realizada a normalização da inclinação da linha de base das palavras, visto que as mesmas, geralmente, encontram-se inclinadas em relação a uma linha de referência horizontal imaginária [56].

A seguir, tem-se uma revisão bibliográfica dos módulos básicos de um sistemas *off-line* de reconhecimento de palavras manuscritas existentes na literatura especializada. Esta revisão concentra-se principalmente nos sistemas que utilizam a abordagem analítica.

2.1 Pré-processamento

Os algoritmos de pré-processamento diminuem a variabilidade entre as classes de objetos ou entre escritores diferentes. Nesta fase, a imagem é geralmente limpa de ruído e dos erros causados pelo processo de digitalização. Uma fase bastante importante do

pré-processamento é a normalização, que visa reduzir as variações observadas em diferentes amostras da mesma palavra, escrita pela mesma pessoa em instantes distintos ou por diferentes escritores, tais como:

- Ângulo de inclinação vertical dos caracteres (*slope* ou *slant*): Cada escritor possui uma forma peculiar de escrever e desta forma é muito natural encontrar em documentos manuscritos palavras ou numerais com diferentes ângulos de inclinação. O ângulo de inclinação vertical dos caracteres é definido como o ângulo entre o eixo correspondente à direção média dos caracteres, calculada como a média dos ângulos de inclinação de cada letra que compõe a palavra, e o eixo vertical [57]. É muito comum encontrar palavras manuscritas em que cada letra possui um ângulo de inclinação distinto. As técnicas mais simples consideram que esta inclinação é uniforme para todas as letras que compõem a palavra, realizando uma correção global. Outras técnicas trabalham isolando segmentos de palavra e efetuando a correção individual de cada segmento [46, 52, 55, 57–65].

- Inclinação da linha de base da escrita (*Skew*): O ângulo da linha de base da escrita é definido como o ângulo entre a linha de base definida pelo contorno inferior da palavra e a linha horizontal de referência. De modo geral, os métodos de correção de linha de base são globais (considerando que a inclinação é válida para a palavra como um todo) ou locais (considerando que a inclinação horizontal não é constante e igual para toda a palavra, realizando pequenas correções localizadas) [15, 48, 55].

- Largura do traçado do caractere: Este tipo de variação deve-se principalmente ao instrumento de escrita utilizado, à pressão aplicada, ao tipo de papel e à inclinação do instrumento de escrita. Pode ser eliminada através de algoritmos de esqueletização ou extração de contornos [66–71], por exemplo.

Linhas de base (*baselines*) são linhas imaginárias que dividem a palavra em três regiões: ascendente, descendente e central [15, 35, 47, 48, 55, 56, 63, 72, 73], conforme mostrado na Figura 2.2. No reconhecimento de palavras manuscritas as linhas de base

bem como as regiões ascendente, central e descendente são utilizadas, geralmente, nas etapas de segmentação e extração de características [15, 35, 47, 48, 55, 56, 63, 72]. Por exemplo, no processo de extração de características de palavras, as linhas de base são utilizadas para detectar os traços ascendentes e descendentes, bem como a região central.

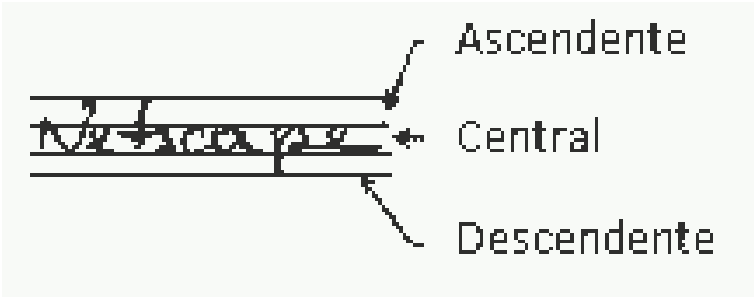


Figura 2.2: Linhas de base.

2.2 Segmentação de palavras em caracteres

Palavras são essencialmente formadas por uma seqüência de caracteres, de acordo com as restrições gramaticais impostas pela língua. Dessa forma, uma possível técnica de reconhecimento de palavras é a segmentação destas palavras em caracteres, seguida pelo reconhecimento destes caracteres como símbolos válidos do alfabeto e, finalmente, pelo casamento entre as seqüências de caracteres identificados e palavras de um dicionário (Figura 2.3). A diversidade de estilos e a grande ambigüidade encontrada na escrita cursiva, torna o processo de reconhecimento automático de manuscritos complexo, devido principalmente à dificuldade de segmentar as palavras em letras.

Um problema clássico em segmentação de palavras é apresentado na Figura 2.4: as imagens (a) e (c) possuem forma similar de representar letras distintas (m e ui). Na Figura 2.4 (a) o traçado do caractere que está circulado não necessita ser segmentado, pois pertence a único caractere. O mesmo não acontece na imagem (c) que contém dois caracteres circulados, necessitando segmentar. Um exemplo de diversidade pode

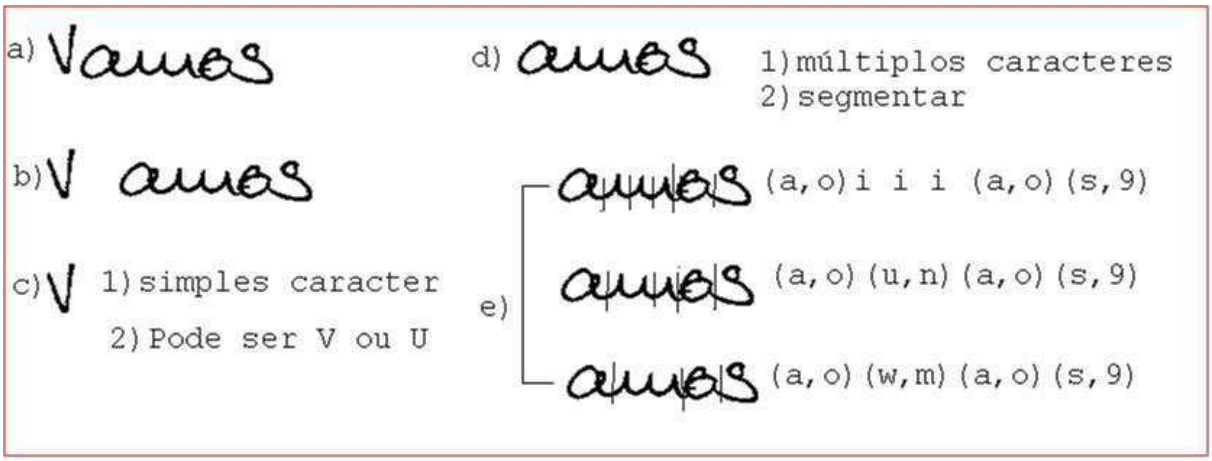


Figura 2.3: Forma intuitiva de reconhecimento de palavras manuscritas: (a) palavra manuscrita; (b) separação dos segmentos; (c) reconhecimento do primeiro segmento; (d) análise do segundo segmento; e) sobresegmentação do segundo segmento.

ser observada na Figura 2.4 imagem (b), em que o caractere "r" é escrito de formas totalmente diferentes.

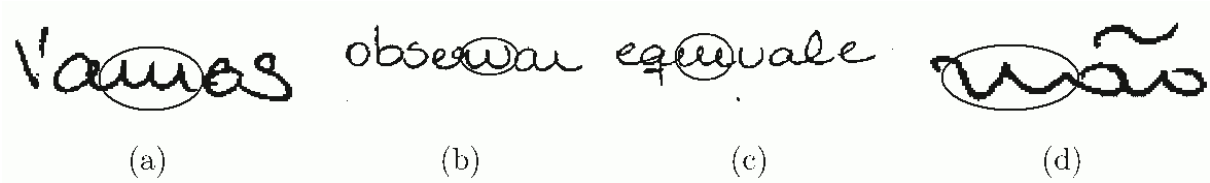


Figura 2.4: Exemplos de palavras manuscritas.

Visando resolver os problemas provocados pela ambigüidade dos caracteres na escrita cursiva, muitos sistemas dividem a imagem em segmentos, ou pedaços da palavra original, gerando várias hipóteses de pontos de segmentação (sobre-segmentação). Cada segmento idealmente consiste de um caractere ou parte de um caractere. O processo de casamento determina a melhor maneira de concatenar estes segmentos de forma a representar palavras contidas em um léxico. Entretanto, se o número de hipóteses de segmentação for excessivo a sobre-segmentação pode se tornar um gargalo. Pois, cada uma das hipótese de segmentação deve ser confirmada pela etapa de classificação,

aumentado a complexidade do sistema e possibilidade de confusão entre as letras.

Entretanto, a leitura de manuscritos pelo computador é algo mais do que simplesmente isolar e identificar os símbolos individuais que formam as palavras. A informação contextual não é apenas útil, ela é necessária [49].

Segmentação de palavras caracteres é uma operação que procura decompor a imagem de uma palavra formada por uma seqüência de caracteres em sub-imagens de caracteres ou pseudo-caracteres [74]. Segmentar corretamente uma palavra sem o conhecimento do rótulo de cada caractere que a compõe pode ser uma tarefa muito difícil. Devido a isto, em geral os processos de segmentação e reconhecimento estão fortemente relacionados [11, 35, 74].

A seguir, tem-se a descrição dos algoritmos que inspiraram o desenvolvimento do módulo de segmentação deste trabalho.

O algoritmo de segmentação utilizado por Kim [61] e Govindaraju [52] é baseado nas hipóteses de que o número de segmentos por caractere deve ser no máximo quatro e que todos os caracteres conectados devem ser separados. Traços de ligação e características de concavidade são utilizados na determinação dos pontos de segmentação. Os traços de ligação, fortes candidatos a possuírem pontos de segmentação, são obtidos através da análise da distância entre as ordenadas y dos contornos superiores e inferiores para cada ordenada x da imagem. Se a distância entre as ordenadas y for menor ou igual a largura média do traçado da palavra, a ordenada x é marcada como sendo um elemento de ligação. As características de concavidade do contorno superior e convexidade no contorno inferior são utilizadas em conjunção com os traços de ligações na determinação dos pontos de segmentação, para o caso de letras de forma conectadas. Regras heurísticas são utilizadas para reduzir o número de pontos potenciais de segmentação.

O algoritmo de segmentação desenvolvido por Favata [75] procura por traços horizontais conectando caracteres em palavras cursivas minúsculas. Conforme estas características são encontradas, pontos de segmentação são dispostos na palavra analisada. Outras características analisadas são uniões especiais de caracteres, tais como dois ca-

racteres unidos sem traço de conexão entre eles, união de cavidades (*duplo o*), união de dois "*t*" e a união de um caractere descendente com um ascendente. Todos os pontos de segmentação são analisados visando à remoção de pontos redundantes.

Bonzinovic e Srihari [11] desenvolveram um algoritmo que segmenta palavras manuscritas em caracteres, que opera em dois passos principais: pré-segmentação e compactação. A etapa de pré-segmentação opera dividindo horizontalmente a palavra em porções denominadas pré-segmentos, baseados nos mínimos locais existentes no contorno inferior da palavra. Para evitar que ao longo dos traços de conexão entre caracteres exista mais de um ponto de pré-segmentação, um procedimento de compactação é aplicado. Este procedimento procura por seqüências de pontos de pré-segmentação cuja distância é menor que um dado limiar. Se tal seqüência for encontrada, é substituída pelo ponto de pré segmentação da seqüência encontrada que possui o valor mais baixo de projeção vertical.

A técnica de segmentação proposta por Chen *et. al* [36] é baseada em características morfológicas e opera com as restrições de que cada caractere completo pode ser segmentado em no máximo três partes (segmentos) e cada segmento pode possuir no máximo dois caracteres. Este algoritmo classifica cada *pixel* ativo ("1") na imagem binária como singularidade (ilhas) ou regularidade (pontes). As ilhas e as pontes são extraídas através de operações morfológicas. A operação morfológica de abertura com um elemento estruturante de linha vertical, cujo tamanho é estimado pela largura do traçado, é utilizado para a obtenção das ilhas. As pontes são encontradas através do complemento das ilhas. Os pontos de segmentação devem existir apenas sobre as pontes. Os segmentos são achados usando o algoritmo de Viterbi [53, 54]. A matriz custo dá o caminho com a menor quantidade de pontes.

Uma segunda técnica de segmentação foi desenvolvida por Chen *et. al* [38], em que cada caractere pode ser segmentado em no máximo quatro segmentos, todos os caracteres conectados devem ser segmentados. O algoritmo, também neste caso, classifica os *pixels* de uma palavra como singularidade ou regularidade. As colunas dos corpos dos caracteres podem ser representadas por traços singulares através de uma operação de

abertura, utilizando um elemento estruturante vertical. Uma dilatação iterativa com um elemento estruturante vertical é utilizada para agrupar esses traços singulares. Os traços regulares são obtidos como resultado da subtração da imagem original da imagem contendo os traços singulares. Um procedimento de afinamento é utilizado para representar cada característica regular por um traço vertical com largura de um *pixel*, que pode ser utilizada para separar os segmentos. Em seguida, um procedimento de verificação é utilizado para encontrar os segmentos mais à direita e o mais à esquerda. O resultado final é obtido aplicando um conjunto de regras heurísticas para conferir cada segmento.

O método de segmentação de palavras manuscritas proposto por Gader *et al.* [76], realiza uma segmentação inicial detectando e agrupando os traços horizontais segundo um critério de distância. Cada segmento inicial é novamente dividido em pelo menos três seqüências de sub-imagens. Antes disso porém, cada segmento inicial é passado pelo módulo de reconhecimento de caractere. Se o valor de confiança para uma classe exceder um limiar, o segmento não é dividido, caso contrário são geradas novas primitivas e o processo é repetido.

Num segundo trabalho, Gader *et al.* [49] propõem algumas alterações no módulo de segmentação [76], como a diminuição das restrições para agrupamento e divisão da imagem da palavra. Todos os segmentos iniciais passam pela etapa de nova divisão gerando um maior número de primitivas e evitando a tentativa de casamento inicial, ou seja, o casamento só é feito após a geração das primitivas.

Kimura e Shiridhar [11] desenvolveram um sistema de segmentação e reconhecimento de palavras manuscritas. O processo de segmentação de caracteres utilizado inclui uma etapa de pré-segmentação, na qual o algoritmo detecta os pontos de mínimo existentes no contorno superior da imagem, chamados de pontos de pré-sementação. A seguir, são extraídos os componentes conectados, ou segmentos primitivos, e o seu menor retângulo envolvente.

Yanokoglu e Sandon [77] elaboraram um algoritmo de segmentação de palavras cursivas utilizando uma função custo. O custo de segmentação em um dado ponto e o

ângulo de inclinação das letras depende de características globais da escrita, como a largura média das letras e a largura média dos traços que formam as letras, bem como das características locais do ponto, como a distância ao último ponto de segmentação determinado. Os pesos para cada característica são determinados utilizando programação linear. Os pontos de segmentação são determinados pelo cálculo da função custo em cada ponto ao longo da linha de base da palavra.

Os algoritmos [11, 36, 38, 49, 61, 75–77] descritos nos parágrafos acima segmentam as palavras em sub-imagens utilizando informações provenientes dos pontos de mínimos locais do contorno da palavra, características de concavidade do traçado da palavra e localização dos traços relativamente horizontais (traços de ligação entre caracteres) que unem os caracteres na escrita manual. Cada sistema determina a quantidade máxima de segmentos em que o caractere pode ser segmentado e se permite ou não a união de caracteres em segmentos de imagens. A Tabela 2.1 sumariza as técnicas utilizadas por cada algoritmo.

Os algoritmos [36, 38, 49, 76], que operam apenas analisando os traços de ligação entre os caracteres, possuem dificuldade para segmentares caracteres fortemente conectados (caracteres conectados sem a presença de traços de ligação). Algoritmos [11] que analisam o contorno da palavra detectando pontos de mínimos e de concavidades localizam pontos de segmentação em caracteres fortemente conectados, entretanto possuem dificuldades para segmentar caracteres unidos por traços de ligação.

2.3 Extração de características e reconhecimento

O desempenho de qualquer algoritmo de classificação e/ou reconhecimento depende da representação escolhida para os padrões a serem classificados, ou seja, das características ou primitivas que são extraídas dos padrões da imagem [47,78]. Estas características devem preservar as informações que são pertinentes e úteis para a classificação e ao mesmo tempo eliminar as informações irrelevantes e desnecessárias [47, 78, 79].

Na definição do conjunto de características é importante considerar alguns critérios básicos [43]:

Tabela 2.1: Características utilizadas pelos algoritmos de segmentação

Técnica	**Característica analisada**
Kim [61] e Govindaraju [52]	Traços de ligação e características de concavidade
Favata [75]	Traços horizontais
Bonzinovic e Srihari [11]	Mínimos locais
Chen *et. al.* [36]	Traços de ligação
Chen *et. al.* [38]	Traços de ligação
Gader *et al.* [76]	Traços de ligação
Gader *et al.* [49]	Traços de ligação
Kimura e Shiridhar [11]	Pontos de mínimo
Yanokoglu e Sandon [77]	Características globais da escrita

- As características devem ser preferencialmente insensíveis à rotação, translação e variações de tamanho da escrita;
- A extração das características deve ser de baixo custo computacional;
- As características devem ser independentes umas das outras.

Os conjuntos de características utilizados nos sistemas de reconhecimento de padrões afetam diretamente vários aspectos daqueles sistemas, tais como: desempenho, tempo de aprendizagem e número mínimo de exemplos para a aprendizagem.

A tarefa do classificador é identificar a classe à qual a imagem analisada pertence, com base nas características extraídas da mesma. Desta forma, a escolha do classificador é dependente do tipo de características extraídas. Várias estratégias de classificação são utilizadas no reconhecimento *off-line* de manuscritos. Casamento de padrões, técnicas estatísticas, técnicas estruturais e redes neurais são algumas das abordagens de reconhecimento de padrões que tem sido investigadas pelos pesquisadores que atuam no reconhecimento de manuscritos [80]. Redes neurais e técnicas de reconhecimento baseadas em HMM têm sido bastante investigadas pelos pesquisadores que atuam no reconhecimento de manuscritos [22,35,81,82]. Para aumentar o desempenho dos sistemas

muitas pesquisas investigam a combinação de diferentes conjuntos de características ou a seleção de características otimizadas [31, 32, 83–86].

Para aumentar o desempenho e a confiabilidade vários esquemas híbridos de múltiplos classificadores têm sido pesquisados. As pesquisas se baseiam no pressuposto de que a decisão combinada de classificadores é melhor do que a obtida pelo melhor classificador individual. [2, 18, 79, 84, 85, 87–89, 89–91].

Interessantes resultados têm sido obtidos pelos pesquisadores utilizando conjuntos de classificadores (*ensemble classifiers*) [32, 91–94]. Conjuntos de classificadores podem ser obtidos através da variação do conjunto de treino, das características ou parâmetros e arquiteturas do classificador base [80].

As estratégias de aprendizagem podem ser classificadas em dependente ou independente do escritor. Técnicas de aprendizagem consideradas como independentes do escritor constroem seus modelos por treinamento a partir de um conjunto de amostras de palavras manuscritas de diversos escritores, enquanto as técnicas de aprendizagem dependentes do escritor utilizam apenas amostras de palavras redigidas pelo mesmo escritor para a construção de seus modelos [95].

A seguir é apresentada uma revisão bibliográfica sobre diferentes tipos de características, bem como os respectivos métodos de reconhecimento.

Gader *et al.* [76] propuseram para o reconhecimento de palavras manuscritas a utilização de dois conjuntos de características, um relacionado a cavidades e o outro, de características de valor de direção. O reconhecimento é realizado por redes neurais MLP (*Multi-Layer Perception*). Um léxico de 746 palavras foi utilizado para avaliar o desempenho do algoritmo produzindo uma taxa de 83,92%.

Em trabalho posterior, Gader *et al.* [50] utilizaram um vetor de características composto por 120 elementos, que representam características direcionais. Estratégias *fuzzy* são utilizadas para o treinamento das redes neurais. Em outra variante do mesmo sistema, Gader *et al.* [49] introduzem uma medida de compatibilidade entre pares de caracteres adjacentes formados durante a segmentação, com o objetivo de reduzir ambigüidade. Para isso, os caracteres são divididos em três categorias, ascendentes (A),

descendentes (D) e centrais (C), resultando em nove possíveis classes de pares de caracteres AA, AC, AD, CA, CC, CD, DA, DC e DD. Redes neurais são treinadas com características de transição e características de enquadramento, calculadas separadamente, uma para cada componente dos pares de caracteres.

Em outro trabalho Gader *et al.* [51] utilizaram os mesmos conjuntos de características, de barra e transição além de duas novas estratégias de reconhecimento: mapas auto-organizáveis com redes neurais (SOFM/MLP) e mapas auto-organizáveis com integral *fuzzy* (SOFM/FI).

Uma segunda geração de sistemas de reconhecimento de palavras manuscritas foi desenvolvida por Gader e Mohamed [41, 42, 78]. As principais características desses sistemas são a não utilização de algoritmos de segmentação e a incorporação dos HMM no cálculo dos escores de casamento. No primeiro trabalho [42], os autores descrevem a imagem da palavra por uma lista de vetores de características calculados para cada coluna da imagem pré-processada, usando características de transição e de gradientes. No reconhecimento da palavra é utilizado um HMM de densidade contínua do tipo esquerda-direita com doze estados (duas misturas de Gaussianas por estado) para cada classe de caracteres, tanto maiúsculos quanto minúsculos. Testes foram efetuados em uma base de dados de 317 imagens de nomes de cidades, utilizando o léxico do CEDAR, com o tamanho médio do dicionário igual a 100. Se a palavra correta não é encontrada no léxico, ela é inserida. As taxas de reconhecimento obtidas para as 10 primeiras palavras mais prováveis foram de 78,9% a 94,3%.

Em outro trabalho, Gader et. al. [78] propuseram um novo modelo de Markov escondido incorporando lógica *fuzzy*. São definidas novas formulações para as variáveis de avanço e retardo, bem como para as fórmulas de reestimação e para o algoritmo de Viterbi, modificando o HMM, e construindo o HMM *fuzzy*. Nesta nova formulação é relaxada a condição de independência estatística de modo que o HMM *fuzzy* pode ser visto como um HMM não estacionário, para o qual as probabilidades não só variam com o tempo, mas também são dependentes das seqüências de observação.

Usando o HMM *fuzzy* Gader e Mohamed [41] realizaram uma avaliação do conjunto

de características, do número de estados por caractere (N), e do número de misturas por estado (M), concluindo que o melhor desempenho é atingido para $N = 12$ e $M = 3$. Também, é sugerida a incorporação de um estágio de refinamento do dicionário, para evitar a necessidade de determinar um escore para cada palavra.

Gillies [96] foi um dos primeiros a utilizar HMM em um sistema de reconhecimento de palavras manuscritas com segmentação implícita. Operações morfológicas são utilizadas para rotular cada *pixel* de acordo com seu valor de pertinência relativo a traços, buracos e concavidades, localizados acima, abaixo e dentro da região central da imagem. A imagem rotulada de características é transformada em uma seqüência de símbolos pela utilização de quantização vetorial. Esta abordagem utiliza HMM de densidade discreta esquerda-direita, em que é construído um modelo para cada letra. Os modelos de letras individuais são combinados em modelos para cada palavra no léxico. Os conjuntos de treinamento e de teste consistem em 2.662 e 296 imagens de palavras cursivas, respectivamente, e foram extraídos de blocos de endereços manuscritos do correio americano. As taxas de reconhecimento obtidas para as 10 primeiras palavras foi de 72,6% a 90,5% para o léxico de 100 palavras e para o léxico de 1.000 palavras foi de 51,0% a 80,1%.

Yacoubi *et al.* [35] propuseram uma técnica de reconhecimento de palavras manuscritas para um vocabulário amplo, usando HMM. Neste sistema, a seqüência de segmentos obtidos por segmentação é transformada em uma seqüência de símbolos, considerando dois conjuntos de características. O primeiro conjunto é formado por características globais: *loops*, traços ascendentes e traços descendentes. O segundo conjunto de características é obtido por uma análise bidimensional do histograma de transição do contorno, para cada segmento, nas direções horizontal e vertical. Além destas características são utilizadas características de segmentação relativas à maneira como os segmentos estão conectados. Para unificar os dois conjuntos de características, dois modelos pequenos (em relação ao número de parâmetros), um para cada conjunto, são combinados por interpolação linear. No final, temos então um HMM interpolado. O modelo foi testado em uma base de imagens de nomes de cidades francesas. O

resultado encontrado para um léxico de 1.000 palavras foi de 88,9%.

Os sistemas de reconhecimento de palavras manuscritas desenvolvidos por Chen *et al.* [36, 37] realizam o reconhecimento de palavras manuscritas utilizando HMMs de densidades discretas. A imagem é primeiramente particionada em uma seqüência de segmentos (caractere completo, fração de um caractere ou união de até dois caracteres completos). Um conjunto de 35 características são usadas para identificar os segmentos: características globais extraídas dos segmentos da imagem e características locais, que representam os traços que compõem os caracteres. O outro sistema de reconhecimento de palavras manuscritas proposto por Chen *et al.* [38] utiliza HMM de duração variável com densidade contínua (*Continuous Density Variable Duration Hidden Markov Model* - CDVDHMM). Um algoritmo de segmentação separa todos os caracteres conectados e garante que cada caractere será segmentado no máximo em cinco partes. O CDVDHMM define as 26 letras do alfabeto como 26 estados diferentes. Na fase de treinamento, grupos consecutivos de segmentos são identificados como estados, e uma mistura de Gaussianas é utilizada para modelar as probabilidades de símbolos para cada estado. Espera-se que cada componente Gaussiana em cada classe represente um estilo particular de escrita.

Gilloux *et al.* [44] desenvolveram duas abordagens de aplicação de HMMs no reconhecimento de palavras manuscritas. As duas abordagens utilizam o mesmo método de extração de características, que realiza a segmentação explícita da palavra em letras ou em pedaços de letras, ambos chamados de segmentos. Em cada segmento de letra são analisados a existência de loops, traços ascendentes e traços descendentes. Símbolos diferentes são atribuídos para cada configuração possível destas características. A primeira abordagem proposta consiste de uma HMM holística para vocabulários pequenos. Cada classe de palavra é representada por um modelo de HMM, do tipo esquerda-direita, em que as observações são emitidas ao longo das transições. Na segunda abordagem, para dicionários dinâmicos, para cada letra do alfabeto é construído um HMM e os modelos para as palavras são obtidos através da concatenação dos HMM de letras.

Kundu *et al.* [43] desenvolveram um sistema de reconhecimento de palavras manuscritas utilizando HMM, o qual assume que a palavra é previamente segmentada em caracteres. Um total de quatorze características são extraídas, relativas à forma (números de junções X e T, número de loops, dentre outras) e à distribuição dos *pixels*. Os vetores de características são transformados em símbolos, através de um algoritmo de quantização vetorial e as 26 letras do alfabeto são identificadas com estados do HMM. Na etapa de reconhecimento, cada letra da palavra é classificada por comparação com os símbolos do dicionário do quantizador vetorial, de acordo com o critério da mínima distância. O reconhecimento é feito utilizando o algoritmo de Viterbi. Num outro trabalho, Kundu *et al.* [39] desenvolveram um sistema de reconhecimento de palavras manuscritas utilizando HMM de duração variável (*VDHMM - Variable Duration Hidden Markov Models*) utilizando HMM não ergódigo (NEHMM) . O sistema NEHMM foi implementado com duas abordagens MD-HMM (*Model discriminant* HMM) e PD-HMM (*Path Discriminant* HMM).

Bunke *et al.* [40] desenvolveram um sistema com HMM, capaz de se adapatar ao estilo de escrita de diferentes escritores. Na fase de treinamento e teste foram utilizadas palavras manuscritas de cinco indivíduos distintos. As características usadas pelo HMM são relativas aos nós e às bordas do esqueleto da palavra a ser reconhecida (bordas são traços formados por *pixels* com dois vizinhos e nós são *pixels* com um, três ou quatro vizinhos). Um total de dez características são utilizadas para descrever uma borda: as primeiras quatro descrevem a localização espacial da borda, a quinta característica é binária indicando se uma borda é incidente ou não a um nó de grau um. As cinco últimas características contêm informação sobre a curvatura das bordas. O vetor de características foi codificado por um quantizador vetorial, um HMM foi treinado pelo algoritmo de Baum-Welch e o reconhecimento feito utilizando o algoritmo de Viterbi.

Côté *et al.* [55] desenvolveram um sistema de reconhecimento de palavras manuscritas que possui arquitetura inspirada no sistema de leitura humano e em concepções perceptuais. Este sistema realiza um reconhecimento holístico utilizando três tipos de características: primárias, secundárias e de vales. As características primárias são utili-

zadas para detectar letras chaves (letras ou parte de letras contendo traços ascendentes, descendentes ou loops) no corpo da palavra. Características secundárias (*b-loops, d-loops*, ou as barras T) são condicionais, porque são apenas detectadas na presença de características primárias. As características de vale com cavidade para cima e/ou para baixo são extraídas do *background* da imagem. No reconhecimento, uma rede neural com três camadas identifica a imagem da palavra utilizando as características extraídas por intermédio de sucessivos ciclos perceptuais. Uma técnica de casamento fuzzy identifica a correspondência entre as zonas na imagem de entrada e letras nas palavras do léxico. A saída do módulo de reconhecimento é uma lista de palavras candidatas dispostas em ordem decrescente de ativação.

Rigoll *et al.* [97] apresentaram um sistema de reconhecimento de palavras manuscritas que utiliza uma técnica de extração de características tolerante ao erro de detecção da linha de base. Para cada letra do alfabeto foi criado um HMM. As palavras contidas no léxico foram modeladas através da concatenação de modelos de letras. Neste sistema, o método de extração de características realiza a codificação em zonas. Em outros trabalhos, Rigoll *et al.* [88, 98] realizaram comparações de sistemas de reconhecimento de palavras manuscritas baseados em HMMs, utilizando técnicas de modelamento discreto e híbrido. Nas abordagens híbridas são realizadas associações de HMMs e redes neurais. As redes neurais são utilizadas como um quantizador vetorial para um HMM discreto, ou para calcular a função de densidade de probabilidade para um HMM contínuo. Outro aspecto interessante é o uso do conhecimento contextual para modelar a linguagem, ao invés de léxicos fechados, com o objetivo de aumentar os resultados do reconhecimento de caracteres para um vocabulário ilimitado. O vetor de características é composto por coeficientes da DCT (*Discrete Cosine Transform*) e 3 características indicando altura, largura e número de transições de preto para branco. Utilizando HMMs discretos, o vetor de características é quantizado por dois dicionários distintos de tamanhos 200 e 100 (correspondendo aos dois tipos de características). Na abordagem híbrida com HMMs discretos, a rede neural *winner-takes-all* substitui o quantizador vetorial K-média. Na segunda abordagem híbrida, o HMM contínuo utiliza uma rede

neural para estimar a probabilidade *a posteriori* para substituir o dicionário. A última abordagem é um HMM discreto convencional. Estes sistemas utilizam 80 HMM diferentes, um para cada caractere (letras, números e caracteres especiais). São utilizados oito estados para letras e números e no máximo quatro estados para caracteres especiais de acordo com sua largura. O treinamento dos sistemas é realizado utilizando o algoritmo de Baum-Welch e o reconhecimento utiliza o algoritmo de Viterbi.

Guillevic e Suen [99] propuseram um método de reconhecimento de palavras manuscritas baseado em HMM a ser integrado no sistema de processamento de cheques bancários desenvolvido pelo CENPARMI. Neste método, para cada palavra do dicionário é construído um modelo. O tipo de HMM escolhido é esquerda-direita, com o número de estados sendo função do comprimento médio da seqüência de observação para cada classe. O número de símbolos de observação é definido pelo tamanho do dicionário (*codebook*). Todas as características extraídas são obtidas do contorno da palavra, por uma janela deslizante (*key-word spotting*) que percorre a imagem da esquerda para direita. O número de características extraídas para cada posição da janela é de 4 características de ângulos para cada 5 sub-janelas, totalizando 20 características. Também são extraídas características (ascendentes ou descendentes) adicionais para cada sub-janela. Estas características adicionais codificam as posições verticais médias dos *pixels* em relação à posição da linha superior (inferior).

Kim *et al.* [100] utilizaram para o reconhecimento de palavras manuscritas uma estratégia híbrida de classificadores baseada em HMM e redes neurais. O reconhecimento com segmentação explícita é realizado pelo HMM, enquanto o reconhecimento holístico é realizado por redes neurais. Esses dois classificadores são combinados utilizando uma estratégia de multiplicação.

Arica *et al.* [101] utilizaram um técnica analítico para reconhecer manuscritos cursivos. Parâmetros globais tais como o ângulo de inclinação vertical, linhas de base, largura e altura dos traços são estimados. O método de segmentação utiliza informações binárias e nível de cinza para segmentar os caracteres. HMM é utilizado para realizar o reconhecimento dos caracteres. A saída do HMM e as informações do espaço

de características são combinadas em um problema de otimização gráfica para realizar o reconhecimento das palavras. O desempenho do sistema é testado utilizando 2000 palavras da base de dados Lancester-Oslo/Bergen.

Koch *et al.* [102] combinaram informações contextuais para o reconhecimento de palavras manuscritas. O reconhecimento é realizado utilizando três diferentes fontes de informação: as saídas de um classificador de caracteres, informações contextuais extraídas da palavra e algum conhecimento prévio. Os experimentos demonstraram o benefício da utilização dessas informações adicionais.

Koerich *et al.* [103] propuseram um algoritmo de HMM em dois níveis, para reconhecer palavras manuscritas pertencentes a um grande dicionário. O algoritmo de reconhecimento separa a decodificação de palavras nos níveis de estados e de caracteres. Isto possibilita o cálculo prévio das probabilidades dos caracteres e sua utilização por um algoritmo rápido de decodificação não heurístico baseado em HMM para a representação dos caracteres. Foi utilizado um mecanismo de rejeição para aumentar a confiabilidade do sistema. Os resultados experimentais mostram que o sistema desenvolvido obtém uma taxa de 91% para um vocabulário de 1000 palavras com um tempo de reconhecimento de 0,2s por palavra.

Uma abordagem multi-vista foi desenvolvida por Oliveira Jr. [16, 83, 84, 104–106] para o reconhecimento de palavras manuscritas. O sistema em questão é formado por três módulos distintos. O primeiro módulo realiza uma pseudo-segmentação de radical que divide as palavras em duas regiões e seu vetor de características é composto por características perceptivas e geométricas. As palavras analisadas são reconhecidas por redes neurais multicamadas (MLP). O segundo módulo é composto por uma pseudo-segmentação fixa, três extratores de características (perceptivas, direcionais e topológicas) e classificadores neurais. O último módulo realiza uma pseudo-segmentação variável, para extrair as mesmas características anteriormente descritas do sistema, embora usando uma representação por alfabetos de símbolos. O classificador utilizado é um HMM com uma topologia esquerda-direita com passo dois, sendo no treinamento aplicado o algoritmo de *Baum-Welch* com validação cruzada e no reconhecimento o

algoritmo *Foward*.

Morita [107] desenvolveu um sistema híbrido baseado em HMM e rede neural MLP (*Multi Layer Propagation*) para segmentar e reconhecer datas manuscritas em cheques bancários. O sistema utiliza um classificador HMM para reconhecer a palavra manuscrita que contém a informação sobre o mês do ano e uma rede neural MLP para reconhecer o dia e o ano. Uma metodologia não supervisionada para seleção de características é proposta.

Kessentini *et al.* [82] realizou o reconhecimento de manuscritos utilizando uma abordagem *multi-stream* com HMM. Dois conjuntos de características são extraídos do contorno inferior e superior da palavra através de uma janela deslizante. Cada modelo do *2-stream* é treinado independentemente utilizando o algoritmo de *Baum Welch*. O último passo é o reconhecimento, que utiliza um algoritmo modificado de Viterbi o qual decodifica dois modelos *strem* em paralelo de acordo com o formalismo *multi-stream*. Os testes experimentais foram realizados com 500 palavras manuscritas. As taxas de reconhecimento obtidas comprovam que a utilização desta estratégia aumenta a taxa de reconhecimento, comparada às taxas obtidas considerando apenas os modelos de contorno inferior e superior isoladamente.

Shetty *et al.* [108] desenvolveram uma abordagem para reconhecer palavras manuscritas que utiliza campos randômicos condicionais (CRF - *Conditional Random Fields*) para realizar a classificação. Cada palavra é segmentada em segmentos contendo no máximo $\frac{1}{4}$ de caractere. Os segmentos são rotulados como caracteres com probabilidade calculada através do modelo CRF. A probabilidade da imagem de entrada representar uma palavra no vocabulário é calculada utilizando programação dinâmica que avalia a melhor combinação de segmentos.

O desempenho de alguns sistemas aqui descritos são reportados nas Tabelas 2.2 e 2.3, em que TR é a , NV é o Número de palavras no Vocabulário. Os sistemas foram avaliados usando diferentes bases de dados e conseqüentemente muitas vezes possuem vocabulários distintos. Comparar o desempenho dos sistemas através de suas taxas de reconhecimento é inviável e imprudente. A dificuldade do sistema não reside apenas no

tamanho do vocabulário mais também na similaridade entre as palavras do vocabulário, a aplicação ao qual se destina o sistema e se o sistema é dependente ou não do escritor. De uma maneira geral, quanto maior o vocabulário em questão, menores são as taxas de reconhecimento.

2.4 Comentários finais

O reconhecimento de palavras manuscritas é um dos problemas mais importantes da área de processamento de documentos e, como conseqüência, tem recebido bastante atenção da comunidade cientifica da área.

Como pôde ser observado nos trabalhos analisados, o reconhecimento de palavras manuscritas é realizado segundo duas abordagens: holística ou analítica. Geralmente, quando a aplicação requer um pequeno léxico utiliza-se a abordagem holística, caso contrário é preferível utilizar a abordagem analítica.

Um sistema genérico de reconhecimento de palavras manuscritas é composto por cinco etapas: pré-processamento, segmentação extração de características, classificação e decisão. As etapas de pré-processamento e segmentação podem ou não ser utilizadas, dependendo da complexidade do problema a ser tratado.

Devido à existência de letras ambíguas e ilegíveis é muito difícil segmentar a palavra em caracteres completos. A solução é realizar o reconhecimento utilizando a abordagem holística ou utilizar a etapa de segmentação para quebrar a palavra em segmentos, que podem ser caracteres completos ou pedaços de caracteres.

Várias técnicas de extração de características e de classificação foram propostas e investigadas, no reconhecimento de palavras manuscritas, dentre as quais HMMs tem ocupado um lugar de destaque. Entretanto, não existe ainda um consenso sobre qual a melhor arquitetura para modelar a escrita cursiva.

Tabela 2.2: Desempenho de alguns sistemas de reconhecimento de manuscritos

Sistema	**Classificador**	**NV**	**TR** (%)	**Base de dados**
Gader *et al.* [76]	NN	746	83,92	nomes de cidades (CEDAR) (Inglês)
Gader e Mohamed [42]	HMM	100	89,3	nomes de cidades (CEDAR) (Inglês)
Gillies [96]	HMM	1000	80,1	nomes de cidades americanas (Inglês)
Yacoubi *et al.* [35]	HMM	1000	88,9	nomes de cidades francesas (Frances)
Chen *et al.* [38]	CDVDHMM	1000	59,6	nomes de cidades, nomes de pessoas, nomes de estados americanos (Inglês)
Gilloux *et al.* [44]	HMM	27	79,0	valores por extenso em cheques bancário (Frances)
Kundu *et al.* [39]	VDHMM	1000	67,0	nomes de cidades, nomes de pessoas, nomes de estados americanos (Inglês)
Bunke *et al.* [40]	HMM	150	98,4	palavras manuscritas (Inglês)
Côté *et al.* [55]	HMM e Fuzzy	32	73,6	valores por extenso em cheques bancário (inglês)

Tabela 2.3: Desempenho de alguns sistemas de reconhecimento de manuscritos (continuação)

Sistema	**Classificador**	**NV**	**TR** (%)	**Base de dados**
Rigoll *et al.* [97]	HMM	1000	98,30	palavras manuscritas (Inglês)
Guillevic e Suen [99]	HMM-KNN	30	86,7	valores por extenso em cheques bancário (Inglês)
Kim *et al.* [100]	HMM-MLP	32	92,2	valores por extenso em cheques bancário (Inglês)
Arica *et al.* [101]	HMM	1000	90,8	palavras manuscritas (Inglês)
Koch *et al.* [102]	MLP	1000	67,8	palavras manuscritas (Frances)
Koerich *et al.* [103]	HMM	1000	91,0	nomes de cidades (Frances)
Oliveira Jr. [16]	NN e HMM	12	97,9	palavras do meses do ano (Português)
Morita [107]	HMM	12	91,5	palavras do meses do ano (Português)
Kessentini *et al.* [82]	HMM	1000	42,6	palavras manuscritas (Inglês)
Shetty *et al.* [108]	campos randômicos condicionais	300	80,0	palavras manuscritas (Inglês)

Capítulo 3

Descrição geral do sistema proposto

O sistema de reconhecimento automático de palavras manuscritas dependente do usuário desenvolvido foi projetado para reconhecer palavras manuscritas isoladas escritas por usuários cadastrados. Além disso, ele pode ser utilizado, com pequenas modificações, em diversas aplicações práticas. Este sistema se caracteriza por ser um sistema que opera *off-line*, utilizando a modelagem por HMMs para a obtenção dos padrões representativos dos caracteres.

O diagrama de blocos do sistema desenvolvido é ilustrado na Figura 3.1. O sistema possui duas fases distintas de operação: treinamento e reconhecimento. As etapas de pré-processamento, segmentação e extração de características são as mesmas em ambas as fases de operação. A seguir são descritos os objetivos gerais de cada uma das etapas do sistema.

3.1 Pré-Processamento

No módulo de pré-processamento são realizadas operações de suavização, normalização da linha de base e normalização da inclinação vertical dos caracteres que compõem a palavra em análise, com a finalidade de diminuir a variabilidade intra classe. Além de adequar as palavras a etapa seguinte de segmentação.

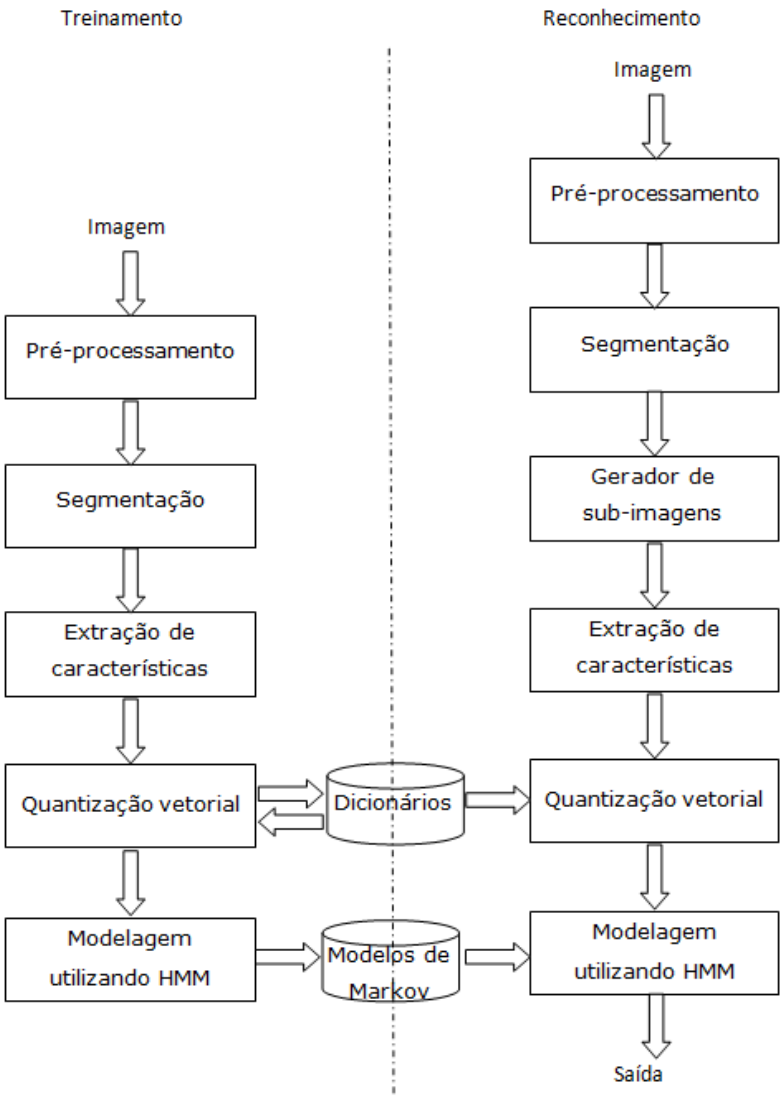


Figura 3.1: Diagrama de blocos do sistema de reconhecimento de palavras manuscritas dependente do usuário.

3.2 Segmentação

A etapa de segmentação é realizada nas imagens pré-processadas e consiste em dividir a palavra em segmentos. O algoritmo proposto baseia-se na hipótese de que cada segmento obtido corresponde a um caractere completo, ou pelo menos 1/5 de um caractere. Os pontos de segmentação são determinados utilizando a análise do perfil de projeção inclinado, análise dos mínimos locais do contorno das palavras e regras heurísticas baseadas nas características de concavidade e convexidade [109, 110].

3.3 Gerador de sub-imagens

Os segmentos determinados na etapa de segmentação são selecionados e combinados (unidos) formando sub-imagens compostas por um, dois, três, quatro ou cinco segmentos consecutivos. Esse processo é ilustrado na Figura 3.2.

3.4 Extração de características

Vetores de características são extraídos de cada sub-imagem. Estes vetores são utilizados, na fase de treinamento, para obtenção do dicionário do quantizador vetorial e da seqüência de observações que geraram os HMMs de referência associados a cada caractere. Na fase de reconhecimento, os vetores de características são usados para determinação das probabilidades associada aos HMMs.

3.5 Quantização vetorial

A quantização vetorial foi utilizada na fase de treinamento para a construção dos dicionários associados a cada vetor de característica para cada usuário. Na fase de reconhecimento a quantização vetorial é utilizada para a obtenção dos vetores de observação. Estes vetores de observação são utilizados na classificação estatística, pelos Modelos Escondidos de Markov (HMMs), sendo obtidos após o processo de representação dos

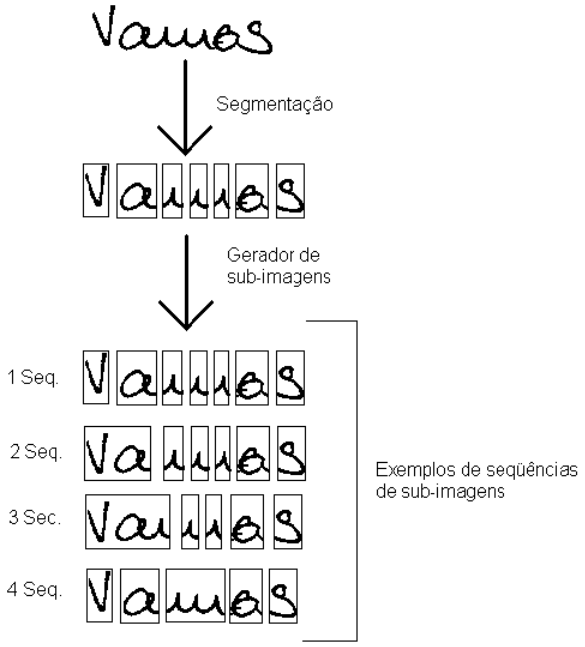


Figura 3.2: Exemplos de seqüências de sub-imagens (hipóteses de segmentação), partindo de pontos fixos determinados pelo algoritmo de segmentação.

padrões de teste pelo dicionário, ou seja, correspondem aos índices dos vetores códigos.

O algoritmo utilizado na construção de dicionários desempenha um papel importante, pois cada dicionário de padrões deve representar, de forma eficiente, as características dos caracteres por ele representados. Desta forma, pode-se afirmar que o desempenho do quantizador vetorial depende da qualidade dos dicionários projetados. O alvo a ser perseguido é a obtenção de dicionários, que minimizem a distorção média introduzida pela aproximação de cada vetor de entrada por um dos vetores-código. O quantizador ótimo deve satisfazer duas condições: a condição de vizinho mais próximo para partição e a condição de centróide para vetores-códigos [111].

Existem diversas técnicas para o projeto de dicionários para a quantização vetorial, dentre as quais o algoritmo LBG (*Linde-Buzo-Gray*) destaca-se pela sua ampla utilização [111–113].

A construção de dicionários necessita da análise de um conjunto de treinamento representativo para os padrões codificados. Especificamente, nos sistemas de reconhecimento de palavras manuscritas independentes do escritor, a grande variedade de estilos presente nas palavras manuscritas dificulta o levantamento de um conjunto de treinamento representativo dos vários estilos de escrita encontrados na prática. O sistema desenvolvido é dependente do escritor, o que diminui a quantidade de estilos de escrita tratados pelo sistema. Por outro lado, isto provocou a necessidade de obter um conjunto representativo de amostras de palavras manuscritas para cada usuário do sistema.

O sistema de reconhecimento de palavras manuscritas foi elaborado para reconhecer palavras escritas por usuários cadastrados do sistema. Para cada usuário foram construídos dicionários particulares (um para cada espaço de característica).

3.6 Reconhecimento estatístico

O reconhecimento de palavras manuscritas dependente do usuário, utilizando HMMs de densidade discreta pode ser assim resumido: classificar as palavras escritas por usuários cadastrados do sistema, utilizando as seqüências de observações extraídas

das sub-imagens das palavras, de acordo com modelos de caracteres para cada usuário. Essa classificação é realizada por meio da determinação da medida de similaridade entre as seqüências de observações de uma dada palavra e todos os modelos de caracteres. Esses modelos foram obtidos durante a fase de treinamento, por meio do cálculo da probabilidade de que cada modelo represente as ocorrências associadas às seqüências de observações da palavra. O resultado final do processo de reconhecimento é a palavra do dicionário que possui a maior probabilidade de gerar a seqüência analisada.

No treinamento, após a determinação das arquiteturas, foi realizada a modelagem dos caracteres. Esta estratégia possui a vantagem de necessitar de um número pequeno de modelos para representar um grande vocabulário. Entretanto, apresenta a desvantagem de perder os efeitos da co-articulação entre caracteres adjacentes [99].

A estimação do modelo foi realizada utilizando o algoritmo de Baum-Welch [53], que iterativamente e automaticamente ajusta os parâmetros segundo o conjunto de seqüências de observações de treinamento.

3.7 Comentários finais

O sistema de reconhecimento automático de palavras manuscritas desenvolvido neste trabalho é constituído das etapas de pré-processamento, segmentação, geração de sub-imagens, extração de características, quantização vetorial e modelagem utilizando HMM. No próximo capítulo tem-se uma descrição mais detalhada de cada algoritmo utilizado neste sistema.

Capítulo 4

Descrição dos módulos implementados

O sistema proposto neste trabalho está ilustrado na Figura 3.1. A seguir serão descritos, em detalhes, os algoritmos que compõem os módulos de pré-processamento, segmentação, extração de características e quantização vetorial e HMMs.

4.1 Pré-processamento

O pré-processamento é projetado para reduzir ruído e variações observadas em diferentes amostras da mesma palavra, escrita pela mesma pessoa em instantes distintos, além de adequar as palavras a etapa de segmentação, conforme anteriormente descrito no Capítulo 3. O resultado deste processamento é um conjunto de dados mais consistente, aumentando assim a precisão do sistema.

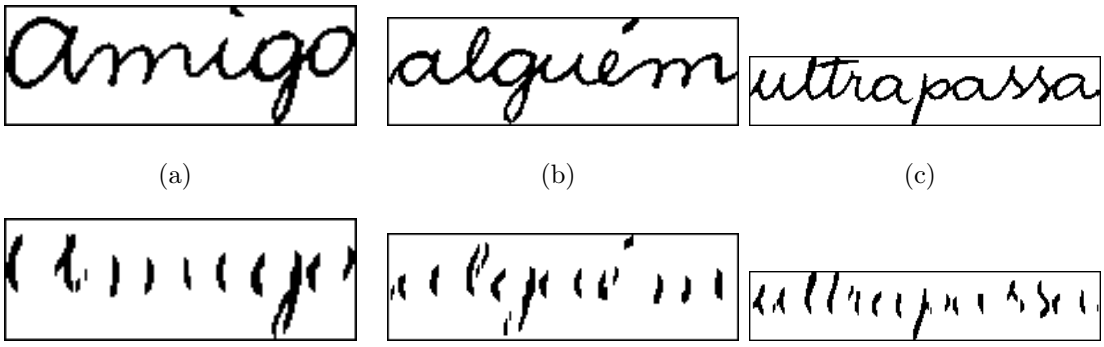
Nesta etapa, operam os algoritmos de normalização da inclinação vertical dos caracteres das palavras, normalização da inclinação da linha de base das palavras e suavização. A seguir serão descritos os algoritmos.

4.1.1 Normalização da inclinação média dos caracteres da palavra

O método desenvolvido (contribuição original deste trabalho) para determinar e corrigir a inclinação média dos caracteres que compõem as palavras foi inspirado no algoritmo

de determinação da linha de base proposto por Côté [55].

Inicialmente é realizada uma operação morfológica de abertura nas imagens [67], com objetivo de prevenir que traços aproximadamente horizontais interfiram na determinação da inclinação média das letras. O elemento estruturante utilizado consiste de uma linha vertical cujo comprimento é determinado pela largura média do traçado da palavra [1]. A imagem resultante desta operação é formada pelos traços das letras que possuem pequena inclinação em relação à vertical. Na Figura 4.1 são ilustrados os resultados da operação de abertura em algumas amostras de palavras. A imagem resultante da operação morfológica de abertura é utilizada na determinação da inclinação média da palavra.



(a) (b) (c)

(d) (e) (f)

Figura 4.1: Resultados da operação de abertura em amostras de palavras. (a), (b) e (c):imagens originais; (d), (e) e (f): imagens resultantes da operação de abertura.

O primeiro passo na determinação da inclinação vertical dos caracteres da palavra é calcular o perfil de projeção inclinado (PPI) da imagem, em ângulos de inclinação que variam de -60 a 60 graus em relação à vertical, com intervalos de 1 grau. O perfil de projeção inclinado indica a quantidade de *pixels* pretos existentes em colunas inclinadas. O algoritmo utilizado para o cálculo do perfil de projeção inclinado, para uma imagem $M \times N$, é descrito a seguir:

[1] A largura média dos traços que compõem o caractere é obtida calculando-se a média da razão entre o número de *pixels* pretos e o número de transições entre *pixels* pretos e brancos em cada coluna da imagem [36].

Para cada pixel (i, j) *da imagem;* $i = 1, 2, ..., M$ *e* $j = 1, 2, ..., N$.

Para cada ângulo de inclinação (θ)

1. *Determine o novo valor* v *da coordenada* j *na imagem, como sendo:*

$$v = \lfloor j - (M - i).tan(\theta) \rfloor, \tag{4.1}$$

   em que o operador $\lfloor . \rfloor$ *indica o maior inteiro mais próximo.*

2. *Se o valor do pixel em* (i, v) *for igual a* "1"*, incremente a* v*-ésima coluna do* θ*-ésimo perfil de projeção inclinado (PPI).*

$$PPI_\theta(v) = PPI_\theta(v) + 1; \tag{4.2}$$

A segunda etapa, no cálculo da inclinação média, é calcular as entropias associadas a cada perfil de projeção inclinado, utilizando a Equação 4.3.

$$E_\theta = -\sum_{v=1}^{L} P_v \log P_v; \tag{4.3}$$

sendo L o número de colunas do perfil de projeção inclinado e P_v a probabilidade de um *pixel* preto ser encontrado na coluna inclinada v.

Este algoritmo assume que a origem da imagem encontra-se no canto superior esquerdo da matriz de *pixels*. Esta fórmula é válida para ângulos positivos e negativos.

O ângulo que proporciona a menor entropia (θ_{min}) será o ângulo desejado, ou seja, o ângulo de inclinação média dos caracteres da palavra (Equação 4.4). O próximo passo é proceder à normalização em inclinação, que é realizada através da transformação, descrita a seguir.

$$\theta_{min} = \arg(\min(E_\theta)); \tag{4.4}$$

Para cada pixel na imagem original com coordenadas (i, j) *são calculadas as suas novas coordenadas* (i', j') *na imagem normalizada, utilizando a Equação 4.5.*

$$
\begin{aligned}
j' &= \lfloor j - (M - i).tan\theta_{min} \rfloor, \\
i' &= i.
\end{aligned}
\tag{4.5}
$$

Na Equação 4.5, θ_{min} é o ângulo pelo qual se deseja rotacionar os *pixels* da imagem com relação à normal, (i, j) são as coordenadas do *pixel* na imagem original e (i', j') são as novas coordenadas do *pixel* na imagem normalizada. A Figura 4.2 mostra o histograma de projeção vertical para a imagem original e para a imagem normalizada.

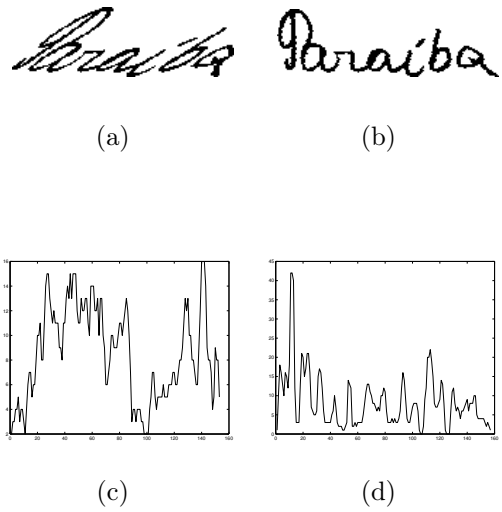
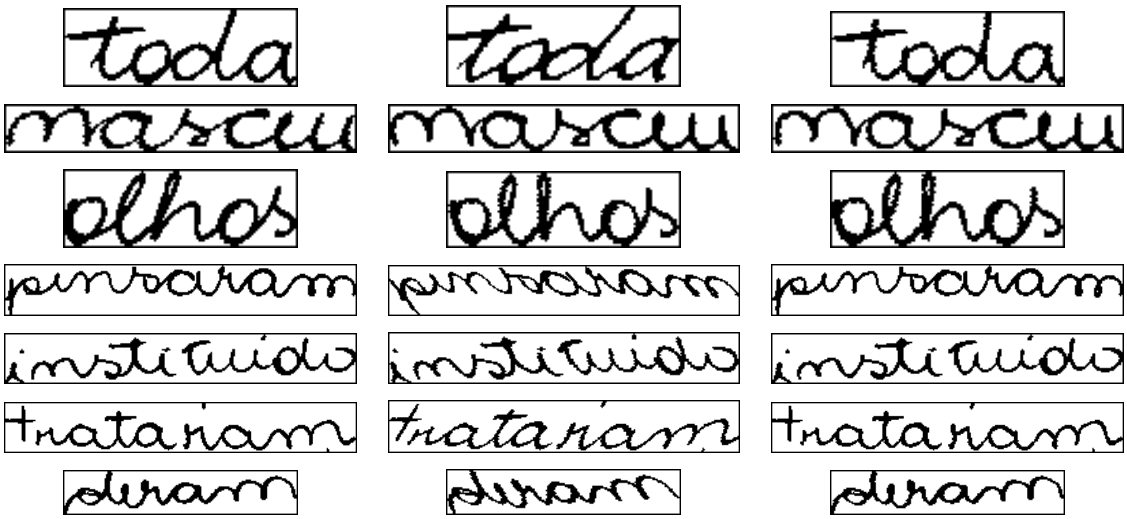


Figura 4.2: Correção da inclinação das letras da palavra utilizando o histograma de projeção vertical: (a) imagem original, (b) imagem normalizada, (c) e (d) são os respectivos histogramas de projeção vertical.

Alguns resultados obtidos com o algoritmo de normalização da inclinação desenvolvido são apresentados na Figura 4.3. Nos testes, é utilizada como entrada a imagem da palavra original ou a imagem dos traços aproximadamente verticais da palavra (imagem produzida pela operação de abertura, ilustrado na Figura 4.1). A análise do resultado do processo de normalização da inclinação vertical das letras nestas imagens mostra que o algoritmo produz melhores resultados quando é utilizada a imagem contendo os traços aproximadamente verticais da palavra, para o cálculo da inclinação média das letras na imagem. Portanto, este será o caso efetivamente utilizado neste trabalho.



(a) (b) (c)

Figura 4.3: Normalização da inclinação vertical dos caracteres das palavras. (a) imagem original, (b) imagem normalizada a partir da imagem original e (c) imagem normalizada a partir dos traços verticais da palavra.

4.1.2 Normalização da inclinação da linha de base da palavra

O método que determina e corrige a inclinação da linha de base analisa o contorno inferior da palavra. Este método é semelhante ao utilizado para a correção da inclinação média das letras de uma palavra, descrito na seção anterior. Esse método foi proposto originalmente por Côté [55]. Neste trabalho o algoritmo trabalha com os pixels pertencentes a linha de base da palavra, o que melhora a eficiência na determinação do ângulo de inclinação da linha de base e o tempo de processamento da imagem.

A primeira etapa é determinar o contorno inferior da palavra com a finalidade de evitar que os pontos que não pertençam à linha de base interfiram no cálculo da inclinação. O algoritmo utilizado para determinar o contorno inferior da palavra, realiza uma busca em todas as colunas da imagem, marcando os *pixels* pertencentes ao traçado da palavra com ordenada i de maior valor em cada coluna, já que a origem da imagem encontra-se no canto superior esquerdo da matriz de *pixels*. Os *pixels* selecionados formam o contorno inferior da palavra. A Figura 4.4 ilustra um exemplo de uma palavra e seu contorno inferior.

(a) (b)

Figura 4.4: (a) Imagem original, (b) Contorno inferior da imagem original.

A imagem resultante do processo de extração do contorno inferior é utilizada para determinação da inclinação da linha de base das palavras. A utilização da imagem contendo apenas o contorno inferior da palavra reduz os cálculos da inclinação da linha de base, uma vez que a mesma possui dimensão menor que a imagem original.

O primeiro passo é calcular o perfil de projeção horizontal inclinado (PPHI) em ângulos de inclinação que variam de -60 a 60° graus, com relação à linha de referência horizontal, com intervalos de 1 grau. O PPHI informa a quantidade de *pixels* pretos existente nas linhas inclinadas.

O algoritmo utilizado para o cálculo do perfil de projeção horizontal inclinado, para uma imagem $M \times N$, é descrito a seguir:

Para cada pixel (i, j) *da imagem;* $i = 1, 2, ..., M$ *e* $j = 1, 2, ..., N$.

Para cada ângulo de inclinação (θ)

1. *Determine o novo valor* l *da coordenada* i *na imagem*:

$$l = i + j \tan(\theta); \tag{4.6}$$

2. *Se o valor do pixel* (l, j) *for igual a* "1", *incremente a* l*-ésima coluna do* θ*-ésimo PPHI.*

$$PPHI_\theta(l) = PPHI_\theta(l) + 1; \tag{4.7}$$

A seguir são calculadas as entropias associadas a cada PPHI, utilizando a Equação 4.8.

$$E_\theta = -\sum_{l=1}^{L} P_l \log P_l. \tag{4.8}$$

em que L é o número de linhas no histograma e P_l é a probabilidade de um *pixel* preto ser encontrado na linha l.

$$\theta_{min} = \arg(\min(E_\theta)); \tag{4.9}$$

O ângulo que proporciona a menor entropia (θ_{min}) é o ângulo desejado, ou seja, o ângulo da inclinação da linha de base da palavra. Finalmente rotacionam-se os *pixels* da imagem original utilizando a transformação:

$$i' = \lfloor \; i + j \tan \theta_{min} \rfloor \tag{4.10}$$

$$j' = j, \tag{4.11}$$

A Figura 4.5, mostra a representação gráfica do cálculo da nova coordenada (i', j'). O ponto A é "transportado" para o ponto B. Note que os pontos A e B pertencem à mesma coluna da imagem (coordenada j não é afetada).

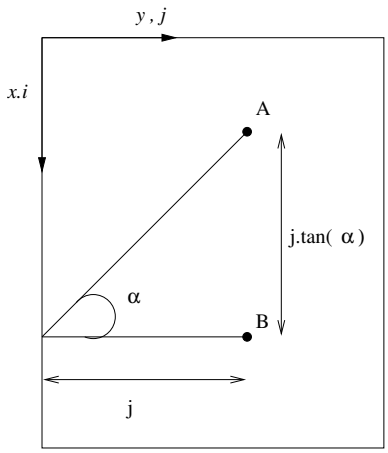
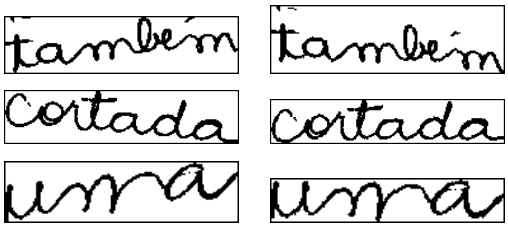


Figura 4.5: Representação gráfica do cálculo da nova coordenada (i', j').

Alguns resultados obtidos com o algoritmo de correção da inclinação da linha de base utilizado são apresentados na Figura 4.6.



(a) (b)

Figura 4.6: Normalização da inclinação da linha de base das palavras. (a) imagem original, (b) imagem normalizada.

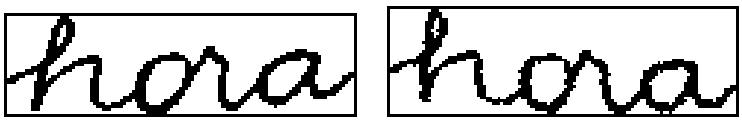
4.1.3 Suavização

No presente trabalho, devido à resolução com que foram digitalizadas as imagens, não foi inicialmente necessária a utilização de qualquer processo de suavização. Entretanto, após as etapas de normalização descritas nas seções anteriores, como mostrado na Figura 4.12, as palavras apresentaram em seus contornos pequenas saliências e buracos, necessitando do processo de suavização.

O processo de suavização tem como objetivo retirar da imagem original pontos isolados (ruído) e reduzir picos e buracos existentes no contorno da imagem. O ruído pode ser resultado da resolução com que o *scanner* capta as imagens, do processo de escrita e/ou de sujeira na tela do *scanner*, como também das operações de normalização da inclinação e da inclinação da linha de base da palavra, descritas nas seções anteriores. A Figura 4.7 ilustra o ruído provocado pelos processos de normalização da inclinação vertical e da inclinação da linha de base da palavra. Observa-se na Figura 4.7 (b) que o contorno da palavra apresenta uma série de saliências (picos) e buracos não existentes na imagem original.

O algoritmo de suavização implementado neste trabalho utiliza o casamento da imagem com máscaras. Estas máscaras estão divididas em duas categorias: as que tratam com *pixels* isolados e as que tratam dois *pixels*.

As máscaras básicas utilizadas, para tratar com *pixels* isolados, são mostradas na Figura 4.8. Além destas, são usadas outras 14 máscaras obtidas pelo espelhamento das



(a) (b)

Figura 4.7: Exemplo de ruído provocado pelos processos de normalização da inclinação média dos caracteres e da inclinação da linha de base. (a) imagem original, (b)imagem normalizada em inclinação vertical e inclinação da linha de base.

máscaras x_l e x_c e rotacionamento das mesmas de 90, 180 e 270 graus. Os x, nestas máscaras, podem representar tanto *pixels* pretos como *pixels* brancos, o número 1 representa os *pixels* pretos e o número 0 os *pixels* brancos. Quando ocorrer o casamento entre qualquer uma dessas máscaras com uma janela da imagem, o elemento central da janela, definido pelas coordenadas (x, y) na imagem, tem seu valor modificado (0 para 1 ou 1 para 0).

x	1	x
1	0	1
x	1	x

x_a

0	0	0
0	1	0
0	0	0

x_e

0	0	x
0	1	1
0	0	1

x_c

1	1	x
1	0	0
1	1	0

x_r

Figura 4.8: Máscaras utilizadas para remoção de ruído pontual.

As máscaras básicas para tratar com ruído de dois *pixels*, desenvolvidas neste trabalho, são ilustradas na Figura 4.9. Quando ocorre o casamento entre qualquer uma destas máscaras com a janela da imagem, os elementos da imagem que correspondem às posições sombreadas das máscaras têm seu valor alterado (0 para 1 ou 1 para 0).

O resultado do processo de suavização é ilustrado na Figura 4.10.

0	0	1	1	0	0
1	1	1	1	1	1

x_a

1	1	0	0	1	1
x	x	1	1	x	x

x_b

1	1	1	1	1	1
0	0	1	1	0	0

x_c

x	x	1	1	x	x
1	1	0	0	1	1

x_d

1	x
1	x
0	1
0	1
1	x
1	x

x_e

x	1
x	1
1	0
1	0
x	1
x	1

x_f

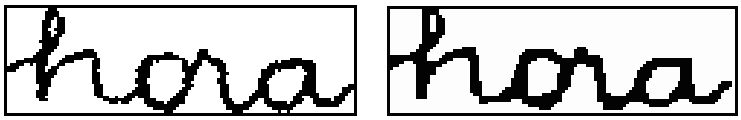
1	0
1	0
1	1
1	1
1	0
1	0

x_g

0	1
0	1
1	1
1	1
0	1
0	1

x_h

Figura 4.9: Máscaras utilizadas para remoção de ruído de 2 *pixels*.



(a) (b)

Figura 4.10: Exemplo de ruído provocado pelos processos de normalização da inclinação média dos caracteres e da inclinação da linha de base. (a)imagem normalizada em inclinação vertical e inclinação da linha de base, (b) imagem suavizada pelo algoritmo desenvolvido.

4.1.4 Análise dos resultados

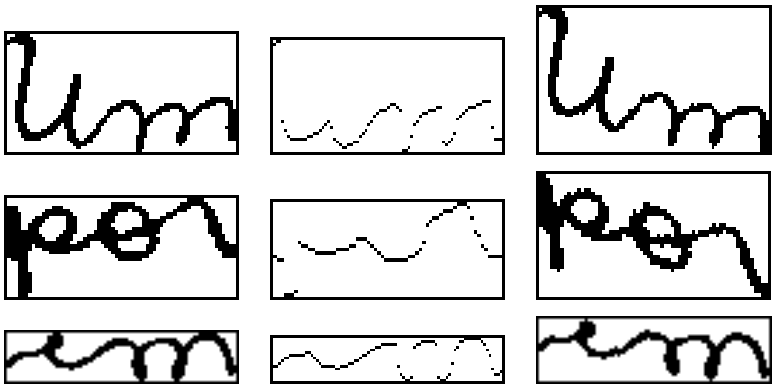
Com a finalidade de testar a eficiência dos algoritmos implementados, foram pré-processadas todas as 16000 imagens de palavras manuscritas no formato PAV contidas na base de dados. A análise dos resultados obtidos foi realizada através de uma inspeção visual subjetiva das imagens.

A partir da análise realizada constatou-se que em apenas duas imagens o processo de normalização da inclinação vertical das palavras não obteve resultados satisfatórios, o que significa que $99,98\%$ das imagens da base de dados tiveram a inclinação corretamente corrigida.

Com relação à normalização da inclinação da linha de base, apenas 58 (0.36%) das imagens não produziram resultados satisfatórios. Foi realizada uma análise visual minuciosa das imagens originais, das imagens contendo os contornos inferiores das palavras e dos resultados obtidos, com a finalidade de determinar a causa do problema. Constatou-se, como ilustrado na Figura 4.11, que o contorno inferior extraído nem sempre contém apenas os *pixels* pertencentes ou localizados próximos à linha de base da palavra. A presença de *pixels* não pertencentes à linha de base ou pixels localizados distantes da linha de base do contorno inferior extraído prejudica a determinação da inclinação da linha de base da palavra. A solução encontrada é realizar uma filtragem, após a determinação do contorno inferior da palavra, com a finalidade de eliminar os *pixels* estranhos.

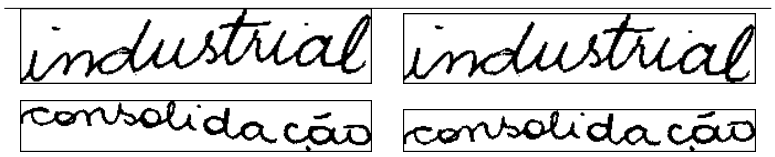
O método de determinação da inclinação da linha de base da palavra foi desenvolvido baseado na hipótese de que a mesma é uma linha reta. A análise dos resultados indica que em 99% dos casos esta suposição é verdadeira. Nos casos em que tal suposição é falsa, como ilustrados na Figura 4.12, a solução é considerar a linha de base como uma sucessão de segmentos, conectando os pontos de mínimo do traçado da palavra.

Os resultados finais obtidos quando a imagem de entrada é submetida às operações de normalização das inclinações vertical e da linha de base, bem como à suavização, são ilustrados na Figura 4.13.



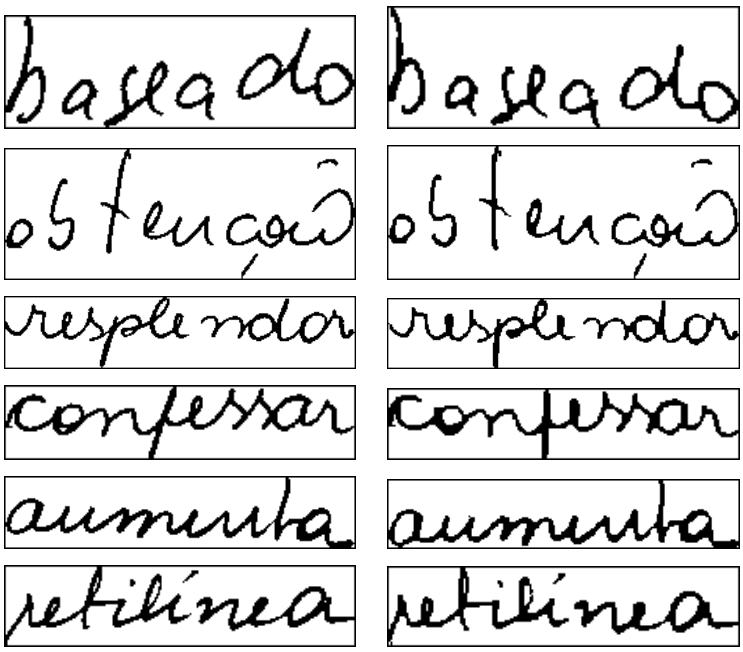
(a) (b) (c)

Figura 4.11: Resultado do processo de normalização da inclinação da linha de base da palavra. (a) imagem original, (b) contorno inferior da imagem original e (c) resultado final do processo.



(a) (b)

Figura 4.12: Resultado do processo de normalização da inclinação da linha de base da palavra. (a) imagem original, e (b) resultado final do processo.



(a) (b)

Figura 4.13: Resultado final do pré-processamento. (a) imagem original, (b)imagem normalizada na inclinação vertical e linha de base e suavizada.

4.2 Segmentação de palavras manuscritas

A concepção de um sistema de reconhecimento de palavras manuscritas está fortemente correlacionada às restrições impostas aos seguintes problemas: como tamanho do vocabulário, dependência do escritor e estilo de escrita, dentre outras. Quando o vocabulário da aplicação em questão é grande (milhares ou mais de palavras), como em aplicações postais, muitas técnicas correntes utilizam uma abordagem analítica. Neste tipo de abordagem, as palavras são inicialmente segmentadas em unidades menores, que podem representar um caractere ou parte deste.

A segmentação das palavras é uma tarefa difícil de ser realizada, devido a fatores como ambigüidade, sobreposição e concatenação de caracteres, bem como devido à possibilidade do mesmo padrão de *pixels* representar distintos caracteres em diferentes palavras.

A escrita manuscrita ocidental é gerada pelo movimento do instrumento de escrita (lápis ou caneta) sobre uma superfície, da esquerda para direita em uma direção horizontal. Esse movimento produz traços horizontais unindo caracteres adjacentes, que são denominados de pontes ou traços de ligação, conforme é ilustrado na Figura 4.14. Estes traços de ligação são candidatos naturais a possuírem pontos de segmentação. Para lidar com traços de ligação foram utilizados três métodos de segmentação. O primeiro método, denominado algoritmo morfológico de segmentação (AMS) foi inicialmente proposto por Chen et al. [36], tendo sido modificado neste trabalho. O segundo e o terceiro métodos são denominados de método morfológico de segmentação utilizando algoritmo genético (AMSAG) e método de segmentação por histogramas de projeções (ASHP), respectivamente, ambos sendo contribuições originais deste trabalho [109, 110].

Os traços de ligação entre caracteres cursivos não são encontrados quando os mesmos estão fortemente conectados ou sobrepostos nas palavras, como ilustra a Figura 4.14. Para estes casos foi desenvolvido um algoritmo de segmentação, que utiliza pontos de alta curvatura (mínimos locais do contorno) para gerar uma lista de todos os possíveis pontos de segmentação.

Figura 4.14: Exemplo de ligações entre caracteres e caracteres fortemente conectados. O ponto de segmentação entre caracteres fortemente conectados é ilustrado através dea seta.

Todos os pontos gerados pelos diversos algoritmos de segmentação são unidos e validados por um conjunto de regras heurísticas, desenvolvidas através de testes com o conjunto de treinamento. Os pontos definitivos de segmentação só serão determinados durante o processo de reconhecimento. O módulo de segmentação fornece como saída as coordenadas cartesiana "x" dos pontos de segmentação e o ângulo de inclinação do traço de segmentação associado ao ponto.

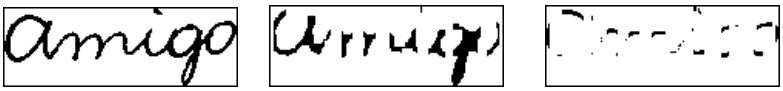
4.2.1 Segmentação nos traços de ligação

Método 1: Algoritmo Morfológico de Segmentação (AMS)

Conforme anteriormente relatado, Chen et al. [36] desenvolveram um algoritmo de segmentação baseado em operações morfológicas, o qual foi adaptado e implementado neste trabalho. A seguir é feita uma descrição detalhada do método de segmentação de Chen e de outros algoritmos desenvolvidos: a) para rotular ilhas, b) algoritmo para a determinação da existência de pontes entre ilhas A e B e c) algoritmo modificado de Viterbi para implementação do algoritmo AMS. Estes algoritmos desenvolvidos constituem contribuições originais desta Tese.

Chen et al. [36, 38] classificam cada *pixel* pertencente ao traçado da palavra em: singular (localizado em ilha) ou regular (localizado em ponte), baseado em sua contribuição à forma do caractere. Ilhas são os traços aproximadamente verticais e pontes são os traços aproximadamente horizontais existentes no traçado da palavra, conforme ilustrado na Figura 4.15. Os pontos de segmentação são posicionados sobre as pontes

e o problema de segmentação, desta forma, se reduz à determinação das pontes.



(a) (b) (c)

Figura 4.15: Exemplo das ilhas e pontes na palavra *amigo*: (a) palavra original; (b)ilhas e (c)pontes.

Chen et al. [36] utilizam as operações morfológicas de abertura e fechamento para obter as ilhas. As pontes são determinadas através do complemento das ilhas em relação à imagem original da palavra.

Considere uma imagem de entrada $X_i(i, j)$ de dimensão $M \times N$. A imagem contendo as ilhas é determinada por:

$$X_{ilhas} = X_i \circ el_1. \tag{4.12}$$

Em que o símbolo $\circ$ representa a operação morfológica de abertura utilizando o elemento estruturante el_1. Este elemento estruturante consiste de uma linha vertical, com comprimento igual à largura média do traçado da palavra. O histograma de projeção vertical e a quantidade de transições entre *pixels* pretos e brancos ao longo das colunas são utilizados para estimar a largura média dos traços que compõem as palavras.

A próxima etapa consiste na fusão de ilhas adjacentes, sendo expressa por:

$$X_o = X_{ilhas} \bullet el_2. \tag{4.13}$$

O símbolo $\bullet$ representa a operação de fechamento e o elemento estruturante el_2 é uma matriz 5×5 contendo uma linha horizontal central de um *pixel* de largura. A imagem contendo as pontes é obtida através do complemento da imagem de entrada X_i com a imagem X_o, ou seja:

$$X_p = X_i \sim X_o; \tag{4.14}$$

na qual símbolo $\sim$ representa a operação de complemento.

A imagem X_p pode conter traços horizontais que não conectam duas ilhas, portanto não pertence ao conjunto de pontes da palavra. Desta forma é necessário identificar as verdadeiras pontes, que conectam caracteres adjacentes. A solução é a determinação do caminho de menor custo, ou seja, o caminho (seqüência de pontes) que permita viajar da ilha mais à direita para a ilha mais à esquerda como menor custo. O algoritmo modificado de Viterbi [2], desenvolvido neste trabalho de Tese, foi utilizado para resolver este problema eficientemente, com a matriz de custo dada por:

$$custo(i,j) = \begin{cases} imóvel(i), & \text{se } i = j; \\ ponte(i,j), & \text{se existe um ponte entre as ilhas } i \text{ e } j; \\ barco(i,j), & \text{caso contrário.} \end{cases} \tag{4.15}$$

Estes três tipos de custo são calculados por [36]:

$$ponte(i,j) = ilhas - (j - i); \tag{4.16}$$

$$imóvel(i) = ilhas^3; \tag{4.17}$$

$$barco(i,j) = \begin{cases} ilhas \times (j - i), & \text{se } j > i; \\ ilhas^2 \times (j - i), & \text{se } j < i. \end{cases} \tag{4.18}$$

O parâmetro *ilhas* representa o número de traços horizontais da imagem X_p. $Ponte(i,j)$ é o custo para ir da ilha i para a ilha j sobre uma ponte, *barco* é custo de barco da ilha i para a ilha j, $imóvel(i)$ é o alto custo de permanência na ilha i. Desta forma, é atribuído o menor custo ao caminho que contém os traços horizontais que conectam as ilhas. Os pontos de segmentação são posicionados sobre o ponto médio de cada ponte percorrida.

Antes de aplicar o algoritmo de Viterbi são descartados todos os traços horizontais na imagem X_p que possuam um comprimento menor que a largura média do traçado, o que não caracteriza um traço de ligação, mas um pedaço do caractere.

[2] A modificação introduzida permite o algoritmo trabalhar com um número não fixo de observações.

Considerações sobre a implementação do algoritmo

Para gerar a matriz de custo utilizada pelo algoritmo de Viterbi é necessário a rotulação de cada ilha e cada ponte encontrados nas imagens X_p e X_o, bem como determinar a existência ou não de pontes entre ilhas. Estes procedimentos, embora fáceis de serem executados por humanos, envolvem certo grau de complexidade quando implementados via *software*. A seguir são descritos os algoritmos utilizados:

Algoritmo para rotular ilhas

Para realizar a rotulação das ilhas encontradas na imagem são necessárias as seguintes considerações:

- A imagem processada é binária: *pixels* brancos (fundo) possuem valor "*0*" e *pixels* pretos (traçado do caractere) possuem valor "1".
- Denota-se p o *pixel* com valor "1" analisado em um momento qualquer do procedimento, denota-se por r o vizinho acima de p e por t o vizinho à esquerda de p, conforme ilustrado na Figura 4.16.

	r	
t	p	

Figura 4.16: Vizinhança de um ponto p.

- Atribui-se um rótulo para cada *pixel* preto p analisado pelo procedimento. O rótulo não representa o valor do *pixel* na imagem, podendo ser, por exemplo, um número ou uma letra do alfabeto, usado para identificar as ilhas.

Estabelecidas as precedentes considerações, a rotulação das ilhas é realizada pelo algoritmo:

1. Percorrer a imagem *pixel*, a *pixel* da esquerda para a direita, de cima para baixo:

2. Para cada *pixel* p, pertencente ao traçado da palavra, verificar os valores dos *pixels* r e t:

a) Se $r = t = 0$, um rótulo ainda não utilizado é atribuído a p;

b) Se $r = 1$ e $t = 0$, o rótulo r é atribuído a p;

c) Se $r = 0$ e $t = 1$; o rótulo t é atribuído a p;

d) Se $r = t = 1$ e seus rótulos são iguais, atribui-se o mesmo rótulo a p;

e) Se $r = t = 1$ e seus rótulos são diferentes, atribui-se a p o rótulo menor. Deste modo os *pixels* r e t têm rótulos equivalentes e estão conectados por p.

No final da varredura todos os *pixels* pertencentes às ilhas estão rotulados.

A Figura 4.17 ilustra o resultado do processo de rotulação das ilhas em uma imagem. Nesta figura, para facilitar a ilustração, foi atribuído um nível de cinza diferente a cada rotulo da imagem (b). Os *pixels* que são representados por elementos com rótulos iguais (um número maior do que zero identifica elementos conexos) pertencem a uma mesma ilha.

(a) (b)

Figura 4.17: Processo de rotulação de ilhas: (a) palavra manuscrita; (b)ilhas.

Determinação da existência de pontes entre as ilhas A e B

Para determinar se existe uma ponte que conecte as ilhas A e B é preciso realizar a seqüência de operações descrita a seguir:

1. Criar uma nova imagem X_{AB}. Esta imagem contém apenas as ilhas rotuladas A e B. Ilhas com rótulos diferentes de A e B são apagados.

2. Unir a imagem das pontes, X_p, com a imagem X_{AB}. O resultado desta união produz uma nova imagem denominada X_u.

3. Rotular as ilhas encontradas na imagem X_u.

Se o rótulo de um *pixel* pertencente à ilha A for o mesmo de um *pixel* pertencente à ilha B na imagem X_u então as ilhas A e B estão conectadas através de uma ponte. Caso contrário não existe ponte que conecte as ilhas A e B.

Algoritmo modificado de Viterbi

O algoritmo modificado de Viterbi (MV) visa encontrar o melhor caminho através de uma treliça, dados os estados inicial e final e a matriz custo. O melhor caminho é definido pela aplicação em questão, por exemplo, no problema de qualificação de pontes o melhor caminho é o de menor custo. Ocorrendo empate entre dois ou mais caminhos, o caminho vencedor será o de menor percurso. A treliça, neste caso, é um grafo que representa o diagrama de estados de uma máquina de estados finita. Cada nó no grafo representa um estado e cada arco representa uma possível transição entre dois estados.

Antes de definir o algoritmo MV, os seguintes símbolos devem ser definidos:

N - Número de estados do grafo.

Y_i - i-ésimo estado do grafo.

Y_o - Estado inicial do grafo.

Y_f - Estado final do grafo.

a_{ij} - Probabilidade ou custo de transição entre o estado i e o estado j.

$\delta_{Y_i}(n)$-Custo do i-ésimo estado do grafo no tempo n.

O algoritmo modificado de Viterbi pode ser assim definido:

Inicialização: $n = 0$

$$
\begin{aligned}
\delta_{Y_i}(0) &= \begin{cases} 0, & \text{se } i = 0 \text{ (estado inicial)} \\ Max & \text{se } 1 \leq i \leq N \text{ e } i \neq 0. \end{cases} \\
\psi_{Y_i}(0) &= 0, \text{se } 1 \leq i \leq N; \\
\sigma(0) &= \delta_{Y_f}(0); \\
\vartheta(0) &= 0.
\end{aligned} \tag{4.19}
$$

Cálculo:$n \neq 0$

$$
\begin{aligned}
\delta_{Y_i}(n) &= \min_{1 \leq j \leq N} (\delta_{Y_j}(n-1) + a_{ij}), 1 \leq i \leq N; \\
\psi_{Y_i}(n) &= \arg \min_{1 \leq j \leq N} (\delta_{Y_j}(n-1) + a_{ij}), 1 \leq i \leq N; \\
\vartheta(n) &= \min_{1 \leq j \leq N} (\delta_{Y_j}(n)); \\
\sigma &= \min_{0 \leq z \leq n} (\delta_{Y_f}(z)) \\
\beta &= n.
\end{aligned} \tag{4.20}
$$

Critério de parada:

$$
\sigma < \vartheta(n). \tag{4.21}
$$

Caminho:

$$
q_i = \psi_{Y_{q_{i+1}}}(i+1), i = \beta - 1, \beta - 2, ..., 1. \tag{4.22}
$$

em que Max é o valor máximo da máquina.

O algoritmo MV, a cada instante de tempo e para cada estado do grafo, analisa qual o melhor caminho para chegar ao estado presente a partir do estado inicial. Desta forma, para cada estado existe apenas um caminho vencedor a cada instante de tempo. A Figura 4.18 ilustra uma estrutura de treliça. A modificação introduzida neste algoritmo permite que o número n de observações não seja fixo. A finalização do algoritmo

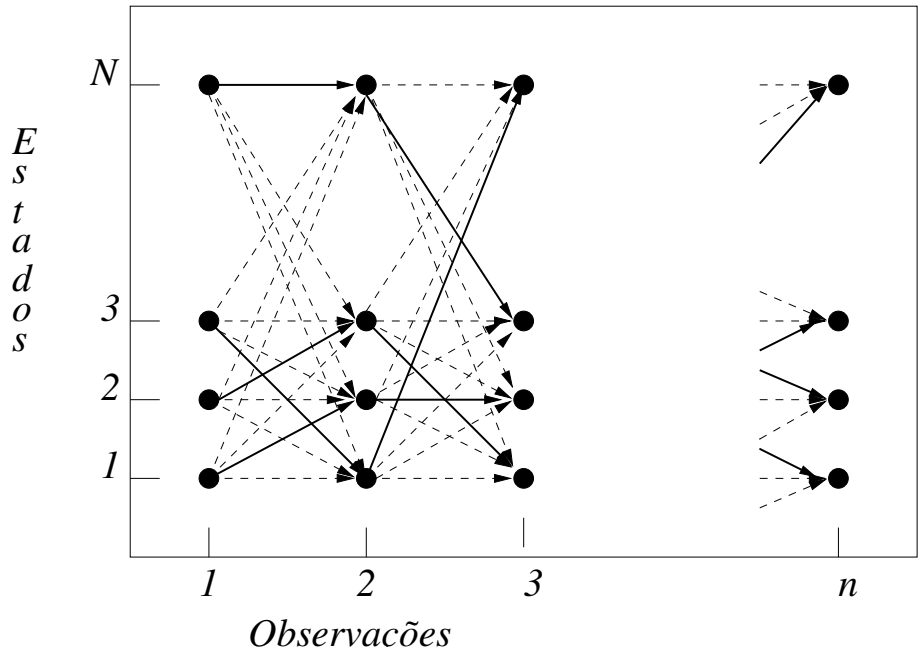


Figura 4.18: Exemplo de uma treliça.

é determinada pelo teste do critério de parada. Enquanto a inequalidade 4.21 for verdadeira o algoritmo continua a ser executado.

O algoritmo MV foi implementado utilizando a linguagem de programação C padrão (ANSI). A uniformidade dos dados a cada instante e o não conhecimento a priori dos recursos computacionais (memória) necessários à execução do algoritmo motivou a utilização de uma lista encadeada à esquerda.

Método 2: Segmentação utilizando algoritmo genético e morfologia matemática

Neste trabalho foi desenvolvido um algoritmo de segmentação de palavras [109], baseado nas seguintes hipóteses: 1) existem pontos naturais de segmentação (SNPs) entre caracteres desconectados e 2) os pontos de segmentação entre caracteres conectados estão localizados sobre os traços de ligações. As operações morfológicas de abertura e fechamento são utilizadas na determinação dos traços de ligações entre as letras. Um algoritmo genético é utilizado na busca dos elementos estruturantes ótimos que

possibilitem a determinação dos traços de ligações [109].

Este método é dividido em duas partes básicas. A primeira parte é dedicada à determinação da existência dos pontos de segmentação natural, correspondendo aos espaços em branco entre caracteres desconectados. A segunda parte objetiva determinar os pontos de segmentação sobre as ligações.

Segmentação Natural

A presença de descontinuidades nos manuscritos é utilizada pelo processo que determina os pontos de segmentação natural. O procedimento adotado é a análise do perfil de projeção vertical, conforme ilustrado na Figura 4.19, também chamado de histograma de projeção vertical, com objetivo de identificar espaços em branco entre caracteres. O histograma é obtido pela contagem dos *pixels* pretos em cada coluna da imagem [11].

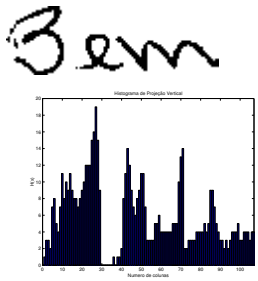


Figura 4.19: Histograma de projeção vertical.

O procedimento utilizado neste trabalho para determinar a existência e a localização dos pontos de segmentação natural utiliza perfis de projeção inclinados (*angled projection profile*), em cinco diferentes ângulos [77]. Durante o pré-processamento a correção da inclinação é utilizada para normalizar as palavras manuscritas, com objetivo de ter palavras com inclinação média zero dos caracteres em relação à vertical. Entretanto, uma inclinação média nula não significa necessariamente que todos os caracteres que compõe a palavra não estejam inclinados. Desta forma, o perfil de projeção inclinado necessita ser calculado. Os ângulos utilizados para o cálculo do perfil de projeção inclinado são -20, -10, 0, 10, 20 graus, com relação à vertical. Estes ângulos foram

determinados empiricamente através de experimentos. O perfil de projeção inclinado indica a quantidade de *pixels* pretos existentes em linhas inclinadas. Valores nulos correspondem a espaço em branco entre caracteres.

O algoritmo proposto para o cálculo do perfil projeção inclinado, para uma imagem de tamanho $M \times N$, é descrito a seguir:

Para cada pixel (i, j) *da imagem;* $i = 1, 2, ..., M$ *e* $j = 1, 2, ..., N$.

Para cada ângulo de inclinação (K)

1. *Determine o novo valor da coordenada* j *na imagem, como sendo*:

$$j = \lfloor j - (M - i).tan(K) \rfloor; \tag{4.23}$$

2. *Se o valor do pixel* (i, v) *for igual a* "1", *incremente a* v*-ésima coluna do* k*-ésimo histograma.*

$$Histograma_K(j) = Histograma_k(j) + 1; \tag{4.24}$$

Em que $Histograma_K(j)$ indica a quantidade de *pixels* pretos existentes na linha j inclinada com ângulo k

Os pontos de segmentação correspondem aos pontos médios das seqüências de zeros detectados pelos perfis de projeção inclinados.

No final deste processo, tem-se a localização dos pontos de segmentação natural, armazenados em 5 vetores, um para cada ângulo K analisado. Estes vetores que armazenam as localizações dos pontos de segmentação (i, j) armazenam apenas a coordenada j de cada os ponto de segmentação, já que a coordenada i é determinada através do cálculo da linha de base da palavra, sendo a mesma para todos os pontos de segmentação.

A Figura 4.20(a)-(f) mostra exemplos de segmentação natural para a palavra *Bem* utilizando linhas de separação com -20°, -10°, 0°, 10° e 20°.

Muitos pontos de segmentação não são determinados em todos os cinco vetores de segmentação (um para cada ângulo). Entretanto, eles são geralmente determinados em

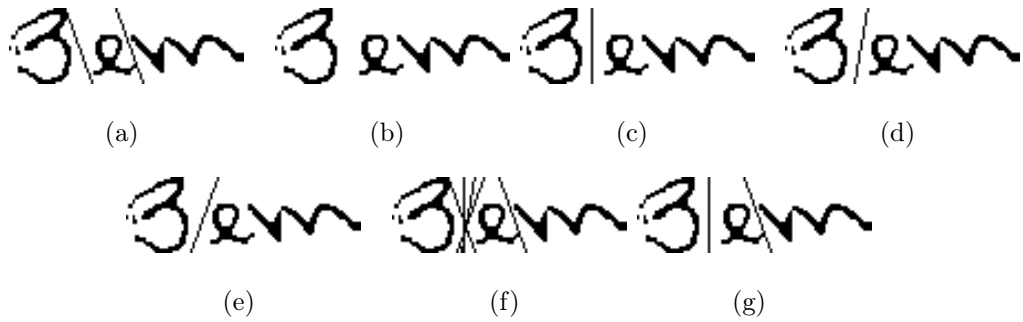


Figura 4.20: Exemplos de segmentação natural nos seguintes ângulos a) -20, b) -10, c) 0, d) 10, e) 20 graus, f) união entre os pontos de segmentação natural g) resultado final do processo de segmentação natural.

pelo menos um deles. Realizando a união do conteúdo de todos os vetores, obtêm-se todos os pontos de segmentação natural, resultando em mais de um ponto de segmentação para a mesma fronteira entre caracteres sucessivos. Estes pontos devem ser identificados e agrupados, evitando assim o que é conhecido como sobre-segmentação (*oversegmentation*). Para realizar esta tarefa foi utilizada a seguinte regra heurística: se a distancia entre dois ou mais pontos de segmentação natural for maior que duas vezes a largura média dos traços que formam a palavra, então os pontos que correspondem às linhas de separação com maiores ângulos são eliminados.

A Figura 4.20(f) mostra o resultado da união entre os pontos de segmentação obtidos utilizando os cinco ângulos de inclinação. Na Figura 4.20(g) é mostrado o resultado final do processo de segmentação natural.

Segmentação sobre ligações

Testes realizados provaram a eficiência dos operadores morfológicos de abertura e fechamento na determinação dos traços de ligação entre os caracteres. A escolha dos elementos estruturantes é fortemente relacionada à largura e à altura dos traços que compõem a palavra analisada, existindo uma grande variedade de elementos estruturantes que podem ser utilizados.

Algoritmos genéticos (AG) são métodos de busca que utilizam processos baseados em mecanismos de seleção natural e genética [114]. Estes algoritmos utilizam operado-

res específicos para encontrar estruturas ótimas em um ambiente particular. Ao longo dos últimos anos, algoritmos genéticos estabeleceram-se como uma técnica eficiente e robusta para problemas requerendo busca em espaços complexos, tendo sido utilizados no projeto de filtros morfológicos [114].

Um algoritmo genético é utilizado neste trabalho para encontrar os elementos estruturantes ótimos que permitem determinar os traços de ligação entre as letras que formam as palavras do conjunto de treinamento. O algoritmo genético implementado é baseado no artigo de Harvey e Marshall [114], com os seguintes parâmetros: tamanho da população de 80, taxa de cruzamento de 0,9, taxa de mutação de 0,050 e o critério de parada sendo o número máximo de gerações (200 gerações). No processo de otimização do algoritmo genético foi utilizada a base de palavras manuscritas dependente do escritor descrita no Capitulo 5. A seguir são mostrados alguns detalhes das operações utilizadas pelo AG.

Elementos estruturantes Os elementos estruturantes utilizados nas operações morfológicas (abertura e fechamento) possuem dimensão fixa (3x3). Desta forma, o cromossomo do AG, ilustrado na Figura 4.21, é formado através da concatenação das duas matrizes binárias que representam os elementos estruturante. Esta situação é ilustrada na Figura 4.21.

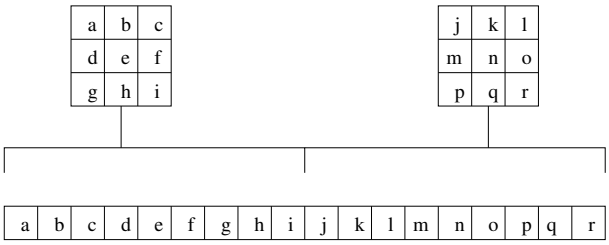


Figura 4.21: Elementos estruturantes e cromossomos.

Algoritmo de otimização Um processo de otimização supervisionado é utilizado para o algoritmo genético. Deste modo, é apresentada uma seqüência de imagens de entrada ao algoritmo e associada a cada imagem de entrada, uma imagem de saída

de referência. A Figura 4.22 mostra um exemplo de uma imagem de entrada e sua imagem de referência. A imagem de referência é uma versão da imagem de entrada na qual foram apagados os *pixels* que pertencem aos traços de ligação entre as letras.

Figura 4.22: Imagem de entrada e de referência.

No processo de otimização as atualizações nos cromossomos são realizadas após apresentação de todos os pares de imagens (entrada e referência) contidos no conjunto de treinamento.

Considere imagens com dimensão $M \times N$. $X_i(i,j)$ e $X_r(i,j)$ são as imagens de entrada e a correspondente imagem de referência, respectivamente. $X_o(i,j)$ é a imagem dada por:

$$X_o = (X_i \circ el_1) \bullet el_2; \tag{4.25}$$

os símbolos $\circ$ e $\bullet$ denotam as operações morfológicas de abertura e fechamento respectivamente.

O custo individual para um par de imagens X_i e X_r é definido como:

$$f_o(x) = \sum_{i=0}^{M} \sum_{j=0}^{N} X_z(i,j); \tag{4.26}$$

com X_z calculado através da seguinte Equação:

$$X_z(i,j) = \begin{cases} 200, & \text{se } X_i(i,j) = 1 \text{ e } X_r(i,j) = 0 \text{ e } X_o(i,j) = 0; \\ 2, & \text{se } X_i(i,j) = 0 \text{ e } X_r(i,j) = 0 \text{ e } X_o(i,j) = 0; \\ 2, & \text{se } X_i(i,j) = 1 \text{ e } X_r(i,j) = 1 \text{ e } X_o(i,j) = 1; \\ -200, & \text{caso contrário.} \end{cases} \tag{4.27}$$

Os valores 2 e 200 da Equação 4.27 foram escolhidos empiricamente para bonificar

e penalizar o processo quando o algoritmo acertar e errar a localização dos traços de ligação, respectivamente.

Finalmente a função objetivo é definida como:

$$f = \sum_{x=0}^{L} f_o(x); \tag{4.28}$$

sendo L o número de imagens de entrada no conjunto de treinamento.

O custo individual impõe uma penalidade proporcional à diferença entre a imagem de referência $X_r(i,j)$ e a imagem de saída $X_o(i,j)$.

Cada cromossomo é reproduzido de acordo com seu valor de adequabilidade, em relação à adequabilidade média de toda a população, usando um esquema conhecido como amostragem determinística [114]. O algoritmo genético utilizado neste trabalho para determinar traços de ligação em palavras, desta forma, visa maximizar a função objetivo. O critério de parada utilizado é o número de ciclos de reprodução (200), ou seja, o número de gerações. A grande vantagem deste método é que nenhuma normalização das dimensões das imagens necessita ser realizada, ao contrario do ocorre quando se utiliza redes neurais, por exemplo.

Determinação dos pontos de segmentação sobre as ligações

As operações morfológicas com os elementos estruturantes encontrados pelo AG podem detectar falsos traços de ligação. Desta forma, é necessário qualificar estes traços de ligação em falsos e verdadeiros. O procedimento adotado é o mesmo utilizado pelo algoritmo de Chen [36].

Após a detecção dos pontos de segmentação sobre os traços de ligação é necessário realizar a união entre estes e os pontos de segmentação natural. Para agrupar estes pontos de segmentação foi utilizada a seguinte regra heurística: se a distancia entre dois ou mais pontos de segmentação for maior que duas vezes a largura média dos traços que formam a palavra, então os pontos correspondentes às linhas de separação com maiores ângulos são eliminados.

Método 3: Segmentação por histogramas de projeção

Nas palavras manuscritas em que os caracteres encontram-se desconectados, os pontos de segmentação são encontrados apenas analisando o perfil de projeção inclinado, como explicado na Seção 3.1. Entretanto, se a palavra manuscrita é puramente cursiva, este procedimento não faz sentido, pois não existe nenhum ponto no histograma de projeção vertical que tenha valor zero, como exemplificado na Figura 4.23.

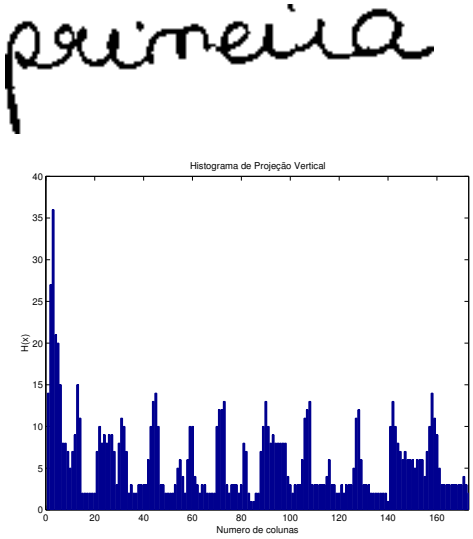


Figura 4.23: Palavra cursiva e seu histograma de projeção vertical.

Analisando o perfil de projeção para as palavras da base de dados, observou-se que a quantidade média de *pixels* pretos nas colunas onde existem os traços de ligação corresponde à largura média dos traços da palavra, calculada através do algoritmo desenvolvido por Chen [36, 38]. Com base nesta constatação e no critério de segmentação utilizado (cada caractere pode ser segmentado no máximo em cinco partes), foi desenvolvido um algoritmo de segmentação baseado no perfil de projeção inclinado e na estimativa da largura média dos traços que compõem a palavra [110].

A primeira etapa da determinação dos pontos de segmentação sobre as ligações é o "preenchimento" de todas as cavidades centrais da palavra (*hole-filling*). Por cavidades centrais entende-se uma região da imagem composta por *pixels* brancos e delimitada por *pixels* pretos. Esta operação é necessária para prevenir que o algoritmo de determinação

das ligações detecte pontos de segmentação no meio de letras como "b", "o" e "a", dentre outras que possuem cavidades centrais.

O perfil de projeção inclinado, em ângulos de -20, -10, 0, 10 e 20 graus com relação à vertical, é calculado para a imagem com cavidades preenchidas. A determinação dos traços de ligação é feita analisando o histograma de projeção inclinado para cada ângulo K, conforme as Equações 4.23 e 4.24. Para cada $Histograma_K$, uma projeção menor ou igual a duas vezes a largura média dos traços que compõem a palavra é considerada como pertencente à ligação e o traço de ligação é marcado. O ponto de segmentação corresponde ao ponto médio de cada traço de ligação marcado. A posição dos pontos de segmentação e o valor do ângulo de inclinação K são armazenados para um posterior processamento, que realizará a combinação com os demais pontos de segmentação.

4.2.2 Segmentação de caracteres fortemente conectados ou sobrepostos

Os estudos realizados com a base de treinamento revelaram que os pontos de segmentação que separam caracteres fortemente conectados ou sobrepostos estão localizados geralmente nos mínimos locais do contorno.

Dois algoritmos distintos foram desenvolvidos. O primeiro método determina os pontos do contorno da palavra onde há uma mudança na orientação da curvatura, através da variação angular do contorno da palavra. Esta abordagem constitui uma alternativa para métodos tradicionais de análise de curvatura. São considerados como possíveis pontos de segmentação aqueles que apresentam uma variação angular acima de um determinado limiar, determinado empiricamente. O segundo algoritmo determina os possíveis pontos de segmentação através da análise do contorno da imagem, procurando localizar mínimos locais do contorno. Testes realizados com ambos os métodos indicaram que o segundo método é mais eficaz e menos complexo. Portanto, a seguir é realizada uma explanação mais detalhada do segundo algoritmo de segmentação.

Para o funcionamento do algoritmo são necessárias as coordenadas dos pixels do

contorno da imagem analisada. Primeiramente é realizada uma operação de abertura na imagem de entrada pré-processada, com o elemento estruturante tipo "cruz" (3×3), seguida por uma operação de diferença ente a imagem resultante da operação de abertura e a imagem de entrada pré-processada com as cavidades centrais preenchidas. A imagem resultante contém apenas o contorno externo da palavra. Finalmente, um algoritmo de rastreamento percorre o contorno extraído, determinando a localização de cada um dos pontos. A técnica usada para representar o contorno foi o código de cadeia [66].

O algoritmo de segmentação proposto obtém os possíveis pontos de segmentação através de análise das coordenadas "y" de todos os *pixels* do contorno da palavra, com o objetivo de detectar os mínimos locais. O j-ésimo ponto do contorno, $P(x_j, y_j)$, é considerado como um possível ponto de segmentação se as coordenadas y dos pontos $P(x_j, y_j)$, $P(x_{j-1}, y_{j-1})$, $P(x_{j+1}, y_{j+1})$, $P(x_{j-2}, y_{j-2})$ e $P(x_{j+2}, y_{j+2})$, obedecerem a uma das seguintes condições:

1. $y_j > y_{j-1}$ e $y_j > y_{j+1}$;
2. $y_j > y_{j-2}$ e $y_j > y_{j+2}$;
3. $y_j > y_{j-3}$ e $y_j > y_{j+3}$.

Se o j-ésimo ponto analisado P_j satisfaz a uma das condições acima, ele é rotulado como um possível ponto de segmentação, com um rótulo que identifica a condição satisfeita. Essa identificação permite estabelecer uma ordem de prioridade, para a realização de uma triagem nesses pontos. Os pontos marcados como satisfazendo à condição número 1 tem uma maior prioridade, seguidos dos pontos que obedecem às condições 2 e 3, respectivamente. Se um ponto satisfaz a mais de uma condição prevalece a condição de maior prioridade.

Testes iniciais indicaram a necessidade de fazer uma triagem no conjunto de candidatos a pontos de segmentação, com a finalidade de identificar e eliminar os falsos pontos de segmentação. Para o correto funcionamento desta análise foi necessário identificar a localização dos caracteres da palavra que possuem cavidades centrais. Para

tanto se utilizou o algoritmo de rotulação de cavidades centrais. Este algoritmo é similar ao algoritmo de rotulação de ilhas anteriormente descrito, rotulando os *pixels* pertencentes ao fundo da imagem. Outras duas informações necessárias são a largura média dos traços verticais e horizontais da palavra, obtida utilizando a imagem resultante da operação morfológica de dilatação anteriormente mencionada nesta seção.

Um candidato a ponto de segmentação é eliminado se estiver posicionado em ponto do contorno tal que a reta imaginária de projeção vertical do ponto atravesse uma cavidade central. Caso a projeção vertical tangencie uma cavidade central, o algoritmo promove uma busca com a finalidade de encontrar um novo ponto, à direita do candidato e dentro de uma vizinhança pré-determinada (duas vezes a largura do traçado horizontal), que possua uma projeção vertical menor que duas vezes a largura do traçado vertical e que não esteja localizado no traçado de uma cavidade central. Caso não encontre esse ponto, o mesmo processo de busca é realizado novamente, só que à esquerda dos candidatos. Este mesmo procedimento de busca é realizado caso o candidato a ponto de segmentação esteja localizado sobre uma coluna que possua um perfil de projeção vertical maior que duas vezes a largura do traçado vertical.

Outra análise dos candidatos a ponto de segmentação, é realizada para unir ou excluir pontos muito próximos entre si ou muito próximos dos pontos inicial e final do contorno, com a finalidade de eliminar sub-imagens de tamanho reduzido. Para tanto é necessário definir a vizinhança de um ponto como sendo duas vezes a largura horizontal média dos traços dos caracteres que compõem a imagem. Para cada candidato a ponto de segmentação, é definida uma vizinhança. Se mais de um ponto estiver posicionado nessa vizinhança é escolhido o ponto com a mais alta prioridade. Caso em uma determinada vizinhança exista mais de um possível ponto de segmentação com a mesma prioridade, a solução é transferida para a etapa seguinte.

Pós-segmentação

Geralmente, os limites do traçado de uma dada letra não são detectados por todos os algoritmos de segmentação, mas são usualmente detectados por pelo menos um deles.

Portanto, a partir da união de todos eles, tem-se uma melhor segmentação. Por outro lado, a união usualmente contém mais de um ponto de segmentação para a mesma ligação entre letras. Estes pontos devem ser identificados e unidos para reduzir o efeito de sobre-segmentação.

A etapa de pós-segmentação consiste na seleção dos pontos finais de segmentação, realizada através da análise da distância entre os pontos detectados, segundo a seguinte ordem de prioridade:

1. Pontos de segmentação natural;
2. Pontos de segmentação provenientes da análise dos mínimos locais;
3. Pontos de segmentação sobre os traços de ligação que possuem linha de inclinação de zero grau;
4. Pontos de segmentação sobre os traços de ligação que possuem linha de inclinação de 10, -10, 20 e -20 graus em relação à vertical, respectivamente.

(a) (b)

Figura 4.24: (a) união dos pontos de segmentação produzidos pela combinação dos resultados dos algoritmos de segmentação e (b) resultado final do processo de segmentação após a etapa de pós-segmentação.

A Figura 4.24 ilustra o resultado da união dos pontos de segmentação gerados por algoritmos de segmentação, e o resultado final após o processo de pós-segmentação. Como resultado, o algoritmo de pós-segmentação pode produzir uma correta segmentação do caractere ou a divisão do caractere em até cinco partes.

4.2.3 Resultados experimentais

Nesta seção são avaliados os quatro métodos de segmentação de caracteres em palavras manuscritas descritos e implementados, dos quais os métodos 2 e 3 são contribuições

originais. Os testes foram realizados utilizando amostras de palavras manuscritas.

Os pontos de segmentação encontrados pelos algoritmos são classificados como corretos quando estão posicionados nos traços de ligação que unem dois caracteres adjacentes ou nos espaços em branco entre os mesmos, quando for o caso. É considerado um erro do sistema quando, entre letras sucessivas, não é detectado nenhum ponto de segmentação. Vale salientar que a sobre-segmentação é uma característica aceitável nos algoritmos de segmentação implementados. Portanto pontos de segmentação posicionados sobre o traçado do caractere não são considerados como errados, embora estes pontos dificultem o processo de classificação.

Na Figura 4.25 são ilustrados alguns exemplos de resultados obtidos com o algoritmo de segmentação de Chen modificado [36]. Estes resultados confirmam a eficiência dos operadores morfológicos na determinação dos traços de ligação existentes na escrita cursiva. Os testes realizados utilizando diferentes elementos estruturantes (aumentando ou diminuindo o tamanho dos elementos estruturantes sem modificar a sua forma) demonstraram uma forte correlação entre as dimensões dos elementos estruturantes ótimos e a largura e a altura dos caracteres que compõem as palavras.

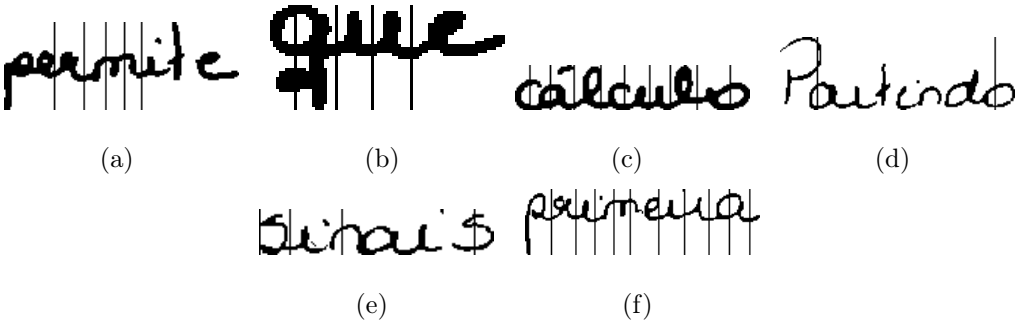


Figura 4.25: Exemplos de segmentação (método 1).

Os testes realizados com o algoritmo morfológico de segmentação proposto (método 2: segmentação de palavras manuscritas utilizando algoritmo genético) foram realizados em duas etapas. Na primeira foi feita a otimização do algoritmo genético e na segunda etapa a segmentação propriamente dita. O experimento inicial com algoritmo genético foi realizado utilizando o erro médio absoluto entre a imagem de saída e a imagem de referência como função objetivo, não tendo produzido resultados satisfatórios. A

pequena diferença entre a imagem de referência e a imagem de entrada (uma diferença de poucos *pixels*) faz com que o AG encontre elementos estruturantes que reproduzam a imagem de entrada na saída. Para resolver tal problema, a função objetivo dada pela Equação 4.28 foi desenvolvida. Nesta função, quando existe alguma diferença entre a imagem de saída e a imagem de referência é atribuída uma alta penalidade. Quando estas duas imagens são exatamente iguais uma alta tolerância é atribuída. Desta forma, os *pixels* que pertencem aos traços de ligação entre as letras possuem os maiores pesos (positivos ou negativos). A inclusão da etapa de segmentação natural com linhas de separação em diferentes ângulos permitiu que fossem detectados pontos de segmentação entre caracteres não acoplados, mesmo que estes estejam inclinados em relação à vertical. Após o pré-processamento das imagens o procedimento de "*Fill-hole*" é utilizado para preencher as cavidades centrais da palavra. Este procedimento evita que sejam localizados pontos de segmentação nos traços que formam os caracteres que possuem cavidades centrais (ex.: "a", "b" e "o") A Figura 4.26 ilustra alguns resultados obtidos utilizando o método 2.

Figura 4.26: Segmentação de palavras manuscritas (método 2).

Na Figura 4.27 são mostrados alguns exemplos dos resultados obtidos utilizando o algoritmo de segmentação por histogramas de projeção (método 3). As linhas de separação indicam os pontos de segmentação. Nestes exemplos, as letras "m", "n" e "u" são segmentadas em mais de um ponto. Esta é uma característica do sistema, que possui como objetivo prevenir problemas devido à ambigüidade da escrita cursiva. Outra característica desta técnica é que nunca são encontrados pontos de segmentação no meio de letras que possuem cavidades centrais. A utilização de linhas de separação inclinadas permite que sejam encontrados pontos de segmentação entre letras que não possuem traços de ligação bem definidos, como nos exemplos c) e f) na Figura 4.27.

Os três algoritmos de segmentação (métodos 1, 2 e 3) implementados neste projeto

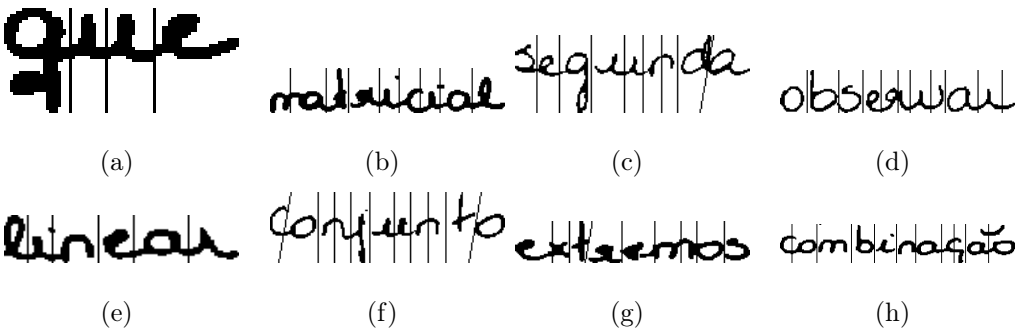


Figura 4.27: Exemplos de segmentação (método 3).

Figura 4.28: Falhas na determinação dos pontos de segmentação entre caracteres fortemente conectados.

possuem desempenho semelhante, detectando corretamente os pontos de segmentação sobre as ligações, mesmo quando é difícil a visualização exata deste ponto. Entretanto, estes algoritmos geralmente falham na detecção dos pontos de segmentação quando os caracteres estão fortemente conectados, conforme é ilustrado na Figura 4.28. Para resolver tal problema utiliza-se associado a estes métodos o algoritmo de segmentação baseado em mínimos locais, como é ilustrado na Figura 4.29.

Figura 4.29: Resultado final do processo de segmentação utilizando o método 3 e o método dos mínimos locais.

Portanto, no sistema de reconhecimento desenvolvido neste trabalho a etapa de segmentação é realizada pela união dos resultados produzidos pelo algoritmo de segmentação baseado em histogramas de projeção inclinado, método 3, juntamente com o algoritmo de segmentação que utiliza mínimos locais. O método 3 foi selecionado por possui uma menor complexidade, não requerendo a realização de operações de abertura

e fechamento. Apenas o cálculo de histogramas de projeção se faz necessário. Após a união dos pontos de segmentação produzidos pelos dois algoritmos, é executada a etapa de pós-segmentação, que tem por finalidade unir ou excluir pontos de segmentação que se encontram muito próximos uns dos outros.

Com o objetivo de testar a eficiência do resultado da combinação dos algoritmos de segmentação foi realizada uma análise manual em 2350 imagens segmentadas. É considerado como um erro de segmentação quando os algoritmos não detectam pontos de segmentação entre letras sucessivas e quando dividem o caractere em mais de cinco sub-imagens. Em 175 imagens (aproximadamente 7%) foi constatado erro na segmentação. Destas imagens, têm-se apenas seis imagens que foram consideradas mal segmentadas devido ao número excessivo (maior ou igual a seis) de pontos de segmentação num único caractere. O restante dos erros foi devido à perda do ponto de segmentação. Exemplos de imagens de palavras mal segmentadas são ilustrados na Figura 4.30. Também, foi a avaliado o número médio de pontos de segmentação gerados pela combinação dos algoritmos de segmentação (método 3 e 4) antes e após o processo de pós-segmentação, obtendo uma taxa média de 28,73 e 6,10 pontos de segmentação por palavra, respectivamente.

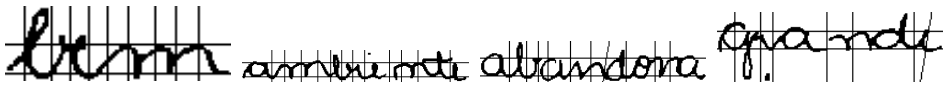


Figura 4.30: Exemplos de palavras mal segmentadas.

4.3 Extração de características

A etapa de extração de características é vital em qualquer sistema de reconhecimento de padrões. Quanto melhor for a representação dos dados em análise, melhor será seu mapeamento pelo classificador.

No sistema descrito neste trabalho, a etapa de extração de características tem como

objetivo fornecer os vetores de características para cada sub-imagem [3] da palavra, tanto na fase de treinamento como na fase de teste. Na fase de treinamento estes vetores são utilizados para obtenção dos dicionários do quantizador vetorial e das seqüências de observações que geram os Modelos Escondidos de Markov (HMMs - *Hidden Markov Models*). Na fase de reconhecimento, os vetores de características são quantizados pelo quantizador vetorial para gerar os vetores de símbolos ou vetores de observação utilizados pelos HMMs para o reconhecimento.

Três conjuntos de características foram extraídos para cada sub-imagem gerada pela etapa de segmentação. O primeiro conjunto denominado características perceptivas contém informações sobre a localização e tamanho dos (*loops*, traços ascendentes e traços descendentes, características essas que são facilmente percebidas pelo olho humano). Este tipo de características é bastante utilizado em sistemas de reconhecimento de palavras [16, 47] e sua utilização é motivada pelo processo de leitura humano, que as utiliza para identificar palavras manuscritas.

Entretanto, características perceptivas não fornecem nenhuma informação quando as imagens analisadas contêm caracteres que não possuem traços ascendentes, descendentes ou loops, como os caracteres "m" e "n", dentre outros. Segundo Grandidier [115] mais de 50% do seu conjunto de treino é composto por palavras que apresentam ausência deste tipo de características. Uma das formas de tratar este problema é utilizar outros tipos de características, construir um classificador para cada espaço de características utilizado e combinar os resultados individuais de cada classificador [16, 91]. Uma segunda possibilidade é obter um novo conjunto, através do produto cartesiano dos conjuntos das características iniciais, o que tem o inconveniente de aumentar o número de parâmetros do sistema. A terceira possibilidade é manter apenas as características mais discriminantes e substituir aquelas com menor poder discriminatório por outras obtidas a partir de um novo espaço de características. Nesta Tese foi adotada a segunda possibilidade, utilizando os vetores de características perceptivas, globais e direcionais. O vetor de características globais contém informações globais sobre o

[3] sub-imagem: são regiões da imagem que contém um único caractere ou pedaços de caracteres, obtidas na segmentação.

traçado da imagem, tais como a sua localização, quanto esse traçado é retangular ou circular. As características direcionais analisam regiões do fundo da imagem.

A seguir, são apresentadas as descrições dos vetores de características utilizados.

4.3.1 Vetor de características perceptivas

O vetor de características perceptivas possui 10 elementos que expressam:

- Altura e posição do traço ascendente;
- Altura e posição do traço descendente;
- Altura e posição do laço localizado na zona ascendente da imagem;
- Altura e posição do laço localizado na zona descendente da imagem;
- Altura e posição do laço localizado na região central da imagem.

O primeiro passo na determinação das características perceptivas é determinar as linhas de base da palavra (linha média (LM), superior (LS) e inferior (LI)) [47], conforme mostrado na Figura 4.31, com a finalidade de determinar as regiões ascendentes, descendentes e central da palavra.

Inicialmente é determinado o histograma de projeção horizontal das transições de pixels branco-preto ao longo de uma linha da palavra. Em seguida, um procedimento de suavização é aplicado para eliminar as descontinuidades do histograma, de acordo com a Equação 4.29:

$$P'_i = \frac{P_{i-2} + P_{i-1} + P_i + P_{i+1} + P_{i+2}}{5} \tag{4.29}$$

em que P_i é o valor do histograma na posição i e P'_i é o novo valor do histograma, obtido após a suavização, na posição i.

A partir do histograma suavizado, são calculadas as linhas média (LM), superior (LS) e inferior (LI). A linha média é aquela a para qual o histograma suavizado de transições branco-preto da palavra é máximo. As linhas superior (LS) e inferior (LI)

são aquelas acima e abaixo da linha média (LM), respectivamente, por valor igual a 70% do valor máximo do histograma. Este percentual foi obtido heuristicamente por Freitas [47], baseado no estudo da diferença entre os picos do histograma de transição branco-preto e os picos do histograma de densidade de *pixels*, para um conjunto de imagens da base de treinamento. Um exemplo deste processo é apresentado na Figura 4.31.

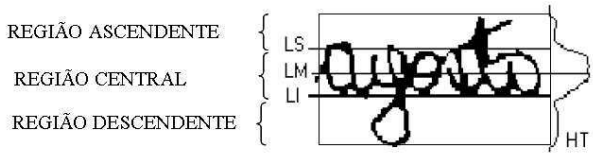


Figura 4.31: Exemplo do processo de detecção das zonas da palavra.

A seguir são determinadas as regiões ou zonas da palavra, definidas por Freitas [47] como:

- Zona ascendente: compreendida entre o limite superior da palavra e a linha superior (LS);
- Zona corpo da palavra: compreendida entre a linha superior (LS) e inferior (LI) da palavra;
- Zona descendente: compreendida entre a linha inferior (LI) da palavra e o limite inferior da imagem da palavra.

Uma vez determinada as zonas da palavra o próximo passo é determinar os traços e os laços localizados nas três regiões, utilizando para tal duas adaptações do algoritmo de rotulação de ilhas. Para a localização dos laços, este algoritmo faz uma busca dos *pixels* brancos da imagem analisando os oito vizinhos mais próximos. Se os *pixels* da vizinhança não tiverem rótulos, é atribuído um novo rótulo (A, B, ...) ao *pixel* em análise, caso contrário ele recebe o mesmo rótulo de um dos seus vizinhos. A seguir

é realizada uma análise da borda da imagem, com o objetivo determinar os rótulos da mesma. As regiões que possuírem os mesmos rótulos da borda da imagem são consideradas como cavidades abertas. Caso contrário, a região é dita como pertencente a um laço, ou seja, uma cavidade completamente cercada pelo traçado da imagem.

Para determinar os traços em cada uma das regiões analisadas, utiliza-se uma segunda adaptação do algoritmo de rotulações de ilhas, agora com análise dos pixels pretos da imagem. Este algoritmo fornece a quantidade de traços em cada uma das regiões e uma matriz de rótulos, em que todos os elementos com rótulos iguais pertencem ao mesmo traço.

Uma vez determinados as zonas, os traços e os laços da palavra, são extraídas as dez características perceptivas. Características semelhantes foram utilizadas por outros trabalhos do grupo [16, 83, 104–106]. As características utilizadas neste trabalho são calculadas conforme descrito a seguir:

- Altura e posição do traço: Inicialmente são calculadas todas as alturas e posições dos traços existentes em cada região (ascendente, central e descendente). Apenas a informação relativa ao maior traço encontrado na região analisada é utilizada no vetor de características. A altura é calculada pela diferença entre as coordenadas do pixel mais inferior e do pixel mais superior do traço, normalizada pela altura da região analisada. A posição é dada pela coordenada Y_{cm} do centro de massa do traço, normalizada pela largura da sub-imagem analisada. As coordenadas do X_{cm} e Y_{cm} centro de massa são definidas de acordo com a Equação 4.30:

$$(X_{cm}, Y_{cm}) = (\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} i.f_{ij}}{M}, \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} j.f_{ij}}{N}), \tag{4.30}$$

  em que M e N são as dimensões da imagem e f_{ij} assume valor 0 quando o *pixel* na posição (i, j) é branco e valor 1, quando o pixel é preto.

- Altura e posição dos laços: A determinação das alturas e das posições dos laços utiliza a mesma lógica usada para o calculo das alturas e das posições dos traços, descrita acima, a diferença é que neste caso a região analisada é delimitada pelo laço.

4.3.2 Vetor de características globais

O vetor de características globais contém seis características que descrevem as características globais da sub-imagem.

- Localização do contorno do traçado das letras presente no segmento analisado;
- Parâmetro de circularidade (C)
- Parâmetro de retangularidade (R)
- Momentos centrais normalizados (η_{02} e η_{20}).

A localização do contorno do traçado da sub-imagem analisada corresponde às coordenadas do centro de massa do traçado do caractere, que são definidas de acordo com a Equação 4.30.

Os parâmetros C e R, calculados a partir da área e do perímetro do contorno das letras, na sub-imagem analisada, descrevem o quão próximo o objeto (traçado da letra) é de um circulo e de um retângulo, respectivamente. O parâmetro C é calculado através da Equação 4.31, em que P e a são o perímetro e a área do traçado da letra (*pixels* pretos), respectivamente. Quanto maior for a semelhança entre o objeto e um circulo mais próximo será o valor de C da unidade.

$$C = \frac{P^2}{4.\pi.a} \tag{4.31}$$

O parâmetro R corresponde à razão entre o perímetro do traçado do objeto e o perímetro do menor retângulo envolvente.

As duas informações restantes que compõem este vetor de características são os momentos centrais normalizados η_{02} e η_{20}, que são obtidos através das Equações 4.32 e 4.33,

$$\mu_{pq} = \sum_{x=1}^{M} \sum_{y=1}^{N} (x - x_c)^p (y - y_c)^q I(x, y), \tag{4.32}$$

Tabela 4.1: Conversão da rotulação dos pixels pelo algoritmo de extração das características direcionais.

Rótulo	Tipo
0	Fechado
1	Aberto abaixo
2	Aberto acima
3	Aberto a direta
4	Aberto a esquerda
5	Aberto a direita e acima
6	Aberto a esquerda e acima
7	Aberto a direita e abaixo
8	Aberto a esquerda e abaixo
9	Sem rótulo

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{1+\frac{i+j}{2}}}, \tag{4.33}$$

em que M e N são as dimensões da imagem, x_c e y_c são as coordenadas do centro de massa do objeto, μ_{pq} é o momento central de ordem (p, q) da função de intensidade $I(x, y)$ de uma imagem. Momentos η são invariantes a translação e a escala.

4.3.3 Vetor de características direcionais

O vetor de características direcionais contém informações sobre a região do fundo da imagem. As características, foram inspiradas num método de rotulação proposto por Parker [66]. Para os *pixels* do fundo da imagem é verificado, em cada uma das oito possíveis direções, a presença ou não de um *pixel* do contorno, como é ilustrado na Figura 4.32. Em função deste teste, os pixels são rotulados de acordo com a Tabela 4.1. Os componentes do vetor de características correspondem ao número de pixels atribuídos a cada rótulo, normalizados pela área de cada sub-região.

O resultado final da etapa de extração de características são três vetores para cada sub-imagem analisada, em que cada componente é um número real entre 0 e 1. Entretanto, o classificador baseado em HMM necessita de uma representação discreta da informação, isto é, o classificador trabalha com vetores de observações, cujos componentes pertençam a um alfabeto de símbolos pré-determinados. Dessa forma, é necessária a utilização de um quantizador vetorial para cada vetor de característica. O objetivo da quantização vetorial é representar os vetores de características por vetores de observação ou vetores de símbolos.

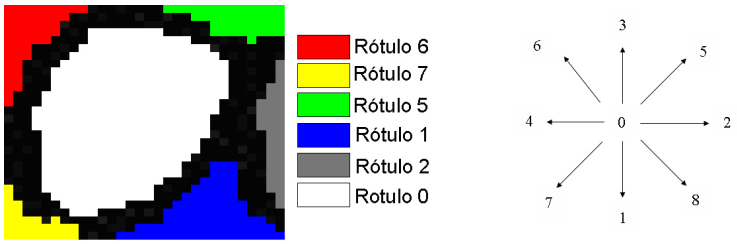


Figura 4.32: Exemplo do processo de rotulação utilizado pelo extrator de características direcionais.

4.4 Quantização vetorial

No presente trabalho, a quantização vetorial antecede os HMMs para o reconhecimento de palavras manuscritas. Na fase de treinamento, a quantização vetorial é utilizada para a construção dos dicionários associados a cada vetor de características do usuário e na fase de reconhecimento, para gerar os símbolos utilizados pelos HMMs. Estes símbolos correspondem aos índices (dos vetores-código) gerados na representação (QV) da seqüência de características com o uso dos dicionários.

A seguir são apresentados, de forma resumida os fundamentos da quantização vetorial, bem sua utilização no sistema de reconhecimento de palavras manuscritas aqui descrito.

4.4.1 Visão geral

A quantização vetorial [112, 113, 116], pode ser entendida como uma extensão da quantização escalar em um espaço multidimensional. Um quantizador vetorial K-dimensional de M-níveis, é um mapeamento Q que assume para cada vetor de entrada, $\overline{x} = \{x_0, ...x_k - 1\}$ um vetor de reprodução $Q(x) = w_i$, extraído de um alfabeto de reprodução finito W. O quantizador Q é completamente descrito pelo alfabeto de reprodução (ou dicionário) W, e pela partição S do espaço vetorial de entrada nos conjuntos S do mapeamento dos vetores de entrada no i-ésimo vetor de reprodução.

O conjunto W é referido como o dicionário de reconstrução, com tamanho M e W_i são os vetores códigos de dimensão K. O tamanho do dicionário é também conhecido como o número de níveis, em analogia com a quantização escalar. Assim, diz-se um quantizador de M níveis ou um dicionário de M níveis.

O objetivo da quantização vetorial é a obtenção de um dicionário ótimo, que minimize a distorção média introduzida pela aproximação de cada vetor de entrada por um dos vetores-código [117]. Neste trabalho, a quantização vetorial é usada para gerar os símbolos a serem utilizados no processo de reconhecimento por meio de HMMs. Obtendo, para cada sub-imagem, um símbolo, que corresponde ao índice (do vetor código) gerado na representação (QV) da seqüência de características com o uso do dicionário específico para a característica analisada.

Existem diversas técnicas para o projeto de dicionários para a quantização vetorial, dentre as quais o algoritmo LBG (*Linde-Buzo-Gray*) destaca-se pela sua ampla utilização [112]. Este algoritmo será descrito a seguir.

4.4.2 Algoritmo LBG

Seja a iteração do algoritmo LBG denotada por n. Dados K, N e um limiar de distorção $\epsilon \geq 0$, o algoritmo LBG [112], consiste da seguinte seqüência de passos:

1) Inicialização: dado um dicionário inicial W_0 e um conjunto de treinamento $X = \{x_m;\ m = 1,\ 2,\ \ldots,\ M\}$, faça $n = 0$ e $D_{-1} = \infty$;

2) Particionamento: dado W_n (dicionário na n-ésima iteração) aloque cada vetor de treino (vetor de entrada) na respectiva classe (célula de Voronoi) segundo o critério do vetor-código mais próximo; calcule a distorção[4]

$$D_n = \sum_{i=1}^{N} \sum_{x_m \in S_i} d(x_m, w_i); \tag{4.34}$$

3) Teste de convergência (critério de parada): se $(D_{n-1} - D_n)/D_n \leq \epsilon$ pare, com W_n representando o dicionário final (dicionário projetado); caso contrário, continue;

4) Atualização do dicionário: compute os novos vetores-código como os centróides das classes de vetores; faça $W_{n+1} = W_n$; faça $n = n + 1$ e vá para *Passo 2*.

Em essência, no algoritmo LBG a função distorção decresce monotonicamente, uma vez que o dicionário é iterativamente atualizado, visando satisfazer as condições de centróide e de vizinho mais próximo. Na dinâmica do algoritmo LBG, a distorção introduzida ao se representar os vetores do conjunto de treinamento pelos correspondentes vetores-código (centróides) é monitorada a cada iteração. A regra de parada (teste de convergência) do algoritmo baseia-se nessa distorção monitorada: o treinamento do dicionário é encerrado quando $(D_{n-1} - D_n)/D_n \leq \epsilon$.

Neste trabalho, como foi anteriormente mencionado, a quantização vetorial foi utilizada para a geração dos dicionários e dos vetores de símbolos associado a cada imagem de palavra. A seguir, é realizada uma discussão sobre a implementação do quantizador vetorial neste trabalho.

4.4.3 Projeto do quantizador vetorial e do seu dicionário

O comportamento de um quantizador vetorial sofre influência de vários fatores, como o número de centróides, ou seja, o número de níveis do quantizador; o algoritmo de projeto de dicionário; o dicionário inicial; a escolha da medida de distância e a dimensão do dicionário.

[4]Em algumas abordagens, é calculada a distorção média $\widetilde{D}_n = D_n/M$. O cálculo da média, no entanto, requer uma operação de divisão (por M), que deve ser evitada, tendo em vista que qualquer das abordagens (D_n ou $\widetilde{D}_n$) tem o mesmo efeito em termos do teste de convergência (*passo 3* do algoritmo).

Neste trabalho, foi projetado um quantizador vetorial para cada vetor de característica utilizado. A dimensão do quantizador vetorial é igual ao número de elementos do vetor de características a ele associado. Portanto, as dimensões dos quantizadores vetoriais associados aos vetores de características perceptiva, global e direcional são 10, 6 e 10, respectivamente. A medida de distância utilizada foi o erro quadrático, por sua simplicidade e longa aceitação.

Para o projeto dos dicionários dos quantizadores vetoriais foi utilizado o algoritmo LBG, por ser o mais amplamente utilizado na literatura especializada. Foram realizados testes a partir de um conjunto de amostras iniciais da seqüência de treinamento e o dicionário inicial foi construído de forma aleatória.

Uma das questões principais no projeto de quantizadores vetoriais é o compromisso entre taxa e distorção [5]. O alvo a ser perseguido é a obtenção de um dicionário ótimo, que minimize a distorção média introduzida pela aproximação de cada vetor de entrada por um dos vetores-código. Uma forma de minimizar a distorção média é aumentar o número de níveis do quantizador, o que, entretanto, aumenta a complexidade computacional. A complexidade computacional de um quantizador vetorial de dimensão K e taxa R é da ordem de $N = 2^{KR}$ de operações por amostra para cada vetor de entrada, se uma busca completa é realizada ao longo do dicionário [117]. Foram realizados vários testes para a escolha dos números de níveis dos quantizadores vetoriais, de forma a obter um resultado que melhor se adaptasse às características do sistema implementado neste trabalho.

4.5 Modelos Escondidos de Markov

Os Modelos Escondidos de Markov (HMM - *Hidden Markov Models*) [53,54] são utilizados em problemas de reconhecimento de padrões. Nestes casos, HMMs são construídos para cada classe de padrão. Para cada seqüência de vetores de características deve-se

[5] Distorção do quantizador é o erro médio de quantização ao se representar o sinal de entrada por sua versão quantizada e a taxa de codificação do quantizador vetorial, que mede o número de bits por componente do vetor, é dada por $R = \frac{1}{K}\log_2 N$.

calcular a probabilidade de que esta seqüência seja produzida por cada um dos modelos. A classe à qual pertence a seqüência analisada é aquela que obtém a maior probabilidade.

No presente trabalho, os Modelos Escondidos de Markov podem ser utilizados para modelar os caracteres. A fase de treinamento tem como objetivo estimar os melhores parâmetros dos modelos de caracteres, e fase de reconhecimento encontrar qual seqüência de caracteres melhor representa a palavra desconhecida.

A seguir são apresentados, de forma resumida os fundamentos dos Modelos Escondidos de Markov, bem como a sua utilização no sistema de reconhecimento de palavras manuscritas aqui descrito.

4.5.1 Fundamentos dos Modelos Escondidos de Markov

Um Modelo escondido de Markov é um processo estocástico gerado por dois mecanismos inter-relacionados. Um canal de Markov básico tem um número finito de estados e um conjunto de funções aleatórias, cada função aleatória associada a cada um dos estados. Para cada instante de tempo discreto, é assumido que o processo está em algum estado e uma seqüência de observação é gerada por uma função aleatória correspondendo ao estado corrente [1, 53].

O canal de Markov básico escolhe o estado de acordo com uma matriz de probabilidade de transição. O observador vê somente a saída (observações) da função de densidade de probabilidade (fdp) associada a cada estado e não pode observar os estados do canal de Markov básico; dai o termo modelo escondido de Markov [1, 53].

Os HMMs podem ser discretos ou contínuos dependendo da natureza das observações. Caso as observações sejam naturalmente discretas ou quantizadas, com os seus símbolos pertencendo a um alfabeto finito, tem-se um HMM discreto. Entretanto, se as observações são contínuas tem-se um HMM contínuo.

Existem três tipos básicos de estruturas para os HMM: modelos ergódigos, seqüencias e paralelos [53, 54, 118]. Nos modelos ergódigos todas as transições entre os estados são possíveis, diferentemente das estruturas seqüenciais e paralelas em que há restrições

entre as transições.

Os HMMs podem ser classificados, também, de acordo com o local onde são localizadas as observações, em modelos de estados e modelos de arcos. Nos modelos de arcos as observações são produzidas nas transições. Já nos modelos de estados as transições são produzidas nos estados. Neste trabalho, os modelos dos caracteres são modelos de Markov discretos com as observações produzidas nos arcos.

Para o reconhecimento de manuscritos utilizando HMM, a primeira questão envolvida na determinação dos HMMs ótimos (HMM ótimo é o conjunto de parâmetros de um HMM que melhor representa uma determinada classe de padrão) é a determinação da estrutura do modelo. Na literatura, diversas arquiteturas de modelos de Markov são consideradas para o reconhecimento de manuscritos. Segundo Koerich [119], a solução para determinar a arquitetura do modelo é fazer suposições iniciais a respeito da estrutura e então utilizar a estimação de parâmetros para melhorar a probabilidade dos modelos gerarem os dados de treinamento.

4.5.2 Parâmetros do modelo

Os parâmetros que caracterizam o HMM de arco, $\lambda = (A, B, \pi)$ são:

1. N, número de estados do modelo. Estados individuais são denotados como $(q_1, q_2, ..., q_N)$.

2. T, comprimento da seqüência de observação O, $O = O_1, O_2, ..., O_T$.

3. M, número de símbolos possíveis de observação.

4. $A = [a_{ij}]$, $1 \leq i, j \leq N$, a matriz transição de estados. Cada a_{ij} corresponde à probabilidade de ocorrer uma transição do estado q_i, no instante t, para o estado q_j, no instante $t+1$ e ao mesmo tempo produzindo uma observação O_t no tempo t.

5. $B = [b_{ij}(k)]$ $1 \leq i, j \leq N$ e $1 \leq k \leq M$, é uma matriz de função de probabilidade das observações. Indica a probabilidade de observar o símbolo k quando ocorre

a transição do estado q_i, no instante t, para o estado q_j, no instante $t+1$.

6. $\pi = \pi_i = Pq_t \mid t = 1,\ 1 \leq i$, vetor de probabilidade do estado inicial. Este vetor indica a probabilidade de iniciar o processo no estado q_i para $t = 1$.

As matrizes A e B obedecem as restrições das Equações 4.35 e 4.36:

$$\sum_{j=1}^{N} a_{ij} = 1, 1 \leq i \leq N. \tag{4.35}$$

$$\sum_{K=1}^{M} b_{ij}(k) = 1, 1 \leq i, j \leq N. \tag{4.36}$$

Os aspectos teóricos dos Modelos Escondidos de Markov podem ser caracterizados em termos da solução de três problemas fundamentais, descritos a seguir:

1. Gerar um HMM dada uma seqüência de observações (treinamento). Ou seja, ajustar os parâmetros do modelo de modo a representar com maior eficiência os dados que estão sendo modelados;

2. Encontrar a probabilidade de uma seqüência de observações dado um HMM (reconhecimento). Ou seja, realizar o cálculo da probabilidade (ou verossimilhança) de uma seqüência de observações dado um HMM específico;

3. Encontrar a seqüência de estados escondidos que mais provavelmente gerou uma seqüência observada (decodificação). Ou seja, determinar a seqüência de estados "ótima".

Para o reconhecimento de palavras manuscritas é necessário solucionar os problemas 1 e 2, que constituem as fases de treinamento e reconhecimento, respectivamente.

O objetivo da fase de treinamento é estimar os melhores valores dos parâmetros dos modelos. Existe um procedimento que ajusta interativamente e automaticamente os parâmetros de um HMMs dado um conjunto de treinamento contendo seqüências de observações. Este algoritmo é o de *Baum-Welch* [53,54], que garante que o modelo converge para o máximo local da probabilidade de observação do conjunto de treinamento de acordo com o critério da estimação máxima da probabilidade [119].

O algoritmo de *Baum-Welch* pode ser descrito por meio dos seguintes passos [1]:

1. Atribuição inicial dos valores para os parâmetros do modelo $\lambda = (A, B, \pi)$ e para a probabilidade P_l.

2. Re-estimação dos parâmetros do modelo através do algoritmo de re-estimação de *Baum-Welch* (as equações deram descritas a seguir, obtendo-se $\bar{\lambda}$);

3. Calculo da probabilidade $\bar{P}_l$ associada ao modelo $\bar{\lambda}$ re-estimado e comparação com a probabilidade anterior calculada P_l;

4. Se $\bar{P}_l - P_l \leq \delta$ (limiar), o processo de re-estimação é finalizado. Caso contrário retorna-se ao passo 2.

Neste trabalho, os parâmetros armazenados do modelo são os que correspondem a iteração que maximiza a probabilidade do conjunto de validação.

Dada uma seqüência de observações $O = (O_1, O_2, ..., O_T)$, é preciso inicialmente calcular as probabilidades *forward* $\alpha_t(i)$ e *backward* $\beta_t(i)$.

Para calcular as probabilidades de *forward* e *backward*, com as observações nos arcos, foram realizadas modificações no procedimento *forward-backward* [35].

As probabilidades $\alpha_t(i)$ e $\beta_t(i)$ são definidas como:

$$\alpha_t(i) = P(O_1, O_2, ..., O_{t-1}, q_t = S_i|\lambda) \tag{4.37}$$

$$\beta_t(i) = P(O_t, O_{t+1}, ..., O_{T-1}, q_t = S_i|\lambda) \tag{4.38}$$

em que q_t é o estado do processo no tempo t.

$\alpha_t(i)$ pode ser calculado como:

1. Inicialização

$$\alpha_1(1) = 1.0 \tag{4.39}$$

$$\alpha_1(j) = 0.0, \;\; j = 2, 3, ..., N \tag{4.40}$$

2. Indução

$$\alpha_t(j) = \sum_{i=1}^{N} a_{ij} b_{ij}(O_{t-1}) \alpha_{t-1}(i), \;\; j = 1, 2, ..., N, t = 2, 3, ..., T \tag{4.41}$$

3. Finalização

$$P(O|\lambda) = \alpha_{T+1}(N) \tag{4.42}$$

Dado que o estado s_N é o estado final.

O procedimento para o cálculo da probabilidade $\beta_t(i)$ considerando a emissão das observações nas transições é apresentado a seguir:

1. Inicialização

$$\beta_{T+1}(N) = 1.0, \tag{4.43}$$

$$\beta_{T+1}(j) = 0.0, \;\; j = 1, 2, 3, ..., N \tag{4.44}$$

Visto que estado s_N é o estado final.

2. Indução

$$\beta_t(j) = \sum_{j=1}^{N} a_{ij} b_{ij}(O_t) \beta_{t+1}(j), \;\; j = 1, 2, ..., N, t = 2, 3, ..., T \tag{4.45}$$

3. Finalização

$$P(O|\lambda) = \beta_1(1) \tag{4.46}$$

Dado que o estado s_1 é o único estado inicial.

Para re-estimar os parâmetros do HMM, define-se $\xi_t(i, j)$ como a probabilidade de estar no estado S_i no tempo t e no estado S_j no tempo $t+1$ produzindo uma observação O_t, dados o modelo e a seqüência de observação O, como:

$$\xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda) \tag{4.47}$$

Utilizando as variáveis *forward* e *backward* pode-se escrever $\xi_t(i,j)$ na forma apresentada na Equação 4.48

$$\xi_t(i,j) = \frac{\alpha_t(i) b_{ij}(O_t) \beta_{t+1}(j)}{P(O|\lambda)} \tag{4.48}$$

É necessário também calcular $\gamma_t(i)$ como a probabilidade de estar no estado S_i no tempo t dada a seqüência de observação O e o modelo λ, como:

$$\gamma_t(i) = P(q_t = s_i | O, \lambda) \tag{4.49}$$

Utilizando as variáveis *forward* e *backward* pode-se escrever $\gamma_t(i)$ como:

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{P(O|\lambda)} \tag{4.50}$$

A re-estimação dos parâmetros do HMM é:

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T} \alpha_t(i) a_{ij} b_{ij}(O_t) \beta_{t+1}(j)}{\sum_{t=1}^{T+1} \alpha_t(i) \beta_{t+1}(i)} \tag{4.51}$$

$$\overline{b_{ij}(k)} = \frac{\sum_{t=1}^{T} \delta(O_t, v_k) \alpha_t(i) a_{ij} b_{ij}(O_t) \beta_{t+1}(j)}{\sum_{t=1}^{T} \xi_t(i,j)} \tag{4.52}$$

em que

$$\delta(x,y) = left\{ \begin{cases} 1, & \text{se } x = y \\ 0, & \text{c.c.} \end{cases} \tag{4.53}$$

Na fase de reconhecimento é realizada a estimação da probabilidade de ocorrência de uma dada seqüência de observação associada ao modelo λ, utilizando a equação 4.42 (algoritmo *Forward*). O modelo que apresentar a maior probabilidade é a classe identificada pelo sistema.

4.5.3 Projeto dos Modelos Escondidos de Markov

Como o processo de segmentação pode produzir tanto a segmentação correta do caractere como a divisão do caractere em até cinco sub-imagens foram considerados modelos esquerda-direita (*Left-Right*) com 6, 11 ou 16 estados, dependendo do número de símbolos considerados por cada sub-imagem (Figuras 4.33, 4.34 e 4.35).

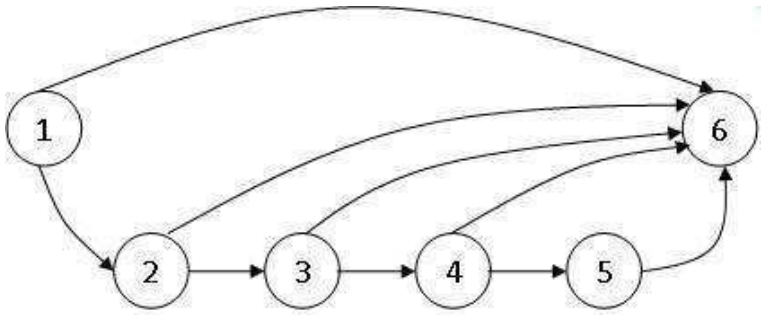


Figura 4.33: Um modelo escondido de Markov de um caractere com 6 estados.

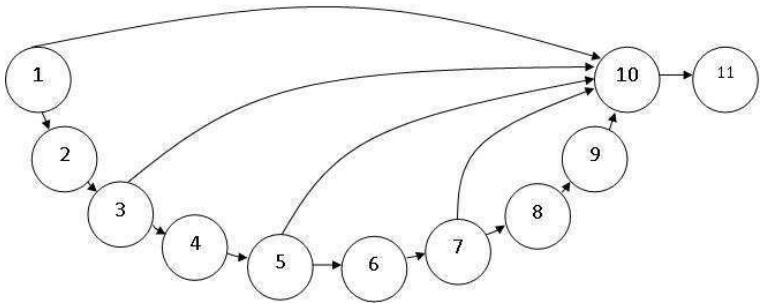


Figura 4.34: Um modelo escondido de Markov de um caractere com 11 estados.

No modelo ilustrado na Figura 4.33 a transição t_{1-6} modela o caso do caractere que foi segmentado corretamente e emite o símbolo que codifica o caractere. As transições t_{1-2} e t_{2-6} modelam o caso onde o caractere foi segmentado em duas partes, sendo representado por dois símbolos, um para cada sub-imagem. As seqüências de transições t_{1-2}, t_{2-3} e t_{3-6} modelam a segmentação do caractere em três partes, produzindo três símbolos de observações. De maneira análoga, as seqüências de transições t_{1-2}, t_{2-3}, t_{3-4} e t_{4-6} e a seqüência t_{1-2}, t_{2-3}, t_{3-4}, t_{4-5} e t_{5-6}, modelam o caso do caractere ser

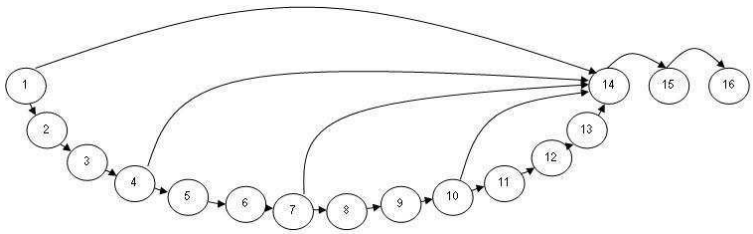


Figura 4.35: Um modelo escondido de Markov de um caractere com 16 estados.

segmentado em quatro ou cinco partes, respectivamente.

O modelo apresentado na Figura 4.34 modela o caso de cada sub-imagem do caractere ser reapresentado por dois símbolos, codificando dois distintos vetores de características. Essa arquitetura é semelhante à anterior. A seqüência de transição t_{1-10} e t_{10-11} modela a segmentação correta do caractere e a observação de dois símbolos, representado dois vetores de características distintos extraídos da mesma imagem, ou seja do caractere corretamente segmentado.

O modelo ilustrado na Figura 4.35 é utilizado quando a sub-imagem é representada por três vetores de características distintos, representados por 3 símbolos do alfabeto. Nesse caso, as seqüências de transições t_{1-14}, t_{14-15} e t_{15-16} modelam a correta segmentação da letra e os três símbolos extraídos dessa imagem.

A inicialização das matrizes A e B, segundo Rabiner [53, 54], não é um processo simples e não existe um único caminho a ser seguido. Nesse trabalho de Tese, após vários testes, a matriz A e B foram inicializadas como equiprováveis.

Para tratar os efeitos da insuficiência de dados no treinamento é adicionada uma restrição aos parâmetros do modelo com o objetivo de garantir que eles nunca sejam menores que um dado ϵ. Para tanto, os valores da matriz A e B são reescalonados de forma a atender as restrições das Equações 4.35 e 4.36. Neste trabalho foram realizados alguns teste, optando-se pelo uso de $\epsilon = 10^{-3}$.

4.6 Cometários finais

Neste capítulo foram descritos os trabalhos realizados no desenvolvimento do sistema proposto, para reconhecimento de palavras manuscritas dependente do usuário. Durante o desenvolvimento deste trabalho, foram implementados quatro algoritmos de segmentação de caracteres em palavras manuscritas. O primeiro, proposto por Chen [36], é baseado nas operações morfológicas de abertura e fechamento. Estas operações são utilizadas na determinação dos traços de ligação existentes entre os caracteres, onde estão localizados os pontos de segmentação. O segundo algoritmo também implementado utiliza as operações morfológicas de abertura e fechamento para realizar a determinação dos traços de ligação, entretanto os elementos estruturantes utilizados são obtidos utilizando algoritmo genético. Na ausência de traços de ligação, pontos de segmentação, denominados naturais, são determinados por um algoritmo que localiza os espaços em branco entre caracteres não conectados. O terceiro algoritmo proposto realiza a segmentação das palavras utilizando histogramas de projeção vertical. Por último, foi implementado o algoritmo de segmentação que analisa os pontos de mínimo locais do contorno da palavra na tentativa de localizar pontos de segmentação entre caracteres fortemente unidos ou sobrepostos Os resultados obtidos comprovam a eficiência dos métodos implementados na segmentação de caracteres em palavras manuscritas. O algoritmo de segmentação baseado em histogramas de projeção possui uma baixa complexidade e apresenta resultados semelhantes aos demais métodos implementados, tendo sido selecionado, juntamente com o algoritmo de segmentação baseado nos pontos de mínimos, para compor a etapa de segmentação do sistema proposto.

O sistema desenvolvido utiliza os vetores de características perceptivas, globais e direcionais para representar cada sub-imagem gerada pelas etapas anteriores do processo. Durante a fase de treinamento, estes vetores de características são utilizados pelo algoritmo LBG para construção dos dicionários de reconstrução do quantizador vetorial. O quantizador vetorial tem a função de transformar cada vetor de característica em símbolos, que são os índices dos vetores códigos.

Modelos Escondidos de Markov são utilizados neste sistema para modelar os ca-

racteres. Os parâmetros dos HMMs são obtidos através do processo de treinamento realizado pelo algoritmo de *Baum-Welch* e o reconhecimento é realizado pelo algoritmo de *forward*.

Capítulo 5

Resultados experimentais

Neste capítulo são descritos os resultados experimentais obtidos para avaliar o sistema de reconhecimento de palavras desenvolvido. O critério utilizado para avaliar a precisão do sistema foi a taxa de reconhecimento (TR), ou seja, a proporção entre as amostras do conjunto de teste reconhecidas corretamente e o total de amostras deste conjunto.

Durante o decorrer deste trabalho foram realizados diversos experimentos com a finalidade de avaliar o desempenho dos algoritmos quando utilizados em diferentes aplicações. Além de determinar qual a melhor estratégia de combinação dos vetores de características e qual topologia dos HMMs melhor se adequa aos caracteres. Para isto, foram utilizadas três base de dados distintas: meses do ano [16,83,104], IRONOFF [120] e base de palavras dependente do escritor.

A seguir tem-se uma descrição das bases de dados utilizadas neste trabalho e dos diversos resultados experimentais produzidas.

5.1 Base de imagens

No desenvolvimento de um sistema de reconhecimento de padrões, em especial no reconhecimento de palavras manuscritas, é vital a utilização de uma base de dados que possua uma grande diversidade de amostras, de modo a fornecer uma representatividade adequada das classes a serem reconhecidas. No reconhecimento de manuscritos,

a base de dados utilizada é normalmente dividida em três conjuntos: treinamento, validação e teste. Os dois primeiros são utilizados durante o treinamento e o terceiro é utilizado na etapa de avaliação do sistema.

Três bases de dados foram utilizadas para realizar a avaliação dos algoritmos implementados. A base de palavras dos meses do ano e a base de caracteres IRONOFF possuem amostras de palavras e caracteres manuscritos produzidos por diferentes pessoas. Entretanto, para realizar a avaliação do sistema desenvolvido neste trabalho foi necessária a criação de uma base de dados, chamada base de palavras manuscritas dependentes do escritor, contendo um número representativo de palavras escritas pela mesma pessoa.

A seguir tem-se uma descrição mais detalhada a respeito da criação e caracterização desta base de dados.

5.1.1 Base de palavras manuscritas dependente do escritor

A criação de uma base de palavras manuscritas é geralmente uma tarefa bastante lenta, que basicamente pode ser dividida em duas etapas: coleta e digitalização das amostras. A base de dados desenvolvida neste trabalho teve como finalidade fornecer amostras de palavras manuscritas para os processos de treinamento, validação e teste do sistema desenvolvido. Atualmente, esta base de dados contém palavras manuscritas coletadas de quatro escritores distintos, cujas amostras foram obtidas de exames, trabalhos escolares, rascunhos, notas de aulas e relatórios, dentre outras fontes. Não foram utilizados formulários ou qualquer outro dispositivo de coleta de dados, nem os escritores tiveram qualquer conhecimento prévio de que seus textos seriam coletados e digitalizados, desta forma garantindo a espontaneidade da escrita. Quando o escritor sabe de antemão que seus textos serão utilizados por algum sistema, eles geralmente redigem com maior cuidado, não preservando a naturalidade da escrita.

A grande dificuldade na coleta do material foi encontrar pessoas que tivessem uma quantidade suficiente de manuscritos em papel sulfite branco. Esta restrição do papel limitou bastante o número de escritores, visto que a maioria das pessoas quando redige

um manuscrito costuma fazê-lo em papel do tipo pautado, que não é adequado para os nossos propósitos, devido às linhas guias que prejudicam o reconhecimento.

Após a coleta dos manuscritos, os mesmos passaram pela etapa de digitalização, em que foi utilizado o scanner HP Scanjet 5200C [121], disponível no LAPS/DEE/UFCG ajustado para uma resolução de 200 dpi (*dots per inch*) com representação binária, no formato PCX (*Zsoft IBM PC Paintbrush*) [122].

Finalizada a etapa de digitalização, todos os arquivos PCX, cada qual contendo uma única palavra, foram convertidos para o formato próprio PAV (a terminação PAV não possui significado, sendo apenas uma abreviação para palavra). O formato PAV, desenvolvido no decorrer deste trabalho, possui um cabeçalho contendo o número de linhas e colunas da imagem, o número de letras que compõem a palavra digitalizada e o rótulo (*label*) da mesma. Não é aplicada nenhuma forma de compressão na imagem: cada pixel é representado pelo código ASCII correspondente ao seu nível de cinza. Este formato permite a identificação da palavra contida na imagem.

Parte do processo de conversão do formato PCX para PAV é realizada manualmente, pois necessita que, para cada imagem, seja fornecido o rótulo da palavra nela contida. Todas as palavras digitalizadas no formato PCX foram convertidas para o formato PAV. A seguir foi feita uma caracterização da base de dados em relação à distribuição das palavras e das letras para os quatro escritores.

Caracterização da base de imagens

A base de palavras manuscritas dependente do escritor do LAPS/DEE/UFCG contém atualmente 17.538 palavras manuscritas digitalizadas, em formato PCX, fornecidas por quatro escritores distintos (três mulheres e um homem), conforme ilustrado na Tabela 5.1. Atualmente, 16.000 destas palavras foram convertidas para o formato PAV.

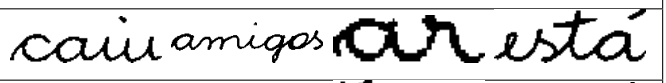
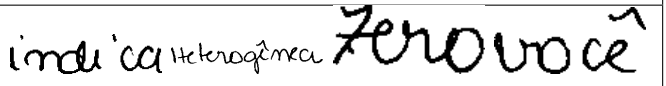
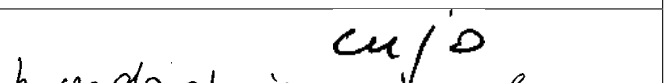
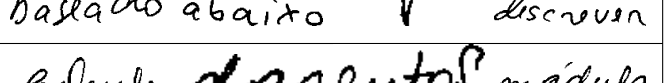
Na coleta dos textos manuscritos não foi imposta nenhuma restrição quanto ao vocabulário utilizado nem quanto ao estilo de escrita. Como conseqüência, o vocabulário de cada conjunto analisado é distinto e complexo, com muitas palavras semelhantes presentes no vocabulário (por exemplo: discreta, discretas, discreto, discretos, apre-

Tabela 5.1: Total de amostras na base de dados em formato PCX e PAV.

	Palavras digitalizadas	Palavras convertidas p/ formato PAV
Escritor 1	4241	4000
Escritor 2	4400	4000
Escritor 3	4501	4000
Escritor 4	4396	4000

sentadas, apresentado, apresentados, apresentando). Como também a base é bastante heterogênea quanto ao estilo de escrita, como ilustrado na Tabela 5.2. Na Tabela 5.3 são apresentados o sexo e o nível de escolaridade de cada escritor.

Tabela 5.2: Exemplos de Palavras Manuscritas.

Escritor	Amostras de palavras
Escritor 1	
Escritor 2	
Escritor 3	
Escritor 4	

Na Tabela 5.4 são ilustradas as maiores e menores palavras encontradas, em número de letras, nos conjuntos de treinamento/validação e teste para os quatro escritores, bem como o tamanho médio das palavras presentes na base. As vogais isoladas não foram consideradas.

Algumas palavras da base apresentam caracteres escritos em letras maiúsculas. Estas palavras não foram utilizadas para compor os conjuntos de treinamento, validação e teste visto que, neste primeiro momento o sistema é dedicado ao reconhecimento de palavras manuscritas escritas utilizando apenas caracteres minúsculos, motivo pelo qual a soma do número de amostras dos conjuntos de treinamento, validação e de teste não

Tabela 5.3: Informações sobre os escritores.

Escritor	Sexo	Nínel de escolaridade
Escritor 1	Feminino	Graduação
Escritor 2	Feminino	Graduação
Escritor 3	Masculino	Pós-graduação
Escritor 4	Feminino	Graduação

Tabela 5.4: Maiores(em número de letras), menores, tamanho médio e o desvio padrão (STD) das palavras na base.

Conjunto	Maior palavra	Menor palavra	Tamanho médio	STD
Treinamento/validação do Escritor1	experimentalmente	ao	4	2,46
Teste do Escritor1	experimentalmente	ao	6	2,63
Treinamento/validação do Escritor2	preferencialmente	ao	4	2,34
Teste do Escritor2	eletronegatividade	ao	7	2,64
Treinamento/validação do Escritor3	características	ao	5	2,71
Teste do Escritor3	Características	a	7	2,67
Treinamento/validação do Escritor4	apropriadamente	ao	4	2,91
Teste do Escritor4	consideravelmente	ao	7	2,68

é igual a 4.000 para cada escritor. O número de palavras por escritor, bem como a quantidade destas que são distintas, é fornecido na Tabela 5.5.

Tabela 5.5: Palavras por escritores nos conjuntos de treinamento/validação e teste.

	Palavas no conjunto de treinamento/validação	Palavras distintas no conjunto de treinamento/validação	Palavras distintas no conjunto de teste
Escritor1	2383	517	1350
Escritor2	2009	489	1380
Escritor3	1998	459	1057
Escritor4	2383	518	1164

O conjunto de teste foi obtido selecionando de forma aleatória uma amostra de cada palavra do vocabulário disponível. O restante das amostras foi utilizado para compor os conjuntos de treinamento e validação. Com esta distribuição, garante-se que toda palavra utilizada no treinamento/validação esteja presente no conjunto de teste. Entretanto, o contrario não é necessariamente verdade, ou seja, uma palavra contida no conjunto de teste pode não estar contida no conjunto de treinamento ou de validação, devido à ocorrência de palavras com apenas uma única amostra na base. Dessa maneira, o conjunto de testes contém uma maior diversidade de palavras, porem apenas uma única ocorrência de cada palavra. Quando uma palavra está presente tanto no conjunto de treinamento/validação quanto no conjunto de teste é porque existem na base varias ocorrências da mesma palavra. Não ocorre nenhuma interseção entre arquivos de imagens de palavras presentes nos conjuntos de treinamento, validação e teste. Os conjuntos de treinamento/validação contêm menos palavras distintas, mas em alguns casos, mais de uma ocorrência da mesma palavra. Esta divisão foi feita de forma a garantir que: 1) o conjunto de teste contivesse uma amostra de cada palavra do vocabulário e 2) o maior número possível de amostras, de palavras distintas ou não, estivesse disponível para treinamento/validação.

Como o sistema descrito neste trabalho tem como finalidade o reconhecimento das palavras através do reconhecimento dos caracteres que as compõem, é importante que conjunto de treinamento possua uma grande quantidade de amostras dos caracteres presentes no alfabeto, de maneira a permitir a construção de modelos fiéis aos caracteres existentes no vocabulário. O número de caracteres presentes nos conjuntos de treinamento/validação e teste é mostrado nas Tabelas 5.6 e 5.7 para cada escritor.

Nenhuma ocorrência das letras "k", "y" e "w" foi encontrada em nenhum conjunto, visto que estas letras não pertenciam ao alfabeto utilizado na língua portuguesa antes de reforma ortográfica no Brasil assinada pelo presidente Luís Inácio Lula da Silva em 29 de setembro de 2008.

5.1.2 Base de dados dos meses do ano

A base de imagens dos meses do ano está disponível no LAPS/DEE/UFCG. Esta base de dados é composta de 6.000 imagens binárias de palavras (nomes) dos meses do ano, com resolução de 200 dpi(*dots per inch*), obtida de 500 escritores [16] . Esta base de dados foi dividida em três conjuntos: treinamento, teste e validação com 3.600, 1.200 e 1.200 imagens, respectivamente. Na Figura 5.1 são ilustrados alguns exemplos de palavras contidas nesta base de imagens.

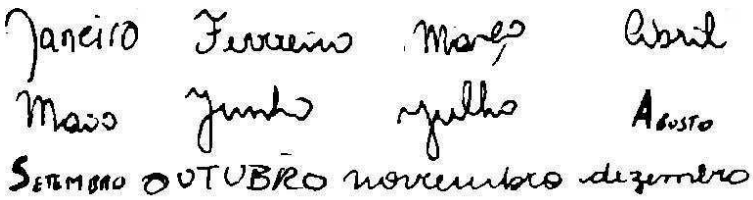


Figura 5.1: Amostras da base de dados dos meses do ano.

5.1.3 Base IRONOFF

A base de caracteres manuscritos IRONOFF [120] da *IRESTE/University of Nantes* (França) foi utilizada neste trabalho devido à sua representatividade de estilos de es-

Tabela 5.6: Número de amostras no conjunto treinamento/validação (Trn/Vld) e teste (Tst) para os escritores 1, 2, 3 e 4. (caracteres normais)

Caracter	Conj. Trn/Vld escritor 1	Conj. Tst escritor 1	Conj. Trn/Vld escritor 2	Conj. Tst escritor 2	Conj. Trn/Vld escritor 3	Conj. Tst escritor 3	Conj. Trn/Vld escritor 4	Conj. Tst escritor 4
a	1117	1105	938	1069	1287	922	1608	973
b	99	89	50	82	76	63	85	59
c	393	471	290	412	349	335	547	351
d	577	411	493	411	665	387	901	441
e	1117	1012	1138	1043	1185	825	1401	884
f	111	126	75	110	87	91	127	95
g	62	106	90	124	114	72	152	105
h	38	66	32	59	31	33	103	46
i	548	713	446	768	644	656	676	637
j	10	25	25	23	25	21	11	15
l	290	316	258	338	334	278	358	286
m	565	387	564	384	575	319	696	342
n	488	547	407	544	539	469	517	466
o	1161	942	1026	927	1158	752	1263	749
p	239	262	260	251	405	209	392	241
q	142	32	167	43	79	47	93	28
r	575	723	504	756	771	552	823	628
s	780	766	711	746	797	626	947	658
t	511	541	397	566	529	466	653	488
u	440	286	462	290	315	239	529	245
v	168	159	101	177	116	125	178	122
x	29	38	15	43	55	45	54	49
z	31	56	16	33	24	35	19	39

Tabela 5.7: Número de amostras no conjunto treinamento/validação (Trn/Vld) e teste (Tst) para os escritores 1, 2, 3 e 4. (caracteres acentuados)

Caracter	Conj. Trn/Vld escritor 1	Conj. Tst escritor 1	Conj. Trn/Vld escritor 2	Conj. Tst escritor 2	Conj. Trn/Vld escritor 3	Conj. Tst escritor 3	Conj. Trn/Vld escritor 4	Conj. Tst escritor 4
á	34	31	38	47	11	31	53	52
â	1	8	1	4	0	1	11	8
ã	112	92	76	78	104	68	15	19
é	52	28	26	39	28	20	15	22
í	25	37	12	43	57	28	53	29
ó	15	23	15	33	11	11	20	18
ô	2	2	8	13	0	0	1	2
õ	13	27	5	19	16	24	13	15
ú	0	1	10	10	0	3	17	9
ç	36	81	43	70	77	73	31	35
ê	24	14	16	29	6	7	21	15

crita e por conter caracteres manuscritos minúsculos isolados. Nos testes realizados neste trabalho foram utilizados os subconjuntos B1, B2,..., B26 que totalizam 10.485 caracteres manuscritos. Para a utilização nos experimentos as imagens dos caracteres manuscritos foram aleatoriamente agrupadas em três conjuntos: treinamento, com 240 imagens por caractere (60%), validação e teste, com 80 (20%) imagens por caractere em ambos os conjuntos. Na Figura 5.2 são ilustradas algumas amostras dos caracteres minúsculos da base IRONOFF utilizados nos experimentos aqui descritos.

Figura 5.2: Amostras de imagens de caracteres presente na base de dados IRONOFF.

5.2 Análise dos experimentos realizados com a base de dados dos meses do ano

A primeira avaliação realizada teve como finalidade analisar o comportamento dos algoritmos de extração de características e quantização vetorial desenvolvidos. O método de reconhecimento utilizado para a realização destes testes foi modelar cada palavra do vocabulário (janeiro, fevereiro,..., dezembro) por Modelos Escondidos de Markov, visto que o tamanho do léxico permite um modelo para cada classe.

Inicialmente foram construídos dicionários de reconstrução (quantizador vetorial) para cada um dos conjuntos de características analisado (perceptivas, globais e direcionais). O projeto dos dicionários foi realizado pelo algoritmo LBG utilizando duas formas distintas de construir o dicionário inicial: por uma função aleatória e através do próprio conjunto de treino. Os dicionários de reconstrução dos vetores de características perceptivas, globais e direcionais possuem vetores códigos com dimensão (K) 10, 6 e 10, respectivamente. Foram realizados testes com quatro diferentes níveis de

Tabela 5.8: Taxa de reconhecimento obtida pelo classificador HMM utilizando diferentes espaços de características.

Características	K	M	Max(%)	Média (%)	STD	DicICCT (%)
Perceptivas	10	27	77,7	76,9	0,6	78.37
Perceptivas	10	32	78,0	77,0	0,7	79.23
Perceptivas	10	64	78,0	77,5	0,4	79.51
Perceptivas	10	128	78,3	77,2	0,6	**80.17**
Globais	6	27	61,2	60,3	0,8	64.69
Globais	6	32	62,3	61,0	1,0	62.42
Globais	6	64	64,9	63,8	1,0	63.37
Globais	6	128	66,3	65,9	0,7	66.45
Direcionais	10	27	69,9	68,9	0,5	72.14
Direcionais	10	32	71,3	70,4	0,9	73.15
Direcionais	10	64	74,9	73,6	1,1	76.01
Direcionais	10	128	75,2	74,2	0,8	78.01

quantização (M): 27, 32, 64, 128.

Os modelos de palavras usados nestes testes são baseados em modelos HMM de topologia esquerda-direita com passo 2, com oito ou nove estados, dependendo se as observações são nos estados ou nos arcos. O número de estados foi determinado de modo empírico depois da vários testes. O treinamento dos modelos foi realizado utilizando o algoritmo Baum-Welch.

Na Tabela 5.8 estão resumidos os resultados experimentais obtidos com as observações nos estados. Os valores Max, Média e STD são os valores máximos, médio e o desvio padrão para a taxa de classificação, calculados a partir de dez repetições do experimento para cada vetor de características com o dicionário inicial construído através de uma função aleatória. A última coluna é a taxa de reconhecimento quando o Dicionário Inicial foi Construído a partir do Conjunto de Treino -(DicICCT).

Observa-se que o melhor desempenho foi obtido pelo classificador usando o conjunto

de características perceptivas, com 128 níveis de quantização e com o dicionário inicial construído a partir do conjunto de treino. Em geral, o aumento do número de níveis do quantizador vetorial tornou a representação dos vetores de entrada pelos vetores de saída mais fiel, ocasionando um aumento da taxa de reconhecimento do classificador. Com relação à inicialização do dicionário do quantizador vetorial, os experimentos mostraram que a utilização do conjunto de treino como dicionário inicial é uma boa estratégia, em geral, produzindo uma maior taxa de reconhecimento do que é obtida ao se inicializar com a estratégia aleatória.

Para avaliar o potencial na combinação dos espaços de características, foram realizados diversos experimentos, com finalidade de identificar qual combinação produz a melhor taxa de reconhecimento para o sistema em teste. Para tanto, foram construídos três novos vetores de características, através de concatenações dos vetores de características perceptivas, globais e direcionais. A mesma metodologia de teste anteriormente aplicada, foi utilizada nestes novos experimentos, cujos resultados são mostrados na Tabela 5.9.

A análise da Tabela 5.9 mostra que a melhor estratégia de combinação de características é unir características perceptivas, globais e direcionais em único vetor, utilizando um dicionário com 128 níveis. Comparando este resultado com o melhor obtido considerando os vetores isolados de características (Tabela 5.8) verifica-se um ganho de $\frac{85,2-80,1}{80,1} = 5,9\%$ na taxa de reconhecimento.

A taxa de reconhecimento sofre uma pequena queda quando são utilizados Modelos Escondidos de Markov com as observações nos arcos. A taxa de reconhecimento com esta estratégia foi de 84,96%, ou seja, uma queda de $\frac{85,2-84,96}{85,2} = 0,28\%$ comparado ao classificador HMM com as observações nos estados.

Além dos testes descritos acima, foram realizados testes com um classificador neural. Nesta estratégia, os vetores de entrada da rede neural são as características perceptivas, direcionais e globais. Diversas redes neurais, cada uma com única camada escondida, foram treinadas e testadas utilizando o simulador do ambiente WEKA (Waikato *Environment for Knowledge Analysis*), version 3.4, desenvolvido pela universidade de

Tabela 5.9: Taxa de reconhecimento do classificador HMM utilizando combinação de vetores de características.

Características	K	M	Max(%)	Média (%)	STD	DicICCT (%)
Perceptivas+globais	16	27	80.9	79,1	1,0	79,2
Perceptivas+globais	16	32	81,0	79,8	0,8	80,0
Perceptivas+globais	16	64	83,0	81,7	0,7	80,6
Perceptivas+globais	16	128	83,6	82,6	1,1	82,1
Perceptivas+Direcionais	20	27	81,4	80,2	1,1	80,57
Perceptivas+Direcionais	20	32	81,8	80,8	1,0	81,01
Perceptivas+Direcionais	20	64	82,7	81,5	1,1	82,7
Perceptivas+Direcionais	20	128	83,2	81,9	1,3	80,9
Perceptivas+globais+Direcionais	26	27	81,0	79,8	1,2	79,34
Perceptivas+globais+Direcionais	26	32	81,2	80,3	0,9	79,76
Perceptivas+globais+Direcionais	26	64	85,2	83,8	1,0	82,8
Perceptivas+globais+Direcionais	26	128	**85,1**	84,3	0,7	83,1

Waikato na Nova Zelândia. WEKA é uma ferramenta largamente utilizada para a aprendizagem automática e mineração de dados, contendo ferramentas escritas em Java, para a classificação, regressão, *clustering* e regras de associação. Os resultados dos testes realizados são apresentados na Tabela 5.10.

Comparando os resultados obtidos pelo sistema com HMM e com NN, percebe-se que a utilização de redes neurais, para problemas de léxico reduzido, é mais eficiente do que a utilização de Modelos Escondidos de Markov.

O resultado obtido utilizando a concatenação das características perceptivas, globais e direcionais, totalizando um vetor de característica com 26 elementos por cada sub-imagem, apresentou uma taxa de reconhecimento de 90,18%.

Para finalizar os testes com a base de dados dos meses do ano, foi realizado um experimento com redes neurais considerando como entrada a concatenação dos vetores de características perceptivas, globais e direcionais. Entretanto, foi realizado um zoneamento em cada sub-imagem analisada (dividindo a sub-imagem em 7 zonas), conforme

Tabela 5.10: Taxa de reconhecimento do classificador NN

Características	Taxa de Reconhecimento (%)
Perceptivas	81,18
Globais	76,97
Direcionais	83,20
Perceptivas+ globais	87,78
Perceptivas+Direcionais	89,09
Perceptivas+globais+Direcionais	**90,18**

descrito por Freitas *et al.* [123]. Entretanto, neste trabalho a sub-imagem foi dividida em sete zonas para só então realizar o processo de extração dos vetores de características 5.3. Vetores de características direcionais são extraídos de cada uma das zonas. Com esta estratégia há um aumento no número de neurônios da camada de entrada da rede neural que passa de 26 para 86 (10 características perceptivas + 6 globais + 7×10 direcionais). A taxa de reconhecimento obtida foi 89,42%, ligeiramente inferior ao obtido sem a estratégia de zoneamento (90,18%). Desta forma, pode-se concluir que o aumento na complexidade introduzida pelo zoneamento, nesta aplicação em específico, não é adequado.

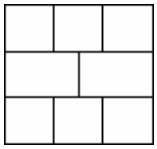


Figura 5.3: Zoneamento em 7 partes

Os resultados obtidos com a base dos meses do ano são compatíveis aos obtidos por Oliveira Jr., Kapp e Freitas, quando eles utilizam este tipo de características em seus sistemas de reconhecimento dos meses do ano. A máxima taxa de reconhecimento obtida foi de 97,9% quando utilizado a combinação de 5 classificadores [16, 83, 84, 104–106, 124].

Tabela 5.11: Taxas de reconhecimento (%) dos caracteres da base IRONOFF quando utilizado HMM.

Modelo	M	Perceptivas	Globais	Direcionais
HMM ERG	16	29,32	26,49	37,47
HMM ERG	32	33,25	30,91	41,92
HMM ERG	64	35,68	33,54	44,41
HMM ERG	128	38,74	34,49	46,04
HMM ED	128	38,23	36,00	47,52
HMM T6	128	39,07	38,20	**47,92**

5.3 Análise dos experimentos realizados com a base de caracteres IRONOFF

Diversos experimentos foram realizados com a base de caracteres IRONOFF. O objetivo foi determinar qual a melhor estratégia de combinação dos vetores de características e qual topologia dos Modelos Escondidos de Markov melhor se ajusta aos caracteres.

Para realizar o treinamento dos modelos dos 26 caracteres presentes na base do IRONOFF foram utilizadas múltiplas seqüências de observações. Cada um dos modelos foi treinado com 323 seqüências de observações.

Nesse trabalho, o número maior de observações (T) é obtido quando a letra é segmentada em 5 sub-imagens (número máximo de sub-imagens geradas pelo algoritmo de segmentação) e representada através de 3 vetores de características. Nesse caso, o tamanho máximo da seqüência de observação é 15 (5 sub-imagens $\times$ 3 símbolos).

Inicialmente, foram realizados experimentos com os vetores de características perceptivas, globais e direcionais isolados, utilizando as topologias de modelos: HMM ergódigo (HMM ERG), HMM esquerda-direita (HMM ED) e o modelo apresentado na Figura 4.33 (HMM T6). Todos os modelos foram treinados utilizando o procedimento *forward-backward* modificado, descrito anteriormente na Seção 4.5, para permitir as observações nos arcos. Os resultados são apresentados na Tabela 5.11.

A melhor taxa de reconhecimento (47,92%) foi obtida utilizando a característica direcional com os modelos de caracteres utilizando a topologia HMM T6. Além dos testes descritos acima, foi realizado um experimento concatenando as três características (perceptivas, globais e direcionais) em um único vetor. O processo de quantização vetorial foi aplicado sobre o vetor resultante da concatenação dos vetores de características. Para a realização da classificação, cada caractere foi modelado por um HMM T6. A taxa de reconhecimento teve uma pequena elevação (52,03%), mas ainda foi insatisfatória.

Uma das razões para estas baixas taxas é que quando a imagem do caractere é corretamente segmentada, ou seja, a sub-imagem é o próprio caractere, ele é representado por apenas um símbolo, o que dificulta o processo de reconhecimento pelos classificadores.

Para solucionar esta dificuldade, foi utilizada a combinação dos vetores de características, com cada vetor de característica quantizado individualmente. Foram gerados três dicionários de reconstrução, um para cada vetor de características. Desta forma, cada sub-imagem é representada por símbolos, representando cada um dos espaços de características. O classificador utilizado foi HMM T11 ou T16. As novas taxas de reconhecimento são apresentadas na Tabela 5.12. Nesta tabela, CCPG representa a concatenação dos vetores de observações referentes às características perceptivas e globais, CCPD a concatenação dos vetores de observações referentes às características perceptivas e direcionais, CCGD a concatenação dos vetores de observações referentes às características globais e direcionais e CCPGD a concatenação dos vetores de observações referentes às características perceptivas, globais e direcionais. O melhor resultado (62,66%) foi obtido utilizando as características perceptivas, globais e direcionais. Neste caso, quando o caractere é corretamente segmentado, ele é representado por três símbolos, um para cada espaço de característica.

Uma nova bateria de testes foi realizada, utilizando como classificador redes neurais. Foram realizados testes com os vetores de características direcionais (CD) e com a concatenação dos três vetores de características (CPGD). Além deste teste, foi realizado

Tabela 5.12: Taxas de reconhecimento dos caracteres da base IRONOFF quando utilizado HMM T11 ou HMMT16.

Modelo	Características	Taxa de reconhecimento (%)
HMM T11	CCPG	52,26
HMM T11	CCPD	55,15
HMM T11	CCGD	52,38
HMM T16	CCPGD	**62,66**

Tabela 5.13: Taxas de reconhecimento dos caracteres da base IRONOFF quando utilizado NN.

Características	Taxa de reconhecimento (%)
CD	65,86
CPGD	74,44
$CPGD_z$	**77,53**

um experimento com a concatenação dos três vetores de características, só que foi aplicado o procedimento de divisão por zonas proposto por Freitas *et al.* [123] em cada uma das sub-imagem dos caracteres antes do procedimento de extração do vetor de características direcionais ($CPGD_z$). Os resultados são apresentados na Tabala 5.13.

O melhor resultado obtido (77,53%) com o classificador neural foi obtido com os três vetores de características concatenados, com o vetor de características direcionais obtidos pela concatenação dos vetores de características direcionais aplicados em cada um das 7 zonas produzidas pelo processo de zoneamento desenvolvido por Freitas *et al.* [123].

Para averiguar a influência do processo de normalização (concebido para palavras) sobre os caracteres isolados, foi realizado um experimento sem o pré-processamento dos caracteres. A taxa de reconhecimento com NN subiu para 80,25%, um ganho de $\frac{80,25-77,53}{80,25} = 3,3\%$. As normalizações da linha de base e do ângulo de inclinação das letras não produzem desempenhos satisfatórios, quando aplicadas a caracteres individuais. Outro problema averiguado é que a determinação da linha de base é comprometida.

Tabela 5.14: Taxas de reconhecimento dos caracteres da base IRONOFF quando utilizados HMM, sem pré-processamento e segmentação.

Características	Tolopogia	N	Taxa de reconhecimento (%)
$CCPGD_z$	HMM T ED	6	57,52
$CCPGD_z$	HMM E ED	6	67,79
$CCPGD_z$	HMM T6	6	65,66
$CCPGD_z$	HMM T16	16	**69,29**

Não havendo como determinar corretamente esta linha de referência. Como os algoritmos de segmentação utilizam estas duas informações, eles têm seus desempenhos também comprometidos. Entretanto, como os caracteres já se encontram segmentados, não é necessário realizar a segmentação, quando usando a base IRONOFF. Os resultados sem os processos de normalização e segmentação para diversas topologias de HMM são apresentados na Tabela 5.14.

Os resultados mostram que a melhor estratégia para classificar os caracteres é utilizar os vetores de características perceptivas, globais e direcionais concatenados com o processo de zoneamento e redes neurais para realizar a classificação, sem as etapas de normalização e segmentação. Usando HMM como classificador a estratégia que melhor se adequa é utilizar os três vetores de características quantizados com dicionários distintos e HMM com as observações nos arcos com a topologia (HMM T16), apresentado na Figura 4.33.

5.4 Análise do resultados com a base de dados dependente do usuário

A seguir são apresentados os resultados obtidos com experimentos com a base de palavras manuscritas dependente do usuário. Os primeiros experimentos foram realizados com os manuscritos do escritor 1. As melhores estratégias para o reconhecimento dos caracteres e das palavras obtidos para o escritor 1 foram utilizados nos experimentos

dos escritores 2, 3 e 4.

5.4.1 Experimento com a base de palavras manuscritas produzidas pelo escritor 1

Inicialmente, cada palavra pré-processada dos conjuntos de treinamento, validação e teste foi segmentada e manualmente rotulada. Embora, este procedimento não seja necessário (o treinamento poderia ter sido realizado através do algoritmo de treinamento *embedded* Baum-Welch [35], que não necessita o rotulamento dos segmentos produzidos pelo algoritmo de segmentação), ele foi realizado para treinar e testar a classificação dos caracteres individualmente, para só então realizar os testes considerando a palavra.

Foram realizados vários experimentos com os caracteres segmentados com o objetivo de determinar qual a topologia que melhor se adequa ao problema em questão. Em todos os experimentos que se seguem o vetor de observação de cada sub-imagem é composto por três símbolos. Cada símbolo representa o índice do vetor código gerado na representação QV da seqüência de características com o uso do dicionário específico.

O projeto dos dicionários do quantizador vetorial foi realizado pelo algoritmo LBG com o dicionário inicial construído através do conjunto de amostras iniciais da seqüência de treinamento (inicialização que obteve uma maior taxa de reconhecimento nos teste realizados com a base dos meses do ano), utilizando 128 níveis de quantização. A dimensão do quantizador vetorial para os vetores de características perceptivas, globais e direcionais com zoneamento é de 10, 6 e 70, respectivamente. Todos os resultados apresentados na Tabela 5.15 foram obtidos utilizando HMM com as observações nas transições, utilizando o mesmo conjunto de dicionários e CCPGD. A melhor taxa de reconhecimento (82,30%) foi obtida com o classificador HMM T16. A partir de então, todos os experimentos com HMM foram realizados com esta topologia de HMM.

A matriz de confusão para o resultado com a topologia HMM T16 é apresentada na Tabela 5.16. Estes resultados mostram que o sistema possui desempenho distinto para os caracteres avaliados. Os caracteres com maior grau de dificuldade no reconhecimento são “h”, “g”, e “x”, com taxas de 62,12%, 69,81% e 50% respectivamente. Analisando a

Tabela 5.15: Taxas de reconhecimento para a base de teste dos caracteres do escritor 1.

Tolopogia	N	Taxa de reconhecimento (%)
HMM ERG	16	79,37
HMM ED	16	81,80
HMM T16	16	**82,39**

Tabela 5.6 percebe-se que os caracteres "h", "g" e "x" possuem uma baixa quantidade de amostras no conjunto de treinamento. Além disso, para os caracteres 'h' e 'g' o número de amostras no conjunto de teste é quase o dobro do conjunto de treinamento, conforme mostrado na Tabela 5.6. Entretanto, o caractere com menor número de amostras no conjunto de treino é o caractere "j", que apresenta uma taxa de reconhecimento de 72%. Não se pode afirmar que a baixa quantidade de amostras no conjunto de treino é determinante para uma baixa taxa de reconhecimento. O importante é que o conjunto de treinamento possua amostras que representem o estilo de escrita daquele caractere na base de dados analisada (representatividade). A maior fonte de confusão na Tabela 5.16 foi reconhecer erroneamente o caractere "g" como sendo o caractere "q" (25,5%). Entretanto, apenas 6,3% dos caracteres "q" foram erroneamente classificados com "g". Existe uma similaridade na forma de escrita dos caracteres "i" e "e", "l" e "e", "m" e "n", "j" e "f" e "j" e "z" para o escritor 1, conforme observado na Tabela 5.16

Além dos experimentos com HMM foram realizados testes com redes neurais. Inicialmente, foi utilizado o vetor de observações normalizados (após a etapa de QV) dos caracteres, sem a etapa de segmentação, como entrada de uma rede neural contendo três neurônios na camada de entrada e na camada escondida. A taxa de reconhecimento obtida foi de 46%. Quando o processo de quantização vetorial não é realizado tem-se um aumento no tamanho do vetor de entrada (3 para 86 elementos). Neste caso, a taxa de reconhecimento aumenta para 93,7%. Em ambos os casos, foi utilizada uma rede neural contendo uma única camada escondida treinada e testada no ambiente de simulação do MATLAB.

Tabela 5.16: Matriz de confusão para o escritor 1 (%).

	a	b	c	d	e	f	g	h	i	j	l	m	n	o	p	q	r	s	t	u	v	x	z
a	91,0	0,0	0,4	1,3	0,5	0,0	0,2	0,1	0,0	0,0	0,3	0,1	0,3	3,0	0,8	0,8	0,4	0,0	0,2	0,5	0,1	0,2	0,0
b	1,1	79,8	1,1	0,0	1,1	2,2	0,0	1,1	0,0	0,0	4,5	0,0	0,0	2,2	0,0	0,0	0,0	0,0	0,0	1,1	5,6	0,0	0,0
c	1,1	0,0	82,0	0,2	1,7	0,0	0,0	0,0	3,4	0,0	1,3	0,0	0,6	0,8	0,8	0,0	2,1	1,9	1,5	0,8	0,4	1,3	0,0
d	1,1	0,0	0,6	96,8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,0	0,2	0,0	0,0	0,0	0,4	0,0	0,0	0,0
e	0,0	0,2	2,0	0,0	79,5	0,1	0,0	0,0	8,5	0,2	2,9	0,0	0,1	0,3	0,3	0,1	1,5	1,5	0,7	1,0	0,2	1,0	0,0
f	0,0	0,8	0,0	2,4	0,0	82,5	0,0	0,0	0,0	1,6	4,0	0,0	0,0	0,8	2,4	0,8	0,0	0,8	2,4	0,0	0,8	0,0	0,8
g	0,0	0,0	0,0	0,0	0,0	0,9	69,8	0,0	0,0	0,9	0,0	0,0	0,0	0,0	0,0	25,5	0,0	0,9	0,0	0,0	0,0	0,0	1,9
h	0,0	6,1	0,0	3,0	3,0	3,0	1,5	62,1	0,0	0,0	1,5	1,5	0,0	0,0	0,0	0,0	9,1	7,6	0,0	1,5	0,0	0,0	0,0
i	0,0	0,0	1,5	0,0	12,3	0,0	0,0	0,0	81,6	0,1	0,1	0,0	0,0	0,1	0,0	0,0	2,0	0,4	0,6	0,6	0,1	0,4	0,0
j	0,0	0,0	0,0	0,0	0,0	12,0	0,0	0,0	4,0	72,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	12,0
l	0,0	1,3	0,3	0,3	10,4	2,2	0,0	0,0	0,0	0,0	82,6	0,0	0,0	0,0	0,0	0,0	0,0	0,6	1,9	0,3	0,0	0,0	0,0
m	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,0	0,0	89,4	9,6	0,0	0,0	0,0	0,3	0,0	0,0	0,5	0,0	0,0	0,0
n	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,0	0,0	0,0	10,1	86,7	0,0	0,2	0,0	0,4	0,7	0,0	0,2	0,7	0,7	0,0
o	4,8	0,1	1,6	0,2	0,3	0,0	0,7	0,0	0,2	0,0	0,0	0,0	0,2	89,6	1,2	0,3	0,0	0,1	0,0	0,1	0,2	0,1	0,2
p	0,0	0,4	0,0	0,0	0,0	1,5	0,0	0,0	0,0	0,0	0,0	0,4	0,0	0,4	95,8	0,4	0,0	1,1	0,0	0,0	0,0	0,0	0,0
q	0,0	0,0	0,0	0,0	0,0	0,0	6,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	93,8	0,0	0,0	0,0	0,0	0,0	0,0	0,0
r	0,0	0,0	1,9	0,1	3,5	0,0	0,0	0,6	2,8	0,0	0,4	0,7	3,2	0,0	0,4	0,0	77,9	5,8	0,8	0,8	0,8	0,3	0,0
s	0,1	0,3	0,5	0,0	3,0	0,3	0,0	0,4	2,3	0,7	0,4	0,4	1,7	0,9	2,0	0,0	5,1	79,4	0,4	0,3	0,4	1,4	0,1
t	0,0	0,4	2,6	0,2	0,7	0,7	0,0	0,0	0,9	0,7	2,6	0,0	0,0	0,0	0,0	0,0	0,4	0,7	90,0	0,0	0,0	0,0	0,0
u	0,7	0,3	0,7	0,0	2,1	0,0	0,0	0,0	1,7	0,0	0,0	0,3	0,3	0,3	0,0	0,0	2,1	1,0	0,3	88,1	0,0	1,7	0,0
v	1,3	2,5	1,9	0,0	0,0	0,0	0,0	0,0	0,0	0,6	0,0	0,6	3,8	1,9	0,0	0,0	1,3	0,6	0,0	3,1	81,8	0,6	0,0
x	2,6	0,0	5,3	0,0	2,6	0,0	0,0	0,0	2,6	0,0	0,0	7,9	15,8	0,0	0,0	0,0	2,6	2,6	2,6	0,0	5,3	50,0	0,0
z	0,0	0,0	0,0	0,0	0,0	1,8	1,8	0,0	0,0	3,6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	92,9

A utilização da quantização vetorial reduz significativamente a quantidade de informações apresentadas à rede neural, o que inviabiliza a classificação por redes neurais. A maior taxa de reconhecimento dos caracteres foi obtida utilizando redes neurais, com a concatenação dos vetores de características como entrada.

O objetivo principal deste trabalho é reconhecer palavras manuscritas produzidas por escritores cadastrados pelo sistema. Para realizar esta tarefa foram construídos os modelos estatísticos dos caracteres (HMM). Diversos experimentos foram realizados para determinar qual a melhor estratégia a ser utilizada para classificar as palavras, partindo do principio que cada palavra é composta de uma combinação de caracteres.

A primeira estratégia utilizada para classificar as palavras da base dependente do escritor foi obter para cada i-ésima seqüência de sub-imagens, fornecidas pelo módulo gerador de sub-imagens, a combinação de caracteres que maximiza a probabilidade a posteriori. Dada uma sub-imagem j de uma seqüência i, representada pela seqüência de observação o_{ij}, foi calculado qual caractere l maximiza a probabilidade $P(o_{ij}|c_l)$. $P(o_{ij}|c_l)$ representa a probabilidade da seqüência de observações o_{ij} ser observada no caractere l. A probabilidade da i-ésima seqüência de sub-imagens, com um vetor observações $O_i = o_{i1}, o_{i2}, ..., o_{im}$, é calculada utilizando a Equação 5.1.

$$P(O_i|w) = \prod_{j=1} \max_{c_l} P(o_{ij}|c_l) \tag{5.1}$$

A probabilidade a posteriori de uma palavra w pode ser calculada através da regra de Bayes:

$$P(w|O) = \max_{w} P(O_i|w).P(w) \tag{5.2}$$

em que $P(w)$ é a probabilidade a priori da palavra ocorrer. Neste trabalho, assumimos que as palavras são equiprováveis.

A taxa de reconhecimento obtida foi de 59,54%. Devido a este baixo desempenho foram realizados testes adicionais com ajuda do vocabulário. A palavra reconhecida é escolhida como aquela pertencente ao vocabulário que obtiver a maior probabilidade a posteriori, admitindo um erro em apenas um caractere na palavra. Com esta nova

estratégia a taxa de reconhecimento foi de 67,68%.

A seguir, foi realizado um novo experimento (estratégia 2), em que apenas eram testadas as combinações de letras permitidas pelo vocabulário. Portanto, dada uma seqüência de m sub-imagem, o objetivo é determinar qual a palavra no vocabulário com m letras que maximiza a $P(w/O)$. A palavra reconhecida é aquela com a maior probabilidade a posteriori entre todas as seqüências de sub-imagens possíveis. A taxa de reconhecimento obtida com esta estratégia foi de 92,96%, um ganho de $\frac{92{,}96-67{,}68}{92{,}96} = 27,19\%$ em relação à estratégia 1.

A estratégia 2 foi utilizada com o classificador neural, utilizando como entrada da rede os três vetores de características concatenados, obtendo uma taxa de reconhecimento de 87,51%. Embora o classificador neural tenha um desempenho superior ao HMM na classificação dos caracteres individualmente, seu desempenho é inferior ao HMM quando utilizado para classificar as palavras através dos caracteres, utilizando a estratégia 2. É difícil identificar o motivo desta perda de desempenho. Entretanto, a análise dos resultados mostrou que quando uma sub-imagem que não representa um caractere (produzida pela sobre-segmentação do caractere) é apresentada ao HMM ele produz uma probabilidade muito baixa (10 a 100 vezes menor do que a obtida quando é apresentado um caractere completo) para todos os caracteres, diferentemente do resultado da NN. Quando é submetido à NN um vetor de características, representando uma sub-imagem que não corresponde a um caractere, este ativa uma das saídas, indicando erroneamente que tal sub-imagem é um dos caracteres para qual ele foi treinado. Devido ao processo de sobre-segmentação das palavras, não é possível a existência de uma classe representando um não caractere, pois a sobre-segmentação de caractere pode gerar pedaços de caracteres que se assemelham a um caractere. Isto prejudicaria o treinamento NN.

5.4.2 Experimento com a base de palavras manuscritas produzidas pelos escritores 2, 3 e 4

Inicialmente, foi realizada a segmentação manual das palavras, dividindo cada imagem em sub-imagens com um único caractere. As taxas de reconhecimento para os testes realizados com os caracteres isolados são apresentados na Tabela 5.20. Todos os experimentos foram realizados considerando a topologia HMM T16. As respectivas matrizes de confusão são apresentadas nas tabelas 5.17, 5.18 e 5.19, com destaque (em vermelho) para as maiores confusões entre os caracteres (em vermelho).

A análise da Tabela 5.17 mostra que a maior fonte de confusão, para o escritor 2, foi reconhecer erroneamente o caractere "e" como sendo o caractere "x" (27,9%). Entretanto, apenas 1,7% dos caracteres "e" foram erroneamente classificados com "x". Os caracteres com maior grau de dificuldade no reconhecimento para o escritor 2 foram os caracteres "b", "c", "f", "j" e "x", com taxas de reconhecimento inferiores a 60%. Enquanto, as maiores de dificuldade no reconhecimento para o escritor 3, Tabela 5.18, foram os caracteres "h", "q", "v", e "x", com taxas de reconhecimento inferiores a 60%.

Os resultados apresentados na Tabela 5.20 mostram que o sistema possui desempenho distinto para cada escritor. Este resultado era esperado. Entretanto, o que chama atenção nesta tabela é o baixo desempenho obtido para escritor o 4. Uma análise visual nas palavras redigidas por este escritor indicou a presença em muitas palavras de caracteres distintos, mas representados pela mesma grafia, como pode ser observado na Figura 5.4. Estes caracteres prejudicam o treinamento e conseqüentemente o desempenho do sistema. Os caracteres com maior grau de dificuldade de reconhecimento para o escritor 4, segundo a matriz de confusão 5.19, foram os caracteres "b", "f", "h", "j" e "x", com taxas de reconhecimento inferiores a 60%. Coincidentemente, estes caracteres possuem baixo número de amostras no conjunto de treinamento.

A análise das matrizes de confusão para os caracteres redigidos pelos escritores 1, 2, 3 e 4 mostra que o caractere mais difícil de ser reconhecido é o caractere "x". Em todos os conjuntos de treinamento (escritores 1, 2, 3 e 4) este caractere possui uma baixa representatividade. Visando inserir mais representatividade a este caractere

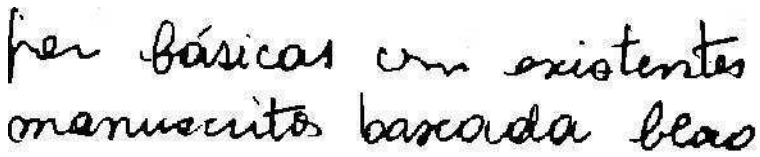


Figura 5.4: Amostras da base de palavras redigidas pelo escritor 4. Estas imagens correspondem as seguintes palavras, no sentido esquerda direita, de cima para baixo: ser, básicas, um, existentes, manuscritos, baseada e bloco.

Tabela 5.17: Matriz de confusão para o escritor 2 (%).

	a	b	c	d	e	f	g	h	i	j	l	m	n	o	p	q	r	s	t	u	v	x	z
a	84,8	0,2	0,7	2,9	0,7	0,0	0,3	0,0	0,1	0,0	0,0	0,0	0,3	4,5	0,0	1,6	0,9	0,9	0,0	1,0	0,6	0,3	0,2
b	1,2	50,0	0,0	0,0	2,4	0,0	2,4	3,7	0,0	0,0	3,7	0,0	0,0	7,3	4,9	0,0	1,2	8,5	0,0	3,7	11,0	0,0	0,0
c	2,9	0,0	60,7	0,2	17,2	0,7	0,5	0,0	4,9	0,0	0,5	0,0	0,2	7,3	0,0	0,0	0,5	1,5	0,2	0,5	0,5	1,5	0,2
d	10,9	0,0	0,5	79,8	1,7	0,0	0,5	0,2	0,0	0,2	0,0	0,2	0,0	0,5	0,0	0,0	0,0	0,7	0,0	3,6	1,0	0,0	0,0
e	1,0	0,1	10,6	0,0	76,3	0,0	0,1	0,0	4,6	0,1	2,2	0,1	0,0	0,5	0,2	0,1	0,7	0,9	0,3	0,3	0,2	1,7	0,1
f	0,0	0,0	0,0	0,0	1,8	51,8	3,6	0,0	1,8	22,7	0,9	0,0	0,0	0,0	1,8	0,0	0,9	0,9	11,8	0,0	0,0	0,0	1,8
g	1,6	0,8	0,8	0,0	0,0	0,8	68,5	0,0	0,0	1,6	0,0	0,0	0,0	0,8	0,0	21,0	0,0	3,2	0,0	0,0	0,0	0,0	0,8
h	0,0	10,2	0,0	0,0	1,7	0,0	0,0	55,9	0,0	0,0	0,0	1,7	1,7	0,0	6,8	0,0	13,6	5,1	3,4	0,0	0,0	0,0	0,0
i	0,4	0,0	2,0	0,0	9,6	0,7	0,0	0,0	81,4	0,1	0,1	0,4	0,0	0,1	0,0	0,0	0,8	0,8	0,1	0,9	0,0	2,6	0,0
j	0,0	0,0	0,0	0,0	0,0	21,7	4,3	0,0	0,0	52,2	0,0	0,0	0,0	0,0	4,3	0,0	4,3	4,3	8,7	0,0	0,0	0,0	0,0
l	0,9	0,3	2,1	0,3	18,9	1,8	0,3	0,0	2,4	1,2	61,5	0,0	0,0	0,0	0,3	0,0	0,6	0,6	8,3	0,0	0,3	0,3	0,0
m	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	93,0	7,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
n	0,0	0,0	0,4	0,0	0,0	0,0	0,0	0,2	0,4	0,0	0,0	31,3	64,0	0,0	0,2	0,0	2,4	0,6	0,0	0,6	0,0	0,2	0,0
o	6,7	0,2	2,4	0,5	0,2	0,1	0,3	0,0	0,1	0,0	0,1	0,0	0,0	83,9	0,1	0,0	0,0	3,8	0,0	0,2	1,0	0,0	0,3
p	0,0	0,0	1,2	0,0	0,0	0,8	0,4	0,4	0,0	0,8	0,0	0,4	0,4	0,4	91,6	0,0	1,6	1,6	0,0	0,0	0,0	0,0	0,4
q	2,3	0,0	0,0	2,3	0,0	0,0	4,7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	90,7	0,0	0,0	0,0	0,0	0,0	0,0	0,0
r	1,2	0,0	0,3	0,1	0,7	0,0	0,3	0,4	0,5	0,1	0,3	9,3	4,1	0,3	1,1	0,0	75,7	3,3	0,1	0,3	0,1	2,0	0,0
s	2,1	0,0	2,1	0,1	1,6	0,0	0,1	0,3	0,3	0,0	0,4	0,7	1,5	4,4	0,8	0,0	8,0	72,7	0,7	1,1	1,5	1,2	0,4
t	0,2	0,4	0,4	0,0	0,9	6,7	0,2	0,2	0,5	2,1	2,8	0,2	0,0	0,0	0,7	0,0	0,2	1,4	83,0	0,0	0,2	0,0	0,0
u	6,9	0,3	0,0	0,3	0,0	0,0	0,0	0,0	0,7	0,0	0,0	0,3	0,0	0,0	0,3	0,0	6,9	0,3	0,0	81,4	4,0	0,0	0,0
v	1,1	0,6	1,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,6	0,6	4,0	0,0	0,0	1,1	5,1	0,0	7,9	78,0	0,0	0,0
x	7,0	0,0	2,3	0,0	27,9	0,0	0,0	0,0	0,0	0,0	0,0	0,0	2,3	0,0	0,0	0,0	4,7	4,7	0,0	4,7	0,0	46,5	0,0
z	0,0	3,0	0,0	0,0	0,0	3,0	6,1	0,0	0,0	9,1	0,0	0,0	0,0	6,1	0,0	0,0	0,0	6,1	0,0	0,0	3,0	0,0	63,6

Tabela 5.18: Matriz de confusão para o escritor 3 (%).

	a	b	c	d	e	f	g	h	i	j	l	m	n	o	p	q	r	s	t	u	v	x	z
a	83,6	0,8	0,9	0,5	2,5	0,0	0,1	0,1	0,2	0,0	0,0	0,9	0,8	4,4	0,1	0,2	1,3	1,2	0,1	1,7	0,0	0,4	0,1
b	0,0	76,2	1,6	1,6	1,6	1,6	0,0	4,8	0,0	0,0	0,0	0,0	0,0	1,6	1,6	6,3	1,6	1,6	0,0	0,0	0,0	0,0	0,0
c	3,9	0,0	78,5	0,0	2,4	0,0	0,3	0,0	0,6	0,3	0,6	0,3	1,2	1,8	0,3	0,0	0,6	0,6	0,0	6,3	0,9	1,5	0,0
d	3,9	0,3	0,6	94,3	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,3	0,0	0,3	0,0	0,3	0,3	0,3	0,0	0,0	0,0
e	2,7	1,0	2,1	0,5	77,5	0,1	1,1	0,0	1,9	0,1	0,8	0,6	0,8	4,2	0,2	0,6	0,7	1,9	0,2	0,8	0,5	1,3	0,1
f	0,0	1,1	0,0	0,0	0,0	69,2	0,0	0,0	0,0	7,7	1,1	0,0	0,0	0,0	1,1	2,2	1,1	0,0	13,2	0,0	0,0	0,0	3,3
g	4,2	1,4	1,4	0,0	0,0	4,2	69,4	0,0	1,4	4,2	0,0	0,0	0,0	0,0	1,4	8,3	0,0	2,8	0,0	0,0	0,0	0,0	1,4
h	3,0	6,1	3,0	0,0	3,0	0,0	0,0	54,5	0,0	0,0	3,0	3,0	9,1	0,0	0,0	0,0	12,1	0,0	0,0	3,0	0,0	0,0	0,0
i	0,0	0,3	0,9	0,0	1,4	0,2	0,0	0,0	88,1	0,6	0,9	0,6	1,4	0,2	0,0	0,0	2,6	0,6	0,2	0,3	0,6	1,2	0,0
j	0,0	0,0	0,0	0,0	0,0	4,8	0,0	0,0	4,8	66,7	0,0	0,0	0,0	4,8	4,8	0,0	0,0	0,0	4,8	0,0	0,0	0,0	9,5
l	0,0	1,8	4,0	1,1	0,4	0,4	0,4	0,7	0,0	0,4	80,9	0,4	0,0	0,0	0,7	1,1	1,4	0,4	5,8	0,0	0,0	0,4	0,0
m	0,0	0,0	0,0	0,0	0,6	0,0	0,0	0,0	0,0	0,0	0,0	92,8	6,0	0,0	0,0	0,0	0,3	0,0	0,0	0,3	0,0	0,0	0,0
n	1,3	0,0	0,2	0,0	0,0	0,0	0,0	0,0	1,1	0,0	0,0	18,6	69,3	0,2	0,0	0,0	1,9	0,0	0,0	6,2	0,6	0,6	0,0
o	2,3	0,3	0,8	0,3	3,9	0,3	0,4	0,0	0,3	0,0	0,0	0,3	0,3	86,2	0,7	0,1	0,7	1,3	0,1	0,8	0,3	0,8	0,1
p	0,0	0,0	0,0	0,5	0,0	0,5	0,5	0,5	0,0	0,5	0,0	0,5	0,0	1,0	89,5	1,0	1,9	1,0	0,0	0,0	1,4	1,0	0,5
q	2,1	0,0	0,0	4,3	0,0	0,0	25,5	0,0	0,0	2,1	0,0	0,0	0,0	2,1	6,4	48,9	0,0	2,1	0,0	0,0	0,0	0,0	6,4
r	2,2	0,2	2,0	0,0	3,4	0,2	0,0	0,2	0,7	0,2	0,0	1,4	3,3	0,4	0,2	0,0	79,0	2,2	0,2	1,6	0,5	2,2	0,0
s	2,6	1,6	2,2	0,3	3,4	0,3	1,0	0,2	1,3	0,5	0,6	0,3	0,6	1,8	1,0	0,3	3,2	72,0	0,5	0,8	0,8	3,2	1,6
t	0,0	0,9	0,0	0,2	0,0	4,5	0,2	0,4	0,4	0,0	5,2	0,2	0,0	0,0	0,2	0,4	0,2	0,9	85,6	0,0	0,4	0,2	0,0
u	0,8	0,4	2,9	0,0	0,8	0,0	0,0	0,0	0,0	0,4	0,0	2,9	8,4	0,8	0,0	0,0	1,3	0,0	0,4	68,2	23,2	0,4	0,0
v	0,8	0,8	2,4	0,0	2,4	0,0	0,0	0,0	2,4	0,0	0,0	0,8	4,0	1,6	0,0	0,0	7,2	3,2	1,6	29,6	39,2	4,0	0,0
x	4,4	0,0	0,0	0,0	8,9	0,0	0,0	2,2	0,0	0,0	0,0	2,2	4,4	2,2	0,0	0,0	20,0	4,4	0,0	4,4	2,2	44,4	0,0
z	0,0	0,0	2,9	0,0	0,0	8,6	0,0	0,0	0,0	14,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	2,9	0,0	2,9	0,0	0,0	68,6

Tabela 5.19: Matriz de confusão para o escritor 4 (%).

	a	b	c	d	e	f	g	h	i	j	l	m	n	o	p	q	r	s	t	u	v	x	z
a	85,9	0,2	1,0	0,8	2,0	0,1	0,0	0,1	0,2	0,0	0,0	0,1	0,6	5,8	0,2	0,1	0,6	0,9	0,0	0,3	0,0	1,0	0,0
b	1,7	39,0	3,4	0,0	0,0	1,7	0,0	8,5	0,0	0,0	16,9	0,0	0,0	0,0	11,9	0,0	0,0	0,0	11,9	3,4	1,7	0,0	0,0
c	0,9	1,1	49,3	0,0	11,7	0,0	0,0	0,0	8,3	0,0	3,4	0,3	0,3	1,7	0,0	0,0	9,1	2,6	0,3	2,6	6,0	2,0	0,6
d	2,0	0,0	0,3	96,4	0,0	0,0	0,0	0,0	0,0	0,0	0,6	0,0	0,0	1,1	0,0	0,3	0,0	0,0	0,3	0,0	0,0	0,0	0,0
e	1,2	0,5	12,7	0,3	54,1	0,0	0,0	0,3	11,3	0,0	2,7	0,2	0,1	3,8	0,1	0,0	5,9	2,3	0,5	0,8	1,9	1,2	0,0
f	0,0	0,0	0,0	0,0	0,0	58,9	1,1	0,0	0,0	12,6	0,0	1,1	0,0	0,0	14,7	4,2	0,0	0,0	5,3	0,0	0,0	0,0	2,1
g	1,0	0,0	0,0	0,0	0,0	2,9	78,1	0,0	0,0	1,0	0,0	0,0	0,0	1,9	0,0	8,6	0,0	1,0	0,0	0,0	1,9	0,0	3,8
h	2,2	15,2	2,2	2,2	0,0	0,0	0,0	54,3	0,0	0,0	10,9	0,0	0,0	0,0	2,2	0,0	2,2	0,0	4,3	4,3	0,0	0,0	0,0
i	0,2	0,0	4,4	0,0	8,0	0,3	0,0	0,0	65,8	0,2	0,5	0,0	1,3	0,3	0,3	0,0	13,2	2,7	0,6	0,9	1,3	0,2	0,0
j	0,0	0,0	0,0	0,0	0,0	26,7	0,0	0,0	0,0	40,0	0,0	0,0	0,0	0,0	20,0	6,7	0,0	0,0	6,7	0,0	0,0	0,0	0,0
l	1,0	5,9	1,4	0,0	5,2	0,3	0,0	4,5	0,0	0,3	71,3	0,0	0,0	0,7	0,3	0,7	0,0	0,3	7,3	0,0	0,0	0,0	0,3
m	1,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,0	87,1	7,0	0,0	0,0	0,0	0,6	0,0	0,0	2,0	1,2	0,6	0,0
n	1,1	0,0	1,3	0,2	0,6	0,0	0,0	0,2	0,4	0,0	0,0	10,7	66,3	0,4	0,4	0,0	3,4	4,1	0,0	5,4	3,2	2,1	0,0
o	8,9	0,3	2,1	1,2	4,5	0,3	0,1	0,0	0,3	0,0	0,8	0,4	0,4	69,6	0,1	0,5	0,9	6,1	0,9	0,3	0,7	1,2	0,3
p	0,0	0,8	0,0	0,0	0,0	4,1	0,0	0,4	0,0	0,8	0,0	0,0	0,8	0,0	88,8	0,0	0,0	0,4	1,7	0,4	0,4	0,0	1,2
q	3,6	0,0	0,0	0,0	0,0	14,3	21,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	3,6	53,6	0,0	0,0	0,0	0,0	0,0	3,6	0,0
r	0,2	0,0	2,9	0,0	6,1	0,2	0,0	0,3	12,3	0,2	0,3	2,2	1,9	0,5	0,5	0,0	52,7	10,7	0,5	2,5	3,5	2,7	0,0
s	2,1	0,5	2,0	0,6	4,6	0,3	0,2	0,2	3,8	0,3	0,2	0,8	3,0	5,2	0,6	0,3	9,0	61,4	0,3	0,2	2,1	0,9	1,7
t	0,0	2,3	2,3	0,6	0,6	2,9	0,0	1,0	0,4	0,8	5,3	0,0	0,6	0,6	3,9	0,2	0,6	0,4	77,3	0,0	0,0	0,0	0,2
u	1,6	0,0	2,4	0,0	2,4	0,0	0,0	0,0	1,2	0,0	0,0	3,3	4,1	0,4	1,2	0,0	2,4	2,0	0,0	73,5	9,8	0,4	0,0
v	0,0	0,8	2,5	0,0	1,6	0,0	0,0	0,8	0,8	1,6	0,8	2,5	4,9	1,6	0,0	0,0	9,0	2,5	0,0	7,4	62,3	0,8	0,0
x	2,0	0,0	12,2	2,0	6,1	0,0	0,0	0,0	6,1	0,0	0,0	4,1	10,2	4,1	0,0	0,0	6,1	18,4	0,0	2,0	8,2	18,4	0,0
z	2,6	0,0	0,0	0,0	0,0	2,6	2,6	0,0	0,0	2,6	0,0	0,0	2,6	0,0	0,0	0,0	2,6	7,7	2,6	0,0	0,0	0,0	74,4

Tabela 5.20: Taxas de reconhecimento para as bases de teste de **caracteres** dos escritores 1, 2, 3 e 4, utilizando o vetor de observações CCPGD e o classificador HMM T16, com a estratégia 2.

Escritor	Taxa de reconhecimento (%)
1	82,39
2	71,61
3	72,17
4	64,28

Tabela 5.21: Taxas de reconhecimento para as bases de teste de **palavras** dos escritores 1, 2, 3 e 4, utilizando o vetor de observações CCPGD e o classificador HMM T16 com a estratégia 2.

Escritor	Taxa de reconhecimento (%)
1	92,96
2	92,85
3	90,78
4	83,31

nos conjuntos de treinamento foi adicionado ao conjunto de treino os caracteres "x" da base de IRONOFF. O resultado não foi satisfatório com uma diminuição da taxa média de reconhecimento do sistema, embora tenha ocorrido um leve aumento na taxa de reconhecimento deste caractere. Este perda de desempenho deve-se à inclusão de amostras de caracteres produzidos por grupo de escritores diferentes, com estilo distinto de representar as letras.

Os resultados obtidos para o reconhecimento das palavras são apresentados na Tabela 5.21 para todos os escritores. Nestes testes foi utilizada a estratégia 2, em que apenas eram testadas as combinações de letras permitidas pelo vocabulário. Para todos os escritores o sistema teve um desempenho superior ao obtido com o reconhecimento dos caracteres individualmente.

5.5 Comentários finais

Neste capítulo foram apresentados os testes efetuados e os resultados obtidos neste trabalho. Inicialmente, foram realizados testes com a base dos meses do ano. Nestes testes, foram avaliados o comportamento do algoritmo de extração de características e de quantização vetorial. O vetor de características perceptivas obteve o melhor desempenho individual. Porém, a combinação dos vetores de características perceptivas, globais e direcionais produziu melhor resultado. Também, foram avaliados o comportamento do sistema com relação ao número de níveis do quantizador vetorial e a forma de construção do dicionário inicial. Os resultados dos testes indicaram que a utilização de dicionários com 128 níveis e a utilização do próprio conjunto de treino como dicionário inicial é a melhor estratégia a ser utilizada para o quantizador vetorial. Foi realizada uma comparação do desempenho do sistema quando os módulos de quantização vetorial e HMM foram substituídos por um classificador neural. O melhor desempenho foi obtido com o classificador neural, utilizando como entrada a concatenação dos vetores de características perceptivas, globais e direcionais.

Os resultados obtidos com os testes realizados com a base de caracteres IRONOFF indicaram que a melhor estratégia para classificar os caracteres isolados é utilizar os vetores de características perceptivas, globais e direcionais concatenados com o processo de zoneamento e redes neurais para realizar a classificação, sem as etapas de normalização e segmentação. Usando HMM como classificador a estratégia que melhor se adequa é a concatenação das observações produzidas pelos dicionários dos três vetores de características avaliados e HMM com as observações nos arcos e a topologia HMM T16.

Para finalizar, foi avaliado o desempenho do sistema utilizando a base de palavras manuscritas dependente do usuário que é o objetivo central deste trabalho. Esta base de dados foi construída, durante o decorrer do trabalho, com palavras redigidas sem que o escritor tivesse qualquer conhecimento que seus textos seriam utilizados pelo sistema. Além disso, os textos coletados foram, na maioria das vezes, produzidos como anotações sem qualquer cuidado na caligrafia, o que dificulta o processo de reconhe-

cimento. Além dos testes com as palavras, foram realizados testes com os caracteres segmentados e rotulados manualmente. As taxas de reconhecimento para os caracteres isolados variaram entre 64,28% a 82,39% utilizando HMM. Quando o classificador utilizado foi NN a taxa de reconhecimento para o escritor 1 foi de 92,96%, superior aquela obtida com HMM que foi 82,39%. Entretanto, quando o sistema foi utilizado para reconhecer palavras manuscritas (objetivo do mesmo) a taxa de reconhecimento com HMM (92,96%), foi superior à obtida com NN (87,51%). O sistema desenvolvido obteve taxas de reconhecimento (com o classificador HMM) que variaram entre 83,31% a 92,96% dependendo do escritor analisado. Os resultados apresentados mostram que o sistema não é adequado para reconhecer caracteres isolados, mais apresenta um ótimo desempenho quando utilizado para reconhecer palavras através dos modelos dos caracteres.

Capítulo 6

Conclusão

A análise de imagens de documentos é um excitante campo de pesquisa que tem atraído interesse de pesquisadores de diversas áreas científicas e tecnológicas. Nesse contexto, o reconhecimento de palavras manuscritas tem despertado bastante interesse devido as suas inúmeras aplicações práticas. Entretanto, devido à grande variedade de formas apresentada pela escrita manual, os sistemas desenvolvidos na prática limitam o dicionário de interesse, ou seja o conjunto de palavras a ser discriminado pelo sistema. Este trabalho foi desenvolvido seguindo uma linha de raciocínio contrária, ou seja, ao invés de limitar o dicionário de interesse, limita-seo número de escritores. O sistema é ensinado a aprender a caligrafia do seu usuário, baseado na hipótese de que o reconhecimento pode ser melhorado se a caligrafia do autor for conhecida a priori.

Neste intuito, foi desenvolvido um sistema de reconhecimento de palavras manuscritas dependente do escritor, que consiste de seis módulos (pré-processamento, segmentação, gerador de sub-imagens, extração de características, quantização vetorial e modelagem dos caracteres utilizando Modelos Escondidos de Markov). Este sistema segmenta as palavras em sub-imagens com base em dois tipos de informações: localização dos pontos de mínimos do contorno da palavra e dos perfis de projeção da palavra. Cada sub-imagem gerada é representada pelos vetores de características perceptivas, globais e direcionais, sendo o vetor de características direcionais o mais representativo, considerando individualmente, no reconhecimento de caracteres. O mapeamento

dos vetores de características pelos vetores de observação é realizado pelo módulo de quantização vetorial utilizando dicionários de reconstrução projetados pelo algoritmo tradicional LBG. Sendo assim, cada sub-imagem é representada por três símbolos que representam os três espaços de características utilizados. As palavras escritas por usuários cadastrados do sistema são reconhecidas de acordo com modelos de caracteres específicos do usuário. Esta classificação é realizada por meio da determinação de medida de similaridade entre as seqüências de observações de uma dada palavra e todos os modelos de caracteres. Estes modelos são obtidos durante a fase de treinamento, por meio do cálculo da probabilidade de que cada modelo represente as ocorrências associadas às seqüências de observações da palavra. O resultado final do processo de reconhecimento é a palavra do vocabulário que possui a maior probabilidade de gerar a seqüência analisada.

Para validar o sistema desenvolvido foram realizados diversos experimentos que comprovaram a eficiência do classificador. Foram realizados testes com caracteres isolados manualmente das palavras contidas na base de palavras dependente do escritor e com as palavras propriamente ditas. Além destes, foram realizados testes comparando o desempenho do sistema quando a classificação dos caracteres é realizada por HMM e NN. Os resultados dos testes confirmaram que a utilização do classificador HMM aumenta o desempenho do sistema de reconhecimento de palavras desenvolvido. A taxa de reconhecimento das palavras produzidas pelo escritor 1 com o classificador HMM tem um ganho de $\frac{92{,}96-87{,}51}{92{,}96} = 5,82\%$ em relação ao classificador NN.

Podem ser citadas as seguintes contribuições originais do trabalho:

- Construção de uma base de dados contendo amostras de palavras manuscritas escritas por quatro escritores diferentes, servindo como suporte ao desenvolvimento deste trabalho, bem como a outros a serem realizados no futuro;

- Desenvolvimento de técnicas de normalização da inclinação vertical dos caracteres;

- Modificação na utilização do algoritmo de normalização da linha de base da pa-

lavra, aplicando-o no contorno inferior da palavra, o que melhora a eficiência na determinação do ângulo de inclinação da linha de base e no tempo de processamento da imagem;

- Desenvolvimento de três algoritmos de segmentação de palavras manuscritas que juntos visam dividir a palavra em segmentos. Cada caractere da palavra poderá ser dividido em no máximo cinco partes, desta forma um segmento poderá ser um completo caractere ou até 1/5 de um completo caractere;

- Avaliação comparativa dos algoritmos de segmentação desenvolvidos com algoritmos propostos na literatura com a finalidade de determinar o mais eficiente;

- Concepção da arquitetura dos Modelos Escondidos de Markov para modelagem de caracteres utilizando os espaços de características perceptivas, globais e direcionais;

- Integração dos algoritmos e modelos desenvolvidos em um sistema de reconhecimento *off-line* de palavras manuscritas dependente do usuário;

- Comparação de classificadores de caracteres, usando NN e HMM, na tarefa de reconhecer caracteres isolados e no reconhecimento de palavras manuscritas.

Estas contribuições originaram diversas publicações em eventos científicos nacionais e internacionais [27, 28, 109, 110, 125]. As contribuições mais recentes serão utilizadas para novas publicações em revistas internacionais da área.

Na seqüência, são apresentadas algumas sugestões para continuidade do trabalho ora apresentado:

- Ampliação da base de dados. Inserindo mais amostras de palavras, principalmente de palavras que contenham caracteres com pouca representatividade na base atual, melhorando assim a sua modelagem;

- Desenvolvimento de algoritmos adicionais à técnica de normalização da linha de base que permitam realizar a correção da linha de base por partes de acordo com a inclinação individual de cada parte;

- Análise de outros espaços de características que possam trazer melhorias à representação das sub-imagens, visando melhorar a modelagem dos caracteres;
- Análise de outras técnicas para projeto de dicionários para o algoritmo de quantização;
- Inclusão de uma etapa de pós- processamento ao sistema desenvolvido para trabalhar com restrições ortográficas da língua, eliminado a necessidade de limitar o vocabulário;
- Inclusão de uma etapa de identificação do autor antes do processo de normalização, extraindo características de cada autor para auxiliar na identificação da autoria do manuscrito. A partir daí selecionar de forma automática os dicionários e os Modelos Escondidos de Markov particulares do autor.

Bibliografia

[1] Fechine, J. M. Reconhecimento automático de identidade vocal utilizando modelagem híbrida paramétrica e estatística. *Tese de Doutorado, Universidade Federal da Paraíba*, 2000.

[2] Suen, C., Kim, J., Kim, K., Xu, Q. and Lam, L. Handwriting Recognition - The Last Frontiers. *Proceedings of the International Conference on Pattern Recognition - ICPR'00*, pp. 1–10, September 2000.

[3] Sabourin, R. . Off-Line Signature Verification: Recent Advances and Perspectives. *Proceedings of the First Brazilian Symposium in Document image analysis - BSDIA'97*, pp. 84–98, November 1997.

[4] Kasturi, R. and Luo, H. . Research Advances in Graphics Recognition: An Update. *Proceedings of the First Brazilian Symposium in Document image analysis - BSDIA'97*, pp. 99–110, November 1997.

[5] Belaid, A. . Future Trends in Retrospective Document Conversion. *Proceedings of the First Brazilian Symposium in Document image analysis - BSDIA'97*, pp. 111–128, November 1997.

[6] Mantas, J. . An Overview of Character Recognition Methodologies. *Pattern Recognition, Vol. 19, No. 6*, pp. 425–430, 1986.

[7] Srihari, S. N. . High-Performance. *Proceedings of IEEE, Vol. 7, No. 7*, pp. 1120–1132, 1992.

[8] Kimura, F. and Shridhar, M. . Handwritten Numerical Recognition Based on Multiple Algorithms . *Pattern Recognition, Vol. 24, No. 10*, pp. 969–983, 1991.

[9] Ezaki, N. and Hikichi, T. A Pen-Based Japanese Character Input system for Blind Person. *Proceedigns of the International Conference on Pattern Recognition - ICPR'00*, pp. 372–375, September 2000.

[10] Gomes, H. M. Investigação de Técnicas Automáticas para o Reconhecimento Off-Line de Assinaturas. *Dissertação de Mestrado, Universidade Federal de Pernambuco*, 1995.

[11] Lu, Y. and Shridhar,M. . Character Segmentation in Handwritten Words - An Overview . *Pattern Recognition, Vol. 29, No. 1*, pp. 77–96, 1996.

[12] Suen, C. Y., Nadal, C., Legault, R., Mai, T. A. and Lam, L. . Computer Recognition of Unconstrained Handwritten Numerals . *Proceedings of the IEEE, Vol. 80, No. 7*, pp. 1162–1180, July 1992.

[13] Lu, Y. . Machine printed Character Segmentation - An Overview . *Pattern Recognition, Vol. 28, No. 1*, pp. 67–80, 1995.

[14] Gomes, N. R. Algoritmo Sequencial para reconhecimento de Numerais Manuscritos Desconectados Utilizando Redes Neurais. *Dissertação de Mestrado, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica - Departamento de Computação*, 1996.

[15] Senior, A. W. . *Off-Line Cursive Handwritting Recognition Using Recurrent Neural Networks* . PhD thesis, University of Cambridge, England, 1994.

[16] Oliveira Jr., J. J. Reconhecimento de palavras manuscritas usando análise multivistas. *Tese de Doutorado, Universidade Federal de Campina Grande*, 2006.

[17] Hu, J., Brown, M. and Turin, W. HMM Based On-Line Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.18, No.10*, pp. 1039–1044, October 1996.

[18] Plamondon, R. and Srihari, S. N. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survery. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, No.1*, pp. 63–84, January 2000.

[19] R. Seiler, M. Schenkel and F. Eggimann. Off-line cursive handwriting recognition compared with on-line recognition. In *In Proc. Int. Conference on Pattern Recognition (ICPR*, pp. 505–509, 1996.

[20] Tappert, C.C., Suen, C. Y. and Wakahara, T. The state of art in on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 787–808, 1990.

[21] Suen,C. T., Berthod,M. and Mori, S. Automatic Recognition of Handprinted Characters - The State of the Art. *Proceedings of IEEE, Vol. 68, No. 4*, pp. 469–487, April 1980.

[22] Koerich, A.L., Sabourin, R. and Suen, C.Y. Recognition and verification of unconstrained handwritten words. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1509–1522, 2005.

[23] Koerich, A. L., Sabourin, R. and Suen, C. Y. Large vocabulary off-line handwriting recognition: A survey. *Pattern Analysis and Applications*, 6:97–121, 2003.

[24] Correia, S. E. N. Reconhecimento de Caracteres Numéricos Manuscritos usando Transformada *Wavelet. Dissertação de Mestrado, Universidade Federal da Paraíba - Departamento de Engenharia Elétrica*, 2000.

[25] Correia, S. E. N. and Carvalho, J. M. Handwritten Numerical Characteres Recognition Using Biorthogonal Spline Wavelets. *CD-ROM of the XVIII Brazilian Telecommunications Symposium (SBrT'2000), Gramado – RS, Brazil*, September 2000.

[26] Veloso, L.R. Reconhecimento de Carateres Numéricos Manuscritos. *Dissertação de Mestrado, Universidade Federal da Paraíba - Departamento de Engenharia Elétrica*, 1998.

[27] Veloso, L. R. and Carvalho, J. M. . Neural versus Syntactic Recognition of Handwritten Numerals. *In Proceedings of the V International Conference on Document analysis and Recognition - ICDAR'99*, September 1999.

[28] Veloso, L. R. and Carvalho, J. M. Reconhecimento de Caracteres Numéricos Manuscritos Utilizando Redes Neurais. *In Proceedings of VIII RPIC,VIII Reunión de Trabajo en Processamiento de la Informactíon*, September 1999.

[29] Correia, S. E. N. and Carvalho, J. M. . Optimizing the Recognition Rates of Unconstrained Handwritten Numerals Using Biorthogonal Spline Wavelets. *Proceedings of the ICPR, Barcelona - Espanha*, September,2000.

[30] Oliveira, L.S. *Automatic Recognition of Handwritten Numerical Strings.* PhD thesis, University of Quebec, ETS, Montreal, QC, Canada, 2003.

[31] Oliveira, L. S., Sabourin, R., Bortolozzi, F. and Suen, C. Y. . A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17:903–930, 2003.

[32] Oliveira, L. S., Morita, M. and Sabourin, R. Feature selection for ensembles applied to handwriting recognition. *Int. J. Doc. Anal. Recognit.*, 8(4):262–279, 2006.

[33] Vellasques, E., Oliveira, L. S., Britto Jr., A., Koerich, A. and Sabourin, R. Filtering segmentation cuts for digit string recognition. *Pattern Recognition*, 41 (10):3044–3053, 2008.

[34] Oliveira L. S., Sabourin R., Bortolozzi F. and Suen C.Y. Impacts of verification on a numeral string recognition system. *Pattern Recognition Letters*, 24(7):1023–1031, 2003.

[35] El-Yacoubi, A., Gilloux, M., Sabourin, R. and Suen, C. Y. 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, August 1999.

[36] Chen, M.-Y., Kundu, A. and Zhou, J. Off-Line Handwritten Word Recognition Using Hidden Markov Model Type Stochastic Network . *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.16 , No. 5*, pp. 481–496, May 1994.

[37] Chen, M.-Y., Kundu, A., Zhou, J. and Srihari, S. N. Off-line handwritten word recognition using hidden markov models. *Proceedings of the Advanced Technology Conference - United States Postal Service, Vol. 1*, 1992.

[38] Chen, M.-Y., Kundu, A. and Srihari, N. . Variable Duration Hidden Markov Model and Morphological Segmentation for Handwritten Word Recognition. *IEEE Transactions on Image Processing, Vol. 4, No. 12*, pp. 1675 – 1687, December 1995.

[39] Kundu, A., He, Y. and Chen, M.-Y. . Alternatives to Variable Duration HMM in Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 11*, pp. 1275–1280, November 1998.

[40] Bunke, H., Roth, M. and Schukatt-Talamazzini, E.G. Off-line Cursive Handwriting Recognition using Hidden Markov Models. *Relatório Técnico,IAM-94-008,Institut für Informatik und angewandle Mathematic, Universität Bern*, 1994.

[41] Mohamed, M. A. and Gader, P. Generalized hidden markov models - part ii: Application to handwritten word recognition. *IEEE Trans. on Fuzzy Systems*, 8(1):82–94, 2000.

[42] Mohamed, M. A. and Gader, P. Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modeling and Segmentation-Based Dynamic Programming Techniques. *Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 5*, pp. 548–554, May 1996.

[43] Kundu, M. and Bahl, P. Recognition of handwritten script: A hidden markov model based approach. *Relatório Técnico*, 1988.

[44] Gilloux, M., Leroux, M. and Bertille, J.-M. . Strategies for Hanwritten Words Recognition Using Hidden Markov Models. *Proceedings of the International Conference on Document Analysis and Recognition - ICDAR'93*, pp. 299–304, 1993.

[45] Park, H.-S. and Lee, S.-G. . A Truly 2-D Hidden Markov Model for Off-Line Handwritten Character Recognition. *Pattern Recognition, Vol. 31, No. 12*, pp. 1849–1864, 1998.

[46] Guillevic, D. . *Unconstrained Handwriting Recognition Applied to the Processing of Bank Cheques* . PhD thesis, Concordia University, 1995.

[47] Freitas, C., El-Yacoubi, A., Bortolozzi, F. and Sabourin, A. Brazilian Bank Check Handwritten Legal Amount Recognition. *Anais do Simpósio Brasileiro de Computação Gráfica e Processamento de imagens- SIBGRAPI'2000*, 2000.

[48] Senior, A. W. and Robinson, A. J. . An Off-Line Cursive Handwriting Recognition System. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3*, pp. 309–321, March 1998.

[49] Gader, P., Mohamed, M. and Chiang, J.-H. Handwritten Word Recognition with Character and Inter-Character Neural Networks. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, Vol. 27, No. 1*, pp. 158–164, February 1997.

[50] Gader, P., Mohamed, M. and Chiang, J.-H. . Comparison of Crisp and Fuzzy Character Neural Networks in Handwritten Word Recognition. *IEEE Transactions on Fuzzy Systems, Vol. 3, No. 3*, pp. 357–363, August 1995.

[51] Chiang, J.-H. and Gader, P. . Hybrid Fuzzy-Neural Systems in Handwritten Word Recognition. *IEEE Transactions on Fuzzy Systems, Vol. 5, No. 4*, pp. 497–509, November 1997.

[52] Kim, G. and Govindaraju, V. A Lexicon Driven Approach to Handwritten Word Recognition for Real-Time Applications . *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 4*, pp. 366–379, April 1997.

[53] Rabiner, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE ,Vol.77 , No. 2*, pp. 257–286, February 1989.

[54] Rabiner,L. and Juang, B.-H. *Fundamentals of speech recognition.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[55] Côté, M., Lecolinet, E., Cheriet, M. and Suen, C. Y. . Automatic reading of cursive scripts using a reading model and perceptual concepts . *International Journal on Document Analysis and Recognition, Vol. 1*, pp. 3–17, 1998.

[56] El-Yacoubi, A., Gilloux, M., Sabourin, R. and Suen, C. Y. Objective Evaluation of the Discriminant Power of Features in an HMM-Based Word Recognition System. *Proceedings of the First Brazilian Symposium*, pp. 60–73, November 1997.

[57] Buse, R., Liu, Z.-Q. and Caelli, T. . A Structural and Relational Approach to Handwritten Word Recognition. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, Vol. 27, No. 5*, pp. 847–861, October 1997.

[58] Kimura F., Shridhar M. and Chen Z. Improvements of a lexicon directed algorithm for recognition of unconstrained handwritten words. In *Proc. International Conference on Document Analysis and Recognition*, pp. 18–22, 1993.

[59] El-Yacoubi, A., Gilloux, M., Sabourin, R. and Suen, C. Y. Unconstrained handwritten word recognition using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993.

[60] Bozinovic, R. M., and Shihari, S. N. Off-Line Cursive Script Word Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.11, No.1*, pp. 68–83, January 1989.

[61] Kim, G. *Recognition of Off-Line Handwritten Words and Its Extension to Phrase Recognition.* PhD thesis, State University of New York at Buffalo, 1996.

[62] Kim, G. and Govindaraju, V. . Handwritten Phrase Recognition as Applied to Street Name Images. *Pattern Recognition, Vol. 31, No. 1*, pp. 41–51, 1998.

[63] El Yacoubi, A. *Modélisation Markovienne de L'Écriture Manuscrite Application à la Reconnaissance des Adresses Postales.* PhD thesis, ' Universite De Rennes 1, 1996.

[64] Ding, Y., Kimura, F., Miyake, Y. and Shridhar, M. Accuracy Improvement of Slant Estimation for Handwritten Words. *Proceedings of the Internacional Conference on Pattern Recognition - ICPR'00*, pp. 527–530, 2000.

[65] Slavík, P. and Govindaraju, V. Equivalence of Different Methods for Slant and Skew Corrections in Word Recognition Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.23, No.3*, pp. 323–326, March 2001.

[66] Parker, J. R. *Algorithms For Image Processing and Computer Vision.* Wiley Computer Publishing, 1997.

[67] Facon, J. *Morfologia Matemática: Teoria e Exemplos.* PUC-PR, 1996.

[68] Frucci, M. and Marcelli, A. . Contour Pixel Classification for Character Skeletonization. *Proceedings of the First Brazilian Symposium*, pp. 141–152, November 1997.

[69] Borgefors, G., Ramella, G. and Baja, G. S. . Using Binary Pyramids to Create Multi-resolution Shape Descriptors. *Proceedings of the First Brazilian Symposium*, pp. 129–140, November 1997.

[70] Gonzalez, R. C. and Woods, R. E. . *Digital Image Processing.* Addison-Wesley, 1992.

[71] Lam, L. and Suen, C. Y. . An Evaluation of Parallel Thinning Algorithms for Character Recognition. *Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 9*, pp. 914–919, 1995.

[72] El-Yacoubi, A., Sabourin, R., Gilloux, M., and Suen. Off-Line Handwritten Word Recognition Using Hidden Markov Models. *Relatório técnico.*

[73] Morita, M. E. Estado da Arte das Principais Etapas de Sistemas de Reconhecimento de Manuscritos. *Relatório Técnico, ppgia-003, Pontifícia Universidade Católica do Paraná*, 1999.

[74] Casey, R.G. and Lecolinet E. . A Survey of Methods and Strategies in Character Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 7*, pp. 690–706, 1996.

[75] Favata, J.T. *Recognition of Cursive, Discrete and Mixed Handwritten Words Using Character, Lexical and Spatial Constraints.* PhD thesis, State University of New York at Buffalo, 1993.

[76] Gader, P., Whalen, M., Ganzberger, M. and Hepp, D. . Handprinted Word Recongition on a NIST Data Set. *Machine Vision and Application, Vol.8*, pp. 31–40, 1995.

[77] Yanikoglu,B. and Sandon, P. A. Segmentation of Off-line Cursive Handwriting Using Linear Programming. *Pattern Recognition, Vol. 31, No. 12*, pp. 1825–1833, December 1998.

[78] Mohamed, M. A. and Gader, P. Generalized hidden markov models - part i: Theoretical frameworks. *IEEE Trans. on Fuzzy Systems*, 8(1):67–81, 2000.

[79] Kim, J. H., Kim, K. K., Nadal, C. and Suen, C. A methodology of Combining HMM and MLP Classifiers for Cursive Word Recognition. *Proceedings of the Internacional Conference on Pattern Recognition - ICPR'00*, pp. 319–322, 2000.

[80] *Recent Advances in Handwriting Recognition*, volume 1, chapter Document Analysis, pp. 1–30. Umapada Pal. Swapan K. Parui. Bidyut B Chaudhuri. (Org.). Chennai: ALLIED PUBLISHERS PVT. LTD., 2005.

[81] Koerich, A. L., Leydier, A. Y., Sabourin, R. and Suen, C. Y. . A hybrid large vocabulary handwritten word recognition system using neural networks with hidden markov models. In *Proc. 8th International Workshop on Frontiers in Handwriting Recognition*, pp. 99–104, Niagara-on-the-Lake, Canada, 2002.

[82] Kessentini, Y., Paquet, T. and Benhamadou, A. A multi-stream approach to off-line handwritten word recognition. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 1*, pp. 317–321, Washington, DC, USA, 2007. IEEE Computer Society.

[83] Freitas, C. O. A., Carvalho, J. M., Oliveira Júnior, J. J., Aires, S. and Sabourin, R. Confusion matrix disagreement for multiple classifiers. *In: 12th Iberoamerican Congress on Pattern Recognition*, 1:387–396, 2007.

[84] Oliveira Jr., J. J. de, Kapp, M. N., Freitas, C. O. de A. , Carvalho, J. M. de and Sabourin, R. Handwritten recognition with multiple classifiers for restricted lexicon. In *SIBGRAPI '04: Proceedings of the Computer Graphics and Image Processing, XVII Brazilian Symposium*, pp. 82–89, Washington, DC, USA, 2004. IEEE Computer Society.

[85] Nunes, C. M., Britto, Ad. S., Jr.; Kaestner, C. A. A.; Sabourin, R. An optimized hill climbing algorithm for feature subset selection: evaluation on handwritten character recognition. In *Ninth International Workshop on Frontiers in Handwriting Recognition*, 2004.

[86] Morita, M., Oliveira, L.S. and Sabourin, R. Geração automática de conjuntos de classificadores através da seleção de características não supervisionada. *IEEE Latin America Transactions*, 3(5):50–56, 2005.

[87] Xu, L., Krzyzak, A. and Suen, C.,Y. Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Vol.22, No.3*, pp. 418–435, May/June 1992.

[88] Brakensiek, A., Kosmala, A., Willet, D., Wang, W. and Rigoll G. . Performance Evaluation of a New Modeling Technique for Handwriting Recognition Using Identical On-Line and Off- Line Data. *Proceedings of the V International Conference on Document Analysis and Recognition - ICDAR'99, Bangalore, India*, 1999.

[89] Dehghan, M., Faez, K., Ahmadi, M. and Shridhar, M. Off-Line unconstrained Farsi Handwritten Word Recognition Using Fuzzy Vector Quantization and Hidden Markov Word Models. *Proceedings of the International Conference on Pattern Recognition - ICPR'00*, pp. 351–354, September 2000.

[90] Knerr, S. and Augustian, E. A Neural Network-Hidden Markov Model Hybrid for Cursive Word Recognition. *Proceedings of the International Conference on Pattern Recognition - ICPR'98, Vol. 2*, pp. 1518–1520, 1998.

[91] Ko A., Sabourin R., Britto Jr. A. and Oliveira L. S. Pairwise fusion matrix for combining classifiers. *Pattern Recognition*, 40(8):2198–2210, 2007.

[92] Oliveira, L. S. Sabourin, R., Bortolozzi, F. and Suen, C. Y. Feature selection for ensembles : A hierarchical multi-objective genetic algorithm approach. In *In*

Proc. of 7 th International Conference on Document Analysis and Recognition, Edinburgh-Scotland, 2003. IEEE Computer Society, pp. 676–680, 2003.

[93] Morita, M., Oliveira, L. S. and Sabourin, R. Unsupervised feature selection for ensemble of classifiers. In *In 9th International Workshop on Frontiers in Handwriting Recognition*, pp. 81–86, 2004.

[94] Liu, C.-L. and Marukawa, K. Normalization ensemble for handwritten character recognition. *International Workshop on Frontiers in Handwriting Recognition,*, 0:69–74, 2004.

[95] Srihari, S. N., Xu, A. and Kalera, M. Learning strategies and classification methods for off-line signature verification. *International Workshop on Frontiers in Handwriting Recognition*, 0:161–166, 2004.

[96] Gillies, A.M. Cursive Word Recognition Using Hidden Markov Models. *Proceedings of the Advanced Technology Conference - United States Postal Service, Vol. 1*, 1992.

[97] Wang, W., Brakensiek, A., Kosmala and Rigoll G. . HMM Based High Accuracy Off-Line Cursive Handwriting Recognition by a Baseline Detection Error Tolerant Feature Extraction Approach. *Proceedings of the IWFHR, Amsterdam, The Netherlands*, 2000.

[98] Brakensiek, A., Rottland, A., Kosmala, A., and Rigoll G. . Off-Line Handwriting Recognition Using Various Hybrid Modeling Techniques and Character N-Grams. *Proceedings of the IWFRH, Amsterdam, The Netherlands*, 2000.

[99] Guillevic, D. and Suen, C. Y. HMM Word Recognition Engine. *Proceedings of the IV International Conference on Document Analysis and Recognition - ICDAR'97, Ulm , Germany*, pp. 544–547, 1997.

[100] Kim, G., and Kim, S. . Feature Selection Using Genetic Algorithms for Handwritten Character Recognition. *Proceedings of the IWFHR, Amsterdam, The Netherlands*, 2000.

[101] Arica, N. and Yarman-Vural, F. T. Optical character recognition for cursive handwriting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):801–813, 2002.

[102] Koch, G., Paquet, T. and Heutte, L. Combination of contextual information for handwritten word recognition. In *IWFHR 04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, pp. 468–473, Washington, DC, USA, 2004. IEEE Computer Society.

[103] Koerich, A. and Sabourin, R. and Suen, C. Y. Fast two-level hmm decoding algorithm for large vocabulary handwriting recognition. In *Proc. Ninth Workshop Frontiers in Handwriting Recognition*, pp. 232–237, 2004.

[104] Freitas, C. O. A., Carvalho, J. M., Oliveira Júnior, J. J., Aires, S. and Sabourin, R. Distance-based disagreement classifiers combination. In *Proceedings of the 20th International Joint Conference on Neural Networks*, 2007.

[105] Oliveira Jr., J. J., Carvalho, J. M., Freitas, C. O. A. and Sabourin, R. Feature sets evaluation for handwritten word recognition using a baseline system. In *Proceedings of International Workshop on Frontiers in Handwritting Recognition*, volume 1, pp. 446–451, 2002.

[106] Oliveira Jr., J. J. ; Carvalho, J. M. ; Freitas, C. O. A. ; Sabourin, R. Evaluating nn and hmm classifiers for handwritten word recognition. In *Proceedings of Brazilian Symposium on Computer Graphics and Image Processing*, volume 1, pp. 210–217, 2002.

[107] Morita, M. *Automatic Recognition of Handwritten Dates on Brazilian Bank Cheques.* PhD thesis, University of Quebec, ETS, Montreal, QC, Canada, 2003.

[108] Shetty, S., Srinivasan, H. and Srihari, S. Handwritten word recognition using conditional random fields. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, pp. 1098–1102, Washington, DC, USA, 2007. IEEE Computer Society.

[109] Veloso, L.R., Sousa, R. P. and Carvalho, J. M. . A New Method for Cursive Word Segmentation. *Proceeding of IEEE International Conference on Image Processing - ICIP'2000*, September 2000.

[110] Veloso, L.R. e Carvalho, J.M. Segmentação de Escrita Cursiva por Histograma de Projeção Vertical. *CD-ROM of the XVIII Brazilian Telecommunications Symposium (SBrT'2000), Gramado – RS, Brazil*, September 2000.

[111] Madeiro,F. Projeto e Avaliação de Dicionários para a Quantização Vetorial de Voz e Imagem. *Proposta de Tese, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, 2001.

[112] Linde, Y., Buzo, A., and Gray, R. M. An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications, Vol.28 , No.1*, pp. 84–95, January 1980.

[113] Gray, R. M. . Vector Quantization. *IEEE ASSP Magazine*, pp. 4–29, April 1984.

[114] Harvey, N. R. and Marshall, S. . The use of Genetic Algorithms in Morphological Filter Design. *Signal Processing: Image Communication, Vol. 8*, pp. 55–71, 1996.

[115] Grandidier, F., Sabourin, R., El-Yacoubi, A., Gilloux, M. and Suen, C. Influence of word length on handwriting recognition, 1999.

[116] Gersho, A. and Gray, R. M. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Boston, MA, 1992.

[117] Madeiro, F. Projeto e avaliação de dicionários para a quantização vetorial de voz e imagem. *Tese de Doutorado, Universidade Federal da Paraíba*, 2001.

[118] Freitas, C. O. A. Uso de modelos escondidos de markov para reconhecimento de palavras manuscritas. *Tese de doutorado, Pontifícia Universidade Católica do Paraná, PUC-PR, Brasil*, 2001.

[119] Koerich, A.L. Reconhecimento de palavras manuscritas usando modelos de markov. *Revista IEEE América Latina*, 2:132–141, 2004.

[120] Viard-Gaudin, C. The ironoff user manual. In *IRESTE, University of Nantes, France*, 1999.

[121] HP ScanJet 5200c Series. *http://www.pandi.hp.com/pandi-db/prodinfo.main?-product=scanjet5200c&Region=non_us.*

[122] Image File Formats List. http:// www.cedar.buffalo.edu/Penman/description.html.

[123] Freitas, C. O. A., Oliveira, L. S., Aires, S. B. K. and Bortolozzi, F. Zoning and metaclasses for character recognition. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pp. 632–636, New York, NY, USA, 2007. ACM.

[124] Kapp, M. N., Freitas, C. O. A. and Sabourin, R. Handwritten brazilian month recognition: An analysis of two nn architectures and a rejection mechanism. In *IWFHR 04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, pp. 209–214, Washington, DC, USA, 2004. IEEE Computer Society.

[125] Oliveira Jr.,J.J., Veloso, L.R. and Carvalho, J.M. . Interpolation/Decimation Scheme Applied to Size Normalization of Characters Images. *Proceedings of the International Conference on Pattern Recognition - ICPR'00*, pp. 577–580, September 2000.