

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA
DA COMPUTAÇÃO**

Andrea Vergara da Silva

**UMA APLICAÇÃO DA TRANSFORMADA
WAVELET À VERIFICAÇÃO ON-LINE DE
ASSINATURAS MANUSCRITAS**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

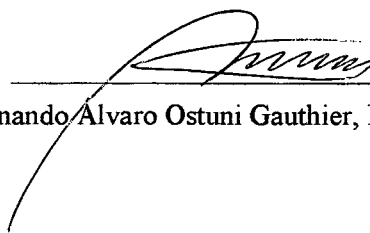
**Prof. Daniel Santana de Freitas, Dr.
Orientador**

Florianópolis, outubro de 2002

UMA APLICAÇÃO DA TRANSFORMADA WAVELET
À VERIFICAÇÃO ON-LINE
DE ASSINATURAS MANUSCRITAS

Andrea Vergara da Silva

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, Área de Concentração Sistemas de Computação, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

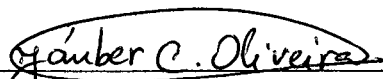


Fernando Alvaro Ostuni Gauthier, Dr. (Coordenador)

Banca Examinadora



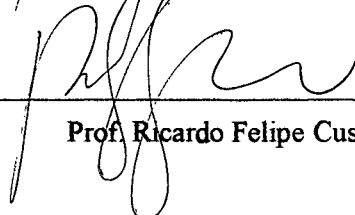
Daniel Santana de Freitas, Dr. (Orientador)



Prof. Jáuber Cavalcante de Oliveira, Ph. D.



Prof. Ricardo Dahab, Ph. D.



Prof. Ricardo Felipe Custódio, Dr.

Agradecimentos

Ao meu marido, Valmir, por todos os momentos em que esteve ao meu lado, sendo fonte das minhas maiores alegrias.

À minha família, em especial aos meus pais, Violeta e Luiz, pela formação e suporte que sempre me deram.

Ao meu orientador, Prof. Daniel, pela ajuda, compreensão e tempo dispensados.

Aos professores do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, em especial aos professores Sérgio Peters, Jáuber Cavalcante de Oliveira e Luiz Carlos Zancanella.

Às funcionárias do Curso de Pós-Graduação em Ciência da Computação, Verinha e Valdete.

À sociedade brasileira, pela manutenção da UFSC pública, gratuita e de excelente qualidade e pelo investimento público concedido através da CAPES, o qual viabilizou o desenvolvimento deste trabalho.

Sumário

1. Introdução	1
2. Revisão Bibliográfica	7
2.1 Métodos Indiretos	7
2.2 Métodos Diretos	12
2.2 Métodos Diretos com Transformadas	15
3. Fundamentos Teóricos	23
3.1 Wavelets	23
3.2 Wavelets como um Espaço Vetorial	25
3.3 Ortogonalidade das Wavelets	29
3.4 Multiresolução	30
3.5 Transformada Wavelet como um Banco de Filtros	31
3.6 Dynamic Time Warping (DTW)	33
3.7 DTW Não Recursivo	39
4. Descrição do Sistema	42
4.1 Metodologia Geral	43
4.2 Aquisição de Dados	44
4.3 Pré-processamento	44
4.3.1 Amostragem	44
4.3.2 Normalização	50
4.4 Extração de Características	55
4.5 Cálculo das Medidas de Dissimilaridade	58
4.6 Cadastramento	61
4.7 Verificação	63
5. Resultados	65
5.1 Resultados com Wavelets	65
5.2 Resultados sem Wavelets	70
6. Conclusão	74
Referências Bibliográficas	75
Apêndice A	78
Apêndice B	81

Lista de Figuras

Figura 1.1: Comportamento das taxas de erro	4
Figura 1.2: Projetos do Laboratório de Segurança em Computação	6
Figura 1.3: Esquema de encirptação de documento com uso da assinatura	6
Figura 2.1: Exemplo de caracteres Tamil	18
Figura 3.1: Exemplo de <i>wavelet</i> mãe	24
Figura 3.2: Base de Haar para V^2	26
Figura 3.3: Representação gráfica de $I(x)$	27
Figura 3.4: <i>Wavelets</i> de Haar para W^1	28
Figura 3.5: Representação intermediária de $I(x)$	28
Figura 3.6: Representação alternativa de $I(x)$	28
Figura 3.7: Decomposição tempo-freqüência para <i>wavelet</i> e Fourier	31
Figura 3.8: Bancos de filtros	32
Figura 3.9: Bancos de filtros com <i>downsampling</i>	32
Figura 3.10: Árvore de decomposição de um sinal	33
Figura 3.11: Exemplo de ajuste de curvas com DTW	34
Figura 3.12: Plano com $T_f \times T_g$ pontos.....	35
Figura 3.13: Exemplos de restrições locais	36
Figura 3.14: Plano de pontos reduzido	37
Figura 3.15: Caminhos locais com pesos	37
Figura 3.16: Plano de pontos com o início do caminho de <i>warping</i>	38
Figura 3.17: Avanço do DTW não recursivo	40
Figura 3.18: Plano de pontos reduzido pelo T_0	41
Figura 4.1:Esquema de um sistema de verificação de assinaturas	43
Figura 4.2: Prancheta Eletrônica	44
Figura 4.3: Curvas originais	45
Figura 4.4: Curvas x e y com o tempo começando em zero	46
Figura 4.5: Curva com pontos não uniformemente espaçados	46
Figura 4.6: Curva tomada como contínua	47
Figura 4.7: Curva com pontos reamostrados uniformemente espaçados	49
Figura 4.8: Curva da DFT tomada sobre o intervalo $[0,N]$	49

Figura 4.9: Curvas normalizadas quanto ao tempo	51
Figura 4.10: Efeito do primeiro passo da normalização	51
Figura 4.11: Retas que melhor aproximam as curvas $x(t)$ e $y(t)$	52
Figura 4.12: Efeito da etapa de rotação	53
Figura 4.13: Assinatura de volta ao intervalo $[-1,1]$	54
Figura 4.14: Efeito do alinhamento pelo centróide	54
Figura 4.15: Sinais $x(t)$ e $y(t)$ pré-processados	55
Figura 4.16: Curvas de aproximação e detalhe.....	57
Figura 4.17: Curvas das características 2 e 3 (padrão e verdadeira)	59
Figura 4.18: Curvas das características 2 e 3 (padrão e falsificação).....	59
Figura 4.19: Área da distorção da função de <i>warping</i>	60
Figura 4.20: Área da distorção da função de <i>warping</i>	61

Lista de símbolos

- cA coeficientes de aproximação da transformada wavelet, Figura 3.10
- cD coeficientes de detalhe da transformada wavelet, Figura 3.10
- \bar{C} comprimento médio de uma assinatura, Equação 4.19
- CAM caminho ótimo obtido com o DTW, relacionando duas curvas, item 4.5
- d_j j-ésima medida de dissimilaridade entre duas assinaturas, item 4.5
- d custo acumulado (distância) entre duas funções, Equação 3.16
- D custo acumulado entre duas funções, ponderado com pesos adequados, Equação 3.14
- Dbn referente à curva Daubechies “n”, item 5.1
- DTW *Dynamic Time Warping* – busca da melhor correspondência entre duas curvas, item 3.6
- FAR *False Acceptance Rate* – percentagem de assinaturas verdadeiras rejeitadas pelo método
- FRR *False Rejection Rate* - percentagem de assinaturas falsas aceitas pelo sistema
- L referente aos lados da região de avanço do DTW não-recursivo, Figura 3.17
- L_i limiares de teste para cada distância “i”, Equação 4.16
- Lzc limiar da quantidade de passagens pelo zero de uma curva WD, tabela 5.8
- m pesos atribuídos aos caminhos locais, Figura 3.15
- nzc quantidade de passagens pelo zero de uma curva WD
- N número de pontos em uma transformada, Equação 4.3
- Nn referente ao nível “n” de decomposição wavelet, item 5.1
- pz_k coordenada da k-ésima “passagem pelo zero” em uma curva WD
- s desvio padrão
- S sinal, item 3.5
- t coordenada tempo [ms] de cada ponto de uma assinatura, Figura 4.3
- \bar{t} coordenada tempo padronizada, Equação 4.1
- t* coordenada tempo normalizada, Equação 4.9
- T referente ao número de pontos no diagrama do DTW, Figura 3.16
- T₀ tamanho da janela de ajuste de Sakoe e Chiba, Figura 3.18
- \bar{T} tempo médio de uma assinatura, Equação 4.19

- va amplitude na mesma abscissa de pz na curva WA do nível anterior, Equação 4.13
- vi valor da integral entre duas passagens pelo zero consecutivas, Equação 4.12
- V espaço vetorial das funções de escala, itens 3.2 e 3.4
- W espaço vetorial das wavelets mãe, itens 3.2 e 3.4
- WA curva de aproximação em um certo nível, após aplicação de transformada wavelet
- WD curva de detalhe em um certo nível, após aplicação de transformada wavelet
- x coordenada horizontal de um ponto de uma assinatura, Figura 4.3
- y coordenada vertical de um ponto de uma assinatura, Figura 4.3
- z variável complexa montada a partir das coordenadas $x(t)$ e $y(t)$

Símbolos Gregos

- ϕ funções escala: servem de base para o espaço V^j , itens 3.2 e 3.4
- μ média estatística
- ψ funções wavelet, Equação 3.12
- ω frequência angular, Equação 4.5
- ζ distância simples (não-acumulada) entre pontos de duas curvas, ponderada pelo peso m , Equação 3.15

Índices

- c referente ao centróide da assinatura, Equação 4.11
- f referente à função “f”, item 3.6
- g referente à função “g”, item 3.6
- H referente à parte horizontal do L de avanço do DTW não-recursivo, Equação 3.17
- V referente à parte vertical do L de avanço do DTW não-recursivo, Equação 3.18

Resumo

Neste trabalho foi implementado um método direto que utiliza a transformada *wavelet* como o principal recurso para a verificação on-line de assinaturas manuscritas. As assinaturas, obtidas na forma de curvas $x(t)$ e $y(t)$ com o auxílio de uma prancheta digitalizadora, passam por uma etapa de pré-processamento e em seguida são submetidas a uma transformada *wavelet*. A partir destas funções transformadas, são obtidos números que caracterizam de maneira única cada assinatura. Em seguida, com base nestes números, são estabelecidas medidas de dissimilaridade (distância) entre cada assinatura a ser testada e uma assinatura padrão, previamente escolhida de uma amostra de dez assinaturas fornecidas pelo usuário durante um processo de cadastramento. Assume-se que toda nova assinatura deverá produzir distâncias até a padrão que estejam dentro de uma faixa aceitável em torno de médias verificadas no cadastramento. A determinação destas distâncias é efetuada de maneira não-linear, com o auxílio de uma técnica típica de reconhecimento de sons conhecida como "*Dynamic Time Warping*" (DTW). Esta técnica permite uma compensação em virtude de variações naturais no ritmo de escrita de cada pessoa a cada nova assinatura. Para recuperar a informação relativa à velocidade, perdida pela uniformização imposta pelo pré-processamento, é utilizada ainda uma outra medida de distância, calculada a partir do desvio das funções de *warping* para as diagonais correspondentes.

A fim de justificar a inclusão da transformada *wavelet*, foi implementado um segundo sistema de verificação, seguindo uma abordagem em que as funções $x(t)$ e $y(t)$ são diretamente confrontadas com a assinatura padrão. Neste caso, as características são as próprias coordenadas dos *pixels* que descrevem cada assinatura e as medidas de dissimilaridade passam a ser simplesmente as normas das distâncias entre as curvas, cuja correspondência entre pontos também é determinada com o auxílio do DTW. A partir dos testes desenvolvidos, foram obtidas taxas de erro aceitáveis para este tipo de biometria nos dois sistemas implementados. O bom desempenho do caso em que as *wavelets* não foram incluídas, no entanto, sugere que, em ambos os sistemas, o principal mecanismo de verificação é a comparação não-linear entre parâmetros efetuada pelo DTW.

Abstract

In this work, a complete system for on-line signature verification is implemented. With the aid of a graphics tablet, data describing a signature to be tested is obtained as two functions of time, $x(t)$ and $y(t)$. Then, these functions are treated with wavelet transforms to generate features to be nonlinearly confronted, through a warping function, with a reference signature chosen among 10 previously stored tryings from the same writer. In order to recover the time dependence lost during the wavelet treatment, the level of departure from the diagonal line in the warping function is included as a complementary measure of distance. For the sake of comparison, a second approach is devised, in which the wavelet step is removed and the functions $x(t)$ and $y(t)$ describing position in time of each pixel of the same test signature are directly (though still nonlinearly) compared to their counterparts in the reference. Both approaches were tested with original signatures from four different writers and achieved highly acceptable error rates (around 15% false acceptance and 20% false rejection) to this kind of biometry. Nevertheless, the low error rates obtained with the system implemented according to the second approach suggested that the wavelet transform was not an essential step in the process and that the warping algorithm can be considered as the main tool for on-line signature verification by direct methods.

Capítulo 1

Introdução

Tradicionalmente, a transmissão e o armazenamento de informações em nossa sociedade têm sido feitos através de documentos escritos. Ao longo do tempo, um elaborado conjunto de mecanismos foi criado para garantir a segurança das informações contidas nestes documentos. O ato de violar correspondências, por exemplo, é crime. No entanto, um volume de dados cada vez maior e equipamentos computacionais cada vez mais acessíveis possibilitaram o uso do meio eletrônico para estas tarefas. Mas, quando o meio é o papel, existe uma ferramenta de eficácia comprovada que garante a segurança da informação escrita, a assinatura. A assinatura é amplamente utilizada devido às seguintes características:

- é única para cada indivíduo;
- serve para identificar, autorizar e validar;
- é totalmente independente da informação contida no documento e
- é muito difícil de ser alterada sem que fiquem marcas no papel assinado.

Entretanto, em meio eletrônico a assinatura é, na realidade, uma seqüência de bits passível de ser copiada e alterada sem que disso restem quaisquer evidências. É necessário, em função disso, que exista uma relação de dependência entre a assinatura eletrônica e a informação contida no documento. A este tipo de assinatura dá-se o nome de assinatura digital.

A assinatura digital funciona como uma assinatura convencional mas é específica para documentos digitais. Ela utiliza técnicas de criptografia assimétrica (ver [STI95]) para garantir a origem e autenticidade de um documento recebido eletronicamente. O

processo consiste em empregar uma senha para criptografar um documento. Um documento assim codificado pode facilmente ser levado de volta à sua forma original com o auxílio de uma outra senha, dual da primeira e publicamente conhecida. Assim, o documento recebe uma assinatura digital se quem o produziu ou enviou utilizar a senha correta. Tal processo pode ser considerado, de certa forma, inseguro, haja vista que qualquer indivíduo de posse da senha, independentemente de como a obteve, pode autenticar um documento.

Essencialmente, uma autenticação pode estar baseada em: algo que o usuário sabe (senha, número de identificação pessoal, informação pessoal), algo que o usuário possui (são os chamados *tokens*, como o *smart card*), ou algo que ele é (impressão digital, formato da mão, do rosto). Para incrementar a confiabilidade do processo de assinatura digital é indicada a inclusão do segundo e terceiro tipos. O segundo tipo já é bastante utilizado, com a gravação de uma parte da senha em um *smart card*, por exemplo. O terceiro tipo se refere à biometria, uma técnica que fornece muita segurança na validação, uma vez que os dois primeiros tipos podem ser "roubados", mas uma característica pessoal não pode.

A biometria se utiliza de características pessoais que são únicas, tais como: impressões digitais, íris, formato do rosto, formato da mão, voz e assinatura. Estas duas últimas são classificadas como biometria mas são consideradas também comportamentais, pois variam de acordo com as circunstâncias onde são produzidas, gerando resultados semelhantes, mas não idênticos.

O fato de a assinatura ser uma característica comportamental representa um desafio a mais no trabalho de reconhecimento, já que qualquer método automático para verificação da autenticidade de uma assinatura deve ser flexível o bastante para levar em conta uma grande variação nos parâmetros descritivos das assinaturas verdadeiras de um indivíduo, mas também deve ser suficientemente restritivo para detectar falsificações treinadas. A escolha desta biometria como complemento a uma assinatura digital se justifica por ser o método mais bem aceito, uma vez que já é muito usual fora do meio eletrônico. As pessoas estão acostumadas a assinar papéis para fornecer autenticidade aos mesmos e também a aceitar papéis assinados como sendo autênticos. Desta forma, a assinatura não gera resistência por parte do usuário, pois não é diferente ou intrusiva, como ocorre com a maioria das biometrias [NEW00]. Além disso, o único

hardware específico necessário é uma prancheta digitalizadora, que hoje em dia é facilmente encontrada no mercado.

Verificação eletrônica de assinaturas

A verificação de uma assinatura pode ser: *off-line* ou *on-line*. O primeiro tipo ocorre quando se analisa uma imagem adquirida através da digitalização da assinatura original, que foi feita em um papel. Neste caso, possui-se apenas as características estáticas da assinatura. Tal tipo é indicado quando se deseja verificar as assinaturas feitas em documentos já assinados, como cheques, por exemplo. A verificação *off-line* consiste em um processo mais difícil por possuir menos informações disponíveis para verificação [PLA00] do que o segundo tipo. Não é possível determinar, por exemplo, a ordem em que os traços foram feitos. Na verificação *on-line* de assinaturas, é possível obter-se também características dinâmicas (temporais), tais como: velocidade média de escrita, velocidade máxima, tempo total gasto, ordem dos traços, etc.. Isto ocorre porque a assinatura é feita em uma prancheta digitalizadora, o que possibilita o armazenamento das suas coordenadas x e y em relação ao tempo ($x(t)$ e $y(t)$).

Para a verificação *on-line* de assinaturas, existem diversos métodos, os quais podem ser divididos em dois grandes grupos: diretos (por funções) e indiretos (por parâmetros). No primeiro caso a curva que caracteriza a assinatura é armazenada como uma função discreta que será comparada com a função da assinatura padrão do mesmo indivíduo, armazenada durante um cadastramento prévio. Este método simplifica a etapa de obtenção de dados, porém a comparação é mais trabalhosa e, normalmente, mais demorada do que no segundo caso. Nos métodos indiretos, não é a assinatura em si que é comparada (ela muitas vezes não é nem armazenada) e sim um conjunto de parâmetros que a representa. Este método requer maior esforço na obtenção dos dados mas, em compensação, o processo de comparação é mais simples.

Tradicionalmente, a medição do desempenho dos métodos é baseada em três taxas de erro. A taxa de erro tipo I representa a percentagem de assinaturas verdadeiras que foram rejeitadas e é conhecida como FRR (*false rejection rate*). A taxa de erro tipo II, conhecida como FAR (*false acceptance rate*), indica a percentagem de assinaturas falsas que foram aceitas como verdadeiras pelo sistema. A terceira taxa de erro deriva

das duas anteriores e é conhecida por taxa de erro global ou EER (Equal Error Rate). Todos os métodos possuem parâmetros de controle que permitem modificar o patamar em que se encontra o limiar de decisão entre aceitação e rejeição. Mudanças nestes parâmetros têm que ser cuidadosamente estudadas, pois têm efeitos opostos sobre FAR e FRR, como mostra a Figura 1.1. O ideal seria projetar o sistema de modo a buscar o ponto de equilíbrio indicado na Figura 1.1 (o ponto da EER). Mas usualmente o que define o sistema é a sua utilização. Dependendo da aplicação, não é admitida nenhuma aceitação de falsificações (FAR deve ser nula), mesmo que para isto seja necessário importunar o usuário com uma alta rejeição de suas assinaturas verdadeiras.

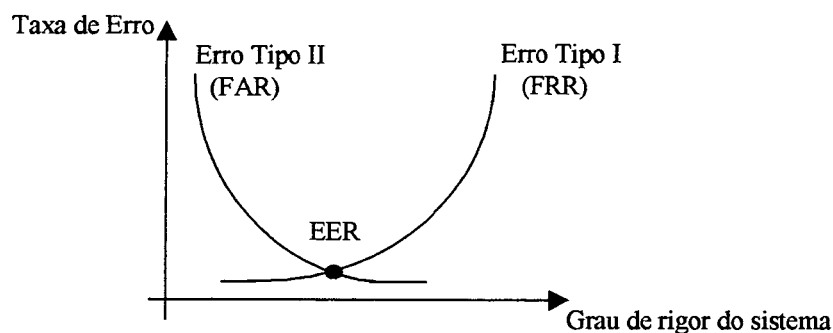


Figura 1.1: Comportamento das taxas de erro em um sistema de verificação de assinaturas.

Estas três taxas de erro constituem a forma padrão para avaliação do desempenho de um sistema de verificação de assinaturas. Embora sejam utilizadas pela maioria dos autores, tais taxas não possibilitam a comparação entre os diferentes métodos propostos. Cada autor chega a estes valores trabalhando com uma base de dados própria. As bases de dados podem diferir de um autor para outro, no número de pessoas que assinaram, no número de assinaturas recolhidas por pessoa, no número de assinaturas falsificadas, no intervalo de tempo em que as assinaturas foram coletadas e na representatividade do grupo que fez as assinaturas (se havia idosos e canhotos, por exemplo [MCC97b]).

Outro fator importante que pode gerar problemas na comparação de resultados obtidos é a definição dos tipos de falsificações, o que também apresenta diferenças de um autor para outro. Segundo Plamondon e Lorette [PLA89], existem três tipos: a falsificação simples, que ocorre quando não se possui nenhuma informação, além do nome, a respeito da assinatura a ser falsificada; a falsificação treinada, onde a informação estática da assinatura está disponível e existe um certo tempo para a prática

da mesma; o terceiro tipo é a falsificação aleatória, conhecida também como esforço-zero pois é feita escolhendo-se aleatoriamente uma outra assinatura da base de dados para ser considerada a falsificação. Este último tipo de falsificação é irreal e gera um decaimento artificial nas taxas de erro.

Presente Trabalho

Trata-se da implementação computacional de um método direto para a verificação eletrônica *on-line* de assinaturas manuscritas baseado em transformadas *wavelet*. Essencialmente, o procedimento implementado consiste em tratar as funções $x(t)$ e $y(t)$, que descrevem a trajetória desenhada em uma assinatura a ser testada, como um sinal de curta duração. Sobre este sinal é feito um pré-processamento e é aplicada a transformada *wavelet* de modo a obter números que sirvam como características, ou parâmetros representativos, da assinatura original. Para comparação, também é implementada uma versão simplificada deste sistema em que a etapa de transformada *wavelet* é eliminada. Com base nestas características, é calculada uma distância em relação a uma assinatura padrão, previamente cadastrada, a fim de fornecer um indicativo quantitativo da autenticidade da assinatura teste. O valor numérico desta distância poderá servir tanto como referência em um sistema de verificação, quanto como uma senha complementar em um sistema de assinatura digital.

Este trabalho faz parte de um projeto atualmente desenvolvido pelo LabSEC (Laboratório de Segurança em Computação - <http://www.labsec.ufsc.br>) no contexto de assinaturas digitais, mostrado esquematicamente na Figura 1.2. Especificamente, as características extraídas da assinatura servirão para, juntamente com um número de identificação pessoal (PIN), compor um número que vai servir de chave para encriptar o resumo obtido pela aplicação de uma função matemática (função *hash*) a algum documento. Este resumo encriptado, anexado ao documento, servirá como garantia de sua integridade em uma verificação posterior. Este processo está mostrado esquematicamente na Figura 1.3.

Organização do Trabalho

No Capítulo 2 é feita uma revisão bibliográfica de trabalhos encontrados na literatura que são relevantes para o sistema desenvolvido. O Capítulo 3 apresenta uma descrição

da transformada *wavelet* e da técnica de programação dinâmica *Dynamic Time Warping* (DTW), elementos matemáticos que constituem a parte fundamental do sistema implementado, descrito no Capítulo 4. No Capítulo 5 são apresentados, então, os resultados obtidos com a aplicação do procedimento desenvolvido a um banco de assinaturas obtido com o auxílio de uma mesa digitalizadora, e são determinadas as taxas de erro (FAR e FRR) para este sistema.

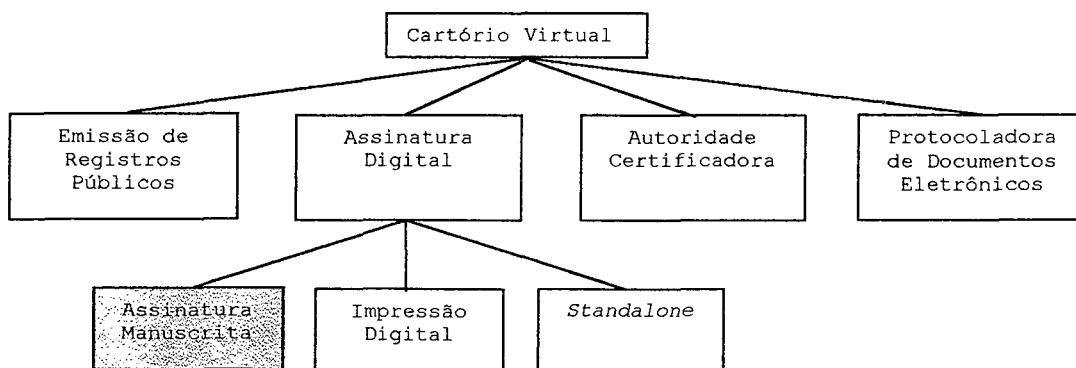


Figura 1.2: Projetos do Laboratório de Segurança em Computação.

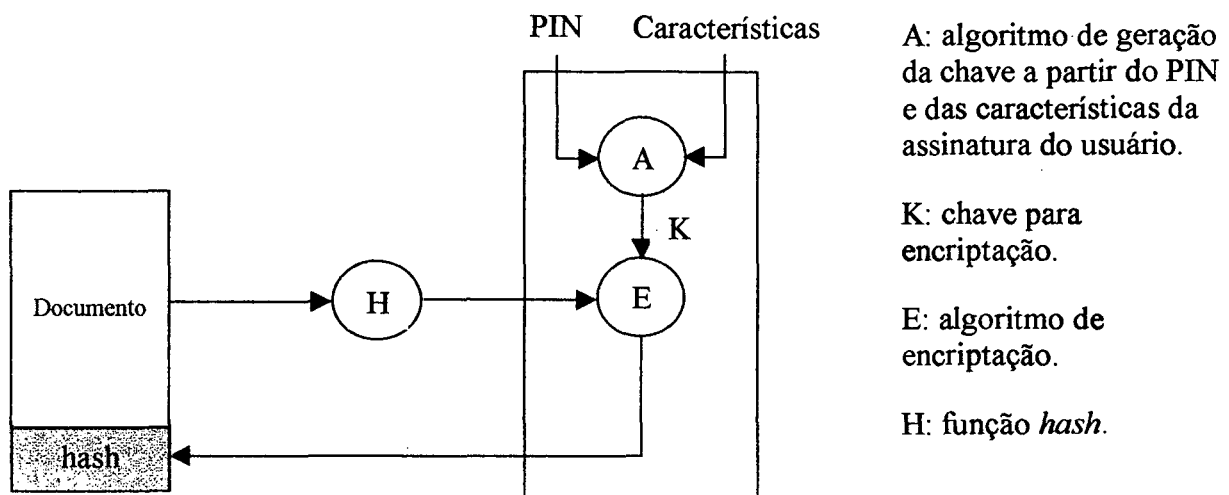


Figura 1.3: Características de uma assinatura usadas como chave para encriptação do hash de um documento.

Capítulo 2

Revisão Bibliográfica

Neste capítulo são apresentados e discutidos alguns métodos diretos e indiretos, presentes na bibliografia, para verificação de assinaturas. Estes trabalhos apresentam argumentos ou técnicas que, de alguma forma, foram utilizados no desenvolvimento do sistema descrito no Capítulo 4. Ainda, é discutido um trabalho relacionado ao tema mais geral de reconhecimento de caracteres escritos a mão, uma vez que também fornece subsídios para a verificação de assinaturas.

2.1 Métodos Indiretos

Conforme mencionado anteriormente, os métodos indiretos demandam maior esforço na obtenção dos dados, que são vetores contendo parâmetros que caracterizam indiretamente as curvas da assinatura. Em compensação, o processo de comparação neste tipo de método é mais simples, consistindo apenas em uma comparação entre vetores.

Griess [GRI00] desenvolveu um sistema *on-line* de verificação de assinaturas indireto que, após um pré-processamento, que consta de suavização, normalização e reamostragem, faz a extração de características. Neste sistema, dois tipos de características são utilizadas: as globais e as locais. O primeiro tipo envolve apenas o número de traços existentes na assinatura. Já as características locais podem ser divididas em espaciais ou temporais. Estas características serão confrontadas com as obtidas durante o cadastramento, quando de três a dez assinaturas por pessoa são recolhidas. Como os tamanhos dos vetores de características locais podem variar, a comparação é feita com o auxílio de uma técnica de otimização conhecida como *Dynamic Time Warping* (ver Capítulo 3), que a autora chama de método de comparação

de *strings*, sendo que cada ponto da assinatura é representado por um vetor de características. Esta comparação resultará em um valor de dissimilaridade que, comparado a um limiar (global ou individualizado), classificará a assinatura como verdadeira ou não.

Para a obtenção das taxas de erros, foram feitos testes com uma base de dados que incluía 520 assinaturas de 52 pessoas, com falsificações de dois tipos: aleatória e treinada. Também foi feita uma comparação da utilização de um limiar de decisão individualizado ou global. Com um limiar global, comum para todos os indivíduos, o erro tipo I (FRR) ficou igual a 2.7% e o tipo II (FAR) igual a 3.3%. Com o limiar individualizado, as taxas de erro caíram para 1.6% no tipo I e 2.8% no tipo II. Estas taxas estão, entretanto, mascaradas porque o autor usou falsificações aleatórias nos seus testes.

Um outro trabalho importante baseado na estratégia de comparação indireta a partir de parâmetros é o de Lee et al [LEE96]. Eles apresentam um sistema de verificação *on-line* indireto no qual a etapa de pré-processamento praticamente inexistente. O problema causado pela possível rotação e deslocamento das assinaturas é eliminado, segundo os autores, colocando-se sobre a prancheta digitalizadora um papel com uma linha horizontal de referência.

O sistema implementado é baseado em um conjunto de 42 características da assinatura, sendo 13 delas estáticas e 29 dinâmicas. Por esse conjunto se mostrar muito sensível a variações no tamanho e na velocidade presentes entre assinaturas verdadeiras, um outro conjunto foi proposto. Este novo conjunto possui 49 características normalizadas, onde a maioria das características temporais, por exemplo, aparece dividida pelo tempo total em que a caneta encosta na prancheta. Os autores destacam três algoritmos para seleção de um subconjunto do total de características apresentadas. O primeiro deles leva em conta que apenas assinaturas verdadeiras estão disponíveis. Neste caso, as melhores características para o sujeito a serão aquelas onde as distâncias d_i , em relação ao resto da população, definidas por:

$$d_i(a, b) = \frac{|m(a, i) - m(b, i)|}{\sqrt{\sigma^2(a, i) + \sigma^2(b, i)}} \quad (2.1)$$

são as maiores, onde $m(a, i)$ e $\sigma^2(a, i)$ são a média e a variância simples da característica i computados a partir da base de dados de referência do sujeito a . Quando, além das

verdadeiras, estão disponíveis assinaturas falsificadas (f), usa-se um segundo algoritmo, no qual a distância d_i passa a ser dada por:

$$d_i(a) = \frac{|m(a,i) - m(f,i)|}{\sqrt{\sigma^2(a,i) + \sigma^2(f,i)}} \quad (2.2)$$

ou seja, a distância não será mais medida de um sujeito para outro, e sim de a para suas falsificações, de maneira que a característica i será considerada mais importante para o indivíduo a do que a característica j se $d_i(a) > d_j(a)$.

Lee et al. comentam que trabalhar com conjuntos individualizados de características produz excelentes resultados mas pode não ser viável por causa de limitações de tempo de resposta ou de tamanho de memória. Assim, um terceiro algoritmo apresenta uma alternativa a esses problemas, sugerindo um *conjunto de características comuns*. Este conjunto será formado pelas características que aparecem como as melhores para a maioria dos indivíduos na população.

Um classificador que implementa uma regra de decisão com base na maioria é utilizado para decidir quando uma assinatura será aceita ou será considerada falsificação. Dadas as n características de uma assinatura:

$$N_\alpha = \left| i : \frac{|t_i - m_i|}{\sigma_i} < \alpha \right| \quad (2.3)$$

onde α é um limiar fixo, m_i e σ_i são a média e o desvio padrão para a característica i e t_i o valor da característica i para a assinatura T que está sendo testada. A regra funciona da seguinte maneira: “ T é declarada verdadeira se $N_\alpha \geq n/2$ e é uma falsificação se $N_\alpha < n/2$, sendo n o número de características utilizadas”.

Selecionando um subconjunto de 15 das 49 características, os autores chegaram às seguintes taxas de erro em seus testes: 1% para o erro tipo I (FRR) e 20% para o tipo II (FAR). As taxas de erro obtidas aqui são maiores que as de Griess, apresentadas anteriormente, contudo são mais confiáveis, uma vez que foram consideradas falsificações simples, estáticas e ainda um terceiro tipo que une as informações estáticas com conhecimento do tempo médio total de assinatura.

Assim como Griess, Huang e Yan [HUA95] adotam uma estratégia de extrair características após um pré-processamento adequado, mas efetuam um tratamento por partes, sobre assinaturas segmentadas. Os autores afirmam que devido a diferenças nas estruturas musculares e a hábitos de escrita de cada pessoa, as características dinâmicas,

como velocidade e tempo, são mais difíceis de imitar. Desta forma, propõem um sistema de verificação baseado em uma segmentação dinâmica da assinatura, onde os pontos em que a velocidade de assinatura é próxima ou igual a zero são escolhidos como pontos de segmentação. Com a assinatura segmentada, são extraídas características como:

- número de segmentos;
- comprimento total;
- tempo de duração;
- velocidade máxima;
- velocidade média.

Estas características são calculadas no nível global, isto é, para toda a assinatura. Algumas delas são calculadas também para o nível local, levando em consideração aspectos individuais de cada segmento, como: tamanho do segmento, tempo de duração do segmento, velocidade máxima, etc.

A verificação é feita com a comparação das características globais e locais da assinatura teste com as características padrão, calculadas a partir de nove assinaturas fornecidas no cadastramento. Na verificação, os parâmetros globais são utilizados preliminarmente para rejeitar falsificações mal feitas ou aleatórias, ficando a verdadeira tarefa de verificação a cargo das características locais, que são mais consistentes. A verificação no âmbito local é feita considerando as medidas de:

- distorção na direção do movimento da caneta;
- variação na forma;
- variação na aceleração;
- variação no tempo de duração e tamanho do segmento.

Os resultados experimentais, segundo os autores, mostram que o sistema é confiável para assinaturas consistentes. Testes feitos num grupo de 20 assinantes indicaram 5% de falsas rejeições (FRR) e 2% de falsas aceitações (FAR), com utilização de falsificações treinadas.

Ainda entre os sistemas *on-line* de verificação de assinaturas por parâmetros, Nalwa [NAL97] apresenta uma proposta inovadora, uma vez que se baseia, principalmente, na forma da assinatura e não na sua dinâmica, como ocorre na maioria dos trabalhos. Segundo o autor, não há evidências substanciais de que o efeito da

dinâmica da caneta seja tão, ou mais, consistente que a forma final da assinatura. Além disso, ele argumenta que, quando uma pessoa verifica uma assinatura, se baseia apenas na forma, não levando em consideração em qual velocidade a assinatura foi feita. O autor credita a importância dada às características dinâmicas à crença difundida de que o ato de assinar é um movimento balístico, uma ação de reflexo (movimento rápido e sem *feedback*) e não uma ação deliberada, na qual há uma tentativa consciente de reproduzir um padrão visual. O autor, entretanto, questiona esta colocação afirmando que indivíduos fazem sua assinatura tanto de uma forma quanto de outra, dependendo da importância e da urgência da tarefa.

Desta forma, é proposto um sistema em que cada assinatura é representada por um modelo local e puramente baseado na forma. Dentre as vantagens deste tipo de modelo, são apontadas:

- rejeita as assinaturas genuínas que não se parecem visualmente com o padrão, o que permite que quem está assinando compreenda uma eventual rejeição da assinatura - além disso, evita que o usuário alegue mais tarde que aquela assinatura seja falsa, embora tenha sido aceita, uma vez que não se parece na forma com a assinatura dele;
- comparações locais permitem, ao contrário das globais, distinguir erros causados por problemas isolados, daqueles causados por desvios sistemáticos, provenientes de diferentes tipos de escrita, por exemplo;
- um sistema deste tipo pode dar destaque a similaridades e discrepâncias entre as formas de duas assinaturas, permitindo que o usuário compreenda por que a assinatura foi aceita ou rejeitada;
- ainda, um sistema baseado na forma não requer que os usuários sejam consistentes quanto ao tempo e força na assinatura - caso isto fosse exigido, talvez levasse a uma alteração na maneira tradicional de se assinar, onde se espera consistência apenas quanto à forma.

O modelo apresentado é baseado em “parametrização sobre comprimento de arco” e consiste em considerar a assinatura como uma linha constituída por massa uniformemente distribuída, ao longo da qual “desliza” uma janela. Em cada pedaço compreendido pela janela são aplicados alguns conceitos da física, como: cálculo do

centro de massa, momento de inércia e torque, para extração de funções que serão as funções características.

Para uma base de dados formada por 904 assinaturas verdadeiras, de 59 indivíduos e 325 falsificações treinadas e incentivadas, inclusive com recompensa financeira, o sistema desenvolvido levou a uma taxa de erro global (taxa do erro tipo I igual a do tipo II) de 3%, usando 6 assinaturas para formar a assinatura padrão. Baixando para 5 ou 4 assinaturas, a taxa de erro global apresentou um ligeiro acréscimo.

2.2 Métodos Diretos

O tratamento computacional de assinaturas tende a ser complicado porque qualquer método deve ser suficientemente preciso e rápido na discriminação de duas assinaturas, mas também suficientemente flexível para considerar pequenas variações em uma mesma assinatura. Desta forma, a decisão do método mais indicado, entre diretos e indiretos, é difícil. É de se esperar, contudo, que métodos que consideram as funções completas que descrevem uma assinatura (curvas $x(t)$ e $y(t)$) obtenham menores taxas de erro. Mas, como foi dito, os métodos diretos são mais trabalhosos no processo de comparação, uma vez que tratar a função toda é custoso e complexo do ponto de vista matemático. Mesmo assim, os métodos deste tipo vêm se tornando cada vez mais eficientes.

Um trabalho que serve como referência para esta linha é o de Sato e Kogure [SAT82]. Neste trabalho, os autores propõem um sistema baseado em três pseudo-distâncias, uma de forma, uma de movimento e uma de pressão de escrita. Elas são chamadas de pseudo-distâncias porque individualmente são uma medida necessária mas não suficiente para verificação. O primeiro passo para obtenção destas distâncias é o pré-processamento dos dados adquiridos, que são: $x(t)$, $y(t)$ e $p(t)$, coordenadas e pressão para os pontos que compõem a assinatura. Tanto a normalização quanto o resto do sistema não trabalham com $x(t)$ e $y(t)$ separadamente, sendo essas duas funções reais unidas em uma função complexa $z(t)$:

$$z(t) = x(t) + iy(t), \quad t \in [0,1] \quad (2.4)$$

Em seguida, a função $z(t)$ é normalizada quanto a:

1. Tempo de duração - Como os tempos de assinatura (T) podem variar, é necessário fazer com que a função varie em $[0,1]$ e não mais em $[0,T]$. Isso é feito da seguinte forma:

$$z_1(t) = z(tT), \quad t \in [0,1] \quad (2.5)$$

2. Localização - Para que as curvas fiquem ajustadas quanto a localização, todas elas são colocadas com o seu centróide na origem. O centróide de uma curva é dado por:

$$c_z = \int_0^1 z(t) dt \quad (2.6)$$

Assim, a nova curva será:

$$z_2(t) = z_1(t) - c_z \quad (2.7)$$

3. Rotação - Para este ajuste é feito:

$$z_3(t) = z_2(t) e^{(-i \tan^{-1} \alpha)} \quad (2.8)$$

Sendo α o ângulo de inclinação do eixo principal que minimiza o erro quadrático de $z_2(t)$, dado por:

$$\alpha = -\beta + \sqrt{1 + \beta^2} \quad (2.9)$$

Onde:

$$\beta = \frac{\operatorname{Re} \left[\int_0^1 z_2^2(t) dt \right]}{\operatorname{Im} \left[\int_0^1 z_2^2(t) dt \right]} \quad (2.10)$$

4. Eliminação do movimento constante - Como geralmente a maioria das pessoas assina da esquerda para direita, há uma tendência a que os valores de x estejam constantemente crescendo com o tempo. Sato afirma que este movimento constante é um fato indesejado e deve ser eliminado da seguinte maneira:

$$z_4(t) = z_3(t) - (\xi t + \eta) \quad (2.11)$$

Onde ξ e η são tais que minimizam a norma:

$$\|z_3(t) - (\xi t + \eta)\| \quad (2.12)$$

5. Tamanho - Finalmente, é feita a normalização quanto ao tamanho da seguinte forma:

$$z_5(t) = \frac{z_4(t)}{\|z_4(t)\|} \quad (2.13)$$

A função $p(t)$ sofre normalização quanto ao tempo de duração e à amplitude, o que é feito da seguinte forma:

$$p^*(t) = \frac{p(t)}{\left[\int_0^1 p^2(t) dt \right]^{1/2}}, \quad t \in [0,1] \quad (2.14)$$

A respeito das medidas que representam a diferença entre duas assinaturas, os autores afirmam que se pode utilizar uma medida simples de distância, dada por:

$$d(A, B) = \|z_A(t) - z_B(t)\| \quad (2.15)$$

onde, $z_A(t)$ e $z_B(t)$ representam as funções pré-processadas de duas assinaturas A e B cuja distância se queira medir.

Esta medida não é, entretanto, considerada ideal, uma vez que é muito sensível a mudanças no eixo do tempo. Para resolver este problema, os autores propõem o uso da técnica de programação dinâmica, geralmente utilizada na área do reconhecimento de fala [RAB93]. Desta forma, é proposta uma nova medida de distância onde o eixo do tempo da assinatura padrão é fixo (t) mas o da assinatura testada é transformado não-linearmente ($\tau(t)$) de forma a minimizar ρ_S , dado por:

$$\rho_S(K) = \min_{\tau} \|z_0(t) - z_k(\tau(t))\| \quad (2.16)$$

A função que representa esta transformação não-linear no tempo é a função de *warping* $c(t, \tau(t))$, descrita no Capítulo 3.

A diferença ρ_S será igual a zero se uma assinatura apresentar uma forma idêntica à da padrão, mesmo que os traços tenham sido feitos em momentos e velocidades distintos. É por causa deste fato que esta distância é considerada necessária mas não suficiente.

A próxima medida, então, tem como objetivo verificar as diferenças no movimento. Estas informações foram eliminadas em ρ_S com o uso da função de *warping*, de modo que podem ser recuperadas desta mesma função. A distância temporal será obtida, então, da diferença entre a função de *warping* e a reta que vai do tempo inicial (0,0) ao tempo final da assinatura:

$$\rho_M(k) = \|\tau_k(t) - t\| \quad (2.17)$$

A última distância que compõe a medida da diferença entre duas assinaturas, chamada de pseudo-distância para pressão, é obtida a partir da diferença da pressão de escrita:

$$\rho_P(k) = \|p_0(t) - p_k(\tau_k(t))\| \quad (2.18)$$

Esta diferença é calculada com as funções $p(t)$ da assinatura padrão e da assinatura de teste normalizadas.

A partir destas três pseudo-distâncias é montado o seguinte vetor:

$$\rho(k) = [\rho_S(k) \quad \rho_M(k) \quad \rho_P(k)]^T \quad (2.19)$$

Finalmente, a medida que representa a diferença do padrão k é dada por:

$$d^2(k) = [\rho_k^T][\Sigma^{-1}][\rho_k] \quad (2.20)$$

onde Σ é a matriz covariância para assinaturas verdadeiras, definida por:

$$\Sigma = E\{\rho_k \rho_k^T\} \quad (2.21)$$

Os testes do sistema foram feitos com assinaturas de 10 japoneses e 1 chinês e falsificações treinadas por 10 minutos, e apresentaram taxa de decisão correta de 96,2% para forma, 92,0% para movimento e 95,5% para pressão. Estas taxas representam os pontos onde as duas taxas de erro, tipo I e tipo II, apresentaram valores iguais.

Diversos autores seguiram a idéia de Sato e Kogure. Dentre eles está Wirtz [WIR95], que propôs um sistema muito semelhante mas com a inovação de segmentar a assinatura. Segundo a autora, o processo de verificação baseado em traços apresentou melhores resultados quando comparado ao sistema original de Sato e Kogure. Já Munich e Perona [MUN99] buscaram melhores resultados com a utilização de um DTW contínuo (CDTW). Os autores concluíram, entretanto que, para sistemas de verificação de assinaturas o uso do CDTW, não representa uma melhoria na performance, em relação ao DTW, que justifique o aumento do custo de processamento.

2.3 Métodos Diretos com Transformadas

Trabalhar com funções tem um custo computacional elevado e uma boa tentativa de melhorar isto é usar *transformadas*. O artigo de Lam e Kamins [LAM89] apresenta o uso da transformada rápida de Fourier (FFT) para levar assinaturas normalizadas para o

domínio de frequência, como alternativa ao uso do *time-warping*, necessário para alinhar as assinaturas teste e padrão no tempo. Em relação às outras transformadas (Hadamard, Hough e Walsh) utilizadas para reconhecimento de padrões, o autor afirma que a transformada de Fourier funciona melhor como um descritor para $x(t)$ e $y(t)$, uma vez que resulta em um número menor de harmônicas, porém mais significativas.

Os autores utilizam o pré-processamento proposto por Sato e Kogure e, além disso, concatenam os segmentos da assinatura, inclusive o início e o final, por interpolação linear. Tal medida é necessária pois a transformada de Fourier exige que a assinatura seja periódica e que não haja descontinuidades.

Aqui, a assinatura é tratada também como uma função complexa, onde:

$$f(t) = x(t) + iy(t) \quad (2.22)$$

Para a obtenção das características, a assinatura normalizada é transformada para o espaço de frequências usando a FFT. São obtidas 1024 harmônicas no domínio de frequência, porque as assinaturas consistiam de 1024 pontos complexos. As frequências mais altas são descartadas por representarem ruído, e para isto é utilizado um limite superior de 20Hz. Como, mesmo assim, ainda restam muitas harmônicas para extração das características, o corte é ampliado de tal forma que restem apenas as 15 harmônicas com as maiores magnitudes normalizadas pela sua variância em várias amostras.

Para testar a utilidade das 15 harmônicas selecionadas, foi utilizada uma análise discriminante para classificar 8 assinaturas autênticas fornecidas por um indivíduo e 152 falsificações feitas por 19 falsificadores treinados. A parte real e a imaginária das 15 harmônicas de todas as assinaturas foram usadas como variáveis para uma análise discriminante. Todas as 8 assinaturas autênticas foram corretamente classificadas, enquanto 4 das 152 falsificações foram classificadas como verdadeiras, o que resulta numa taxa de erro do tipo II de 2,5%. Desta forma, Lam conclui que a FFT se mostrou um método útil para redução de dados e identificação de características para o reconhecimento de assinaturas. Os autores frisam ainda a importância das características dinâmicas no processo de verificação de assinaturas, dada a dificuldade de reprodução das mesmas por parte dos falsários.

Não foram encontrados na literatura outros artigos utilizando a transformada de Fourier. Uma possível explicação para isto é a argumentação apresentada em um artigo de Sundaresan e Keerthi [SUN99]. Os autores apresentam um rápido estudo das

possíveis maneiras de se representar, num sistema de reconhecimento de escrita, os caracteres do idioma indiano Tamil escritos a mão. Qualquer que seja a representação, é necessário que seja feito um pré-processamento do caracter adquirido de maneira *on-line*. Na análise de Sundaresan e Keerthi, este pré-processamento consistiu de normalização e reamostragem. A normalização é necessária pois os caracteres podem ser escritos em tamanhos diferentes, sendo necessário, então, um redimensionamento. Foi necessário ainda um terceiro passo, que consistiu na aplicação de um filtro do tipo médias móveis (filtro passa-baixa), que serviu para a remoção de ruídos possivelmente inseridos durante a normalização e reamostragem. Sundaresan e Keerthi afirmam que as coordenadas x e y pré-processadas poderiam ser utilizadas como entrada para uma rede neural reconhecadora de caracteres escritos a mão. Entretanto, os autores argumentam que, para maior acurácia, o classificador (rede neural) teria que ser mais complexo ou mais dados de entrada seriam necessários. Assim, a forma de representação do caracter é de grande importância, já que, se representado por “boas” características, o processo de reconhecimento torna-se mais simples. Neste contexto, quatro formas de representação de caracteres são propostas:

1. seqüência de direções e curvaturas;
2. seqüência de ângulos;
3. coeficientes da transformada de Fourier e
4. características *wavelet*.

A primeira proposta indica que cada ponto n que compõe a linha da assinatura deve ser representado por sete características:

1. $penup(n)$: indica se a caneta está tocando a prancheta ou não;
2. $\tilde{x}(n)$: coordenada $x(n)$ pré-processada;
3. $\tilde{y}(n)$: coordenada $y(n)$ pré-processada;
4. $\cos\theta(n)$: cosseno da direção local no n -ésimo ponto;
5. $\sin\theta(n)$: seno da direção local no n -ésimo ponto;
6. $\cos\phi(n)$: cosseno da curvatura do n -ésimo ponto;
7. $\sin\phi(n)$: seno da curvatura do n -ésimo ponto.

Se na nova amostragem restarem n pontos, o caracter será representado, então, por $7.n$ números, o que é considerado alto.

Na representação que utiliza seqüência de ângulos, é armazenado apenas o cosseno do ângulo formado entre duas linhas consecutivas, que, por sua vez, são formadas por três pontos consecutivos da reamostragem. Segundo os autores, esta abordagem parece apropriada, mas no caso das coordenadas x e y estarem muito próximas ou possuírem ruídos, os resultados não serão satisfatórios, uma vez que pequenas variações serão responsáveis por grandes diferenças nas características extraídas. Também haverá problemas por causa da reamostragem, que reduz o número de pontos e, conseqüentemente, gera perda de informação, tornando a representação pobre para ser usada na classificação.

As duas outras representações propostas estão baseadas no fato de que, em muitos momentos, uma assinatura pode ser vista como um sinal e um sinal faz mais sentido no domínio de freqüência do que no de tempo. Assim, os coeficientes da transformada de Fourier são apresentados como a terceira forma de representação de caracteres. Os coeficientes de Fourier de baixa freqüência são apontados como uma boa maneira de se representar a forma básica de um caracter, pois é uma representação pouco sensível a pequenas variações, como aquelas que ocorrem em função do estilo de escrita de cada pessoa. Porém, no caso da escrita Tamil, ocorre que dois caracteres distintos não são facilmente diferenciados no domínio de freqüência. Isto porque uma pequena mudança na seqüência (no domínio do tempo) num pequeno intervalo de tempo ficará distribuída em todo o domínio de freqüência, diminuindo, assim, a diferença na representação de caracteres distintos. Pode-se observar na Figura 2.1 que alguns caracteres Tamil diferem apenas em pequenos detalhes.

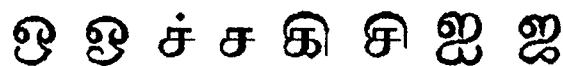


Figura 2.1: Exemplos de caracteres Tamil.

A fim de captar adequadamente as sutis diferenças entre os caracteres, os autores sugerem que as características sejam obtidas com a aplicação da transformada *wavelet* nas seqüências x e y separadamente.

A transformada *wavelet* permite uma análise semelhante àquela proporcionada pelos coeficientes de Fourier, com o conceito de “escalas” ocupando o lugar das freqüências. Mas apresenta a vantagem de também preservar informações relativas ao *momento* em que mudanças importantes ocorrem na função original (no domínio de

tempo). No caso específico do trabalho de Sundaresan e Keerthi, dois caracteres Tamil diferindo apenas por algum detalhe teriam transformadas de Fourier praticamente idênticas, mas suas transformadas *wavelet* apresentariam um nítida mudança de padrão nas regiões de $x(t)$ e $y(t)$ correspondentes a este detalhe.

A representação por transformada *wavelet* é, dentre as estudadas pelos autores, a que apresenta os melhores resultados nos testes efetuados (96.54% de sucesso para 12 caracteres e 94.30% de sucesso para 135 caracteres). Embora os autores não descrevam completamente seus procedimentos de teste, seus resultados sugerem fortemente que a transformada *wavelet* é uma ferramenta útil para a caracterização de símbolos escritos.

Seguindo a idéia de que a transformada *wavelet* é a que melhor consegue captar as sutilezas em uma assinatura, surge o trabalho de Deng et al. [DEN97]. Neste trabalho é proposto um sistema de verificação de assinaturas *off-line* que utiliza a transformada *wavelet* para decomposição e análise dos sinais obtidos a partir da imagem da assinatura.

Por ser um sistema *off-line*, estes sinais não podem ser armazenados em função do tempo. Assim, as coordenadas x e y e o ângulo tangencial de cada ponto da imagem são gravados de acordo com uma convenção pré-estabelecida. Para isso, a imagem da assinatura, que é feita em um papel de tamanho fixo, passa por um processo de dilatação. Na assinatura dilatada são isolados os contornos fechados e são armazenadas as coordenadas de cada contorno em sentido anti-horário, partindo do canto noroeste do contorno.

É necessário obter, a partir dos dados armazenados, características estáveis e que representem bem a assinatura padrão. A transformada *wavelet* (ver Capítulo 3) foi escolhida pelos autores porque permite decompor o sinal em baixa e alta frequências, além de propiciar uma análise em multiresolução. A baixa frequência conteria a informação geral da curva, uma aproximação grosseira, sendo estável porém pouco discriminante. Já a alta frequência conteria informações dos detalhes das curvas e também um alto nível de ruído. Possui um maior poder discriminante, mas, por outro lado, é instável.

Dos dados pré-processados são extraídas as seguintes informações:

- Número total de contornos fechados na assinatura;

- Os seguintes atributos para cada passagem pelo zero das curvas em um nível de detalhe (alta frequência) pré-determinado, para cada um dos três sinais:
 - a abscissa da passagem pelo zero (ver Capítulo 3);
 - a integral entre a passagem pelo zero atual e a anterior;
 - a amplitude, na mesma abscissa, da curva de aproximação (baixa frequência) no nível anterior.

Os autores escolhem estes três atributos das passagens pelo zero como características, pois, segundo Mallat [MAL91], estas informações são suficientes para uma completa reconstrução do sinal original.

A partir destes valores é calculado um grau de dissimilaridade, que representa a distância entre duas assinaturas quaisquer. Para isso, é necessário o uso do *Dynamic Time Warping* (DTW, ver Capítulo 3), pois o número de passagens pelo zero pode variar mesmo entre assinaturas verdadeiras. O grau de dissimilaridade é dado por:

$$d = \sum_{k=1}^K \sum_{s \in S} d^L(k, s) \quad (2.23)$$

Ou seja, a distância utilizada no DTW para comparar dois contornos distintos é a soma das distâncias Euclidianas entre os atributos de passagem pelo zero correspondentes. O parâmetro K representa o número de contornos fechados mais estáveis em todas as assinaturas do indivíduo, ou seja, o conjunto de contornos que resulta na menor distância intra-pessoal (com mais repetibilidade). O conjunto S pode ser formado por até três elementos: coordenada x, coordenada y e ângulo tangencial correspondente. Como o processo de extração de características é feito com uma transformada *wavelet* em múltiplos níveis de resolução, L representa o nível de resolução do detalhe (correspondente a um filtro passa-alta) que minimiza a dissimilaridade.

O processo de verificação da assinatura é feito através da comparação deste grau de dissimilaridade com um limiar (“*threshold*”) individualizado, dado por:

$$T = \mu + \delta\sigma \quad (2.24)$$

onde μ é a média dos graus de dissimilaridade obtidos a partir das n assinaturas de referência (do cadastramento) do indivíduo e σ é o seu desvio padrão. Aqui, cada grau de dissimilaridade é calculado entre cada assinatura de referência e a assinatura mais

próxima a ela dentre as outras $n-1$ assinaturas de referência daquele indivíduo. O parâmetro δ serve para regular o erro tipo I (FRR) e foi fixado pelo autor em 2,0.

O método resulta num limiar ótimo para cada indivíduo. Como μ e δ são funções de d , e d é função de K e L , tem-se que T é função de K e L . O limiar ótimo é calculado com a aplicação de programação não-linear para determinar a combinação de K e L que leva a um valor mínimo para T (função objetiva), o que corresponde a minimizar o erro tipo II (FAR). Na prática, porém, os autores utilizam no lugar da programação não-linear a força-bruta, testando K num intervalo de 1 a 7 e L de 1 a 5. A Tabela 2.1 exemplifica a busca, para três indivíduos, do nível de resolução L que minimiza T .

Os autores trabalham com valores normalizados por l_j , que representa o comprimento total (em pixels) dos K maiores contornos da assinatura de referência j . Segundo os autores, um T normalizado se torna invariante ao número de maiores contornos K e ao melhor nível de resolução L de cada indivíduo.

Tabela 2.1: Resultados parciais na busca do nível de resolução ótimo para a aplicação da transformada wavelet no cálculo do grau de dissimilaridade [DEN97].

Nível L	T Indivíduo 1	T Indivíduo 2	T Indivíduo 3
1	0.003509	0.015381	0.029989
2	0.006410	0.014994	0.016269
3	0.003121	0.009800	0.021100
4	0.002508	0.028672	0.027075
5	0.022350	0.172897	0.154069

Para testar o sistema desenvolvido, Deng et al. construíram uma base de dados própria, uma vez que não há uma base de dados padrão para sistemas de verificação de assinaturas *off-line*. A base é composta por assinaturas de 50 pessoas, incluindo 25 chineses. Das assinaturas fornecidas, 10 foram utilizadas como assinaturas de referência. Foram feitos cinco tipos diferentes de falsificações. As taxas de erro obtidas são de 6% para o erro tipo I e de 7.8% para o tipo II nas assinaturas chinesas e de 5.6% para o tipo I e 10.98% para o tipo II nas assinaturas ocidentais.

A idéia de utilizar a transformada *wavelet* em reconhecimento de padrões também aparece no trabalho de Boles [BOL98], no qual é apresentado um sistema de segurança que utiliza a transformada *wavelet* para reconhecimento da íris. De modo semelhante ao

que é feito por Deng et al., a fim de extrair características únicas obtidas a partir dos perfis de níveis de cinza da imagem da íris, o autor trabalha com as passagens pelo zero da transformada *wavelet* e também com as integrais entre duas passagens pelo zero consecutivas.

Capítulo 3

Fundamentos Teóricos

Neste capítulo são descritas as ferramentas matemáticas que servem de base para o sistema que foi implementado no presente trabalho: a transformada *wavelet* e o *Dynamic Time Warping*. Inicialmente, é discutida a definição de transformada *wavelet*, bem como sua interpretação e propriedades. Em seguida, é apresentado o algoritmo do *Dynamic Time Warping* na forma como aparece na literatura e também uma adaptação para uma versão não-recursiva, utilizada neste trabalho, que viabilizou a comparação entre conjuntos com um grande número de características.

As descrições apresentadas nos itens 3.1 a 3.5 a seguir estão baseadas nos trabalhos de Strang e Nguyen [STR96] e Stollnitz et al. [STO95].

3.1 Wavelets

Wavelets são uma ferramenta matemática utilizada para decompor funções hierarquicamente. Elas permitem que a função seja descrita em termos de uma forma aproximada (grosseira) mais os detalhes.

Wavelets são funções base $\psi_{jk}(t)$ em um tempo contínuo. Uma base é um conjunto de funções linearmente independentes que podem ser usadas para formar todas as funções admissíveis $f(t)$ de um dado espaço:

$$f(t) = \sum_{j,k} b_{jk} \psi_{jk}(t) \quad (3.1)$$

A característica especial da base *wavelet* é que todas as funções $\psi_{jk}(t)$ são construídas a partir de uma única *wavelet* mãe $\psi(t)$. As *wavelets* mãe são, como mostra

a Figura 3.1, pequenas ondas (pulsos). Normalmente começam no tempo $t=0$ e terminam em $t=N$.

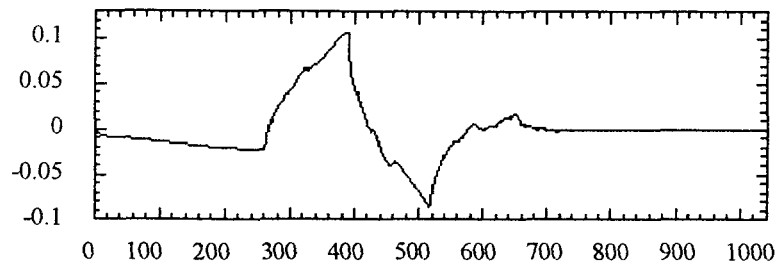


Figura 3.1: Exemplo de wavelet mãe (Daubechies 4) [PRE92].

Exemplo Numérico: decomposição de uma imagem unidimensional na base de Haar [STO95]

A base de Haar é a mais simples das bases *wavelet* e permite apresentar diversos conceitos relacionados a *wavelets* de maneira intuitiva.

Seja uma imagem unidimensional com uma resolução de 4 pixels, com os seguintes valores: 9 7 3 5.

Para representar esta imagem na base de Haar, primeiro faz-se a média dos pixels dois-a-dois, para conseguir um novo nível inferior de resolução da imagem, com os seguintes valores de pixels: 8 4.

É claro que alguma informação foi perdida neste processo. Para recuperar os quatro valores de pixel originais destes dois valores de média, é necessário o armazenamento de alguns *coeficientes de detalhe*, que capturam a informação ausente. Neste caso, o primeiro coeficiente de detalhe será 1, uma vez que a média computada é 1 a menos que 9 e 1 a mais que 7. Este único número permite recuperar os dois primeiros pixels dos quatro originais. De maneira similar, o segundo coeficiente de detalhe é -1, dado que $4 + (-1) = 3$ e $4 - (-1) = 5$.

Assim, o sinal da imagem original foi decomposto em uma versão num nível inferior de resolução (com dois pixels) e um par de coeficientes de detalhe. A repetição recursiva deste processo sobre os valores das médias resulta na completa decomposição, mostrada na Tabela 3.1.

Então, para a base unidimensional de Haar, a transformada *wavelet* da imagem original deste exemplo com quatro pixels é dada pelo coeficiente único que representa a

média total, seguido pelos coeficientes de detalhe para aumento de resolução; ou seja: 6 2 1 -1.

Tabela 3.1: Médias e coeficientes de detalhe para os diferentes níveis de resolução da imagem.

Resolução j	Médias	Coefficientes de Detalhe
4	9 7 3 5	
2	8 4	1 -1
1	6	2

Note que nenhuma informação foi ganha ou perdida neste processo. A imagem original possui quatro coeficientes e a “transformada” também. Dada a transformada, é possível reconstruir a imagem em qualquer resolução, através de somas e subtrações sucessivas dos coeficientes de detalhe nos níveis inferiores de resolução.

3.2 Wavelets como um Espaço Vetorial

Ao invés de se tratar as imagens como seqüências de coeficientes, pode-se pensar nelas como sendo funções constantes por partes no intervalo semi-aberto $[0,1)$. Para isso, usa-se o conceito de espaço vetorial da álgebra linear. Uma imagem de um pixel é apenas uma função que é constante sobre todo o intervalo $[0,1)$. Seja V^0 o espaço vetorial de todas estas funções. Uma imagem de dois pixels possui dois pedaços constantes sobre os intervalos $[0,1/2)$ e $[1/2,1)$. O espaço contendo todas estas funções é chamado V^1 . Desta forma, o espaço V^j inclui todas as funções constantes por partes definidas no intervalo $[0,1)$ com pedaços constantes sobre cada 2^j subintervalos iguais.

Cada imagem unidimensional com 2^j pixels pode ser vista, então, como um elemento, um vetor, em V^j . Como estes vetores são todos funções definidas no intervalo unitário, cada vetor em V^j está também contido em V^{j+1} . Por exemplo, pode-se sempre descrever uma função constante por partes em dois intervalos como uma função constante por partes em quatro intervalos, com cada intervalo na primeira função correspondendo a um par de intervalos na segunda função. Assim, os espaços V^j estão aninhados, isto é: $V^0 \subset V^1 \subset V^2 \subset \dots$. A teoria matemática da análise por multiresolução requer este “aninhamento” dos espaços V^j . As funções base para os espaços V^j são chamadas *funções escala*, e são usualmente denotadas por ϕ .

O produto interno escolhido para este espaço é o padrão:

$$\langle f | g \rangle := \int_0^1 f(x)g(x)dx \tag{3.2}$$

Desta forma, pode-se definir um novo espaço vetorial W^j como o complemento ortogonal de V^j em V^{j+1} . Em outras palavras, W^j será o espaço de todas as funções em V^{j+1} que são ortogonais a todas as funções em V^j sob o produto interno escolhido. Informalmente, pode-se dizer que W^j conterà o detalhe de V^{j+1} que não pode ser representado em V^j .

As funções linearmente independentes $\psi_i^j(x)$ cobrindo o espaço W^j são chamadas *wavelets*. Estas funções base possuem duas propriedades importantes:

1. as funções base ψ_i^j , junto com as funções base ϕ_i^j de V^j , formam uma base para V^{j+1} , e
2. cada função base ψ_i^j de W^j é ortogonal a cada função base ϕ_i^j de V^j sob o produto interno escolhido.

Exemplo: espaço vetorial de wavelets construído sobre a base de Haar

Uma base simples para V^j é dada por um conjunto de funções retangulares transladadas e modificadas quanto a escala (base de Haar):

$$\phi_i^j := \phi(2^j x - i), \quad i = 0, \dots, 2^j - 1 \tag{3.3}$$

onde:

$$\phi(x) := \begin{cases} 1 & \text{para } 0 \leq x < 1 \\ 0 & \text{caso contrário} \end{cases} \tag{3.4}$$

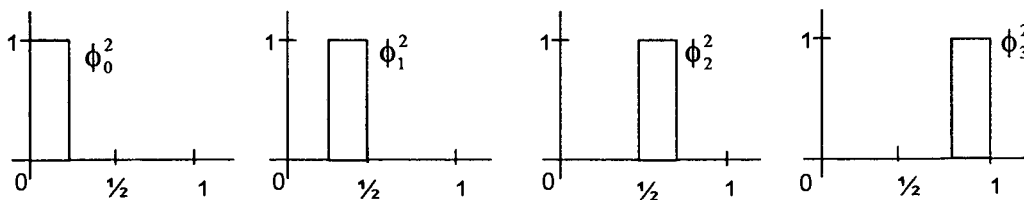


Figura 3.2: Base de Haar para V^2 .

Os coeficientes de detalhe são na verdade coeficientes das funções base *wavelets*.

As *wavelets* de Haar são dadas por:

$$\psi_i^j(x) := \psi(2^j x - i), \quad i = 0, \dots, 2^j - 1 \quad (3.5)$$

Onde:

$$\psi(x) := \begin{cases} 1 & \text{para } 0 \leq x < \frac{1}{2} \\ -1 & \text{para } \frac{1}{2} \leq x < 1 \\ 0 & \text{caso contrário} \end{cases} \quad (3.6)$$

Exemplo numérico: decomposição de uma imagem unidimensional sobre o espaço das funções de Haar

Vimos que o sinal 9 7 3 5 pode ser descrito por uma média (6) e três detalhes (2 1 -1). No contexto de espaços vetoriais, isto é o mesmo que representar a imagem original, $I(x)$, como uma combinação linear das funções base em V^2 , mostradas na Figura 3.2; ou seja:

$$I(x) = c_0^2 \phi_0^2(x) + c_1^2 \phi_1^2(x) + c_2^2 \phi_2^2(x) + c_3^2 \phi_3^2(x) \quad (3.7)$$

Utilizando-se as funções base em V^2 , uma representação gráfica de $I(x)$ teria a forma mostrada na Figura 3.3.

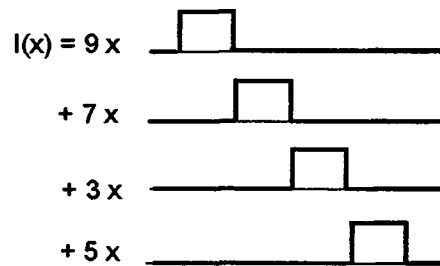


Figura 3.3: Representação gráfica de $I(x)$, onde os coeficientes c_0^2, \dots, c_3^2 são exatamente os quatro valores originais dos pixels: 9, 7, 3, 5.

Mas a expressão para $I(x)$ também pode ser escrita em termos das funções base em V^1 e W^1 , conforme mostra a Figura 3.4, usando médias e diferenças dois-a-dois, ou seja,

$$I(x) = c_0^1 \phi_0^1(x) + c_1^1 \phi_1^1(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad (3.8)$$

o que levaria a uma segunda representação gráfica de $I(x)$, mostrada na Figura 3.5.

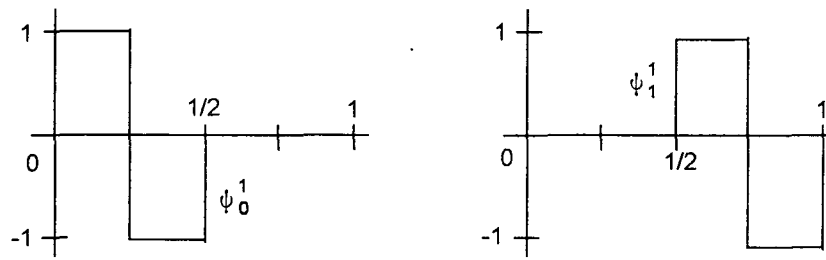


Figura 3.4: Wavelets de Haar para W^1 .

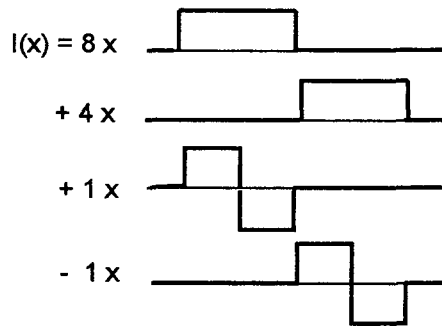


Figura 3.5: Representação intermediária de $I(x)$.

Finalmente, $I(x)$ pode ser reescrita como a soma das funções base em V^0 , W^0 e W^1 :

$$I(x) = c_0^0 \phi_0^0(x) + d_0^0 \phi_0^0(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad (3.9)$$

o que é uma representação gráfica alternativa, mas equivalente à original. A Figura 3.6 ilustra esta última forma de representar a imagem.

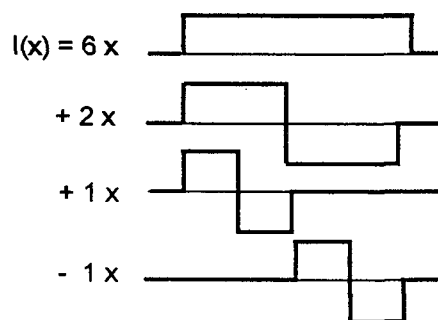


Figura 3.6: Representação alternativa de $I(x)$.

Estes quatro coeficientes são a *transformada wavelet* de Haar para a imagem original. As quatro funções mostradas na Figura 3.2 constituem a base de Haar para V^2 . Ao invés de usar as quatro funções usuais, pode-se usar $\phi_0^0, \psi_0^0, \psi_0^1$ e ψ_1^1 para representar a média total, o detalhe grosseiro e os dois tipos de detalhe mais refinados possíveis em uma função em V^2 . A base de Haar para V^j com $j > 2$ inclui estas funções bem como versões transladadas e mais comprimidas da *wavelet* $\psi(x)$.

3.3 Ortogonalidade das Wavelets

De um modo geral, as *wavelets* deslocadas ψ_{0k} começam no tempo $t=k$ e terminam no tempo $t=k+N$. As *wavelets* modificadas e reescaladas ψ_{j0} começam em $t=0$ e terminam no tempo $t=N/2^j$ (seus gráficos estão comprimidos pelo fator 2^j). Os gráficos de ψ_{0k} são transladados k unidades para a direita:

$$\text{compressão: } \psi_{j0} = \psi(2^j t), \quad \text{deslocamento: } \psi_{0k} = \psi(t - k)$$

Uma *wavelet* típica ψ_{jk} sofre compressão j vezes e é deslocada k vezes. Sua fórmula é:

$$\psi_{jk} = \psi(2^j t - k) \tag{3.10}$$

Uma propriedade alcançada por muitas *wavelets* é a ortogonalidade. Duas *wavelets* são ortogonais quando seu produto interno, dado pela Equação 3.2, é zero:

$$\int_{-\infty}^{\infty} \psi_{jk}(t) \psi_{JK}(t) dt = 0 \tag{3.11}$$

Neste caso as *wavelets* $\phi_0^0, \psi_0^0, \psi_0^1$ e ψ_1^1 , formam uma base ortogonal para o espaço de funções admissíveis. A ortogonalidade leva a uma fórmula simples para cada coeficiente b_{JK} na expansão para $f(t)$. Multiplicando a expansão apresentada na Equação 3.1:

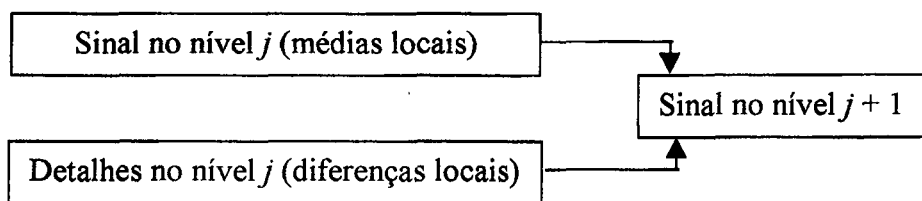
$$f(t) = \sum_{j,k} b_{jk} \psi_{jk}(t) \tag{3.12}$$

por ψ_{JK} e integrando, obtém-se b_{JK} por meio de:

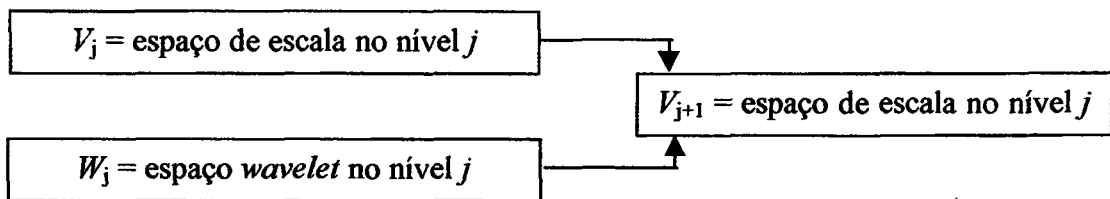
$$\int_{-\infty}^{\infty} f(t) \psi_{JK}(t) dt = b_{JK} \int_{-\infty}^{\infty} (\psi_{JK}(t))^2 dt \tag{3.13}$$

3.4 Multiresolução

Em uma dada resolução de um sinal ou imagem, as funções escala $\phi(2^j t - k)$ são base para a representação de sinais. O nível de resolução é dado por j , e o número de passos no tempo daquele nível é dado por 2^j . Os detalhes no nível j são representados pelas *wavelets* $\psi(2^j t - k)$. Então, o sinal suavizado mais os detalhes, os coeficientes de ϕ mais os coeficientes de ψ , combinam em várias resoluções do sinal em um nível mais refinado $j+1$. As médias vêm das funções escala e os detalhes das *wavelets*:



Isto é multiresolução para um sinal. Quando aplicada a todos os sinais, tem-se multiresolução para espaços de funções:



O sinal é então dividido em diferentes escalas de resolução, ao invés de ser dividido em diferentes frequências. O plano tempo-escala está para *wavelets* assim como o plano tempo-frequência está para filtros. A multiresolução divide as frequências em bandas de ω a 2ω , ao invés de bandas uniformes de ω a $\omega + \Delta\omega$. A compressão de um gráfico, quando $f(t)$ é substituída por $f(2t)$, significa expansão da sua transformada de Fourier de $F(\omega)$ para $(1/2)F(\omega/2)$. Frequências são deslocadas quando a escala do tempo é duplicada. É possível ver, na Figura 3.7, como o plano tempo-frequência é particionado naturalmente em retângulos de área constante.

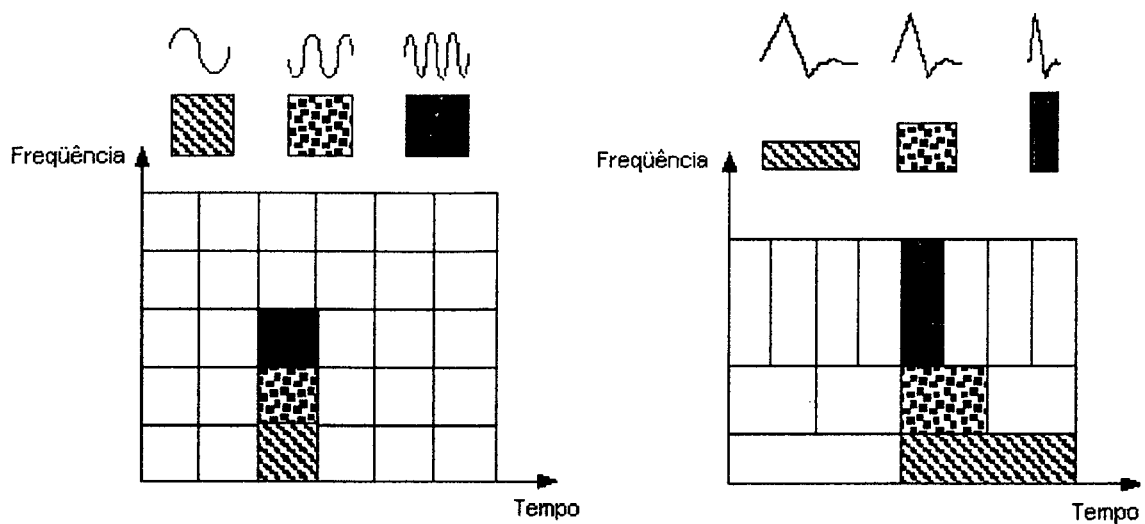


Figura 3.7: Quadrados no plano tempo-frequência para a decomposição de Fourier se tornam retângulos para wavelets. Pequenos intervalos de tempo são naturais para alta frequência.

3.5 Transformada Wavelet como um Banco de Filtros

O resultado da aplicação de uma transformação *wavelet* a um sinal pode ser visto como o efeito da passagem deste sinal por um banco de filtros passa-alta e passa-baixa [STR96].

Filtros, em especial o filtro passa-alta, são utilizados em reconhecimento de padrões a fim de ressaltar detalhes importantes nos sinais que devem ser analisados. O uso de um filtro passa-alta destaca os detalhes do sinal; entretanto, não permite a reconstrução do mesmo. Assim, para que a reconstrução do sinal original seja possível, são utilizados bancos de filtros, que são conjuntos de filtros. Um banco de análise é, geralmente, formado por dois filtros: um passa-alta e um passa-baixa, que separam o sinal de entrada em alta e baixa frequências, como mostra a Figura 3.8.

Em muitos sinais, o conteúdo da baixa-frequência é a parte mais importante, pois é ele que dá a identidade do sinal (aproximação). O conteúdo da alta-frequência, por outro lado, fornece os detalhes.

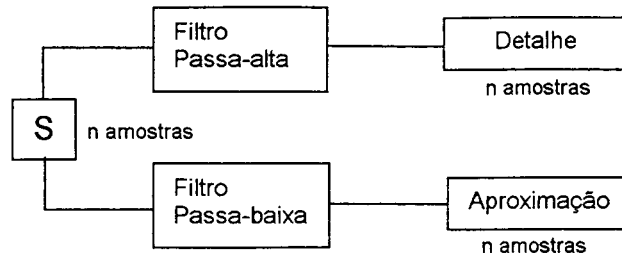


Figura 3.8: O sinal S passa por dois filtros complementares que produzem dois sinais.

Tanto a aproximação, quanto o detalhe obtidos possuirão o mesmo número de amostras do sinal original. Ou seja, após a aplicação dos filtros o sinal estará representado por duas vezes mais amostras que o sinal de entrada.

Não é necessário preservar toda esta saída dos filtros de análise. Para redução do número de amostras é feito um *downsampling*, onde são mantidas apenas as componentes pares das saídas dos filtros passa-baixa e passa-alta. O efeito do *downsampling* pode ser visto na Figura 3.9.

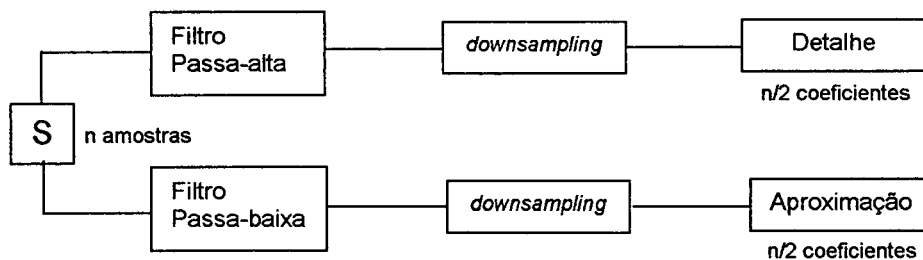


Figura 3.9: Aplicação dos filtros seguida pelo *downsampling* produzindo os coeficientes de detalhe e aproximação, que somados equivalem em quantidade às amostras do sinal de entrada.

As curvas de aproximação e detalhe assim obtidas correspondem às mesmas que seriam obtidas como resultado da aplicação de transformadas *wavelet* com *wavelets* mãe adequadas. Os filtros passa-baixa, de onde saem apenas as baixas frequências do sinal (uma aproximação), equivalem a uma decomposição sobre uma base de *funções de escala* $\phi(t)$. No tempo contínuo, a função de escala vem da sucessiva aplicação do filtro passa-baixa. A *wavelet* $\psi(t)$ consiste de uma única aplicação do filtro passa-alta à função de escala $\phi(t)$ assim obtida. Este processo está mostrado esquematicamente na Figura 3.10.

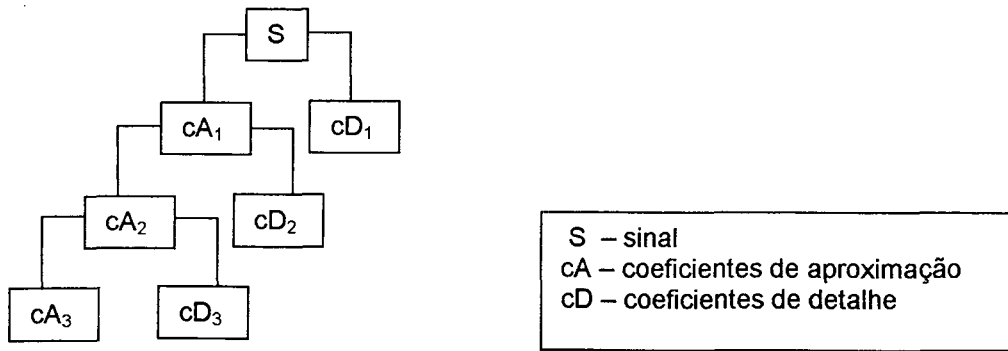


Figura 3.10: Árvore de decomposição do sinal em aproximação e detalhe sucessivas vezes.

3.6 Dynamic Time-Warping (DTW)

Na verificação de assinaturas é necessário obter medidas do grau de dissimilaridade entre duas assinaturas. No presente trabalho, as assinaturas são representadas por curvas $x(t)$ e $y(t)$, de modo que o grau de dissimilaridade entre duas versões de uma assinatura fica bem representado por uma medida adequada da distância euclidiana (norma) entre estas curvas. Entretanto, como mostra o exemplo da Figura 3.11, duas configurações absolutamente compatíveis (e que teriam que ser aceitas com idênticas) de uma mesma assinatura podem não estar alinhadas em relação ao tempo, já que é quase impossível uma pessoa assinar sempre com a mesma cadência. Este efeito deve ser levado em conta na determinação de qualquer medida de distância entre as curvas. Para um dado ponto em uma curva A, o ponto de uma curva B “mais próximo” a ele não deve ser tomado como aquele que se encontra no mesmo instante de tempo e sim como o que vai levar a uma distância mínima entre as curvas de maneira global.

Desta forma, o cálculo do grau de dissimilaridade entre duas curvas depende da solução de um problema de determinação da *correspondência ótima* entre pontos selecionados nas mesmas. Este tipo de problema pode ser resolvido com a técnica de programação dinâmica conhecida como *Dynamic Time Warping*, em que o eixo do tempo da curva de referência (padrão) permanece fixo, mas o eixo do tempo da curva em teste é ajustado não linearmente, de modo a minimizar a norma da distância global entre as curvas. O processo de DTW é realizado entre duas curvas, com o mesmo número de pontos ou não, para encontrar a melhor correspondência entre pontos selecionados nas mesmas. De uma maneira geral, o DTW é um recurso de otimização

que pode ser usado sempre que for necessário encontrar a correspondência ótima, ou seja, aquela em que a soma das distâncias entre os pontos selecionados de duas curvas é mínima. É um recurso bastante utilizado no reconhecimento de fala [RAB93] para comparar pronúncias diferentes de um mesmo fonema.

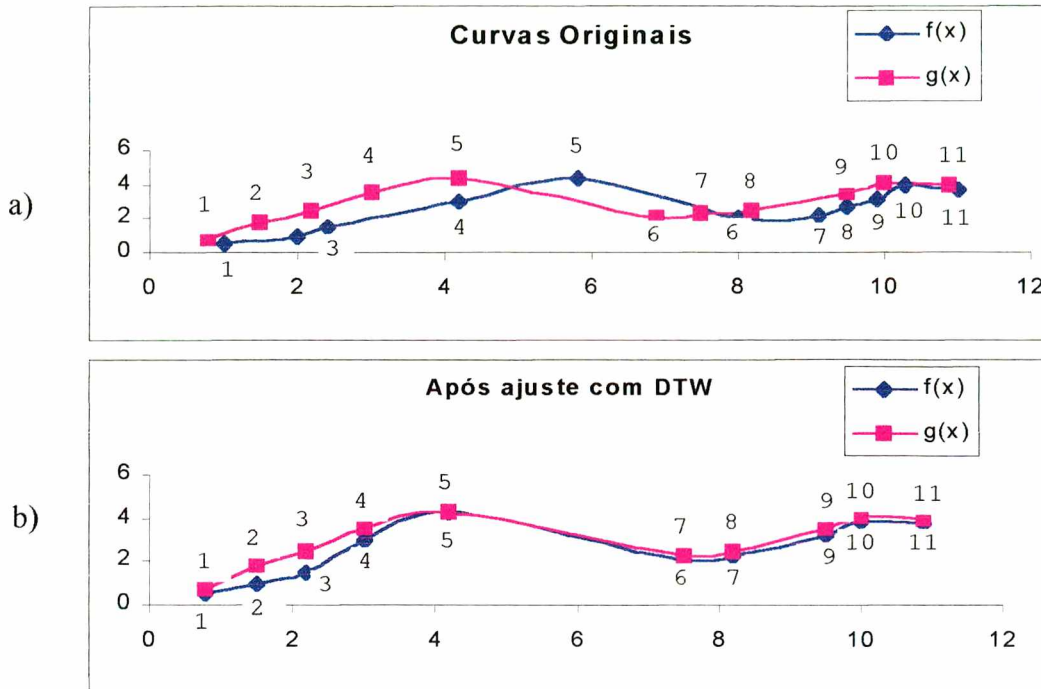


Figura 3.11: (a) Duas curvas, $f(x)$ e $g(x)$, que devem ser consideradas iguais num processo de comparação, embora possuam pequenas diferenças, principalmente em relação ao alinhamento no tempo. (b) Resultado da aplicação do DTW entre as duas curvas, mostrando que, embora os pontos 4 da $f(x)$ e 5 da $g(x)$ tenham ocorrido praticamente no mesmo instante de tempo, não devem ser relacionados.

Para que seja encontrada a melhor relação entre as duas curvas, é necessário atribuir "custos" para a associação entre dois pontos. Estes custos são organizados como uma matriz, onde o elemento $a_{i,j}$ contém o valor da distância euclidiana entre o ponto i da curva $f(x)$ e o ponto j da $g(x)$. A partir desta matriz, é montado um plano com $T_f \times T_g$ pontos, como o da Figura 3.12, onde a cada ponto deste plano está associado um custo dado pelo elemento de mesma posição na matriz e T_f é o número total de pontos da curva f e T_g é o total de pontos da curva g . Encontrar a correspondência ótima equivale a isolar uma seqüência de pontos que leve de $(1,1)$ a (T_f, T_g) tal que a soma dos custos seja mínima.

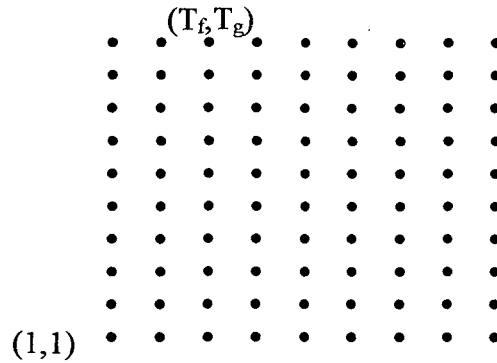


Figura 3.12: Plano com $T_f \times T_g$ pontos, sendo $T_f = 9$ e $T_g = 10$.

Descrição do Processo de Otimização

Com o plano montado a partir da matriz de custos, inicia-se o processo para encontrar a função de *warping* que indica o melhor caminho, levando em consideração que:

1. As funções devem iniciar em $f(1)=1$ e $g(1)=1$ e terminar em $f(T)=T_f$ (no exemplo $T_f=9$ pontos) e $g(T)=T_g$ (no exemplo $T_g=10$ pontos).
2. Um ponto da função $f(x)$ não pode estar relacionado com um ponto da $g(x)$ anterior no tempo. Isto elimina a possibilidade de associações reversas no tempo, mesmo em pequenos intervalos (condição de monotonicidade). Esta condição é essencial no reconhecimento de fala, pois sem ela, conforme Rabiner [RAB93], o DTW poderia levar a uma medida de dissimilaridade pequena mesmo entre palavras tão distintas como “we” e “you”. No caso de verificação de assinaturas, esta condição evita o estabelecimento de correspondências “cruzadas” entre pontos, o que parece ser adequado a uma combinação natural entre curvas.
3. Existem outras restrições ao conjunto de caminhos possíveis que são derivadas de observações práticas. Para simplificar, estas restrições estão organizadas em grupos de “restrições de caminho local”, que indicam quais os movimentos permitidos (que obedecem às restrições impostas) na busca do caminho ótimo. Alguns dos conjuntos de restrições locais mais utilizados são apresentados na Figura 3.13.

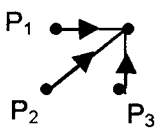
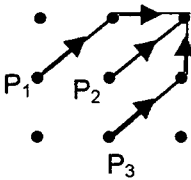
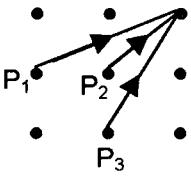
<p>a)</p> 	<p>$P_1 \rightarrow (1,0)$ – um movimento para o lado e nenhum para cima. $P_2 \rightarrow (1,1)$ – um movimento para o lado e um para cima. $P_3 \rightarrow (0,1)$ – nenhum movimento para o lado e um para cima.</p>
<p>b)</p> 	<p>$P_1 \rightarrow (1,1)(1,0)$ – um movimento para o lado e um para cima mais um movimento para o lado e nenhum para cima. $P_2 \rightarrow (1,1)$ – um movimento para o lado e um para cima. $P_3 \rightarrow (1,1)(0,1)$ – um movimento para o lado e um para cima mais nenhum movimento para o lado e um para cima.</p>
<p>c)</p> 	<p>$P_1 \rightarrow (2,1)$ – dois movimentos para o lado e um para cima. $P_2 \rightarrow (1,1)$ – um movimento para o lado e um para cima. $P_3 \rightarrow (1,2)$ – um movimento para o lado e dois para cima.</p>

Figura 3.13: Exemplos de conjuntos de restrições locais (movimentos permitidos sobre o diagrama de pontos durante o processo de busca do caminho ótimo).

Tais conjuntos podem variar, mas terão sempre em comum o movimento $P_2 \rightarrow (1,1)$, necessário para que o alinhamento linear seja possível. Estas restrições são expressas em termos de caminhos permitidos para se atingir um determinado ponto no plano em um único passo.

4. As restrições locais fazem com que parte do plano seja excluída da região onde é possível que passe o caminho de *warping*. Ou seja, tais restrições tornam certas regiões inatingíveis, pois só são válidas regiões que podem ser alcançadas a partir $(1,1)$ e podem terminar em (T_f, T_g) . O efeito disto é mostrado no gráfico da Figura 3.14.
5. Para que seja possível dar preferência a um certo passo ou penalizar outro, é feita uma ponderação nos passos. Assim, cada um dos conjuntos de passos P_i recebe um peso (m), de tal forma que para ir de um ponto a outro o custo passará a ser o custo existente na matriz de custos multiplicado pelo peso atribuído ao passo que está sendo utilizado. Para o conjunto de caminhos locais apresentado, uma ponderação possível é apresentada na Figura 3.15.

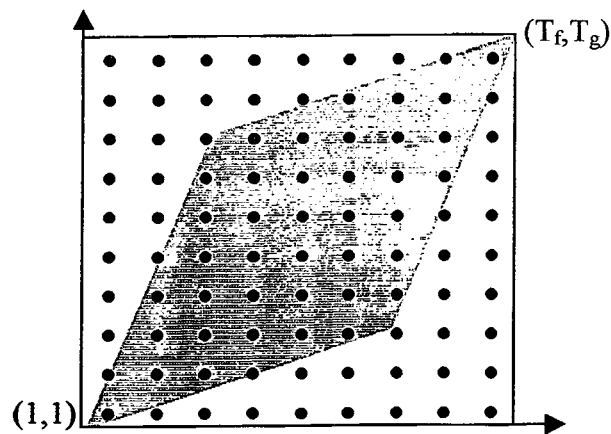


Figura 3.14: O espaço por onde o caminho de *warping* pode passar está reduzido à área sombreada por causa das restrições de caminho aplicadas.

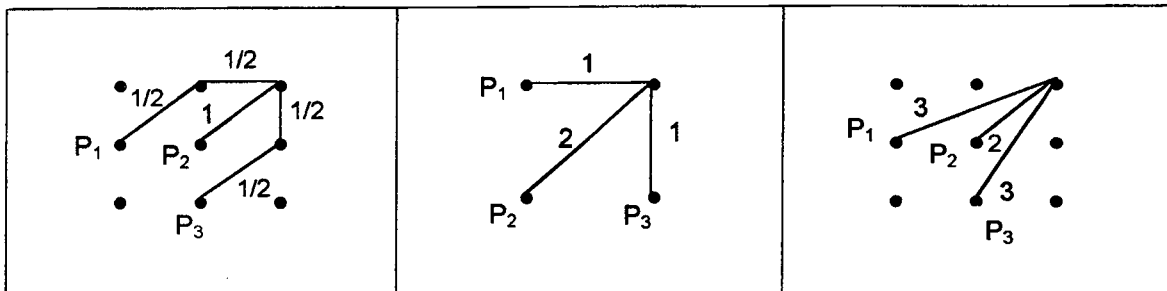


Figura 3.15: Caminhos locais com seus respectivos pesos (m).

O algoritmo de programação dinâmica utilizado para encontrar o melhor caminho através de uma malha T_f por T_g , iniciando em (1,1) e terminando em (T_f, T_g) pode ser resumido nos três passos a seguir:

1. *Inicialização*:

$$D_A(1,1) = d(1,1)m(1) \tag{3.14}$$

Onde $d(1,1)$ é a distância para ir do ponto 1 de uma função para o ponto 1 da outra e $m(1)$ é o peso.

2. *Recursão*: A recursão monta a árvore de caminhos possíveis. Começando do último ponto (T_f, T_g) , para cada ponto (i_f, i_g) deve-se procurar entre os candidatos permitidos (i'_f, i'_g) , conforme ilustra a Figura 3.16, qual aquele que, após a soma da distância entre (i_f, i_g) e (i'_f, i'_g) , representada por $d((i'_f, i'_g), (i_f, i_g))$, $(i_f,$

i_g)), leva a uma distância total mínima. Apenas valores (i_f, i_g) que podem ser alcançados a partir de $(1,1)$ e podem terminar em (T_f, T_g) , respeitando as restrições de caminho local, são avaliados neste passo de recursão. Matematicamente, isto pode ser descrito como:

Para $1 < i_f < T_f, 1 < i_g < T_g$ compute

$$D_A(i_f, i_g) = \min_{i'_f, i'_g} [D_A(i'_f, i'_g) + \zeta((i'_f, i'_g), (i_f, i_g))] \quad (3.15)$$

Onde, ζ é o custo para ir diretamente do ponto (i'_f, i'_g) para o ponto (i_f, i_g) multiplicado pelo peso m referente ao caminho local utilizado. A determinação de $D_A(i'_f, i'_g)$, por sua vez, depende da verificação do caminho mínimo até (i'_f, i'_g) , de modo que será preciso repetir este passo para cada um dos (i'_f, i'_g) permitidos, conforme as restrições de caminho local escolhidas para a implementação. Este processo se repete recursivamente até que o único candidato a verificar seja o ponto $(1,1)$.

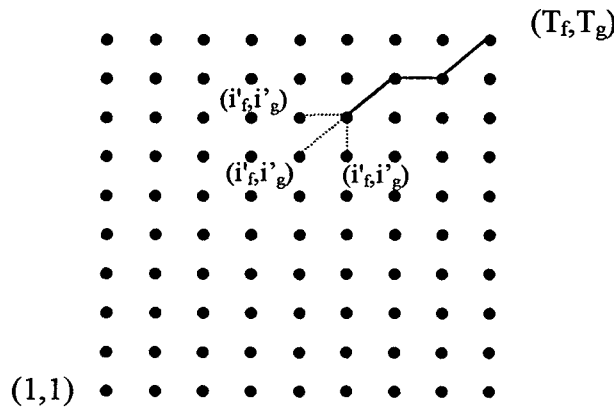


Figura 3.16: Plano de pontos com o início do caminho de warping.

3. *Término*: Ao final, atribui-se à distância entre as duas funções o custo acumulado mínimo, normalizado por M_ϕ , para compensar o efeito da aplicação dos pesos m ao longo do caminho (ver Figura 3.15).

$$d(F, G) = \frac{D_A(T_f, T_g)}{M_\phi} \quad (3.16)$$

3.7 DTW Não Recursivo

Na forma recursiva descrita anteriormente, o algoritmo para *Dynamic Time Warping* teria complexidade $O(3^n)$, onde n é o maior número de pontos das curvas. Isto seria muito lento e não permitiria comparações que incluíssem centenas de pontos de cada lado, como é o caso da comparação direta entre as funções. Por esta razão, para que fosse possível considerar as coordenadas $x(t)$ e $y(t)$ como características, foi implementada uma versão não recursiva deste mesmo algoritmo. Basicamente, este novo algoritmo consiste em dois passos:

1º passo: Geração da matriz de custos (Igual ao recursivo)

Dadas duas curvas, uma com T_f pontos e outra com T_g , o primeiro passo necessário para se encontrar a melhor correlação entre elas é atribuir custos para a associação entre seus pontos. Estes custos são organizados na forma de uma matriz, onde o elemento $a_{i,j}$ contém o valor da distância euclideana entre o ponto i da curva 1 e o ponto j da curva 2. Encontrar a correspondência ótima equivale a isolar uma seqüência de elementos (posições) da matriz que leve de (1,1) até (T_f, T_g) tal que a soma dos custos seja mínima.

2º passo: Avanços com o L

Para que o DTW possa ser feito de maneira não recursiva, ao invés de se iniciar pelo ponto final da matriz (T_f, T_g) , começa-se pelo extremo oposto (1,1) e são feitos sucessivos avanços até que (T_f, T_g) seja atingido. Tais avanços são feitos na forma de um L invertido (\lrcorner ou \neg), com lados de tamanho L_H (parte horizontal) e L_V (parte vertical) dados por:

$$L_H = \frac{T_0 * \sqrt{T_f^2 + T_g^2}}{T_g} \quad (3.17)$$

$$L_V = \frac{T_0 * \sqrt{T_f^2 + T_g^2}}{T_f} \quad (3.18)$$

onde T_0 é um parâmetro de controle adicional, definido a seguir.

Este L invertido percorre a matriz até que seu canto alcance a posição (T_f, T_g) , em um avanço mostrado esquematicamente na Figura 3.17. Assim, com base nos caminhos

locais permitidos, são calculados os custos acumulados dos elementos que compõem o L. No primeiro L, elementos que não alcançam o (1,1) têm atribuído custo infinito, pois não podem vir a fazer parte do caminho ótimo. Por exemplo, no primeiro L da Figura 3.17, a restrição indicada na Figura 3.13 (b) faz com que seja atribuído um custo diferente de infinito somente a $a_{2,2}$. Cada vez que o L avança, todos os resultados anteriores necessários ao cálculo dos custos associados aos elementos internos ao L já estão disponíveis, o que elimina a necessidade de recursão. O tamanho do L depende do número de pontos das duas curvas e é definido de maneira que, ao avançar, o L invertido percorra apenas as regiões permitidas pelas restrições de caminho local e pelo parâmetro T_0 .

O parâmetro T_0 é escolhido com base nas restrições globais dos caminhos de *warping* e limita a quantidade de elementos que podem vir a fazer parte do caminho ótimo. Este parâmetro, acrescentado ao algoritmo DTW por Sakoe e Chiba [SAK78], é empiricamente definido sobre o plano de pontos (ver Figura 3.18) e sua função é aumentar a eficiência do algoritmo de busca pelo caminho ótimo. Por exemplo, elementos que, respeitando a restrição de caminhos locais escolhida (item b da Figura 3.13), não permitem atingir (T_f, T_g) são automaticamente eliminados do espaço de busca através do T_0 . Além disso, T_0 elimina elementos que têm pouca chance de fazer parte do caminho ótimo e define o que Sakoe e Chiba [SAK78] chamam de janela de ajuste. É justamente o parâmetro T_0 que viabiliza a implementação de uma versão não-recursiva do algoritmo para o DTW.

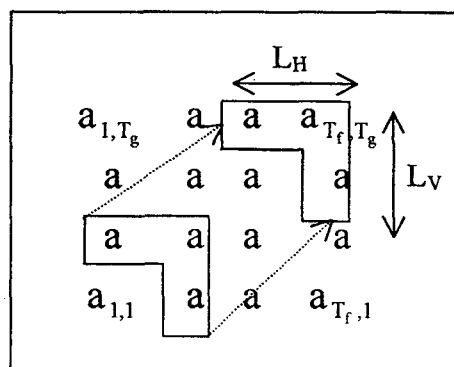


Figura 3.17: Sentido de avanço dos cálculos com o DTW não recursivo.

Com esta versão mais eficiente, que não repete cálculos já efetuados, o algoritmo adquire complexidade linear no número de pontos da maior curva e permite a comparação entre funções discretas com centenas de pontos sem a preocupação com tempos de processamento excessivamente grandes.

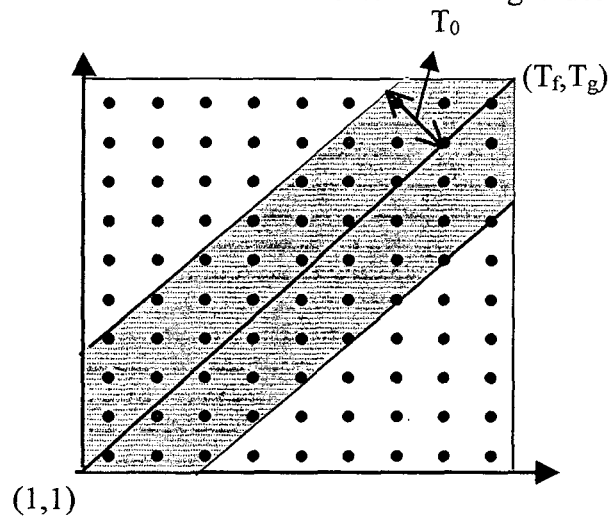


Figura 3.18: O espaço por onde o caminho de *warping* pode passar é reduzido à área sombreada pelas restrições de caminho e pelo parâmetro T_0 .

Capítulo 4

Descrição do Sistema

Com base na revisão de literatura (ver Capítulo 2), optou-se pela adoção de um sistema para verificação *on-line* de assinaturas incluindo:

- assinatura representada por funções (método direto): para garantir que todos os detalhes originais sejam repassados ao sistema;
- tratamento em um plano transformado: a aplicação de uma transformada permite que os esforços computacionais sejam concentrados apenas sobre os aspectos mais relevantes de uma assinatura.

O sistema desenvolvido captura as assinaturas como funções discretas $x(t)$ e $y(t)$ e utiliza a transformada *wavelet* antes da análise dos dados. A transformada *wavelet* foi escolhida por:

- ser completa, possibilitando que a assinatura seja analisada em diversos graus de resolução;
- ser flexível, possibilitando a escolha de um nível de resolução adequado;
- ser compacta, uma vez que a curva original pode ser resumida por apenas alguns parâmetros (as “passagens pelo zero”) de uma curva de detalhe adequadamente escolhida da transformada;
- apresentar cálculo eficiente, pois o algoritmo da transformada rápida *wavelet* (FWT) é muito rápido (praticamente $O(n)$ [STR96]);

- já ser utilizada na caracterização de outras biometrias, como íris [BOL 98] e retina [LEA 01].

4.1 Metodologia Geral

Os itens a seguir deste trabalho estão organizados de forma a contemplar os aspectos apontados por Plamondon [PLA 89] como relevantes no desenvolvimento de um sistema típico para verificação de assinaturas:

- aquisição de dados;
- pré-processamento;
- extração de características;
- processo de comparação;
- avaliação do desempenho.

Um sistema de verificação de assinaturas, abrangendo as etapas acima, pode ser organizado de acordo com o esquema apresentado na Figura 4.1. A avaliação do desempenho, que consiste na obtenção das taxas de erro FAR e FRR, é realizada posteriormente, na fase de teste do sistema.

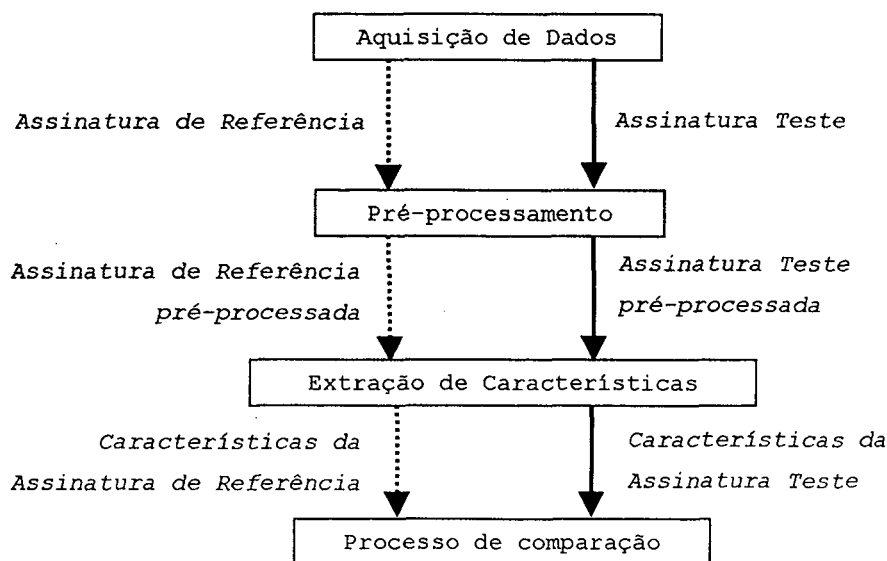


Figura 4.1: Esquema de um sistema de verificação de assinaturas, onde os passos indicados pelas setas pontilhadas são executados uma única vez, pois se referem ao processo de cadastramento.

4.2 Aquisição de Dados

A aquisição de dados foi feita com o auxílio de uma prancheta eletrônica do tipo *Graphire*, da marca Wacom (<http://www.wacom.com>), como a da Figura 4.2. Esta prancheta possui uma área de captura de 12,76 x 9,28 cm e taxa máxima de transmissão de dados de 100 pontos por segundo.



Figura 4.2: Prancheta eletrônica *Graphire* utilizada para captura da assinatura.

Por não ser de cristal líquido, esta prancheta apresenta a característica indesejada de não possibilitar a visualização da assinatura, o que a torna diferente do meio usual (papel e caneta) e prejudica uma possibilidade de *feedback* durante a escrita. Entretanto, a qualidade do dispositivo de aquisição de dados deve influenciar apenas nas taxas de erro obtidas na fase experimental do trabalho.

4.3 Pré-processamento

Antes de submeter os dados para a etapa de extração de características, é necessário que estes sejam discretizados e normalizados. Este pré-processamento é necessário para que seja enviado para a transformada *wavelet* um sinal discreto, com espaçamento uniforme e sem influências de possíveis rotações e deslocamentos da assinatura, decorrentes de diferentes situações ou posições no momento da execução da assinatura.

4.3.1 Amostragem

A amostragem consiste da obtenção do sinal discreto, que será submetido à transformada *wavelet*, a partir dos dados fornecidos pela prancheta. Este processo

envolve muitos detalhes e exige muitos cuidados, de modo que a sua descrição foi dividida nas 11 etapas mostradas a seguir:

1. Os dados que chegam da prancheta produzem duas curvas de coordenadas, x (mm) e y (mm), em função de uma coordenada t (ms), obtida a partir de um relógio interno do computador. Além disto, pelas características técnicas da prancheta (origem no alto à esquerda), a coordenada y cresce de cima para baixo, ou seja, a prancheta produz duas curvas conforme apresentado na Figura 4.3.

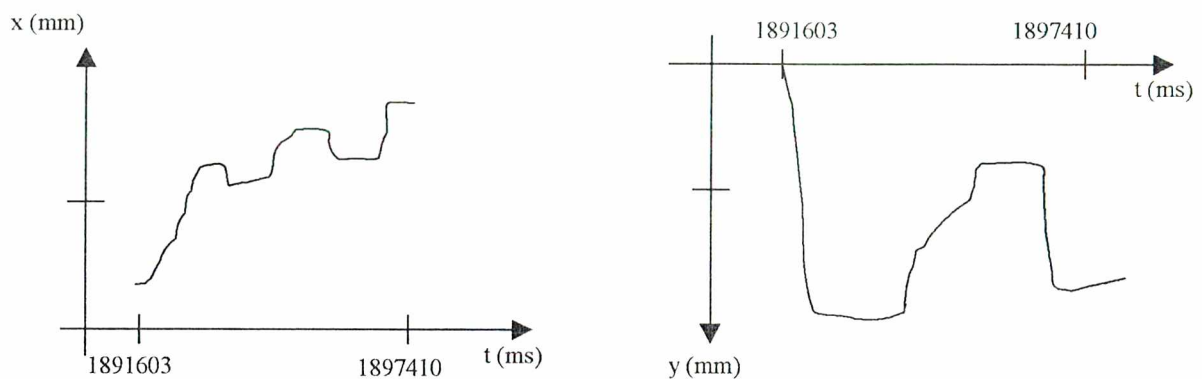


Figura 4.3: Curvas, capturadas através da prancheta eletrônica, representando as posições das coordenadas x e y durante a assinatura.

2. Em função disto, uma primeira correção a ser feita consiste em rebater a curva y (apenas para que a figura da assinatura dada por $x(t)$ x $y(t)$ apareça na posição correta) e padronizar a coordenada t (para que inicie em zero), da seguinte forma:

$$\bar{y} = y_{\text{máx}} - y(t) \quad \text{e} \quad \bar{t} = t - t_{\text{mín}} \quad (4.1)$$

Isto produz o efeito apresentado na Figura 4.4.

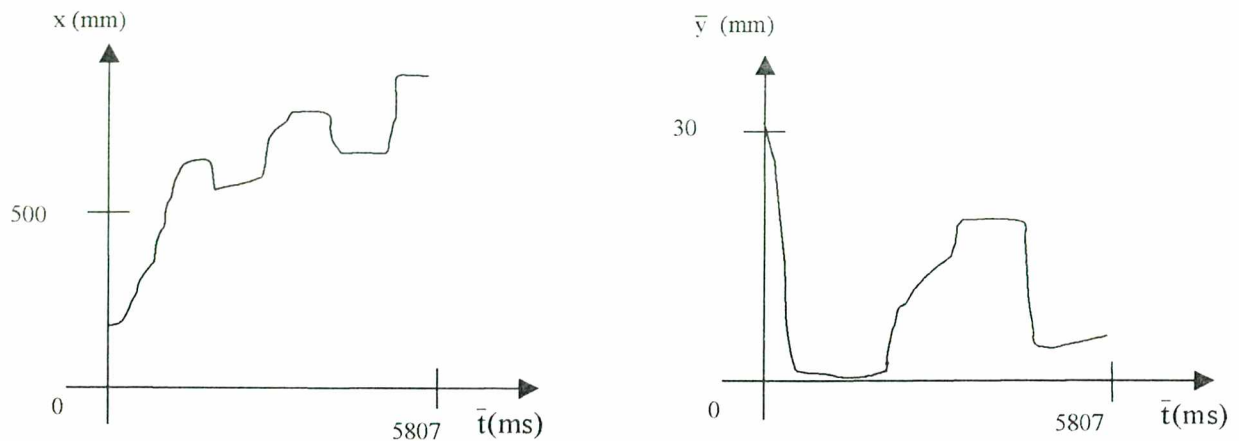


Figura 4.4: Curvas x e y com o tempo começando em zero.

3. O próximo passo consiste em *normalizar* as curvas, a fim de submetê-las ao tratamento com *wavelets*, mas antes é preciso realizar uma *reamostragem* dos pontos, pois, apesar da forma discreta, os pontos fornecidos pela prancheta (167 pontos, no exemplo) não estão uniformemente espaçados, como mostra a Figura 4.5.

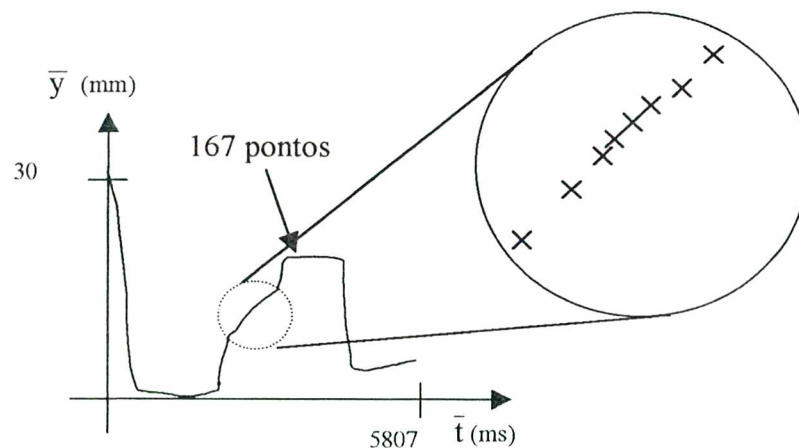


Figura 4.5: Função discreta original fornecida pela prancheta com pontos não uniformemente espaçados.

4. Esta reamostragem deve ser feita em uma frequência tal que obedeça ao teorema de Nyquist, o qual determina que "*a frequência de amostragem de qualquer curva*

contínua deve ser maior do que duas vezes a máxima freqüência presente em sua transformada (contínua) de Fourier" [OPP 89].

5. A curva é fornecida pela prancheta em forma discreta (não uniforme), mas os pontos estão suficientemente próximos para justificar que o seu conjunto, *com interpolações lineares nos espaços vazios*, possa ser considerado uma curva *contínua*, como mostra a Figura 4.6.

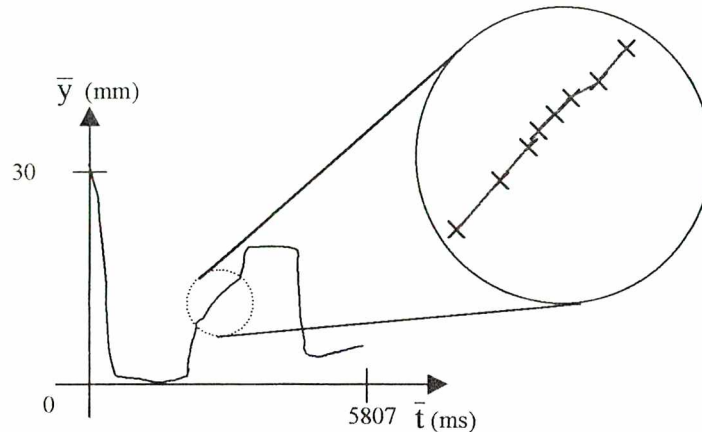


Figura 4.6: Representação da função discreta original, com pontos não uniformemente espaçados, como uma curva contínua.

6. É a transformada de Fourier da curva contínua assim obtida que deve ser utilizada para a obtenção da máxima freqüência presente neste sinal e, por conseguinte, da freqüência mínima com que ela pode ser amostrada para que o sinal discreto resultante represente fielmente a curva contínua.
7. Mas, dada a forma desta curva, a sua transformada de Fourier *contínua*:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (4.2)$$

somente pode ser obtida *numericamente*. Porém, conforme Briggs e Henson [BRI95], uma aproximação numérica para esta integral é proporcional à sua DFT (transformada discreta de Fourier), da seguinte forma:

$$\hat{f}(\omega_k) \approx AF_k \quad \text{onde} \quad F_k = \frac{1}{N} \sum_n f_n e^{-j2\pi nk/N} \quad (4.3)$$

onde A é o tamanho total do domínio do tempo (neste caso, $A=5807$) e os f_n são pontos igualmente espaçados, escolhidos na $f(t)$ original.

8. Mas a obtenção de uma aproximação numérica para a transformada deve se basear em um número suficiente de pontos de $f(t)$ (uma *amostragem* adequada) para que a DFT realmente se constitua em uma boa aproximação numérica. Pode-se adotar um processo iterativo, mas, considerando-se a quantidade de pontos das $x(t)$ e $y(t)$ originais, assume-se que é suficiente tomar o dobro do número de pontos, reposicionados sobre a curva interpolada de modo a gerar intervalos iguais, como mostra a Figura 4.7. Com isto, obtém-se uma curva \bar{y}_N idêntica à original, mas composta de pontos igualmente espaçados, o que permite que ela seja escrita como uma *seqüência* como a mostrada na Tabela 4.1, e a seqüência $\bar{y}_N[n]$ assim obtida pode então ser submetida a uma DFT, obtendo-se:

$$(\bar{Y}_N)_k = \frac{1}{N} \sum_n (\bar{y}_R)_n e^{-j2\pi nk/N} \tag{4.4}$$

que é uma aproximação para $\hat{y}(\omega)$ = transformada contínua de Fourier de $y(t)$.

Tabela 4.1: Seqüência de pontos igualmente espaçados.

N	\bar{y}_N (mm)
0	30,0
1	29,8
2	27,5
3	...

Devido à aproximação, este processo permite conhecer $\hat{y}(\omega)$ apenas nos pontos utilizados para a DFT, e a relação entre os pontos numéricos e contínuos é a seguinte:

- k = freqüências *angulares* adimensionais;
- $\Delta f = 1/A$ = menor freqüência dimensional (em Hz) que pode ser associada com $[0;A]$;
- $d\omega = 2 \times \pi \times \Delta f$ = menor freqüência *angular* dimensional (rad/s) possível, logo: $\omega_k = k \cdot 2 \cdot \pi \cdot (1/A)$

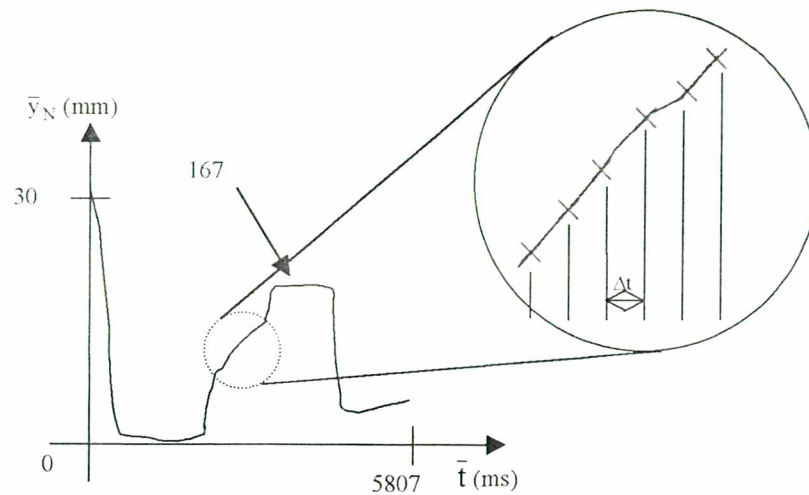


Figura 4.7: Curva com pontos reamostrados uniformemente espaçados.

9. Agora é preciso buscar o valor de ω_k que delimita a parte do domínio transformado que não pode ser considerada nula:

$$\omega_c = k_c \cdot 2\pi \cdot (1/A) \quad (4.5)$$

onde k_c é o máximo índice da parte (considerada) não nula. Dada a periodicidade da DFT, quando ela é tomada sobre um intervalo $[0;N]$ (e não $[-N/2;N/2]$), as altas frequências ficam concentradas em torno de $k=N/2$, como mostra a Figura 4.8.

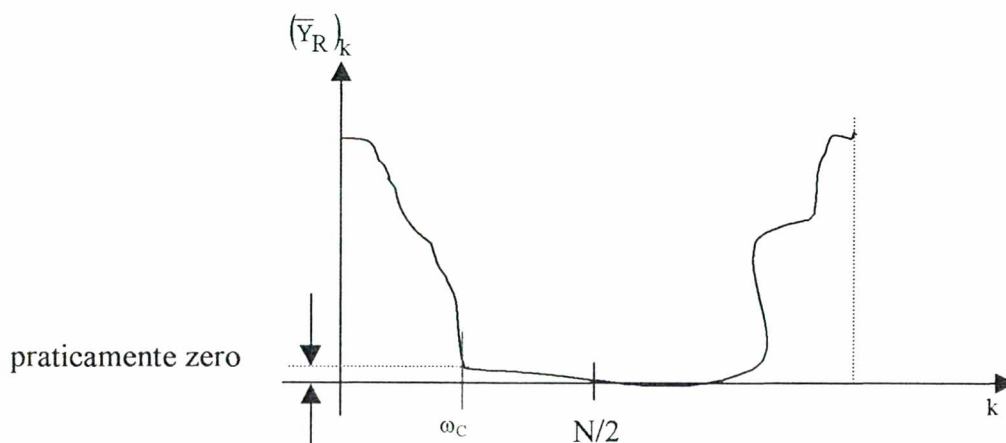


Figura 4.8: Curva da DFT tomada sobre um intervalo $[0,N]$, com ω_c indicando a frequência máxima.

10. Finalmente, chega-se à frequência angular de amostragem que será usada, dada por:

$$\omega_{am} = 1,2 \times (2 \times \omega_c) \quad (4.6)$$

Esta frequência é 20% maior do que o valor mínimo exigido por Nyquist e implica um período de amostragem (s) dado por:

$$\Delta t_{am} = \frac{2\pi}{\omega_{am}} \quad (4.7)$$

Como a amostragem deverá ser feita sobre um sinal adimensional, este valor precisa ser adimensionalizado:

$$\Delta t_* = \frac{\Delta t_{am}}{t_{m\acute{a}x} - t_{m\acute{i}n}} \quad (4.8)$$

Na prática, Δt_* será um valor padrão, calculado como o máximo obtido entre os Δt_* obtidos para as assinaturas cadastradas e será o mesmo para todas as assinaturas a serem verificadas (depois de adimensionalizadas).

11. O sinal interpolado agora deve ser amostrado respeitando o valor máximo de Δt_* dado pela Equação 4.8. Na prática verificou-se que é suficiente fazer uma amostragem com o dobro do número original de pontos e é isto que é, de fato, feito no sistema. Sobre este sinal deve ser efetuada ainda uma *normalização* com o objetivo de eliminar diferenças elementares afins (basicamente rotação e translação), antes de submeter o sinal a uma transformada *wavelet*. O processo de normalização utilizado está descrito no item a seguir.

4.3.2 Normalização

Neste trabalho, a normalização é feita da seguinte forma:

1. Inicialmente, o tempo é adimensionalizado, de modo a eliminar a influência de pequenas variações no tempo total para produzir uma assinatura - assume-se que "eventos semelhantes ocorrem em momentos correspondentes relativamente à duração total da assinatura" [LEE 96]. Assim, o novo intervalo de tempo passará a ser [0,1], o que é feito da seguinte forma:

$$t_* = (t - t_{m\acute{i}n}) / (t_{m\acute{a}x} - t_{m\acute{i}n}) = \bar{t} / (t_{m\acute{a}x} - t_{m\acute{i}n}) \quad (4.9)$$

As curvas x e y ficarão como as apresentadas na Figura 4.9 (aonde já aparece também o efeito do próximo passo).

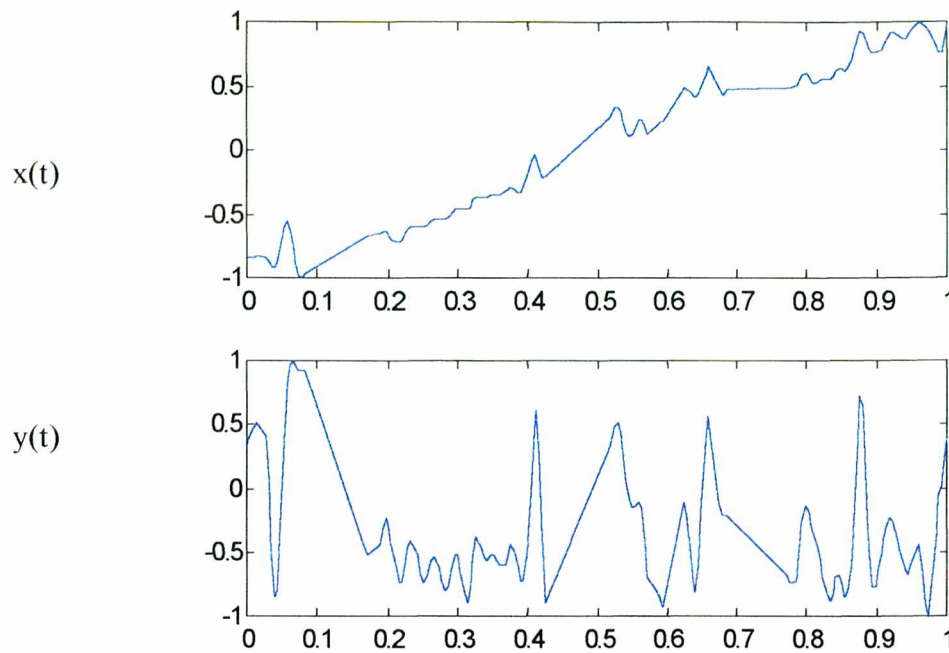


Figura 4.9: Curvas normalizadas quanto ao tempo.

- Depois é efetuada a normalização no espaço, fazendo com que x e y variem no intervalo de $[-1,1]$. O efeito dos passos 1 e 2 sobre duas assinaturas verdadeiras é mostrado na Figura 4.10.

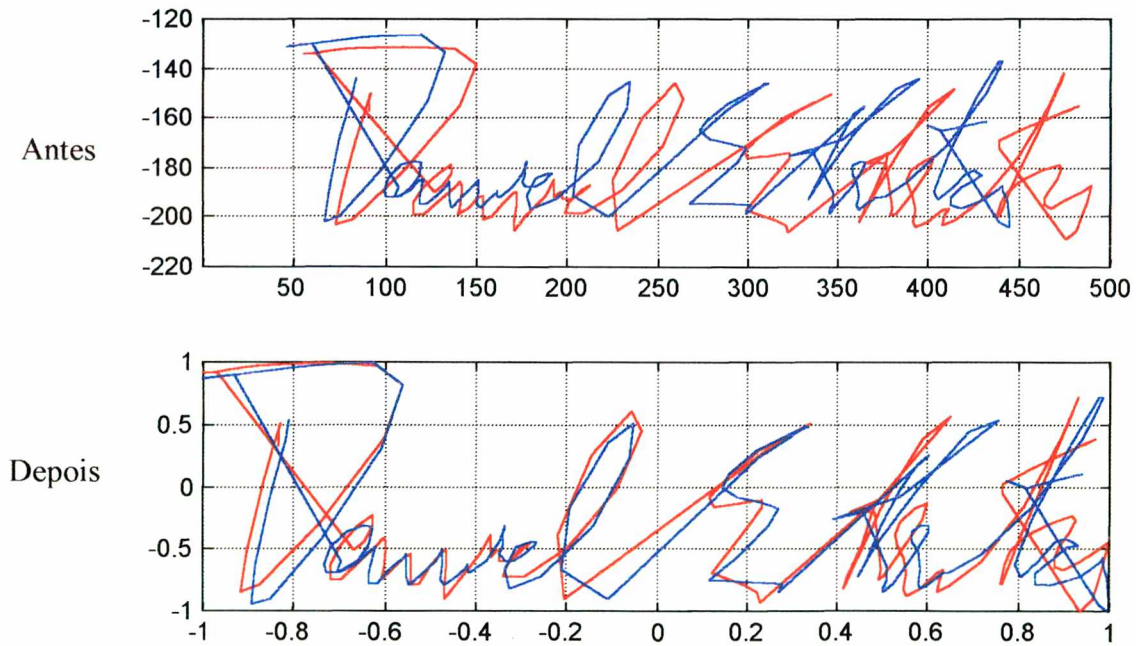


Figura 4.10: Efeito dos dois primeiros passos da normalização sobre duas assinaturas verdadeiras.

3. Para eliminar a rotação, primeiro são encontradas as retas que melhor se ajustam às curvas $x(t)$ e $y(t)$, segundo o critério dos mínimos quadrados, como aparece na Figura 4.11. Tais retas produzirão no plano (x,y) uma reta que representa o eixo principal da assinatura. O ângulo calculado entre este eixo principal e o eixo horizontal fornece a medida da rotação que é preciso aplicar à assinatura para levá-la para a horizontal. Isto é feito com os seguintes passos:

- a) montar a variável complexa $x + i.y$;
- b) aplicar $\bar{z} = ze^{-i\omega}$, onde ω é o ângulo (em radianos) que define a rotação a ser aplicada.

O efeito desta etapa é mostrado na Figura 4.12.

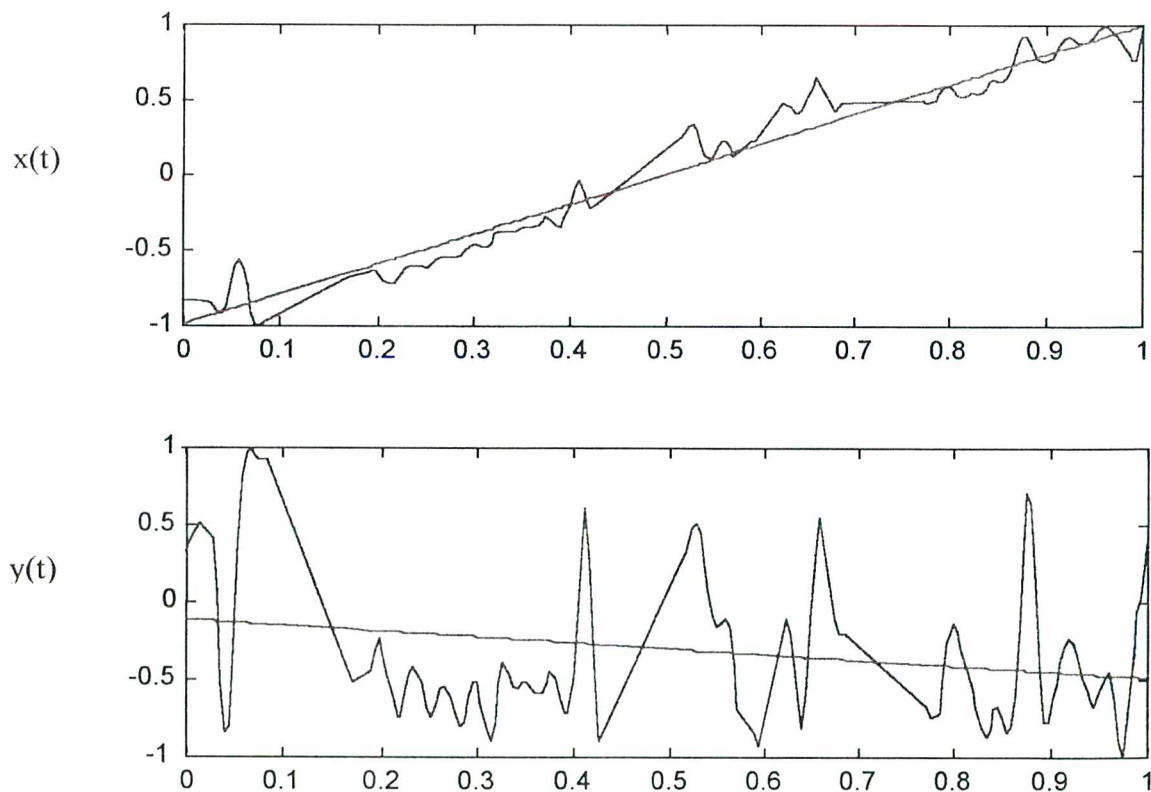


Figura 4.11: Retas que melhor aproximam as curvas $x(t)$ e $y(t)$.

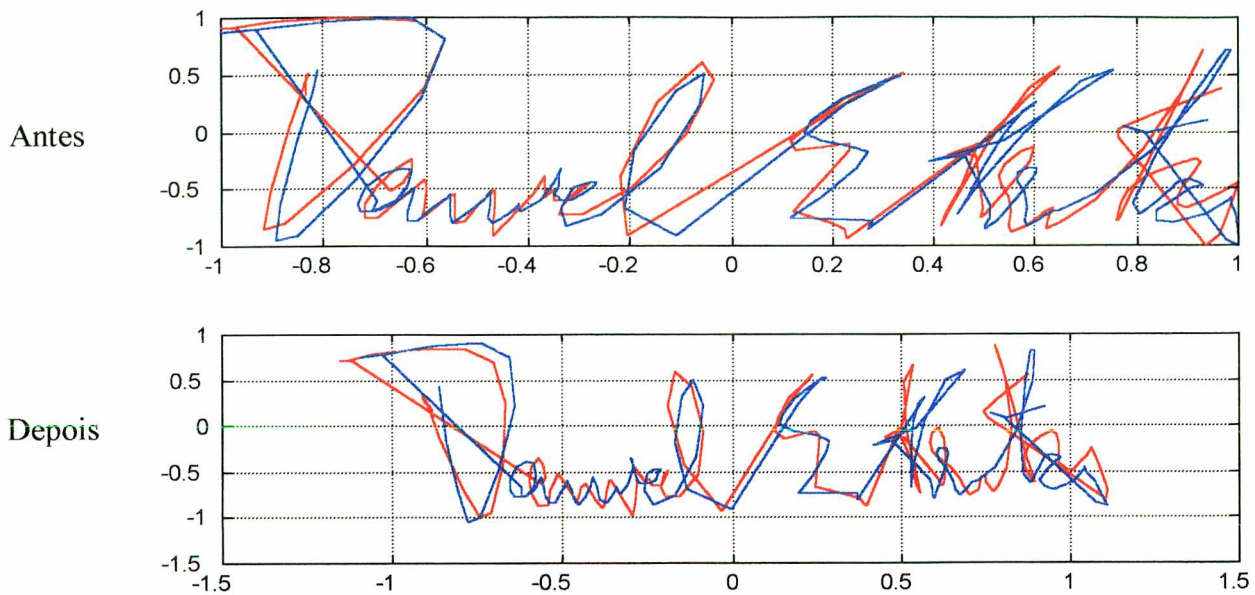


Figura 4.12: Efeito da etapa de rotação do pré-processamento.

4. É possível verificar na Figura 4.12 que as assinaturas rotacionadas estão um pouco deslocadas do intervalo de $[-1,1]$. Quanto maior for o ângulo de rotação, maior será este deslocamento. Para eliminar este possível desajuste, as curvas são novamente normalizadas, voltando ao intervalo $[-1,1]$, como mostra a Figura 4.13.
5. Finalmente, é calculado o centróide da assinatura (x_c, y_c) , para que este seja colocado na origem do sistema coordenado x-y, fazendo:

$$x = x - x_c \quad \text{e} \quad y = y - y_c \tag{4.10}$$

onde x_c e y_c são dados por:

$$x_c = \int_0^1 x(t) dt \quad \text{e} \quad y_c = \int_0^1 y(t) dt \tag{4.11}$$

O efeito do alinhamento pelo centróide é mostrado na Figura 4.14.

6. Os sinais $x(t)$ e $y(t)$ assim obtidos, apresentados na Figura 4.15, são submetidos a uma análise com transformada *wavelet*.

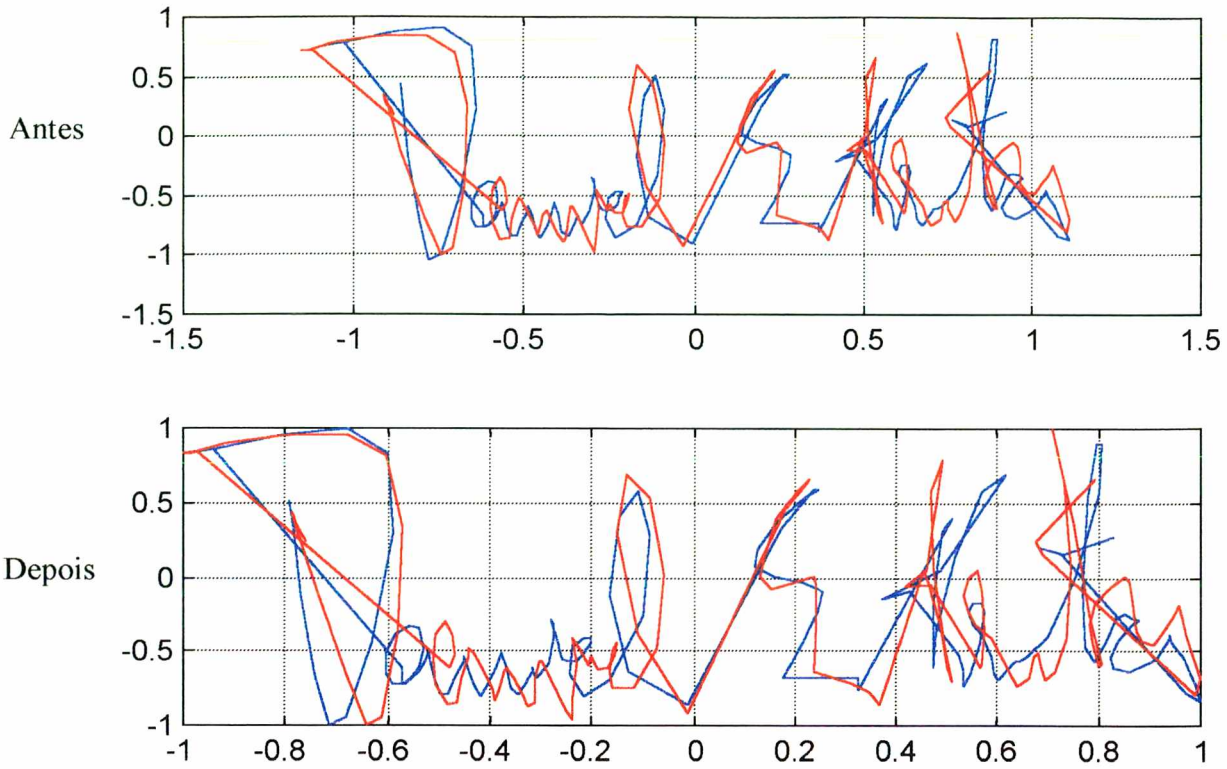


Figura 4.13: Após a rotação, a assinatura é trazida de volta para o intervalo $[-1, 1]$.

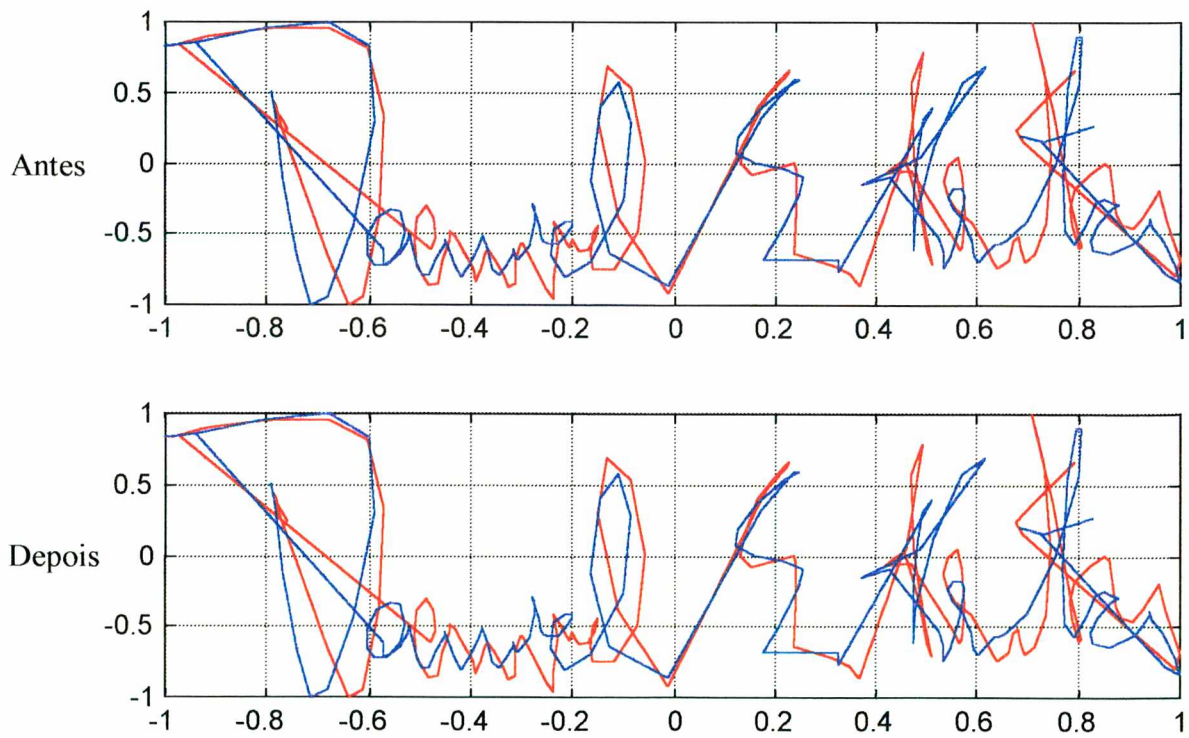


Figura 4.14: Efeito do alinhamento pelo centróide.

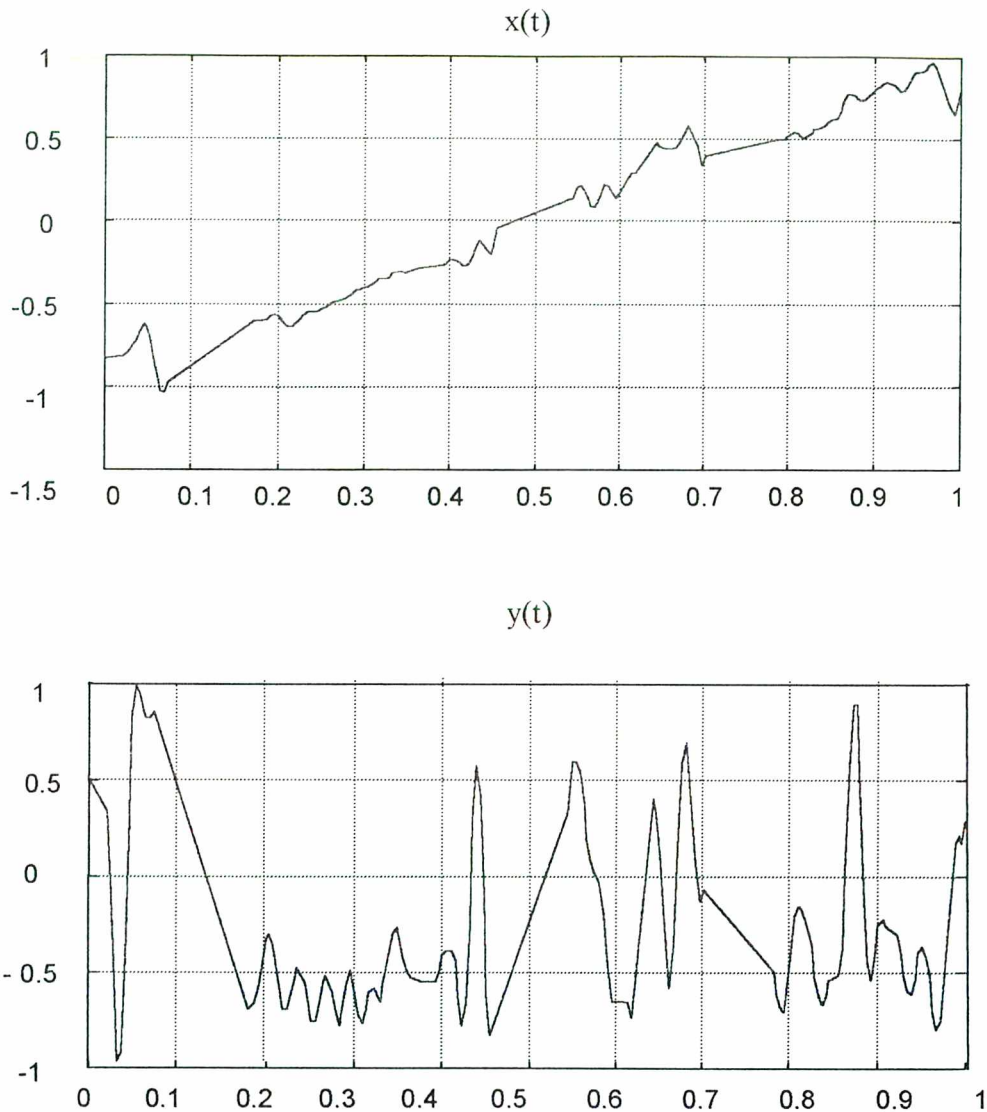


Figura 4. 15: Sinais, $x(t)$ e $y(t)$, pré-processados.

4.4 Extração de Características

Utilizando as Wavelets

Levando-se em consideração os bons resultados obtidos por Deng et al. [DEN97], as *wavelets* foram utilizadas para a extração das características que representarão as assinaturas. A transformada *wavelet* se mostra uma boa opção dada a sua capacidade de decompor hierarquicamente as funções em diferentes níveis de resolução, separando-as em curvas de aproximação e detalhe.

Desta forma, a decomposição *wavelet* das funções $x(t)$ e $y(t)$ pré-processadas gera curvas de aproximação e detalhe, como as mostradas na Figura 4.16, das quais serão extraídas as características. O nível escolhido para a obtenção das informações foi o 4º para todos os usuários. A escolha de um nível único tem como fundamentação os testes realizados por Deng et al., os quais apontaram que a utilização do 4º nível resulta em taxas de erro tão boas (FAR 5,6 e FRR 12,24) quanto aquelas obtidas utilizando-se um nível ótimo para cada indivíduo (FAR 5,6 e FRR 10,98). Assim, para cada passagem pelo zero (pz) da curva de detalhe do quarto nível de resolução, três parâmetros são extraídos:

1. a sua abscissa;
2. o valor da integral entre ela e a passagem pelo zero anterior:

$$v_i = \int_{PZ_{k-1}}^{PZ_k} WD4(t) dt, \quad (4.12)$$

onde $WD4$ indica a curva de detalhe de quarto nível da transformada *wavelet*;

3. a amplitude correspondente a mesma abscissa na curva de aproximação do nível anterior (terceiro nível):

$$v_a = WA3(pz_k), \quad (4.13)$$

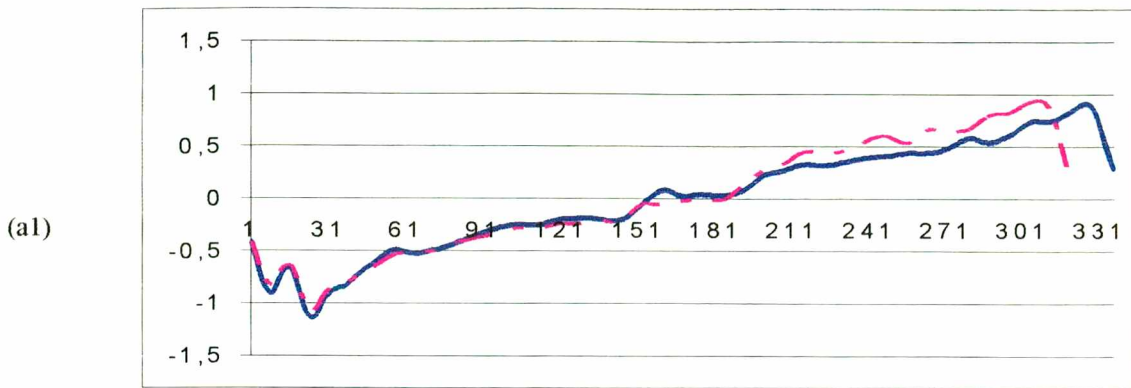
onde $WA3$ indica a curva de aproximação do terceiro nível da transformada *wavelet*.

Estes três atributos foram escolhidos porque são suficientes para a completa reconstrução do sinal original [MAL91] e resultam em características bem significativas.

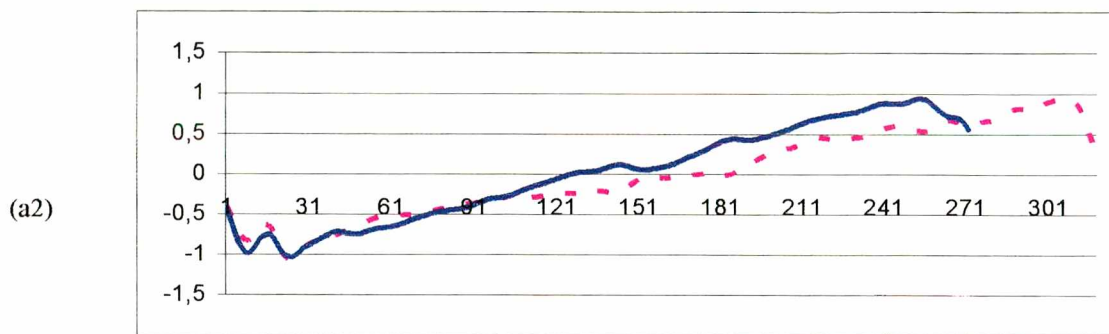
Utilizando as Funções sem Transformadas

A fim de se avaliar a importância da utilização das *wavelets* no processo de verificação de assinaturas manuscritas, foi testado também um método de verificação baseado diretamente nas funções $x(t)$ e $y(t)$. Neste caso, não há a etapa de extração de características, porque as próprias coordenadas das funções (Figura 4.15) são encaradas como características.

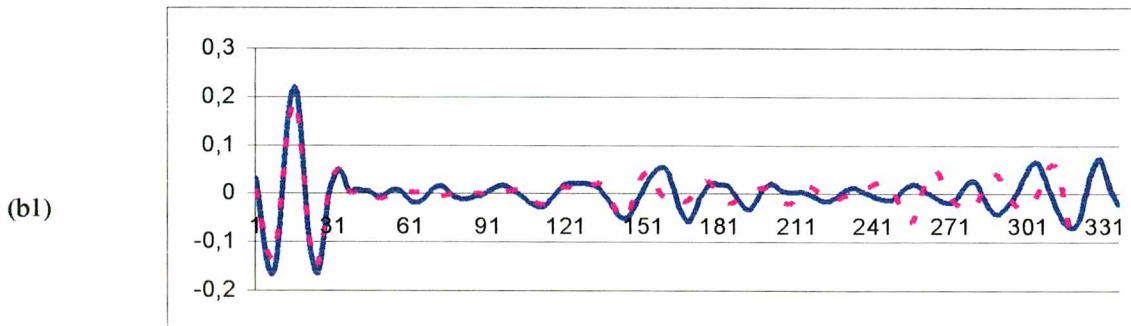
Curvas de Aproximação da Assinatura Padrão e de uma Verdadeira



Curvas de Aproximação da Assinatura Padrão e de uma Falsificação.



Curvas de Detalhe da Assinatura Padrão e de uma Verdadeira.



Curvas de Detalhe da Assinatura Padrão e de uma Falsificação

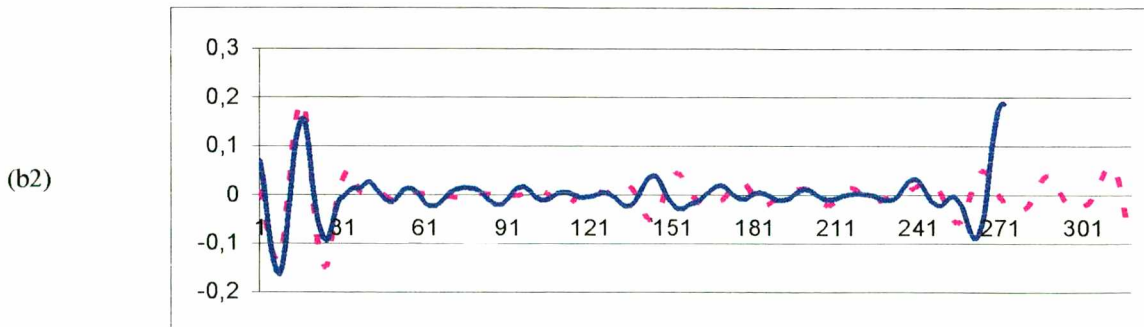


Figura 4.16: Curvas de aproximação e detalhe da assinatura padrão (curvas pontilhadas) de um usuário confrontadas com as curvas de uma assinatura verdadeira e de uma falsificação.

4.5 Cálculo das Medidas de Dissimilaridade

Utilizando as Wavelets

As características obtidas na etapa anterior serão confrontadas a fim de se obter as distâncias da assinatura que está sendo testada (*assinatura teste*) para a assinatura padrão. Contudo, como duas assinaturas originais nunca são idênticas e a quantidade de passagens pelo zero da curva de detalhe é variável, é necessário que o confronto seja feito de maneira não linear, acarretando o melhor ajuste possível entre os vetores de características, ou seja, a menor distância global entre eles. Para isso é utilizada a técnica de programação linear descrita no Capítulo 3, o DTW.

O DTW é aplicado apenas entre as primeiras características (vetores de abscissas das passagens pelo zero das curvas de detalhe da transformada *wavelet* do quarto nível da assinatura teste e da assinatura padrão. Como resultado, será fornecida uma distância ($d1$), que representa a distorção entre as curvas após o ajuste não linear. Além disso, é fornecida a correspondência ótima entre os pontos da curva teste e da padrão. Tal correlação recebe o nome de *caminho*. O caminho é dado por uma matriz de duas colunas $[CAM_{i,j}]$, onde cada linha contém um par de pontos associados pelo DTW. O caminho é necessário para que sejam, então, obtidas as outras distâncias ($d2$, $d3$ e $d4$). As características 2 e 3 são dadas pela norma da diferença da 2ª e 3ª características, respectivamente, para as suas correspondentes na assinatura padrão, de acordo com o caminho, conforme as Equações abaixo:

$$d2_x = \left\| vi_{CAM_{k,1}}^{x,1} - vi_{CAM_{k,2}}^{x,2} \right\|, \quad d2_y = \left\| vi_{CAM_{k,1}}^{y,1} - vi_{CAM_{k,2}}^{y,2} \right\| \quad (4.14)$$

$$d3_x = \left\| va_{CAM_{k,1}}^{x,1} - va_{CAM_{k,2}}^{x,2} \right\|, \quad d3_y = \left\| va_{CAM_{k,1}}^{y,1} - va_{CAM_{k,2}}^{y,2} \right\| \quad (4.15)$$

A Figura 4.17 mostra as curvas das características 2 e 3 da assinatura padrão de um usuário e uma assinatura verdadeira do mesmo usuário, enquanto a Figura 4.18 mostra as mesmas curvas da assinatura padrão, só que comparadas com as obtidas de uma falsificação.

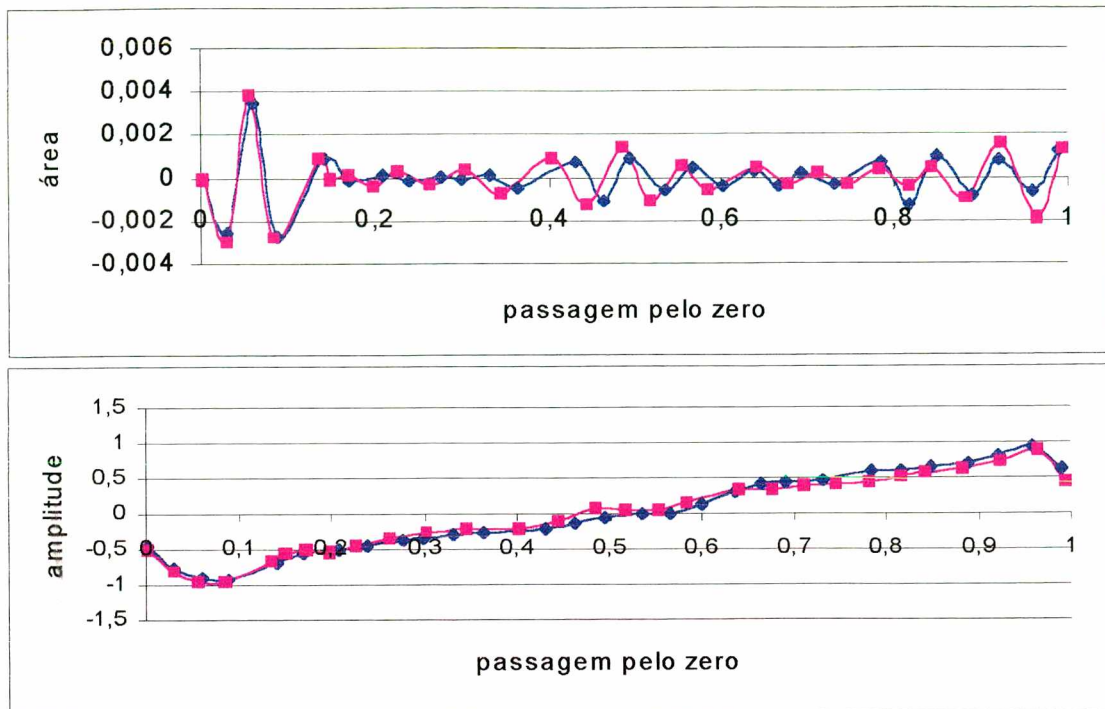


Figura 4.17: Curvas das características 2 (área entre duas passagens pelo zero consecutivas) e 3 (amplitude na curva de aproximação no terceiro nível) da assinatura padrão e de uma verdadeira.

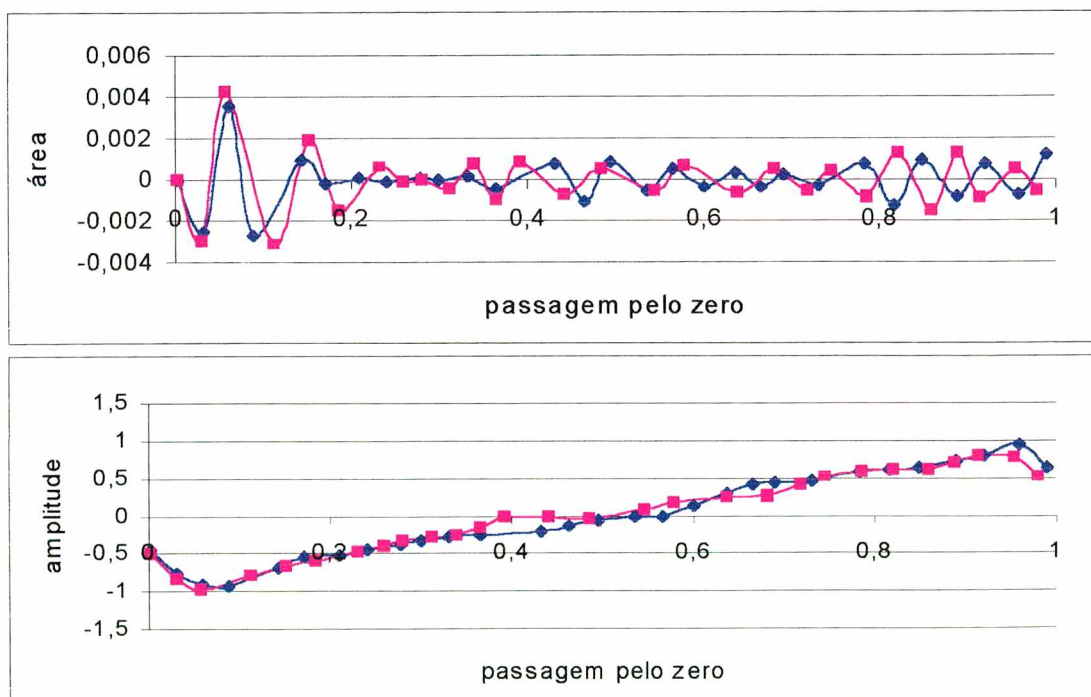


Figura 4.18: Curvas das características 2 (área entre duas passagens pelo zero consecutivas) e 3 (amplitude na curva de aproximação no terceiro nível) da assinatura padrão e de uma falsificação.

A quarta distância ($d4$) é dada pela área da diferença entre a função de *warping* e a diagonal correspondente, como mostra a Figura 4.19. Assume-se que este parâmetro recupera a informação relativa à velocidade, perdida no processo de amostragem em pontos igualmente espaçados. Esta recuperação é possível porque uma grande diferença na velocidade de escrita da assinatura teste em relação à velocidade padrão produzirá uma função de *warping* com maior desvio da diagonal.

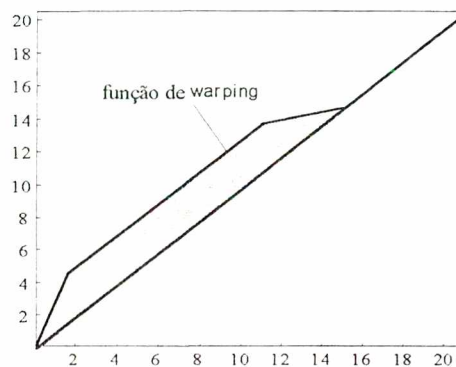


Figura 4.19: A área em destaque, indicando o desvio da função de warping para a diagonal, fornece a quarta distância.

Utilizando as Funções sem Transformadas

No caso da verificação da função, o DTW é efetuado diretamente entre a curva $z = x+y.i$, montada a partir de x e y normalizados, da assinatura teste e a curva z da assinatura padrão. Para esta curva serão obtidas apenas duas distâncias. A primeira é justamente a distorção fornecida pelo DTW (equivalente a $d1$ no caso anterior) e a segunda, Figura 4.20, é a área da diferença entre a função de *warping* e a diagonal correspondente (equivalente a $d4$ no caso anterior). Aqui o DTW é efetuado sobre um número maior de pontos (em torno de 200), daí a necessidade do DTW mais eficiente (não recursivo), apresentado no Capítulo 3. Novamente, leva-se em consideração que duas assinaturas com grande diferença na velocidade de escrita darão origem a uma função de *warping* com maior desvio da diagonal.

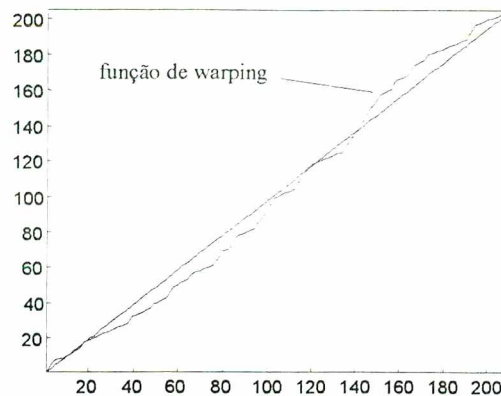


Figura 4.20: A área entre a diagonal e a curva de warping fornece a segunda medida de distância para a comparação da função sem transformada.

4.6 Cadastramento

O sistema verifica se uma assinatura é verdadeira ou não mediante comparação com uma assinatura padrão, escolhida entre as assinaturas que formam a base de referência do indivíduo. Durante o cadastramento serão fornecidas as assinaturas que servirão como referência para cada indivíduo. Neste trabalho, são utilizadas 10 assinaturas de referência, todas coletadas nas mesmas condições das assinaturas de teste. Para as 10 assinaturas de referência são realizados os passos amostragem, normalização e obtenção de características.

Utilizando as Wavelets

Uma vez obtidas as características, são calculadas, duas a duas, com o uso do DTW, as 8 distâncias (4 para x e 4 para y) entre estas assinaturas. Estas distâncias são, então, armazenadas em uma matriz. Desta forma cada assinatura estará associada a uma matriz com oito colunas (uma para cada tipo de distância) e 9 linhas, onde cada linha representa a distância daquela assinatura para cada uma das outras 9 da base de referência. A assinatura que possuir a matriz com menor norma será considerada a padrão e terá suas características armazenadas para serem usadas no processo de verificação.

Além disso, serão armazenados os 8 limiares (um para cada tipo de distância d_i), todos obtidos da seguinte forma:

$$L_i = \bar{d}_i + 1,96.s_i \quad (4.16)$$

onde \bar{d}_i é a média da distância i ($1 \leq i \leq 8$) da assinatura padrão para as outras 9 assinaturas e s_i é o desvio padrão. O valor de 1,96 é escolhido de forma que, assumindo uma distribuição normal, 95% das distâncias das assinaturas verdadeiras até o padrão (d_v) estejam dentro dos limites:

$$\bar{d} - 1,96.s \leq d_v \leq \bar{d} + 1,96.s \quad (4.17)$$

Se o valor de 1,96 fosse substituído por um menor, a taxa de erro de falsas aceitações (FAR) cairia. Isto, porém, causaria um incômodo para o usuário, uma vez que mais assinaturas verdadeiras estariam fora do limite de aceitação. Para diminuir a FAR sem que isto ocorra, é preciso agir sobre \bar{d} e s , o que implica em um processo de otimização baseado na escolha da escala mais adequada da transformada *wavelet*.

Além destes limiares, ainda são armazenados outros dois, referentes à quantidade de passagens pelo zero na curva de detalhe do nível escolhido. Neste caso, são gravados os limiares superior e inferior, calculados da seguinte forma:

$$\begin{aligned} L_{nzcx} &= \overline{nzcx} - 3.s_{zc}, & L_{nzcx} &= \overline{nzcx} + 3.s_{zc}, \\ L_{nzcyc} &= \overline{nzcyc} - 3.s_{zc}, & L_{nzcyc} &= \overline{nzcyc} + 3.s_{zc} \end{aligned} \quad (4.18)$$

onde, \overline{nzcx} e \overline{nzcyc} são as médias da quantidade de passagens pelo zero das curvas WD4x e WD4y, respectivamente, das dez assinaturas do cadastramento.

O uso destes limiares é indicado para que o DTW não seja efetuado entre curvas que tenham uma diferença muito grande na quantidade de passagens pelo zero. Isso deve ser feito para evitar a comparação de curvas tão diferentes que o DTW não consiga achar uma correspondência entre elas. O limiar foi calculado com três desvios padrão para que apenas curvas com uma quantidade muito diferente de passagens pelo zero fossem eliminadas, de forma a não cortar assinaturas verdadeiras, que podem estar sujeitas a pequenas variações.

Utilizando as Funções sem Transformadas

Aqui, os passos para a escolha do padrão são os mesmos, porém a matriz possuirá apenas duas colunas, uma vez que, neste caso, só existem duas distâncias. Conseqüentemente, serão armazenados apenas dois limiares.

Em ambos os casos (curvas transformadas e não transformadas), são gravados ainda outros dois limiares, referentes às características globais da assinatura. Estes dois últimos limiares serão usados na implementação de um filtro detector de falsificações grosseiras. As características globais escolhidas contêm as mesmas informações daquelas apontadas como as mais significativas no primeiro conjunto de características estudado por Lee [LEE96] (ver Capítulo 2). As características globais escolhidas foram:

- Comprimento da assinatura pré-processada (C).
- Tempo total de duração da assinatura (T).

Neste caso, como se deseja um filtro grosseiro, os limiares são calculados da seguinte forma:

$$L_C = \bar{C} \pm 3.s \quad \text{e} \quad L_T = \bar{T} \pm 3.s \quad (4.19)$$

onde \bar{C} e \bar{T} representam o comprimento médio e o tempo médio, respectivamente, considerando-se as 10 assinaturas de referência.

4.7 Verificação

Para poder ser verificada, a assinatura teste deve passar antes pelas etapas de pré-processamento e extração de características, para que, então, sejam calculadas as medidas de dissimilaridade em relação à assinatura padrão do indivíduo. Contudo, só passará para esta última etapa a assinatura que, no caso de se trabalhar com a transformada *wavelet*, apresentar uma quantidade de passagens pelo zero na curva de detalhe do quarto nível compatíveis com aquelas apresentadas pelo indivíduo durante o cadastramento. Desta forma, uma assinatura que gerar uma quantidade de passagens pelo zero que fuja dos limiares daquele usuário será automaticamente considerada falsa, nem sendo efetuado o DTW. Este limiar fornece uma margem ampla de modo que nenhuma assinatura original, dos quatro usuários, foi eliminada por este teste. Contudo, mesmo não sendo rigoroso, este limiar consegue excluir várias falsificações. Isto ocorre

porque mesmo que uma falsificação seja consistente quanto à forma geral, sua curva de detalhe tende a ser mais inconsistente.

A verificação em si consiste em avaliar se todas as 8 distâncias (ou 2 para o caso da função não transformada) obtidas estão dentro dos limiares calculados para aquele indivíduo durante seu cadastramento no sistema.

Capítulo 5

Resultados

O método descrito no capítulo anterior foi implementado em Matlab e seu desempenho foi testado com 4 usuários (2 destros e 2 canhotos), com cada um deles fornecendo 30 assinaturas originais. Entre estas, 10 foram aleatoriamente selecionadas para a escolha de uma assinatura padrão. Além disto, foram produzidas mais 30 falsificações treinadas para cada um dos usuários. As falsificações foram obtidas sem limite de tempo para treino e com acesso a todas as informações, tais como forma e seqüência de execução dos traços.

5.1 Resultados com Wavelets

Em função de testes preliminares, foi escolhida a configuração padrão como sendo: *wavelet-mãe* Daubechies 6 e nível de resolução 4. A partir desta configuração inicial, foi feita uma investigação sobre a influência da escolha da *wavelet mãe* nas taxas de erro, durante a qual foram testadas, além da Daubechies 6, a Db. 1, a Db. 5, a Db. 10 e a Biortogonal 5.5. Os resultados obtidos com estes testes estão apresentados abaixo na Tabela 5.1 e na Tabela 5.2. Além disso, foi feita uma busca pelo nível de resolução mais adequado a este tipo de aplicação da transformada *wavelet*. Os níveis de resolução mais baixos apresentariam uma aproximação muito grosseira, facilitando as falsificações, enquanto que os níveis mais altos normalmente contém um excesso de detalhe, o que inclusive dificultaria a aceitação de assinaturas verdadeiras. Desta forma, os únicos níveis testados foram aqueles ao redor do nível que indicou melhores resultados nos testes preliminares (o nível 4 da configuração padrão). As tabelas 5.3 e 5.4 apresentam os resultados obtidos nesta investigação.

Tabela 5.1: Taxas de falsa aceitação (FAR) para diferentes tipos de *wavelet mãe* (em negrito está a configuração padrão).

FAR				
<i>Wavelet mãe</i>	Usuário 1	Usuário 2	Usuário 3	Usuário 4
Db. 1 – N.4	0%	30%	0%	0%
Db. 5 – N.4	0%	20%	0%	0%
Db. 6 – N.4	0%	17%	0%	3%
Db. 10 – N.4	0%	23%	0%	20%
Bior. 5.5 – N.4	0%	30%	0%	0%

Tabela 5.2: Taxas de falsa rejeição (FRR) para diferentes tipos de *wavelet mãe*.

FRR				
<i>Wavelet mãe</i>	Usuário 1	Usuário 2	Usuário 3	Usuário 4
Db. 1 – N.4	20%	10%	47%	23%
Db. 5 – N.4	20%	20%	47%	40%
Db. 6 – N.4	20%	23%	30%	33%
Db. 10 – N.4	33%	17%	47%	27%
Bior. 5.5 – N.4	20%	30%	33%	30%

Tabela 5.3: Taxas de falsa aceitação (FAR) para diferentes níveis de resolução da transformada wavelet.

FAR				
<i>Nível</i>	Usuário 1	Usuário 2	Usuário 3	Usuário 4
Db. 6 – N.3	0%	43%	3%	23%
Db. 6 – N.4	0%	17%	0%	3%
Db. 6 – N.5	0%	67%	0%	13%

Tabela 5.4: Taxas de falsa aceitação (FRR) para diferentes níveis de resolução da transformada wavelet.

FRR				
Nível	Usuário 1	Usuário 2	Usuário 3	Usuário 4
Db. 6 – N.3	20%	10%	40%	43%
Db. 6 – N.4	20%	23%	30%	33%
Db. 6 – N.5	30%	17%	33%	30%

Os resultados acima ratificam a escolha da configuração padrão inicial. Trocando-se as *wavelets-mãe* e os níveis, algumas taxas de erro até apresentam melhoras, mas apenas de forma localizada. No conjunto das taxas, não há nenhuma mudança significativa que justifique a troca da configuração padrão escolhida inicialmente.

Uma vez confirmada a configuração padrão, foi feita uma pesquisa em cima dos limiares de cada indivíduo nesta configuração, apresentados na Tabela 5.5, e também da relação entre a média e o desvio padrão usados para o cálculo dos limiares, Tabela 5.6.

Tabela 5.5: Limiares individuais, sendo $L = \bar{x} + 1,96.s$.

	Usuário 1			Usuário 2			Usuário 3			Usuário 4		
	L	\bar{x}	s	L	\bar{x}	S	L	\bar{x}	s	L	\bar{x}	S
L1x	0,0012	0,0009	0,0001	0,0005	0,0003	0,0001	0,0007	0,0004	0,0001	0,0005	0,0004	0,0001
L1y	0,0014	0,0009	0,0003	0,0006	0,0004	0,0001	0,0005	0,0004	0,0001	0,0005	0,0004	0,0001
L2x	0,0349	0,0215	0,0068	0,0092	0,0057	0,0018	0,0427	0,0206	0,0113	0,0126	0,0067	0,0030
L2y	0,1238	0,0611	0,0320	0,0385	0,0307	0,0039	0,0291	0,0192	0,0051	0,0555	0,0313	0,0122
L3x	1,3275	0,7752	0,2818	0,6727	0,4909	0,0928	1,2842	0,8728	0,2099	1,1904	0,5213	0,3524
L3y	1,1240	0,7361	0,1979	2,0192	1,2287	0,4033	1,3206	1,0077	0,1596	2,1829	1,3396	0,4284
L4x	0,1578	0,0870	0,0361	0,0862	0,0430	0,0220	0,1134	0,0693	0,0225	0,1027	0,0510	0,0254
L4y	0,0827	0,0417	0,209	0,1160	0,0761	0,0203	0,1117	0,0719	0,0203	0,0591	0,0369	0,0113

A atuação de cada um dos limiares na rejeição de assinaturas verdadeiras é apresentada na Tabela 5.7, enquanto a Tabela 5.8 apresenta a atuação na rejeição de falsificações. Desta forma, procurou-se identificar se algum dos limiares não atuava, podendo ser descartado, ou se atuava de forma a aumentar alguma das taxas de erro. Buscou-se, ainda, relacionar a medida desvio padrão/média (Tabela 5.6) com as

características específicas de cada assinatura e com as taxas de erro, verificando, por exemplo, se assinaturas mais inconstantes tendem a apresentar maiores desvios em relação à média e se isso tem consequência nas taxas de erro.

Tabela 5.6: Relação desvio padrão sobre média de cada um dos quatro usuários.

Desvio Padrão * 100/ Média				
Limiar	Usuário 1	Usuário 2	Usuário 3	Usuário 4
L1x	11	33	25	25
L1y	33	25	25	25
L2x	32	32	55	45
L2y	52	13	27	39
L3x	36	19	24	68
L3y	27	33	16	32
L4x	41	51	32	50
L4y	56	27	28	31
Média	36	29	29	39

A proporção entre desvio padrão e média é apresentada na Tabela 5.6 como um indicativo do grau de variabilidade da assinatura de cada indivíduo. Quanto maior o valor deste parâmetro para um certo limiar, maior é possibilidade de uma falsificação passar como verdadeira e menor é a probabilidade de ocorrer a rejeição de uma assinatura verdadeira com base neste mesmo limiar. Porém, esta última característica pode ser melhor controlada pela própria definição de limiar (Equação 4.16), de modo que o desejável é trabalhar com limiares que apresentem valores baixos para a proporção mostrada na Tabela 5.6. Neste caso, os resultados mostram que os limiares mais consistentes seriam aqueles relativos às curvas $y(t)$. As distâncias calculadas a partir das curvas $x(t)$ variam mais e tendem a ser menos efetivas na identificação de uma falsificação do que aquelas que são obtidas de $y(t)$. Os resultados da Tabela 5.6 mostram ainda que os usuários 1 e 4 possuem assinaturas mais inconstantes. Para os casos mostrados, isto não teve grande influência na aceitação de falsificações, mas certamente as assinaturas dos usuários que mantêm um

maior grau de coerência em sua escrita ao longo de várias amostras tendem a ser mais difíceis de falsificar.

Tabela 5.7: Quantidade de assinaturas verdadeiras cortadas pelos limiares.

Usuário	Lzcx	Lzcy	L1x	L1y	L2x	L2y	L3x	L3y	L4x	L4y
1	0	0	1	0	0	3	0	1	0	1
2	0	0	1	2	0	1	2	1	0	0
3	0	0	1	2	4	1	0	1	1	0
4	1	0	2	5	1	0	0	0	2	1
Total	1	0	5	9	5	5	2	3	3	2
Total			14%	26%	14%	14%	5.5%	8.5%	8.5%	5.5%
Total			40%		28%		14%		14%	

Observação: uma assinatura pode ser cortada por um ou mais limiares, mas quando é cortada pelo(s) limiar(es) da passagem pelo zero, os outros limiares não são nem verificados.

Tabela 5.8: Quantidade de assinaturas falsas cortadas pelos limiares.

Usuário	Lzcx	Lzcy	L1x	L1y	L2x	L2y	L3x	L3y	L4x	L4y
1	26	9	0	0	2	2	0	0	0	2
2	6	3	2	4	2	6	6	18	0	2
3	27	29	0	0	1	1	0	1	0	0
4	25	28	0	0	0	0	0	0	0	1
Total	78	66	2	4	5	9	6	19	0	5
Total			4%	8%	10%	18%	12%	38%	0%	10%
Total			12%		28%		50%		10%	

Observação: uma assinatura pode ser cortada por um ou mais limiares, mas quando é cortada pelo(s) limiar(es) da passagem pelo zero (Lzcx e Lzcy), os outros limiares não são nem verificados.

A partir de uma análise das tabelas 5.7 e 5.8, pode-se notar que o limiar L3 apresenta melhor desempenho, dado que dos quatro limiares (excluindo-se o de passagens pelo zero) é o que corta menos verdadeiras (junto com L4) e mais falsas. Já o comportamento de L1 não é satisfatório, uma vez que corta uma grande quantidade de assinaturas verdadeiras e poucas falsas (em comparação com L3). Tendo isto em vista, foi efetuado um teste no qual L1 foi excluído, os resultados obtidos estão na Tabela 5.9.

Tabela 5.9: Taxas de erro com a configuração padrão, mas sem o limiar L1.

Tipo de Erro	Usuário 1	Usuário 2	Usuário 3	Usuário 4
FRR	17%	17%	23%	17%
FAR	0%	17%	0%	3%

Tais taxas confirmam a idéia de que é melhor trabalhar sem o L1, de forma a diminuir a quantidade de testes efetuados e, ao mesmo tempo, reduzir as taxas de falsa rejeição (FRR), em comparação com aquelas apresentadas com a configuração padrão. Ainda na investigação a respeito dos limiares, a Tabela 5.10 apresenta os resultados de um teste no qual, além do L1, o L4 foi removido. Como o L4 apresentou mais cortes de assinaturas verdadeiras do que falsas (ver tabelas 5.7 e 5.8), as taxas de FRR baixaram ainda mais. Contudo, a falsificação de assinatura do Usuário 4 que este limiar cortava (ver Tabela 5.8) passou a ser aceita, acarretando um acréscimo na taxa de FAR.

Tabela 5.10: Taxas de erro com a configuração padrão, sem os limiares L1 e L4.

Tipo de Erro	Usuário 1	Usuário 2	Usuário 3	Usuário 4
FRR	13%	13%	20%	7%
FAR	0%	17%	0%	7%

Se a aplicação do sistema de verificação de assinaturas não necessitar de segurança máxima, esta última configuração é bastante indicada, pois atinge taxas de erro em um patamar bem razoável para este tipo de biometria.

5.2 Resultados sem Wavelets

As assinaturas coletadas também foram utilizadas para testar um sistema de verificação que utiliza diretamente a função pré-processada $z(z(t) = x(t) + i.y(t))$ como característica, sem que esta passe por nenhuma transformada. As taxas obtidas com este tipo de verificação estão expostas na Tabela 5.11.

Tabela 5.11: Taxas de erro da verificação de assinaturas com base na função não transformada.

Tipo de Erro	Usuário 1	Usuário 2	Usuário 3	Usuário 4
FRR	23%	10%	17%	20%
FAR	0%	17%	0%	0%

Também neste caso foi feita uma investigação acerca dos limiares, expostos nas tabelas 5.12 e 5.13, e o papel de cada um deles na rejeição de assinaturas, tanto verdadeiras quanto falsas (tabelas 5.14 a 5.17). Os testes foram feitos de forma a verificar também a relevância do pré-teste, baseado no tempo e no comprimento. Este pré-teste cortou, de maneira indesejada, algumas assinaturas verdadeiras. Uma investigação a respeito da taxa de falsa rejeição sem a utilização deste pré-teste, mostrada na Tabela 5.18, mostra, contudo, que a maioria das assinaturas rejeitadas pelo pré-teste seria, de qualquer forma, rejeitada pelos outros limiares, de forma que a FRR cairia, mas não significativamente. Somado a isto, o desempenho em relação às falsificações (Tabela 5.17) justifica a utilização de tal teste.

Tabela 5.12: Limiares individuais para verificação com base na função não transformada.

	Limiar 1			Limiar 2		
	L1	$\bar{d}1$	S	L2	$\bar{d}2$	s
Usuário 1	0.0407	0.0237	0.0086	0.0113	0.0078	0.0018
Usuário 2	0.0356	0.0208	0.0076	0.0065	0.0046	0.0009
Usuário 3	0.0389	0.0217	0.0088	0.0068	0.0051	0.0009
Usuário 4	0.0212	0.0131	0.0041	0.0069	0.0038	0.0016

Tabela 5.13: Limiares do pré-teste.

	Tempo (ms)				Comprimento (<i>adimensional</i>)			
	T min.	T máx.	Média	Desvio	C min.	C máx.	Média	Desvio
Usuário 1	2753,8	3835,6	3294,7	180,3	19,32	23,87	21,6	0,75
Usuário 2	3726,7	6302,7	5014,7	429,3	20,60	31,35	26,0	1,79
Usuário 3	4366,6	7544,6	5955,6	529,6	16,80	26,63	21,7	1,64
Usuário 4	3257,1	9863,3	6560,2	1101,0	17,34	23,86	20,6	1,09

Tabela 5.14: Assinaturas verdadeiras cortadas pelo pré-teste.

	Usuário 1	Usuário 2	Usuário 3	Usuário 4
Verdadeiras cortadas no pré-teste pelo Tempo:	1	1	1	0
Verdadeiras cortadas no pré-teste pelo Comprimento:	0	0	1	1
Total de verdadeiras cortadas no pré-teste:	1	1	2	1

Observação: As assinaturas podem ser rejeitadas por apenas um critério do pré-teste ou pelos dois simultaneamente, sendo que as assinaturas que são cortadas aqui não passam para a comparação com o dtw.

Tabela 5.15: Assinaturas verdadeiras cortadas pelas distâncias do dtw.

	Usuário 1	Usuário 2	Usuário 3	Usuário 4
Verdadeiras cortadas pelo L1:	6	0	3	4
Verdadeiras cortadas pelo L2:	0	2	0	2
Total de verdadeiras rejeitadas	7	3	5	6

Observação: As assinaturas podem ser rejeitadas por um ou pelos dois critérios, sendo que as cortadas no pré-teste nem passam por este teste.

Tabela 5.16: Assinaturas falsas cortadas pelo pré-teste.

	Usuário 1	Usuário 2	Usuário 3	Usuário 4
Falsas cortadas no pré-teste pelo Tempo:	17	4	25	1
Falsas cortadas no pré-teste pelo Comprimento:	1	0	16	13
Total de falsas cortadas no pré-teste:	17	4	30	13

Observação: As assinaturas podem ser rejeitadas por apenas um critério do pré-teste ou pelos dois simultaneamente, sendo que as assinaturas que são cortadas aqui não passam para a comparação com o dtw.

Tabela 5.17: Assinaturas falsas cortadas pelas distâncias do dtw.

	Usuário 1	Usuário 2	Usuário 3	Usuário 4
Falsas cortadas pelo L1:	13	8	0	16
Falsas cortadas pelo L2:	12	21	0	13
Total	13	21	0	17

Observação: As assinaturas podem ser rejeitadas por um ou pelos dois critérios, sendo que as cortadas no pré-teste nem passam por este teste.

Tabela 5.18: Taxas de falsa aceitação sem o pré-teste baseado no tempo e no comprimento da assinatura.

Tipo de Erro	Usuário 1	Usuário 2	Usuário 3	Usuário 4
FRR com Pré-teste	23%	10%	17%	20%
FRR sem Pré-teste	20%	7%	13%	17%

Observa-se que o resultado sem a transformada *wavelet* é ligeiramente superior, inclusive zerando a taxa de FAR para mais um usuário (o Usuário 4). Tal fato sugere que a comparação não linear no tempo entre as características (DTW) é o que realmente importa no processo de verificação de assinaturas manuscritas. Além disso, a verificação com *wavelets* utiliza uma quantidade maior de operações em ponto flutuante (aproximadamente 185 *Mflops*) que a verificação sem a aplicação desta transformada, que utiliza em torno de 135 *Mflops*.

Capítulo 6

Conclusão

Neste trabalho, foi implementado um método direto para a verificação eletrônica *on-line* de assinaturas manuscritas baseado em transformadas *wavelet*. Essencialmente, o procedimento implementado consistiu em tratar as funções que descrevem a trajetória da assinatura teste como um sinal de curta duração e aplicar a transformada *wavelet* de modo a obter números que servem como características, ou parâmetros descritivos, da assinatura original. Após serem comparadas de maneira não-linear, com o auxílio do *Dynamic Time Warping* (DTW), com uma assinatura padrão, estas características fornecem um indicativo quantitativo da autenticidade de uma assinatura teste. A fim de verificar a necessidade da utilização da transformada *wavelet*, foi implementado um segundo sistema de verificação, seguindo uma abordagem em que as funções são diretamente confrontadas com a padrão. Neste caso, as características foram as próprias coordenadas dos *pixels* que descrevem cada assinatura e as medidas de dissimilaridade passaram a ser simplesmente as normas das distâncias entre as curvas, cuja correspondência entre pontos também é determinada com o auxílio do DTW.

Ambos os métodos foram testados utilizando-se um banco de assinaturas obtido com o auxílio de uma mesa digitalizadora, e foram determinadas as suas taxas de erro (FAR e FRR). A comparação entre as duas abordagens sugeriu que a *wavelet* não é uma ferramenta essencial na verificação e que é o DTW que exerce o papel fundamental em todos os casos testados. De uma forma geral, no entanto, ambos os sistemas se mostraram razoavelmente eficientes, com taxas de erro aceitáveis para este tipo de biometria.

Referências Bibliográficas

- [BOL98] BOLES, W. W. A Security System Based on Human Iris Identification Using Wavelet Transform. **Engineering Applications of Artificial Intelligence**, [S.I.], p.77–85, 1998.
- [BRI95] BRIGGS, W. L.; HENSON, N. E. **The DFT, An Owner's Manual for the Discrete Fourier Transform**. SIAM - Society for Industrial and Applied Mathematics, 1995.
- [DEN97] DENG, Peter Shaohua; LIAO, Hong-Yuan Mark; HO, Chin Wen; TYAN, Hsiao-Rong. Wavelet-Based Off-Line Signature Verification. **Proc. IEEE**, [S.I.], 1997.
- [GRI00] GRIESS, Friederike Dorothea. On-Line Signature Verification. Project Report. Michigan State University, Department of Computer Science and Engineering, May, 2000.
- [HUA95] HUANG, Kai; YAN, Hong. On-Line Signature Verification Based on Dynamic Segmentation and Global and Local Matching. **Optical Engineering**, [S.I.], v.34, n.12, p.3480–3487, December, 1995.
- [KAS97] KASHI, R. S.; HU J.; NELSON, W. L.; TURIN, W. On-Line Handwritten Signature Verification Using Hidden Markov Model Features. **Proc. IEEE**, [S.I.], p.253–257, 1997.
- [LAM89] LAM, Chan F.; KAMINS, David. Signature Recognition Through Spectral Analysis. **Pattern Recognition**, [S.I.], v.22, n.1, p.39–44, 1989.
- [LEA01] LEANDRO, J. J. C.; CESAR JR., R. M.; JELINER, H. F. Blood Vessels Segmentation in Retina: Preliminary Assessment of the Mathematical Morphology and the Wavelet Transform Techniques. **XIV SIBGRAPI 2001 - Brazilian Symposium on Computer Graphics and Image Processing**, [S.I.], p.84–90, 2001.
- [LEE96] LEE, L. L.; BERGER, T.; AVICZER E. Reliable On-Line Human Signature Verification Systems. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.I.], v.18, n.6, p.643–647, June, 1996.
- [MAL91] MALLAT, Stephane. Zero-Crossings of a Wavelet Transform. **IEEE Transactions on Information Theory**, [S.I.], v.37, n.4, p.1019–1033, July, 1991.
- [MCC97] MCCABE, Alan. Implementation and Analysis of a Handwritten Signature Verification Technique, December, 1997.

- [MUN99] MUNICH, Mario E.; PERONA Pietro. Continuous Dynamic Time Warping for Translation-Invariant Curve Alignment with Applications to Signature Verification. 1999.
- [NAL97] NALWA, Vishvjit S. Automatic On-Line Signature Verification. **Proc. IEEE**, [S.I.], v.85, n.2, p.215–239, February, 1997.
- [NEW00] NEWHAM, Emma. Survey: Signature Verification Technologies. **Bit**, [S.I.], p.8–10, April, 2000.
- [OPP89] OPPENHEIM, A. V.; SCHAFER, R. W. **Discrete-Time Signal Processing**. Prentice-Hall, 1989.
- [PLA00] PLAMONDON, R; SRIHARI, S. N. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.I.], v.22, n.1, p.63–84, 2000.
- [PLA89] PLAMONDON, Rêjean; LORETTE, Guy. Automatic Signature Verification and Writer Identification: The State of the Art. **Pattern Recognition**, [S.I.], v.22, n.2, p.107–131, 1989.
- [PRE92] PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P. **Numerical Recipes in C - The Art of Scientific Computing**. Cambridge University Press, 2nd edition, 1992.
- [RAB93] RABINER, L; JUANG, B. **Fundamentals of Speech Recognition**. Prentice-Hall, 1993.
- [SAK78] SAKOE, H.; CHIBA, S., Dynamic Programming Algorithm Optimization for Spoken Word Recognition. **IEEE Trans. Acoust., Speech, Signal Processing**. Vol. ASSP-26, pp. 43-49, Feb. 1978.
- [SAT82] SATO, Y.; KOGURE, K. On-Line Signature Verification Based on Shape, Motion and Handwriting Pressure. **Proc. 6th. Int. Conf. on Pattern Recognition**, [S.I.], v.2, p.823–826, 1982.
- [STI95] STINSON, D. R. **Cryptography, Theory and Practice**. CRC Press, 1995.
- [STO95] STOLLNITZ, E. J.; DEROSE, Tony D.; SALESIN, David H. Wavelets for Computer Graphics: A Primer. **IEEE Computer Graphics and Applications**, [S.I.], p.76–84, May, 1995.
- [STR96] STRANG, Gilbert; NGUYEN, Truong. **Wavelets and Filter Banks**. Wellesley-Cambridge Press, 1996.

- [SUN99] SUNDARESAN, S.; KEERTHI, S. S. A Study of Representations for Pen based Handwriting Recognition of Tamil Characters. **Proceedings of the Fifth International Conference on Document Analysis and Recognition**. 1999.
- [WIR95] WIRTZ, Brigitte. Stroke-Based Time Warping for Signature Verification. **Proc. IEEE**, [S.I.], p.179–182, 1995.

Apêndice A

Estrutura do Programa

Este apêndice trata da estrutura dos programas desenvolvidos em Matlab para a implementação do método descrito no Capítulo 4. A Figura A.1 apresenta a organização geral do programa de verificação e a Figura A.2 expõe a estrutura do programa de comparação entre duas assinaturas, onde:

- 1) WAVE: programa que calcula a transformada *wavelet* (ver Capítulo 3), obtendo as curvas de aproximação e detalhe nos níveis desejados.
- 2) CHARACTERIST: extrai, a partir das curvas de aproximação e detalhe, as características definidas pelas equações 4.12 e 4.13.
- 3) PADRONIZA: executa a padronização exposta no item 4.3.1.
- 4) NORMALIZA: executa a normalização exposta no item 4.3.2.
- 5) DTW: executa o ajuste não-linear apresentado na seção 3.7.

VERIFICA.M

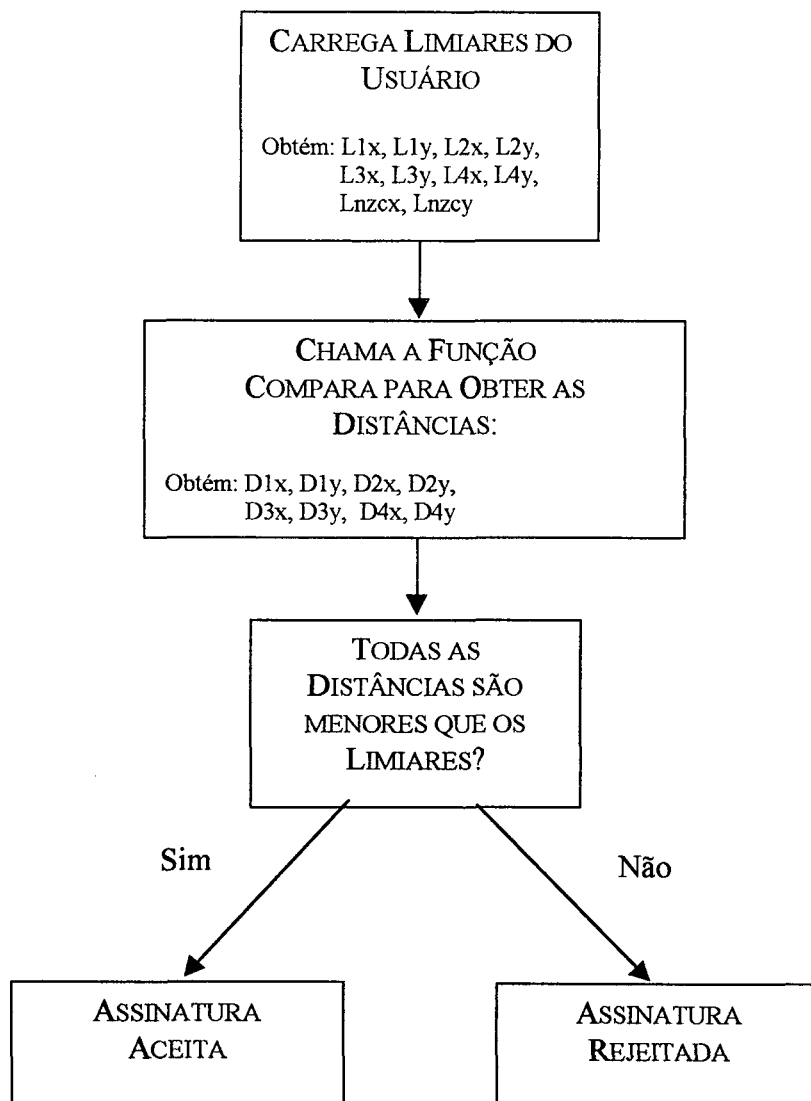


Figura A.1: Estrutura do programa de verificação de assinaturas (verifica.m), que recebe dois parâmetros de entrada: a assinatura teste (ass_teste) e a identificação do usuário, para que se defina o padrão a ser usado na comparação.

COMPARA.M

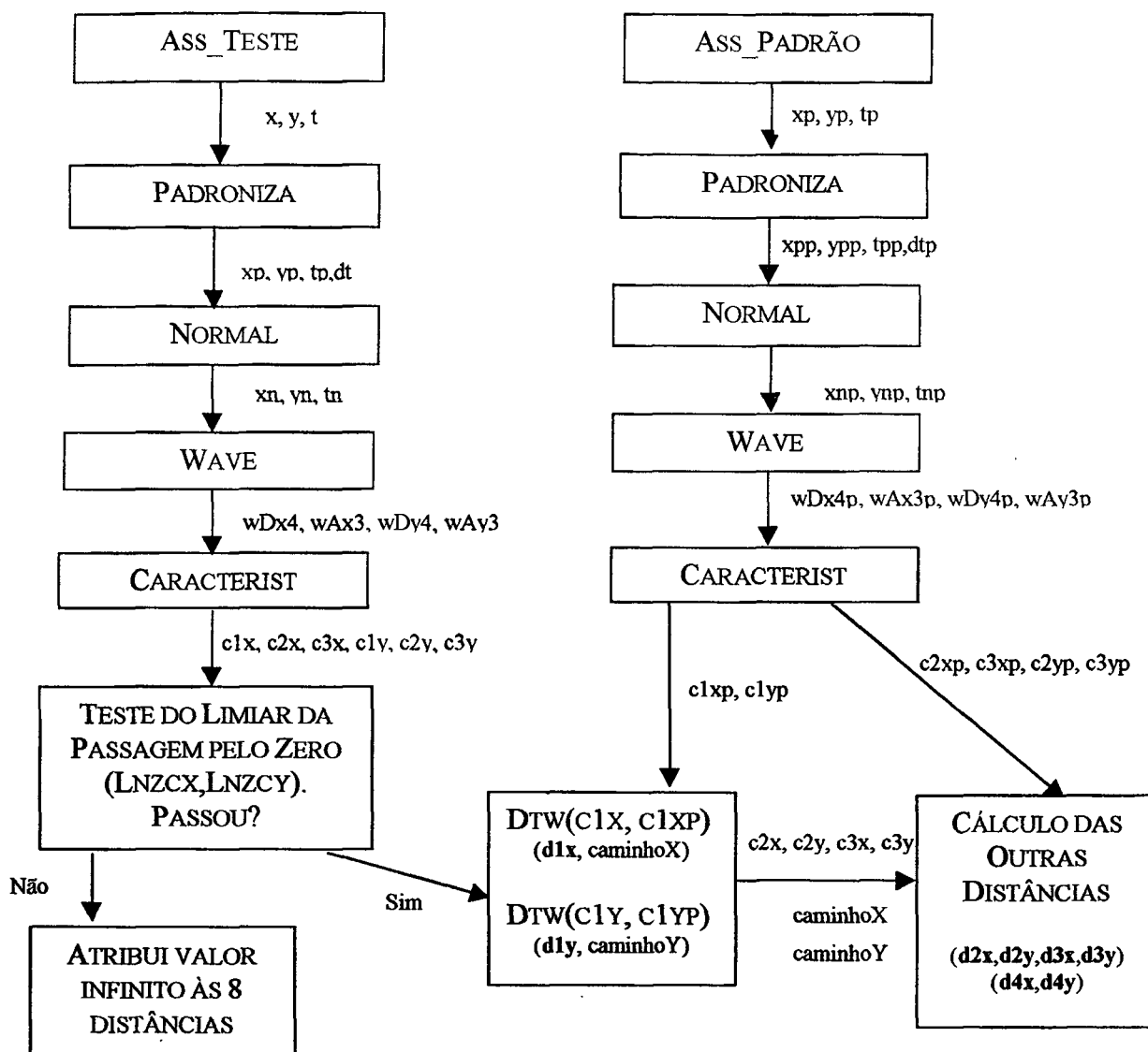


Figura A.2: Estrutura do programa de comparação entre duas assinaturas, que recebe como parâmetro de entrada as assinaturas teste e padrão (ass_teste e ass_padrao) e os limiares das passagens pelo zero (Lanzcx e Lanzcy) e retorna ao *verifica* as oito distâncias calculadas (d1x, d1y, d2x, d2y, d3x, d3y, d4x e d4y).

Apêndice B

Artigo Submetido ao SIBGRAPI 2002

Neste apêndice é apresentado um artigo, produzido a partir desta dissertação, que foi submetido ao Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens. O artigo foi aceito e será publicado nos anais do simpósio que ocorrerá em outubro de 2002.

Wavelet-based Compared to Function-based On-line Signature Verification

ANDRÉA VERGARA DA SILVA, DANIEL SANTANA DE FREITAS

UFSC – Universidade Federal de Santa Catarina - Depto. de Informática e Estatística
Caixa Postal 476, 88049-900 Florianópolis, SC, Brasil
{vergara,santana}@inf.ufsc.br

Abstract. We implemented two direct methods for on-line signature verification. First, data produced by a graphics tablet describing a signature to be tested are treated with wavelet transforms to generate features to be nonlinearly confronted with a reference signature chosen among 10 previously stored tryings from the same writer. In order to recover the time dependence lost in the wavelet treatment, we included the level of departure from the diagonal line in the warping function as a complementary measure of distance. In a second approach, the functions $x(t)$ and $y(t)$ describing position in time of each pixel of the same test signature were directly (though nonlinearly) compared to their counterparts from the reference. We concluded that both approaches showed good fidelity to all details in the signatures, with acceptable false rejection rates (we obtained around 30% FRR) to this kind of biometry. On the other hand, the inclusion of the wavelet transform turned out to be an essential step for the achievement of low false acceptance rates. It was only with the inclusion of the wavelet transform, at the right level of resolution, that we managed to completely prevent trained forgeries to be accepted (0% FAR) in the cases studied.

1. Introduction

This work deals with practical issues related to the safety of digital signatures. Under normal conditions, a digital signature is just an alphanumeric password which confirms authenticity of a document in electronic form based upon solely on the information that this document contains. All the security associated with this scheme relies on the assumption that only the person who signed the document knows the right password (Schneier [10]). However, since the copy of an electronic password cannot be easily distinguished from the original, it is highly recommended the inclusion of additional mechanisms to connect a digital signature and its author, such as the use of tokens (smart cards, for example) or the verification of some physical or behavioral trace (biometrics) collected at the moment of signing. Among all methodologies related to biometrics, we decided to choose signature verification. This choice can be justified because it is very well accepted outside electronic media. People are used to sign papers to confirm authenticity and the signature verification process is not different or intrusive, as the most part of other biometrics (Newham [1]). On the other hand, Ruggles [13] states that a person's signature is prone to variability and this would imply that systems for signature recognition must allow for a wide range of possibilities, being not very reliable.

Signature verification can be performed on-line or off-line. These two approaches and several other practical aspects related to the more general theme of handwriting

recognition are discussed at length in the survey from Plamondon [12]. On-line verification signature verification methods can be further divided into two groups (see Plamondon [12]): direct methods (using the raw functions of time) and indirect methods (using parameters). In the first case, the signature is stored as a discrete function to be compared to a standard from the same writer, previously computed during an enrolment stage. Such methods simplify data acquisition but comparison can become a hard task. On the other hand, indirect methods require a lot of effort preparing data to be processed, but the comparison is quite simple and efficient. Examples of the implementation of indirect methods can be found in the papers from Lee et al. [15] and Griess[16].

Sato and Kogure [3] present one of the first direct methods to be successful. They propose a system that relies on three pseudo-distance measures (shape, motion and writing pressure) derived from coordinate and writing pressure functions through the application of a technique known as Dynamic Time Warping (DTW). They report reasonable error rates (over 90% success) applying their system to Japanese signatures, using forgeries trained for 10 minutes. Some authors tried to improve this idea. Wirtz [4] presents a very similar system but with verification based on strokes (rather than points) as the structural units of the signature. There is also Munich and Perona [8], who propose a continuous DTW as an improvement of precision in the comparison process and Huang and Yan [11], who segment the signature based on the writing

velocity before applying the DTW. None of the modifications seems to represent any significant improvement over the original idea of Sato and Kogure.

Lam and Kamins [5] propose the Fast Fourier Transform as an alternative to time warping. Basically, they suggest that working in the frequency domain would eliminate the need to worry about temporal misalignments between the functions to be compared. They achieve good results and conclude that the FFT can be useful as a method for the selection of features for signature verification. On the other hand, Sundaresan and Keerthi [6], while studying possible ways of representing characters of an Indian language called Tamil, discovered that, because differences among Tamil characters can be very subtle, the best way to represent them is with the aid of wavelet transforms. They noted that Fourier coefficients would not be a good alternative because they are not very sensitive to small variations in style or shape.

The idea of employing wavelet transforms as a means of generating features from signatures appears also in the work from Deng et al. [9]. They propose an off-line verification system that uses wavelet transforms for the decomposition and analysis of the coordinate functions and the tangential angle of points obtained from a signature image. Deng et al. acquire their data from scanned samples of signatures, so they must use image processing techniques to identify closed contours which conveniently represent each signature. Coordinates of the points that compose each contour are stored as polar functions of angles measured counterclockwise with respect to some reference internal to the contour. The functions obtained are submitted to a wavelet transform and the authors take zero-crossings of the detail function in a certain level as features. They complement this set of features with integrals between consecutive zero-crossings and corresponding amplitudes to the same abscissa in the approximation function one resolution up. Experimenting with occidental signatures, these authors were able to achieve error rates as low as 5,6 % false rejection rate (FRR) and 10,98% false acceptance rate (FAR), thus showing the applicability of the wavelet transform to this kind of problem.

Wavelet transforms can also be found in works related to other biometrics. Boles [7], for example, presents a security system that uses the wavelet transform for iris recognition. In order to extract features that are unique to the gray level profiles of the iris image, the author uses (similarly to Deng et al. [9]) the zero-crossings of the detail function and the integrals between consecutive zero-crossings in a certain level of the wavelet transform.

The present work describes an extension of the work from Deng et al. [9] to the case of on-line signature verification. Though a powerful technique, their method for off-line verification cannot be directly applied to the on-line case, mainly because it was developed to deal with situations in which the only available information is the complete signature image. Data obtained on-line do not demand image processing techniques to be used but, besides shape, they provide important additional information which should be considered, such as the order in which points were created in the original signature and the writing velocity. Specifically, we compare the functions $x(t)$ and $y(t)$ describing a test signature with their counterparts in a reference signature, previously chosen in an enrolment stage. Our aim is to determine if this test signature can be taken for original. This comparison is based on dissimilarity measures between features that are obtained from the wavelet transform of the functions $x(t)$ and $y(t)$ through the calculation of certain parameters related to the zero-crossings of the detail function in the right level of resolution to this problem (level 4). These parameters (described below) contain, theoretically, the same amount of information associated to the complete set of coordinates $(x(t), y(t))$. If these dissimilarity measures between the test signature and the reference lie into an acceptable range, established in a previous enrolment stage with 10 original signatures, the test signature is accepted as true.

Before the calculation of the dissimilarity measures, the functions $x(t)$ and $y(t)$ from the test signature must be aligned in time with the corresponding functions in the reference signature. This is accomplished with the Dynamic Time Warping from Sato and Kogure [3]. The information related to the writing motion, lost in a preparation process to the wavelet transform, is recovered with the inclusion of an additional dissimilarity measure, directly extracted from the graphic of the warping function between zero-crossings, given by the area between this function and the diagonal in the dynamic programming diagram. It is similar to the second pseudo-distance of Sato and Kogure, but including the wavelet transform and represents a good measure because forgeries tend to produce warping functions stronger deviated from the diagonal than original samples of the same signature.

Finally, in order to confirm the need of such a sophisticated tool as a wavelet transform in the verification process, we implemented a second method, in which the features are simply the coordinates $x(t)$ and $y(t)$, normalized but not transformed. Similarly to the wavelet approach, the main tool in this case is the warping function that results after the use of the DTW to

dynamically align features. Besides the distance that naturally comes out of the alignment process, we also included the area between the warping function and the corresponding diagonal as a dissimilarity measure. This should take into account, in an indirect way, the information associated with the writing motion in this analysis.

2. Data acquisition

Data acquisition is accomplished with a graphics tablet Graphire from Wacom [14], with 12.76x9.28 cm² active area and maximum data transmission rate of 100 points per second. Budget restrictions prevented us from testing with a LCD pad, which would allow the same kind of feedback provided by usual media (paper and pen). However, the effectiveness of the proposed system does not depend on this factor and the only influence that could be expected from a better quality data acquisition device would be the acquirement of better error taxes in the experimental phase of this work.

The data consisted of original signatures from 4 people (2 right-handed and two left-handed). We took 30 samples from each person and we generated also 30 trained forgeries to each signature. These forgeries were created with free access to all available information, such as shape and sequence of writing of each original signature and with no limitation in time for training. False acceptance rates were computed only with trained forgeries. We used random forgeries in preliminary tests, just to be sure that our system was correctly implemented, but we did not include these results in the FARs reported. Random forgeries are not representative of real situations and their inclusion would tend to dissimulate the error taxes computed. Furthermore, a system that presents a good performance in preventing trained forgeries to be accepted must be effective with random forgeries too.

3. Overview of the system

The signature to be tested is collected from an electronic pad as two functions in time ($x(t), y(t)$) and is numerically processed to generate numbers that represent the distance between it and a reference signature (standard), computed in a previous enrolment stage. If this distance lies inside a statistically acceptable range, the test signature is recognized as being original. The numerical treatment includes resampling to a uniform mesh, correction of elementary distortions between curves (such as spurious displacements and rotations), applying wavelet transforms to produce features and finally nonlinear comparison in time (Dynamic Time Warping).

3.1 Preprocessing

Initially, raw coordinate functions $x_*(t_*)$ and $y_*(t_*)$ are obtained as sequences over a non-uniform mesh in time, presenting an average of 200 points each. These functions are immediately resampled to a uniform mesh with twice the number of nodes in time. Then these resampled data are submitted to the same normalization to location and rotation used by Sato and Kogure [3], producing an effect like the one showed in Figure 1.

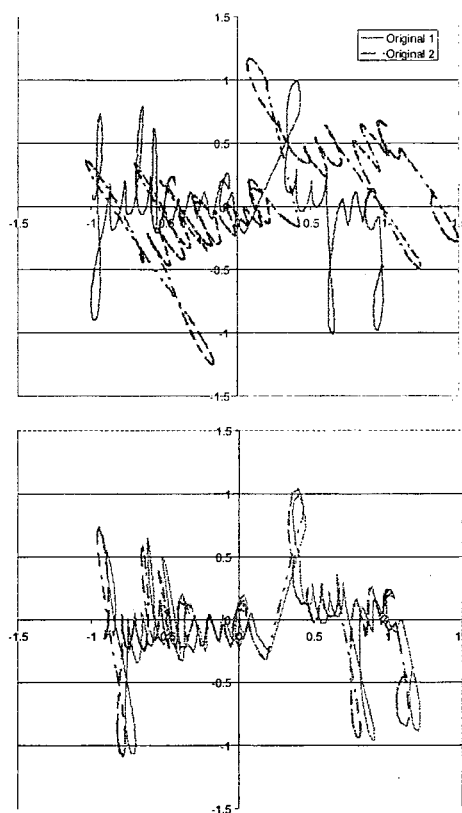


Figure 1 Effect of preprocessing over spurious displacement and rotation.

Specifically, time is changed to the unit interval $[0;1]$ and $x(t)$ and $y(t)$ are changed to $[-1;1]$. In the next step, an auxiliary complex valued function $z(t)=x(t)+i.y(t)$ is defined, whose origin is then situated over the centroid with $\bar{z}(t) = z(t) - z_C$, where z_C is defined as:

$$z_C = \int_0^1 z(t)dt \quad (1)$$

The signature is then aligned with one of its principal axes, with:

$$\hat{z}(t) = \bar{z}(t)e^{-i.\arctan \alpha} \quad (2)$$

where $\alpha = -\beta + \sqrt{1 + \beta^2}$ is the inclination angle of the principal axis, for a value of β given by:

$$\beta = \frac{\operatorname{Re} \left[\int_0^1 (\hat{z}(t))^2 dt. \right]}{\operatorname{Im} \left[\int_0^1 (\hat{z}(t))^2 dt. \right]} \quad (3)$$

3.2 Wavelet-based feature extraction

Considering the good results presented by Deng et al. [9], we decided to use the wavelet transform as the main tool for feature extraction. The wavelet transform is also a good choice because of its capability to allow hierarchical decomposition of functions in different levels of resolution, separating in each level a an approximated shape from the details that complement it. It is a reasonable assumption to suppose that an authentic signature must be consistent with a standard pattern in the details, at a suitable level of resolution. We considered also to use Fourier Transform in place of wavelets, but this option was discarded in view of the remarks from Sundaresan and Keerthi [6]. Fourier coefficients would tend to increase the (false) acceptance of forgeries, since they would not be able to indicate changes of style during the signing process, while these changes can be identified in a natural way in the detail functions of a wavelet transform.

The decomposition of the functions $x(t)$ and $y(t)$ with wavelet transform generates approximations and details like those showed in Figure 2 to an original example of $x(t)$. To each zero-crossing of the detail curve at the 4th level of resolution (this level was chosen empirically, by trial and error), three parameters are extracted: its abscissa, the integral between consecutive zero-crossings (WD4 is the wavelet detail function at the 4th level):

$$vi_k = \int_{z_{k-1}}^{z_k} WD4(t) dt \quad (4)$$

and the corresponding amplitude to the same abscissa in the approximation function one resolution up (WA3 is the wavelet approximation function at the 3rd level):

$$va_k = WA3(z_{k-1}) \quad (5)$$

As it has been demonstrated that this information suffices to a complete reconstruction of the non-

transformed curve (see Deng et al. [9]), these parameters can be used as highly significant features.

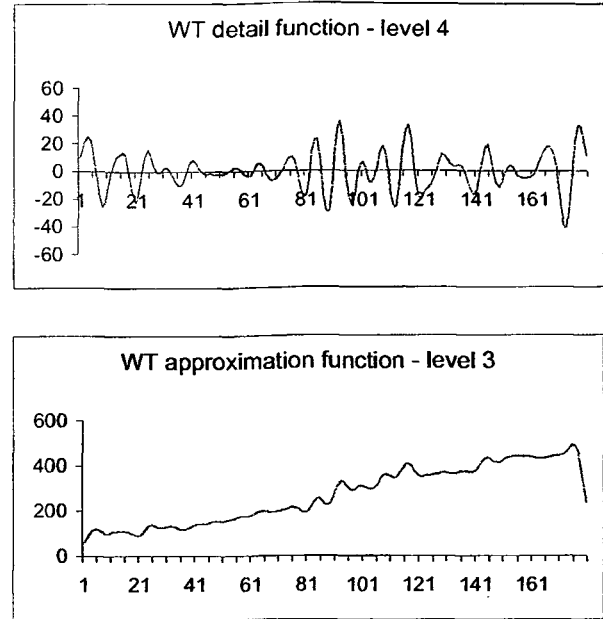


Figure 2 Example of a function $x(t)$ after the wavelet transform.

3.3 Calculation of the dissimilarity measures

The main goal of the feature extraction is to allow for the dissimilarity between two signatures to be quantified. However, even for original samples, it is almost impossible to have two original signatures with exactly the same shape and velocity profiles, so that the right pairs of zero-crossings to be associated will never occur at the same time and sometimes they will not be even close to each other. So, before measuring distances, it is necessary to identify a suitable correspondence between zero-crossings, which is accomplished with the Dynamic Time Warping (DTW) algorithm from Sato and Kogure [3]. It consists of a linear programming technique, in which the time axis of the reference curve is fixed, while the time axis of the test curve is nonlinearly adjusted, so as to minimize the norm of the global distance between the curves. This optimization tool, described by Rabiner [2], is also used in speech recognition to identify different utterances from the same phoneme.

Typically, this DTW is executed between two discrete functions, having or not the same number of points, to find the best correspondence between these points. In our context, the best correspondence is the one in which the sum of the euclidean distance between selected points is minimum. This is accomplished through

the assignment of costs (distance between points) to each association between points in different curves. These costs are organized as a matrix $[A]$, whose element a_{ij} contains the value of the euclidean distance between point i in some curve $f(x)$ and point j in some $g(x)$. This matrix can be viewed as a diagram with $T_f \times T_g$ points and an algorithm of linear programming indicates which pairs of points compose the path that leads to the best global correspondence. Typically, the optimization solution deviates from the main diagonal of the diagram, meaning that some "warping" had to be imposed.

A recursive algorithm that finds the best path in a diagram with $T_f \times T_g$ points, starting at $(1,1)$ and finishing at (T_f, T_g) , can be described by the three steps below (d_{ij} means local cost between points i and j and D means global cost from $(1,1)$ until (i,j) , following an optimal path), extracted from Rabiner [2]:

1. Initialization: $D_{(1,1)} = d_{(1,1)}m_{(1)}$, where m contains weights that empirically impose some preference among possible paths (see Figure 3).
2. Recursive step: search on a tree of possible paths. Starting by the last point (T_f, T_g) , for each intermediary combination (i_f, i_g) , search among allowed candidates (i'_f, i'_g) , the one which, after the addition of the local cost between (i_f, i_g) and (i'_f, i'_g) , leads to a global distance from $(1,1)$ to (i_f, i_g) that is minimum. Since the alternative paths tend to grow exponentially, only the 3 paths that conform to the restriction rule showed in Figure 3 are considered. Mathematically, this can be stated as: for $1 < i'_f < T_f$, $1 < i'_g < T_g$, compute, over all (i'_f, i'_g) allowed by the restriction,

$$D(i_f, i_g) = \min_{i'_f, i'_g} [D(i'_f, i'_g) + d((i_f, i_g), (i'_f, i'_g))] \quad (6)$$

The computation of $D(i'_f, i'_g)$ depends on the verification of the optimal path that reaches (i'_f, i'_g) itself, so that this step must be recursively repeated until point $(1,1)$ is reached.

3. At the end, the distance between functions receives the value of the accumulated minimum cost, normalized in order to take weights into account:

$$\text{dist}(T_f, T_g) = \frac{D(T_f, T_g)}{T_f + T_g} \quad (7)$$

Figure 4 shows a typical result of an application of DTW. In a practical sense, the nonlinear alignment

suggested by the DTW algorithm allows for abstracting from the analysis small behavioral variations during writing. The effect of the alignment produced by the DTW between the zero-crossings of two functions $x(t)$ derived from authentic samples of signatures from the same writer is showed in Figure 5.

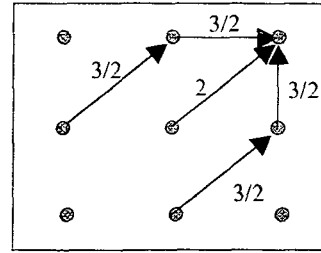


Figure 3 Local restriction used in the DTW algorithm.

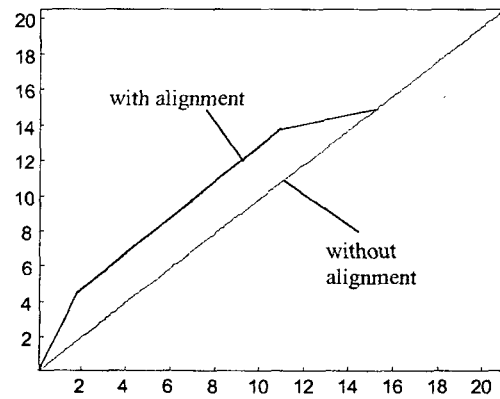


Figure 4 Typical optimal path suggested by DTW to align features originated from wavelet transforms.

The output of the DTW algorithm includes the optimal correspondence between points in the two curves and the value of the minimal distance. The optimal path is the two-column matrix $[CAM_{(i,j)}]$, where each line contain one pair of points associated by DTW.

Once the optimal correspondence is established, distances can be conveniently measured. The comparison process is based on four distances, defined as:

1. The minimum produced by the DTW algorithm, $d(T_f, T_g)$, given by equation (7).
2. The norm of the differences of integrals between consecutive zero-crossings, d_i^x for $x(t)$ and d_i^y for $y(t)$, calculated from:

$$di^x = \left\| vi_{CAM_{k,1}}^{x,1} - vi_{CAM_{k,2}}^{x,2} \right\| \quad (8)$$

$$di^y = \left\| vi_{CAM_{k,1}}^{y,1} - vi_{CAM_{k,2}}^{y,2} \right\| \quad (9)$$

where $[CAM_{i,j}]$ is the matrix containing the optimal path and $[vi_k]$ is given by equation (4).

3. The norm of the differences between amplitudes in the approximation function at the 3rd level, da^x , for $x(t)$ and da^y , for $y(t)$, calculated from:

$$da^x = \left\| va_{CAM_{k,1}}^{x,1} - va_{CAM_{k,2}}^{x,2} \right\| \quad (10)$$

$$da^y = \left\| va_{CAM_{k,1}}^{y,1} - va_{CAM_{k,2}}^{y,2} \right\| \quad (11)$$

where $[va_k]$ is given by equation (5).

4. The area of the difference between the warping function and the corresponding diagonal (see Figure 3). We assume that this last parameter recovers the information related to the velocity that is lost during the resampling process to an uniform mesh. This assumption is based on the simple idea that bigger differences in speed during signing between two signatures demand bigger deviations from the diagonal in the warping function.

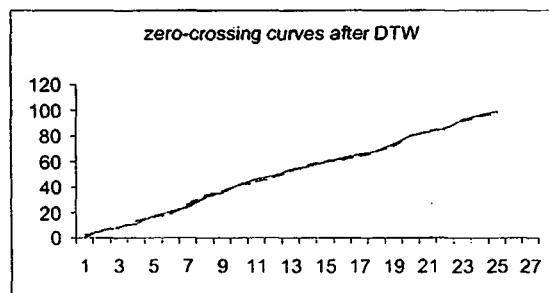
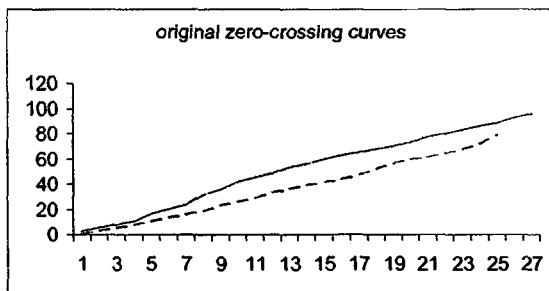


Figure 5 Effect of DTW over zero-crossings extracted from original samples of signatures from the same writer.

3.4 Function-based feature extraction

Basically, the method that includes the wavelet transform consists of the computation of distances between sets of features aligned with DTW. In this second approach, we extract the features directly from the normalized curves $x(t)$ and $y(t)$ that describe the signature, without any transformation. In this case, there are only two distances that can be defined: the minimum that returns from the DTW between the non-transformed curves and the area of the difference between the warping function and the corresponding diagonal. This drastically simplifies feature extraction, but the DTW has now to be applied over a much bigger number of points than when the only points to be considered were the zero-crossings from function WD4. To deal with this great amount of points in an efficient manner, we developed a non-recursive version of the DTW algorithm. Accordingly, in this case the warping function presents shapes like the one showed in Figure 6 (in comparison to Figure 4). We are again assuming that two signatures produced with too different writing velocity profiles tend to present a much more pronounced warping effect.

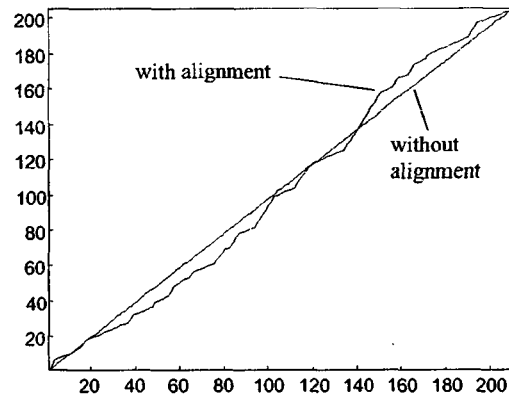


Figure 6 Typical optimal path obtained with DTW directly applied to the coordinate functions $x(t)$ and $y(t)$.

3.5 Reference signatures

During an enrolment stage, 10 sample signatures from each writer to be enrolled are collected and pairwise distances between them are computed. Based on these distances, a reference signature is selected as the one that presents minimal overall distance to the others. Actually, to each of the 10 signatures, we create a 9×8 matrix containing the 8 distances defined (4 for $x(t)$ and 4 for $y(t)$) to each of the other signatures. The reference is the signature whose distance matrix presents the minimal norm.

The distances calculated in this step are also used to produce the 8 thresholds to be tested during the verification stage. Each threshold is computed from the averages (\bar{d}) and standard deviations (s) of all possible pairwise distances associated with the 10 sample signatures, according to:

$$L = \bar{d} + 1.96 \times s \quad (12)$$

where the value 1.96 is chosen to assure that, assuming normal distribution for the distances originated from the same writer, 97.5% of the distances between original signatures and references lie under this threshold.

3.6 Verification

A test signature, after passing through preprocessing and feature extraction, is compared with the reference signature to that writer. The verification consists of checking if all defined 8 distances thus obtained lie under the thresholds given by equation (12).

4. Experimental results

The proposed methodology was tested with the original signatures and the forgeries previously acquired. The tests consisted in determining false acceptance rates (FAR) and false rejection rates (FRR) to each of the 4 writers enrolled during data acquisition. Tests were executed with the mother wavelets Daubechies 1 (Haar), Daubechies 6 (db6), Daubechies 10 (db10) and Biorthogonal 5.5 (Bior. 5.5), at different levels of resolution (3, 4 and 5). We included also a test in which the 4th dissimilarity measure (area between warping function and diagonal) was not considered.

According to preliminary tests, we chose a standard configuration as: db 6 and level of resolution 4. Results for this configuration are presented in Table 1. Although the FRR may be significant, there was no acceptance of forgeries at all. It should be noted that, according to practical considerations, it would be possible to decrease FRR by relaxing the thresholds, but in this case the FAR would increase. The standard configuration was chosen in order to provide maximum security against forgeries, without taking into account any troubles to the writer by having original signatures rejected.

Table 1. db 6, level 4, including distance 4.

	Writer 1	Writer 2	Writer 3	Writer 4
FAR	0%	0%	0%	0%
FRR	37%	23%	30%	40%

The tests showed in Table 2 justify the adoption of level 4 as the reference. Although presenting a light decreasing in FRR on the average, the data in Table 2 show unacceptable values to the FAR, meaning that at levels other than the 4th, this verification system would be unsafe for the cases studied.

Table 2. Mother wavelet db6, levels 3 and 5, including distance 4.

Resolution level	Error taxes	Writer 1	Writer 2	Writer 3	Writer 4
3 rd	FAR	0%	23%	3%	23%
	FRR	23%	13%	40%	37%
5 th	FAR	0%	53%	0%	7%
	FRR	30%	33%	33%	27%

We also performed an investigation to check if a substitution of the mother wavelet Daubechies by others would produce better results. To this end, the same tests from the standard case were re-executed, just replacing db6 successively by db1, db10 and bior 5.5. Our results, showed in Table 3, indicate that, even though it is possible to find some improvement in FRR, there are again some important security failures due to high FAR, meaning that db6 was effectively the most indicated in this case.

Table 3. Mother wavelets db1, db10 and bior 5.5, level 4, including distance 4.

Mother wavelet	Error taxes	Writer 1	Writer 2	Writer 3	Writer 4
Db1 (Haar)	FAR	0%	50%	0%	13%
	FRR	20%	7%	30%	20%
Db10	FAR	0%	27%	0%	23%
	FRR	7%	23%	30%	37%
Bior 5.5	FAR	0%	20%	0%	0%
	FRR	40%	27%	30%	20%

To check if the information associated with writing motion is really relevant in the methodology with wavelets, a test was devised, in which the 4th dissimilarity measure (area between warping and diagonal) was ignored. The results to this case, shown in Table 4, reveal that the 4th measure of distance is an essential parameter to prevent forgeries to be accepted.

Table 4. Mother wavelet db6, level 4, without distance 4.

	Writer 1	Writer 2	Writer 3	Writer 4
FAR	0%	3%	0%	3%
FRR	37%	17%	23%	27%

Finally, the wavelet transform was removed and the distances were obtained directly from the coordinate functions $x(t)$ and $y(t)$. In this case, the difference in the number of points between a test signature and the reference can become very large, so we had to include a "filter", based on two global parameters: total duration time of the signature and its total length. Even with the inclusion of this filter the FAR obtained in this case were not satisfactory at all and showed that, for the cases studied, the wavelet step is essential to ensure security.

Table 5. Function-based, with filter.

	Writer 1	Writer 2	Writer 3	Writer 4
FAR	3%	70%	0%	3%
FRR	13%	13%	23%	17%

5. Conclusions

We concluded that the error taxes obtained were at acceptable levels for this kind of biometry. This shows that the chosen methodology, consisting basically in the application of the Dynamic Time Warping algorithm on features extracted with the application of wavelet transforms, is suitable to on-line signature verification. The need of a wavelet transform step was tested by comparison with a simplified system based solely on the coordinate functions $x(t)$ and $y(t)$. We concluded that it is worth to include the wavelet transform in the analysis. The main tool to comparison of signatures is undoubtedly the DTW, but the inclusion of the wavelet transform was an essential step for the achievement of low false acceptance rates. It was only with the inclusion of the wavelet transform, in the right level of resolution, that we managed to completely prevent trained forgeries to be accepted (0% FAR) in the cases studied. This result was expected, since, through wavelet transforms, it is possible to analyze the problem in a resolution level suitable to the problem of comparison of signatures.

6. Acknowledgements

During this work, the first author was supported by a scholarship from CAPES (the Brazilian Agency for Post-Graduate Studies).

7. References

- [1] E. Newham, "Survey: Signature Verification Technologies", *Bit* (2000), 8--10.
- [2] L. Rabiner, B. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.
- [3] Y. Sato, K. Kogure, "On-line signature Verification Based on Shape, Motion, and Writing Pressure", *Proc. 6th Int. Conf. on Pattern Recognition* (1982), 823--826.
- [4] B. Wirtz, "Stroke-based Time Warping for Signature Verification", *Proc. IEEE* (1995), 179--182.
- [5] C.F. Lam, D. Kamins, "Signature Recognition through Spectral Analysis", *Pattern Recognition* 22 (1989), 39--44.
- [6] C.S. Sundaresan, Keerthi, "A Study of Representation for Pen Based Handwriting Recognition of Tamil Characters", *International Conference on Document Analysis and Recognition (ICDAR)* (1999).
- [7] W.W.Boles, "A Security System Based on Human Iris Identification using Wavelet Transform", *Engineering Applications of Artificial Intelligence* (1998), 77--85.
- [8] M. E. Munich, P. Perona, "Continuous Dynamic Time Warping for Translation Invariant Curve Alignment with Applications to Signature Verification", (1999), Available at: <http://citeseer.nj.nec.com/munich99continuous.html>.
- [9] P. S. Deng, H. M. Liao, C.W. Ho, H.Tyan, "Wavelet-based Off-Line Signature Verification", *Proc. IEEE* (1997).
- [10] B. Schneier, *Applied Cryptography: Protocols, Algorithms and Source Code in C*, John Wiley & Sons, 1996.
- [11] K. Huang, H. Yan, "On-line signature Verification Based on Dynamic Segmentation and Global and Local Matching", *Optical Engineering* 34 (1995), 3480--3487.
- [12] R. Plamondon, "On-line and Off-line Handwriting Recognition: A Comprehensive Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), 63--84.
- [13] T. Ruggles, "Comparison of Biometric Techniques", *Technical Report for The Biometric Consulting Group* (1998), Available at: <http://biometric-consulting.com/bio.htm>
- [14] <http://www.wacom.com>
- [15] L. L. Lee, T. Berger, E. Aviczer, "Reliable On-line Human Signature Verification Systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (1996), 643--647.
- [16] F. D. Griess, "On-line Signature Verification", *Projet Report*, Michigan State University, Department of Computer Science and Engineering, 2000.