

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMAÇÃO E ESTATÍSTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM MODELO PARA IDENTIFICAÇÃO DE NEMATÓIDES
BASEADO NA ESTRUTURA DE ESTILETE**

por

Luis Alberto Martins Palhares de Melo

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

Prof. Rogério Cid Bastos, Dr.
Orientador

Florianópolis, Fevereiro de 1996

**UM MODELO PARA IDENTIFICAÇÃO DE NEMATÓIDES
BASEADO NA ESTRUTURA DE ESTILETE**

LUIS ALBERTO MARTINS PALHARES DE MELO

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE

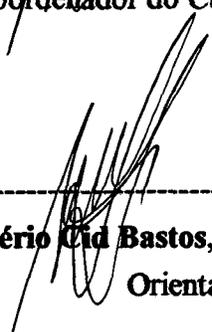
MESTRE EM CIÊNCIA DA COMPUTAÇÃO

**ESPECIALIDADE SISTEMAS DE CONHECIMENTO E APROVADA EM SUA FORMA FINAL
PELO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO.**

BANCA EXAMINADORA:



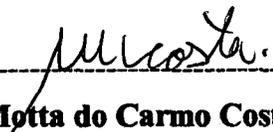
Prof. Rogério Cid Bastos, Dr.
Coordenador do Curso



Prof. Rogério Cid Bastos, Dr.
Orientador



Prof. Luiz Fernando Jacintho Maia, Dr.



Marcos Motta do Carmo Costa, Dr.

"Olhamos para o nosso tempo para constatar que a ciência e a tecnologia enriqueceram imensamente o volume de dados que temos sobre a realidade; é impensável a quantidade de informações de que hoje dispomos sobre o ambiente do homem (...). Aí estão os computadores empanturrados de verdade fragmentária, e nós, ao lado deles, abismados com nossa ciência que gerou tal pluralidade e rapidez e, ao mesmo tempo metidos numa impotência horrorosa de reestruturarmos uma visão coerente ou pelo menos plausível de nós mesmos como seres humanos." [MOR91]

RÉGIS DE MORAIS

A meus pais, Luiz Alberto e Aila,
que sempre me apoiaram
em todos os momentos da vida.

AGRADECIMENTOS

A meus pais, Luiz Alberto e Aila, que sempre me estimularam a realizar o curso de mestrado.

Ao professor e orientador Rogério Cid Bastos, pela orientação, principalmente quando soube "bater na mesa" para mostrar o caminho de um trabalho estimulante.

Ao Eduardo Cajueiro, do CENARGEN/EMBRAPA, que sugeriu o tema e me deu grande apoio quando estive em Brasília para a coleta de informações acerca do trabalho.

Aos meus chefes na EMBRAPA, Antenor Turazi, Adilson Maestro e Marcos Motta que me apoiaram e respaldaram minha vinda à Florianópolis para realização do Curso de Mestrado.

Ao colega Leonardo Hosaka, da EMBRAPA/SEDE, pelo apoio administrativo prestado durante minha permanência aqui em Santa Catarina.

A todos os colegas do DIN, em especial, o Chico, Reinaldo, Tonico, Bruno, Leonardo, Viegas, Rogério, Éder, Pedro, Daniel, Valdemar e Adelina, que de uma forma ou de outra, durante os anos de trabalho, contribuíram para meu aprimoramento técnico.

Ao meu pai, pelo incansável apoio na busca de referências bibliográficas e servir de "ponte" de contato com o pessoal dos nematóides em Brasília.

Ao Dario, meu irmão, pelo apoio na coleta de material bibliográfico na UnB.

Ao meu primo Rogério Saldanha, pelas dicas de tópicos de matemática.

Ao Dr. Juvenil Cares, professor da UnB, pela atenção dispensada para explicações sobre nematóides, e pela bibliografia de nematóides indicada.

À Dra. Edna Costa Manso, do CENARGEN/EMBRAPA, pela atenção dispensada para explicações sobre nematóides, e pelas fotos de nematóides cedidas.

Aos professores da UFSC Rogério Cid Bastos, Luiz Fernando Maia, Raul Wazlawick, Fernando Gauthier e Édís Lapolli, com os quais cursei as disciplinas durante o programa de mestrado.

Aos professores Paulo Roberto Hofmann e Tânia Rauh de Azevedo do Departamento de Biologia Celular, Embriologia, e Genética da UFSC e professor Talmir da Silva do Departamento de Fitotecnia da UFSC pela atenção e apoio dado sobre referências de nematóides.

Aos colegas de mestrado Nildo, Anita, Romerito (Angel), Carla, Ana Lúcia, Carlos Montez, Neilor, Angelita, Marcelo (MMA), Valéria, César pelo companheirismo.

À Cárita Regina, que compartilhou comigo de meus melhores momentos em 1995 aqui em Florianópolis.

À Verinha e Valdete, secretárias do Curso da Pós-Graduação em Ciência da Computação, pela atenção e apoio administrativo prestados durante minha permanência no curso.

A EMBRAPA, que investiu em meu aperfeiçoamento técnico, permitindo que eu realizasse o curso de pós-graduação.

A CAPES, que me concedeu uma bolsa de estudos.

A UFSC, onde realizei o curso de mestrado.

RESUMO

O surgimento de computadores mais potentes em termos de processamento e a um custo comparativo cada vez menor, vem permitindo que cada vez mais estas máquinas sejam popularizadas e utilizadas nas mais diversas atividades econômicas, desde as mais simples como, controle administrativo, até as mais complexas como, suporte à pesquisa de novas tecnologias.

Visão computacional é um ramo da área de Inteligência Artificial que tem se beneficiado com a potência de processamento dos computadores. Inspeção visual na indústria em geral, navegação de robôs, interpretação de fotos, identificação de estruturas 3D para processos de simulação e realidade virtual, inspeção de componentes eletrônicos tais como chips VLSI e placas de circuito impresso, são exemplos típicos de aplicações de visão por computador que obtiveram sucesso.

Em linhas gerais visão por computador é uma tarefa em que se efetua extração de parâmetros em um cenário e utiliza estes parâmetros para suporte a outras tarefas, tais como reconhecimento de padrões.

Uma aplicação possível, de grande interesse econômico baseada em visão computacional é a interpretação de fotos de nematóides, que são animais vermiformes microscópicos, muitos deles parasitas de plantas cultivares e que trazem enormes prejuízos às lavouras dado seu potencial destrutivo.

O presente trabalho apresenta uma proposta de identificação de nematóides, analisando-se imagens destes, e daí tentando-se observar os aspectos morfológicos de um órgão típico do nematóide, o *estilete*.

ABSTRACT

The arrival of more powerful computers with stronger processing capability and with lower comparative cost allows these machines to become more and more popular and to be used in the most variety economic activities, since simple management control for exemple, until complex ones like new technologies support.

Computer vision is a branch of Artificial Intelligence that has taken advantage on the power of processing capability of computers. Visual inspection on industry, robot navigation, photo interpretation, extraction of 3D structure to simulation processes and virtual reality, electronic components inspection such as VLSI chips and circuit board, all of them are successful computer vision applications examples.

Computer vision, roughly speaking, is a task that seeks to extract scene parameters, and to make use of them to support another tasks, such as pattern recognition. A possible application of great economic interest that makes use of computer vision is photo interpretation of nematoid, that are microscopic vermiform animals, a lot of them plant-parasit, and cause great harm to the farming because of their destructive potencial. The present work presents an approach to nematoid identification analysing nematoid images and then observing the morphologic features of *stylet*, a tipic nematoid organ.

SUMÁRIO

	Pág
1. Introdução	1
1.1 Introdução	1
1.2 Objetivos	2
1.3 Justificativa	3
1.4 Importância	3
1.5 Estrutura deste trabalho	4
2. Fundamentação Teórica	5
2.1 Introdução	5
2.2 Reconhecimento de Padrões	5
2.3 Abordagens Utilizadas no Reconhecimento de Padrões	8
2.3.1 Abordagem de Decisão-Teórica ou Discriminante	8
2.3.2 Abordagem Sintática ou Estrutural	14
2.3.3 <i>Template Matching</i>	21
2.3.4 Comparação entre as Abordagens Discriminante e Sintática	26
2.4 Processamento de Imagens	27
2.5 Conjuntos Difusos	28
2.5.1 Conceitos Introdutórios de Lógica	28
2.5.2 Sentenças e Conjuntos	30
2.5.3 Lógica Difusa	32
2.5.4 Sentenças e Variáveis Linguísticas	33
2.6 O Modelo AFT ([SHA92])	34
2.6.1 Descrição do Modelo AFT	35
2.6.2 Sensibilidade à Rotação no Modelo AFT	41
2.6.3 Exemplos dos Cálculos envolvidos na Distância D Para Grafos AFT's	42
2.6.3.1 Definição dos Grafos AFT's dos Padrões P1 e P2	42
2.6.3.2 Cálculo da Distância D Entre os Grafos AFT dos Padrões P1 e P2	44
3. Caracterização e Identificação de Nematóides	48
3.1 Introdução	48
3.2 Taxionomia	48
3.3 Nematóides	52

3.3.1	Caracterização dos Nematóides	52
3.3.2	As Estruturas do Nematóide Como Base de Classificação	54
3.3.3	Nematóides Fitoparasitos (Fitonematóides)	57
3.3.3.1	Caracterização de Fitonematóides	57
3.3.3.2	A Estrutura do Estilete em Fitonematóides	57
3.4	Importância Econômica dos Nematóides	63
3.5	Fitonematóides das Principais Plantas Cultivadas	64
3.6	Aspectos Relativos à Identificação de Nematóides	68
3.7	Escopo da Aplicação - Identificação de Nematóides Através do Formato do Estilete	70
4.	Modelo para Identificação de Nematóides Através do Estilete	72
4.1	Introdução	72
4.2	Esquema Geral do Sistema de Identificação	72
4.3	Considerações Sobre as Imagens de Nematóides Tratadas pelo Modelo	75
4.4	Pré-Processamento	76
4.5	Segmentação da Imagem	77
4.6	Indicação da Estrutura de Estilete	80
4.7	Aproximação Poligonal para o Contorno	81
4.8	Digitação das Características do Nematóide	84
4.9	Representação do Polígono por Grafo AFT	84
4.10	<i>Template Matching</i> Baseado na Distância para AFT	90
5.	Implementação dos Algoritmos	92
5.1	Introdução	92
5.2	Algoritmo para Segmentação de Imagem	92
5.3	Algoritmo para Aproximação Poligonal para o Contorno	114
5.3.1	Envelope Convexo (<i>Convex Hull</i>)	119
5.3.2	Um Critério Para Definição das Arestas do Polígono	124
5.4	Algoritmo para Calcular a Distância entre 2 AFT's	128
5.5	Um Teste do Programa-Protótipo	133
6.	Conclusões e Recomendações	138
6.1	Conclusões	138
6.2	Recomendações	139

REFERÊNCIAS BIBLIOGRÁFICAS	141
ANEXOS	149
Anexo I - Fitonematóides em Cultura de Soja	149
Anexo II - Um Exemplo do Cálculo da Distância Entre 2 Grafos AFT	159

CAPÍTULO 1

Introdução

1.1 - Introdução

Com a popularização de computadores a um preço cada vez menor e com potencial de processamento cada vez maior, aplicações que esbarravam na limitação do hardware, puderam se beneficiar dessa nova realidade e serem implementadas na prática com a performance mais próxima que se esperavam delas. Por exemplo, as aplicações na área de Inteligência Artificial (IA), cujo paradigma geral apresenta tarefas que em sua maioria requerem alto grau de paralelismo, passam a ter um hardware que, se ainda não atende plenamente os requisitos ideais para uma aplicação em IA, permite um desempenho relativamente satisfatório de tais aplicações.

Entre as aplicações de Inteligência Artificial que se relacionam com a atividade de reconhecimento de padrões encontram-se o reconhecimento de caracteres escritos à mão, imagens de satélites e mapas, impressões digitais, etc. O processo para estes casos relaciona-se com algum dispositivo "visual" [GRI94].

A área de Visão Computacional, um ramo da Inteligência Artificial, destina-se em estabelecer modelos de tomada de decisão aplicados em máquina, baseados na interação de dispositivos sensoriais da máquina com o ambiente.

Em linhas gerais, sistemas de visão assim se processam: O sensor da máquina recebe a imagem, que para ele nada mais é do que um conjunto de medidas de intensidade de brilho refletidas, e daí tentará efetuar um mapeamento dessa imagem para o cenário real que a imagem representa. Cada "célula" componente do sensor registra uma intensidade luminosa que reflete uma complexa interação de vários parâmetros: (a) posição e orientação do objeto e fonte de luz em relação à "célula" do sensor; (b) formato tridimensional do objeto; (c) propriedades reflectivas da superfície do objeto (aspectos físicos apresentados pelo raios da fonte luminosa ao incidir na superfície do objeto); (d) propriedades espectrais da fonte luminosa; (e) sensibilidade espectral e eficiência do sensor ([GRI94]).

Uma vez que se tenha extraído os devidos parâmetros do cenário, pode-se então efetuar o reconhecimento de padrões propriamente dito. Neste sentido, sistemas de visão têm a finalidade de servir de suporte à atividade de reconhecimento de padrões. E para tal atividade, várias técnicas podem ser aplicadas. Fu ([FU76]) indica que diversas técnicas para reconhecimento de padrões podem ser vistas como sendo ou de natureza *discriminante* ou de natureza *sintática*.

Uma aplicação de relevante interesse econômico que se baseia no binômio *visão-reconhecimento* é a obtenção da identificação de nematóides a partir de suas imagens digitalizadas. Nematóides são animais vermiformes microscópicos, muitos deles parasitas de plantas cultivares e que trazem enormes prejuízos às lavouras dado seu potencial destrutivo.

Uma correta identificação desses animais baseia-se enormemente nos aspectos morfológicos e morfométricos de seus órgãos internos. Isto implica em realizar inspeção visual na imagem digitalizada do nematóide, onde uma vez localizadas as estruturas, e sabendo-se quais órgãos elas representam, caracterizar, mediante seus aspectos morfológicos e morfométricos, o nematóide na medida do possível dentro de sua apropriada categoria taxionômica.

1.2 - Objetivos

Os objetivos do presente trabalho são:

- 1) Estabelecer um modelo para inspeção de imagens digitalizadas de nematóides onde, uma vez localizada a estrutura do *estilete* - um órgão específico dos nematóides - caracterizá-los adequadamente, de modo que tal caracterização sirva de subsídio para uma identificação. O modelo leva em consideração aspectos morfológicos do *estilete*, não levando em conta seus aspectos morfométricos. Em linhas gerais o modelo assim se compõe:
 - a) Procedimento de localização e isolamento da estrutura do *estilete* dentre as demais estruturas na imagem digitalizada do nematóide.
 - b) Codificação adequada da estrutura morfológica do *estilete* como sentença de alguma linguagem para posterior identificação.
 - c) Identificação/classificação do tipo de *estilete*: Utilizando-se raciocínio aproximado, efetuar *template matching* contra uma base de conhecimento que contém (i) descrição codificada do padrão morfológico do *estilete* e (ii) descrição de uma caracterização para os nematóides com formato de *estilete* deste padrão.
- 2) Implementar computacionalmente um protótipo para tal proposta de modelo.

1.3 - Justificativa

Nematóides são animais vermiformes de corpo longo, fino e afilado nas duas extremidades que se encontram no solo e em plantas. Os nematóides que parasitam plantas são denominados *fitonematóides*. Patógenos de difícil controle e fácil disseminação, atacam praticamente todas as culturas de importância econômica, causando prejuízos que vão desde a destruição de mudas até a redução drástica da produtividade e qualidade das lavouras, que refletem em prejuízos para o produtor e elevação dos preços para o consumidor.

O primeiro passo no combate a esses organismos é o reconhecimento adequado de sua existência, ou seja, o diagnóstico correto. Muitas vezes, o produtor confunde, por exemplo, o ataque do nematóide com a deficiência de nutrientes e com isso, jamais consegue resolver o problema em sua lavoura ([VIL92]).

Em termos de tratamento computacional, encontram-se esparsas referências sobre nematóides ([BOA88], [PEE90], [FRA92], [BLA92], [EIS93], [BRA94], [ROB94]). Nestas referências, os autores citam de modo geral a importância que o suporte computacional traz em geral para a nematologia, destacando principalmente o auxílio em termos de catalogação (classificação taxionômica) de nematóides, um grupo do reino animal com uma vasta diversidade de exemplares.

1.4 - Importância

Conforme colocado na seção 1.3, os fitonematóides apresentam grande importância econômica, haja visto serem parasitas capazes de trazer prejuízos a todo tipo de lavoura. As referências bibliográficas que tratam de sistemas computacionais para controle de nematóides em geral, apresentam limitações, em sua concepção. Em geral, estas aplicações ([BOA88], [BRA94], [ROB94]) efetuam tratamento estritamente numérico dos nematóides, não efetuando tratamento da imagem.

Algumas aplicações ([PEE90], [BLA92], [CAB96]) incorporam o enfoque de processamento da imagem de nematóides. A incorporação do processamento da imagem dos nematóides aos modelos computacionais inteligentes, propicia uma ampliação do potencial dos mesmos, uma vez que a caracterização em geral dos nematóides baseia-se fortemente em aspectos morfológicos (e morfométricos) de seus órgãos.

Desta forma, o modelo proposto neste trabalho é uma contribuição a mais ao esforço desenvolvido para se projetar sistemas computacionais inteligentes destinados ao tratamento de nematóides.

1.5 - Estrutura deste trabalho

Este trabalho constitui-se de 6 capítulos, Introdução, Fundamentação Teórica, Caracterização e Identificação de Nematóides, Modelo para Identificação de Nematóides Através do Estilete, Implementação dos Algoritmos, e Conclusões e recomendações, respectivamente capítulos 1, 2, 3, 4, 5 e 6.

O capítulo 1 apresenta uma introdução, objetivos, justificativa, importância e a estrutura deste trabalho.

Apresenta-se no capítulo 2, uma revisão bibliográfica sobre os aspectos teóricos utilizados no modelo de identificação de nematóides referentes a reconhecimento de padrões e processamento de imagens, conjuntos difusos e o modelo de grafo *Atributed Fuzzy Tournament* (AFT), utilizado no processo de aferição de similaridade entre estruturas.

O capítulo 3, abrange aspectos gerais relativos ao objeto de estudo, os nematóides. Apresentam-se conceitos correlatos das ciências biológicas no que concerne à classificação dos seres vivos, comentam-se aspectos gerais que caracterizam os nematóides e, sua importância econômica.

O capítulo 4, Modelo para Identificação de Nematóides Através do Estilete, apresenta, a proposta para o sistema identificador de nematóides mediante análise da estrutura do *estilete*. Comentam-se os aspectos referentes ao aprendizado do sistema sobre os nematóides e os aspectos ligados ao reconhecimento/identificação de um padrão desconhecido de estilete, por parte do sistema

O capítulo 5 apresenta os algoritmos em pseudo-código e as principais estruturas de dados utilizadas para implementar computacionalmente os conceitos do modelo proposto no capítulo 4, além de um teste do programa-protótipo desenvolvido.

Finalmente o capítulo 6 encerra o trabalho com as conclusões e recomendações para o modelo proposto.

CAPÍTULO 2

Fundamentação Teórica

2.1 - Introdução

Neste capítulo apresentam-se aspectos que de alguma forma servem de suporte teórico ao modelo proposto de identificação de nematóides.

Comentam-se sobre aspectos relativos à atividade de reconhecimento de padrões e suas abordagens. Em seguida descrevem-se aspectos sobre processamento de imagens, conjuntos difusos e o grafo *Atributed Torunament Model* (AFT), uma importante estrutura para o modelo proposto, através da qual se caracteriza a morfologia do estilete do nematóide.

2.2 - Reconhecimento de Padrões

Uma variedade de definições pode ser encontrada na literatura onde se procura caracterizar o que vem a ser um *padrão*, e o que vem a ser a atividade de *reconhecimento de padrões*.

Beale e Jackson ([BEA91]) voltam a atenção para o termo *padrão* e procuram realçar seu entrelaçamento com o termo *informação*: a maior parte de conteúdo informativo que nossas 'redes neurais' biológicas absorvem se apresentam na forma de padrões. Assim, este texto escrito por exemplo, é apresentado em uma forma complexa de cadeias de caracteres.

Uhr ([UHR73]) procura caracterizar reconhecimento de padrões, analisando inicialmente os aspectos sobre a captação de informação que os organismos vivos de modo geral apresentam. Para Uhr, é de fundamental importância as atividades de *sensação* e *percepção*, pois é através delas que ocorre qualquer interação entre indivíduo e meio ambiente. É através da sensação-percepção que um organismo é capaz de se defrontar com um determinado objeto (padrão) é associá-lo ao seu significado dentro de um determinado contexto. O propósito da sensação-percepção é o *re-conhecimento*.

Percepção refere-se ao processo de transformação de uma informação de entrada complexa em apropriados nomes, descrições, comportamentos ou mesmo outras saídas. Reconhecimento de padrões pode ser encarado como a atividade de mapeamento de um conjunto de instâncias de objetos/entidades para um conjunto de nomes-caracterizadores dos referidos objetos/entidades. Tipos de padrões de

interesse não se limitam apenas a objetos concretos tais como figuras, letras, dígitos, mas também entidades abstratas como por exemplo, a beleza de uma mulher, uma partitura de uma música, a profundidade de um teorema matemático ou mesmo a aceitabilidade a um determinado aroma para uma determinada pessoa.

FU ([FU76]) relata o fato de que reconhecimento de padrões está intimamente relacionado com a descrição e classificação de medidas obtidas de processos físicos ou mentais. Beale e Jackson ([BEA91]) também colocam que o objetivo da tarefa de reconhecimento de padrões é a *classificação*. Jain e Mao ([JAI94]), Andrews ([AND72]), Bezdek ([BEZ81]) entre outros também destacam a classificação como objetivo primordial da tarefa de reconhecimento de padrões.

Bezdek ([BEZ81]) apresenta de forma simples e direta, uma definição para a atividade de *reconhecimento de padrões*:

"Quite simply, pattern recognition is a *search for structure in data*." [BEZ81, p. 1]

Na definição acima, Bezdek ressalta três aspectos: a tarefa de procura ou busca (*search*), a estrutura (*structure*) e os dados (*data*). Os *dados* podem ser de natureza qualitativa, quantitativa ou ambas; podem ser numéricos, pictoriais, texturais, linguísticos ou qualquer dessas combinações. Exemplos de dados: registros médicos, fotos aéreas, posições das galáxias, impressões digitais, perfil psicológico, componentes químicos, aspectos demográficos, etc.

A *procura* ou *busca* refere-se às técnicas que se utilizam para processar os dados componentes do padrão. Uma exemplo de técnica de busca, que é baseada em heurística, é a do "olhômetro" (*eyeball technique*, [BEZ81]). Outro exemplo de técnica, porém mais rigorosa em termos algorítmicos baseia-se no uso da análise estatística.

A *estrutura* representa a maneira em que os dados estarão organizados, ou seja, como eles estão dispostos. Está, portanto, ligada à caracterização das *relações* que envolvem os dados componentes do padrão. Segue-se abaixo uma exemplificação de um padrão e os aspectos que o envolvem citados por Bezdek. É uma figura representando uma "borboleta".

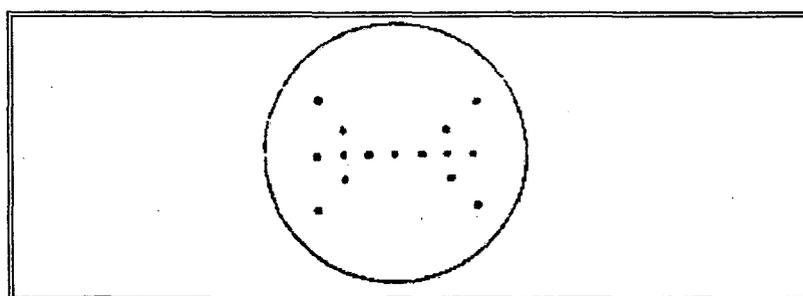


Fig. 2.1 - Uma padrão representativo de "Borboleta" - Fonte: [BEZ81]

Observando-se o padrão nota-se que os *dados* são de natureza pictorial. Os *dados* são os pontos. Mas os *dados* por si só não dizem nada, não veiculam informação alguma. A *estrutura* do padrão indica a disposição geográfica dos pontos (*dados*), caracterizando as intra e inter-relações; através da estrutura poderiam também ser representadas as diferentes texturas (caso houvessem) dos *dados*, etc.

É, então, o binômio *dados-estrutura* que apresentará teor informativo, isto é, permitirá que se faça um mapeamento do padrão em questão para algum universo representativo de objetos/entidades. No caso, duas estruturas de formato triangular dispostas lado a lado horizontalmente, ligadas por um "filete" horizontal conduzem ao mapeamento do padrão para um universo de discurso caracterizador de borboletas (a palavra "borboleta", uma foto de uma borboleta, ou mesmo o próprio animal).

A *busca* ou *procura* de estruturas em *dados* seria a ponte, o processo de mapeamento do padrão até o teor informativo intrínseco a ele. Existem diversas técnicas de procura, cada uma apresentando suas vantagens e desvantagens.

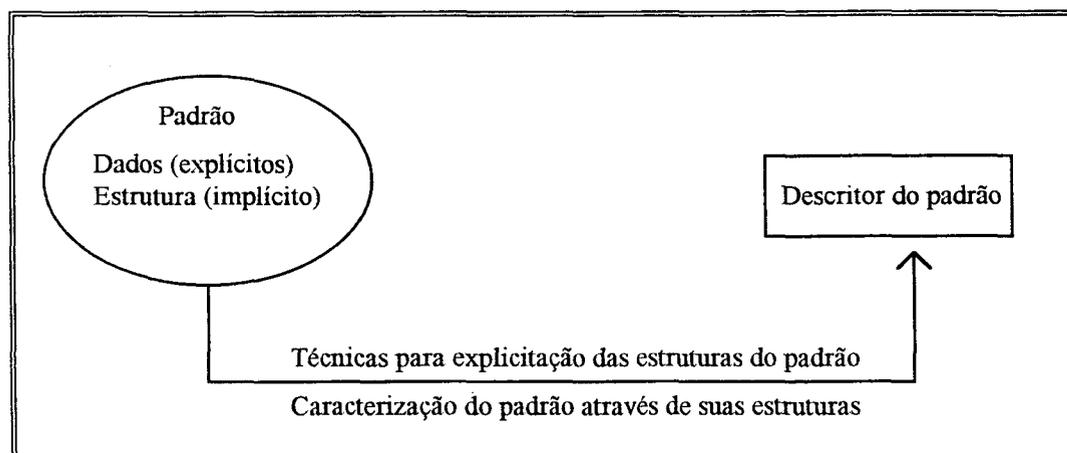


Fig. 2.2 - O mapeamento do padrão para um determinado descritor de classe do padrão

Partindo-se da definição acima apresentada por Bezdek para reconhecimento de padrões, percebe-se o seguinte encadeamento: inicialmente tem-se um determinado padrão não identificado, que em princípio é visto meramente como um conjunto de *dados*. Uma ou mais *estruturas* (entidades abstratas, caracterizadas principalmente por *relações* intrínsecas aos *dados*), que estão implícitas no padrão servem de suporte para uma adequada caracterização do mesmo. Para se encontrar as estruturas caracterizadoras utiliza-se alguma ou algumas *técnicas de procura de estruturas*, tais como "olhômetro" (para padrões pictorias), análise estatística, entre outros.

2.3 - Abordagens Utilizadas no Reconhecimento de Padrões

Fu ([FU82]) relata que as técnicas usadas em reconhecimento de padrões podem ser agrupadas em duas categorias: abordagem de *decisão-teórica* ou *discriminante* e abordagem *sintática* ou *estrutural*, embora admita que em certos casos esta distinção não seja tão clara. E de fato em algumas referências bibliográficas esta distinção é até mesmo mais acentuada. Jain e Mao ([JAI94]) por exemplo, apontam para três abordagens: estatística, sintática e de redes neurais artificiais, enquanto que para Fu, as redes neurais artificiais e modelo estatístico se enquadrariam no modelo de decisão-teórica.

2.3.1 - Abordagem de Decisão-Teórica ou Discriminante

Para descrever a configuração básica de um *problema de reconhecimento de padrão* através da abordagem de *discriminante*, FU ([FU76]) apresenta o seguinte diagrama de blocos:

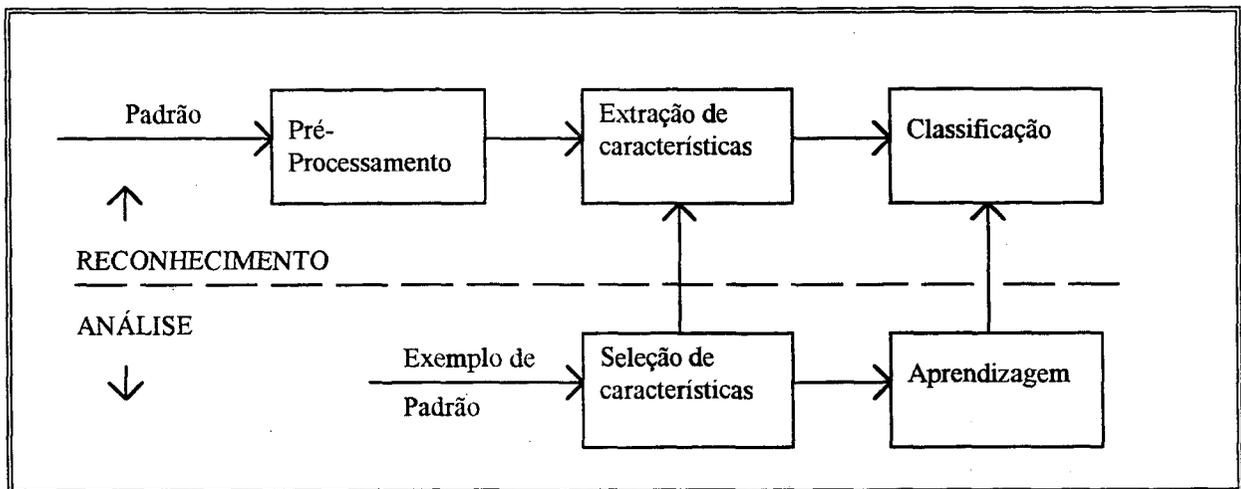


Fig. 2.3 - Diagrama de blocos para um sistema de reconhecimento de padrões - abordagem de decisão teórica (discriminante) - Fonte: [FU76]

A proposta da configuração apresentada na figura 2.3 é a mesma em essência apresentada por outros autores. Andrew ([AND72]) descreve o problema de reconhecimento de padrões como uma transformação a partir do universo P do padrão em questão, para o universo de características F e daí para o universo de classificação C .

$$P \rightarrow F \rightarrow C$$

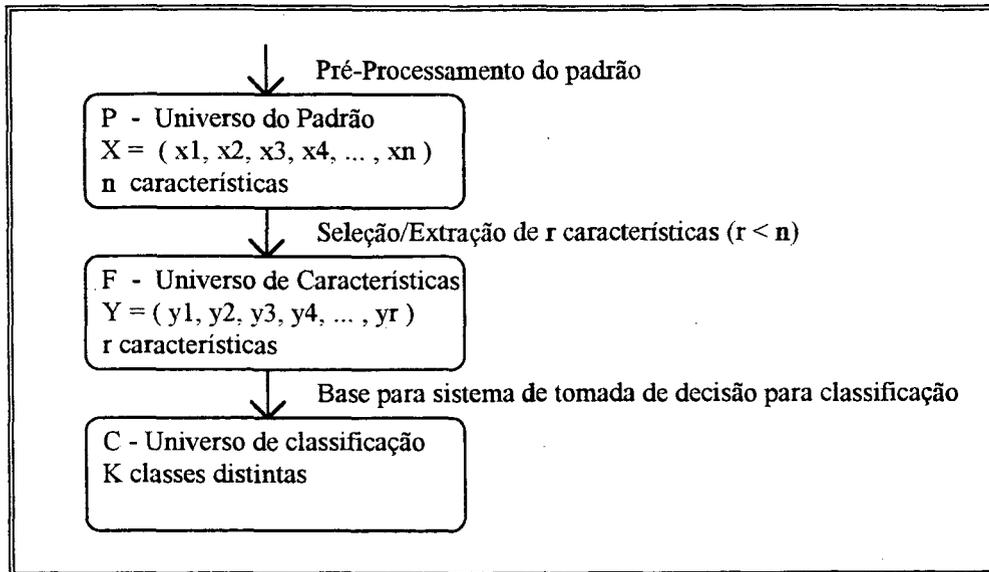


Fig. 2.4 - O problema de reconhecimento de padrões. Mapeamento $P \rightarrow F \rightarrow C$

O universo do padrão é essencialmente um conjunto de elementos identificados no mundo real através de algum meio de observação. Supondo que o padrão (objeto) em questão sejam candidatos a um determinado emprego, os elementos deste universo de padrão seriam: nome, idade, altura, peso, formação profissional, experiência profissional, nacionalidade, naturalidade, preferências musicais, número da carteira de identidade, estado civil, estado geral de saúde, time de futebol de preferência, tipos de filmes que gosta de assistir, etc. Um vetor coluna X representaria então o universo P do padrão em questão, em termos de seus atributos:

$$X = (x_1, x_2, x_3, x_4, \dots, x_k, \dots, x_n)$$

Cada x_k representa um particular valor associado com a k -ésima dimensão do vetor. O vetor X pode ser interpretado como um ponto no espaço de dimensão n do padrão. O universo de características F por sua vez, pode ser visto como um domínio intermediário entre os dados do padrão P e o universo de classificação C . Os elementos do universo de características F são aqueles mais marcantes do Universo P , que servirão de base para se efetuar alguma classificação. No exemplo, como se deseja classificar um indivíduo com perfil adequado ou não ao emprego, certamente aspectos tais como, idade, formação e experiência profissional são imprescindíveis, e serão elementos integrantes do universo de características F . Um vetor Y representaria o universo de características, de forma semelhante ao vetor X para o universo P :

$$Y = (y_1, y_2, y_3, y_4, \dots, y_k, \dots, y_r)$$

O universo de classificação C é relativamente mais fácil de se construir em relação ao universo de características F . Isto porque C é constituído basicamente por um conjunto de regras de decisão que abordam aspectos bem definidos das características discriminantes dos padrões que se desejam classificar.

Jain e Mao ([JAI94]) propõem que um sistema de reconhecimento de padrões envolve essencialmente 3 etapas:

- (a) aquisição de dados e pré-processamento,
- (b) representação/extração de características e
- (c) tomada de decisão ou clusterização.

Inicialmente todo padrão deve sofrer algum tipo de pré-processamento. Esta é uma atividade realizada primordialmente para retirar "ruídos" provenientes do dispositivo original que produziu um padrão. Uma imagem de satélite, ou o som gerado por algum dispositivo, comumente apresentam este tipo de problema. Daí a importância de se efetuar o pré-processamento da imagem, no sentido de retirar estruturas indesejadas do padrão.

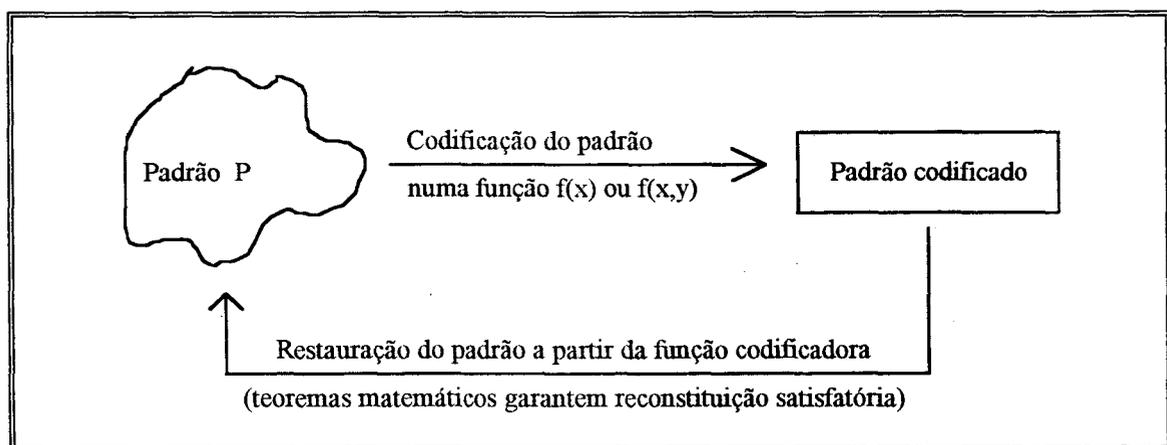


Fig. 2.5 - Codificação do padrão e reconstituição do padrão (aproximação) a partir da função codificadora

Fu ([FU82]) cita como atividades de pré-processamento as atividades de *codificação e aproximação*. Em computadores digitais os padrões são digitalizados e daí então representados através de formas mais compactas (compressão de dados) usando esquemas de codificação e aproximação. Se um padrão é codificado por uma função unidimensional ou bidimensional, então teoremas que se baseiam no uso de transformadas de Fourier garantem que tal função representativa de tal padrão pode ser reconstruída exatamente, através exemplos diversos do mesmo.

Cita ainda outras atividades relativas ao pré-processamento da imagem: operações invariantes de posição que podem ser usadas para *filtrar*, *restaurar* ou *realçar* um padrão degradado por aproximação, de modo a aumentar sua qualidade. Transformadas de Fourier e transformadas de Hadamard são exemplos típicos dessas operações.

Uma vez que se tenha realizado o pré-processamento da imagem, obtém-se explícita ou implicitamente o vetor X característico do universo do padrão P . A dimensão deste vetor pode ser exageradamente grande. Supondo que o padrão seja um determinado animal, pode-se ter o universo P apresentando como características a cor, peso, tamanho, formato, hábito alimentar, tempo médio de vida, altura, etc. Daí a importância de se tentar identificar inicialmente quais os aspectos mais relevantes que permitem caracterizar aquele padrão como o animal em questão.

É na fase de *seleção de características* durante o processo de *análise* do problema de reconhecimento de padrões (ver figura 2.3), que se efetuam os procedimentos necessários à formação do vetor Y representativo do universo F de características (aspectos) mais marcantes do padrão.

Tendo delineado o vetor de características, cumpre agora efetuar a *classificação*. Segundo Beale e Jackson ([BEA91]) existem 2 categorias de técnicas de classificação: numéricas e não numéricas. Técnicas numéricas incluiriam medidas estatísticas e uso de redes neurais, enquanto que técnicas não numéricas seriam representadas pelo uso de conjuntos difusos.

Utilizando-se técnicas numéricas (redes neurais ou medidas estatísticas) o problema de classificação poderia ser assim sumarizado ([JAI94]): Tendo um padrão "A" representado pelo seu vetor de características $A = (a_1, a_2, a_3, a_4, \dots, a_k, \dots, a_n)$ de dimensão n deve-se associá-lo a uma das c categorias do espaço C de classificação $C = (\omega_1, \omega_2, \omega_3, \dots, \omega_c)$.

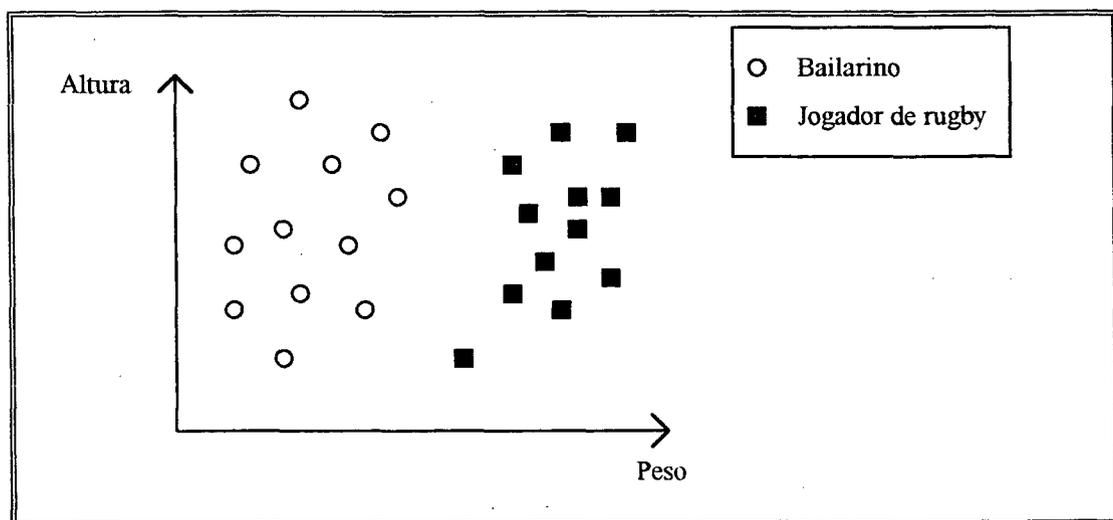


Fig. 2.6 - Características utilizadas para a classificação - Adaptado de [BEA91]

Beale e Jackson apresentam uma situação de classificação baseada nesta abordagem: Tem-se um vetor de características de dimensão 2 (altura e peso) e um vetor do espaço de classificação C também de dimensão 2 (jogador de rugby ou bailarino).

Deseja-se classificar o conjunto, separando-os em jogadores de rugby ou bailarinos, alcançando a situação mostrada na figura 2.7:

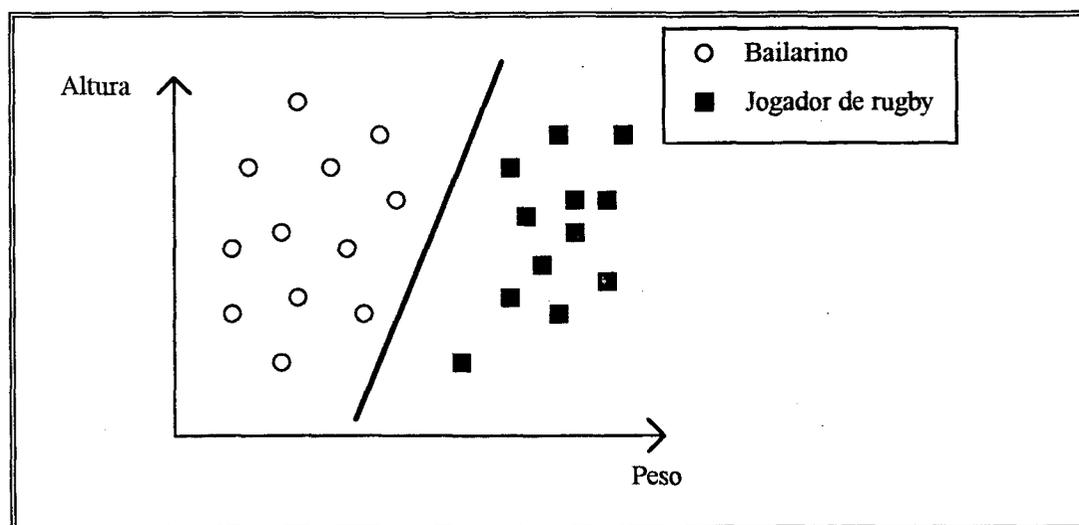


Fig. 2.7 - Classificador cuja "linha divisória" das classes tem formato de reta - Adaptado de [BEA91]

Ao se conseguir construir a linha divisória consegue-se classificar qualquer padrão apresentado colocando-o "à esquerda" ou "à direita" desta linha. No caso de se ter apenas 2 características, tem-se uma "reta" como linha divisória. Se se tiver 3 características, a linha divisória será um plano. Tendo-se 4 ou mais características um hiperplano é a linha divisória.

A linha divisória, seja ela uma reta, plano, ou hiperplano, é em última instância, o *classificador* do padrão. Ela é o delimitador de fronteiras no espaço do vetor de características. Problemas como o acima exemplificado em que o tipo de linha divisória seja uma reta, plano ou hiperplano, são ditos *linearmente separáveis*. Existem porém situações em que as classes em que se podem enquadrar os padrões não serão separadas por uma linha divisória de "formato" de reta, plano ou hiperplano.

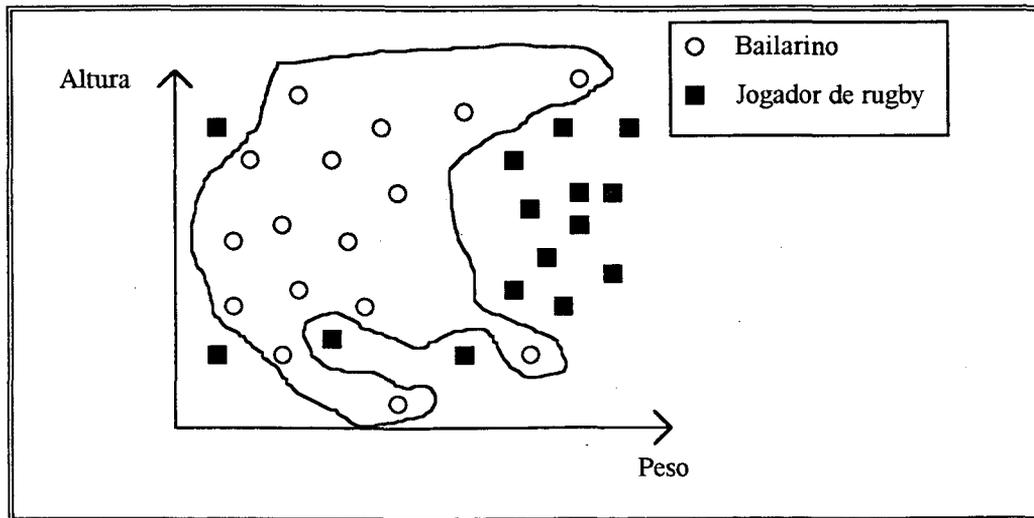


Fig. 2.8 - Classificador cuja "linha divisória" é não linear

Redes neurais vêm sendo largamente empregadas como meio de geração da linha divisória, ou seja, do *discriminador* de classes ([BEA91]). Outra técnica utilizada para classificação é a da análise dos K vizinhos mais próximos (K-NN = *K nearest neighbour*). Em essência um padrão será tomado como pertencente a uma classe em que ele apresente menor "distância". A figura 2.9 exemplifica.

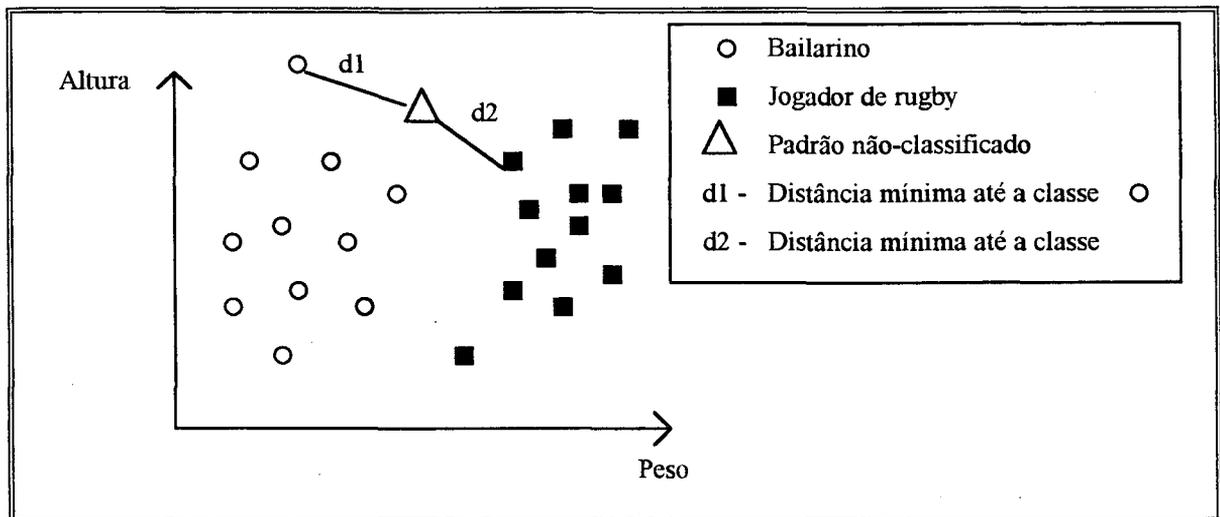


Fig. 2.9 - Classificação por comparação com os K vizinhos mais próximos - Adaptado de [BEA91]

A "distância" permite medir a similaridade entre os exemplos de padrões, com base em suas configurações do espaço de características. Dados 2 exemplares representados por seus respectivos vetores de características

$$A = (a_1, a_2, a_3, a_4, \dots, a_k, \dots, a_n) \text{ e}$$

$$B = (b_1, b_2, b_3, b_4, \dots, b_k, \dots, b_n)$$

algumas das seguintes normas para distância são:

$$\text{Distância de Hamming: } \Sigma(|a_i - b_i|) \quad (2.1)$$

$$\text{Distância Euclideana: } (\Sigma(a_i - b_i)^2)^{1/2} \quad (2.2)$$

As técnicas mais utilizadas para classificação baseiam-se no uso de medidas estatísticas e redes neurais. Classificadores bayesianos são os mais aplicados quando se quer efetuar um tratamento estatístico. A essência do classificador bayesiano repousa na base da teoria de probabilidades e probabilidade condicional. Deseja-se associar um padrão "A" representado pelo seu vetor de características $A = (a_1, a_2, a_3, a_4, \dots, a_k, \dots, a_n)$ de dimensão n a uma das c categorias do espaço C de classificação $C = (\omega_1, \omega_2, \omega_3, \dots, \omega_c)$.

Para decidir que A pertence a classe ω_i deve-se ter que

$$P(\omega_i|A) > P(\omega_k|A) \quad k = 1, 2, \dots, c \quad \text{onde } i \neq k \quad (2.3)$$

Na prática porém, para um perfeito classificador bayesiano é necessário conhecer de antemão a probabilidade de ser o padrão A dado que se sabe que a classe é ω_i (ou seja, $P(A|\omega_i)$), o que é uma situação muito rara. Baseado em padrões de exemplo (treinamento) pode-se ainda trabalhar com a estatística, se os padrões de exemplo ajudarem a fornecer ao menos a forma da função densidade de probabilidade.

Redes neurais por sua vez, se apresentam como uma boa alternativa às limitações do classificador estatístico: Uma estrutura de dados e um algoritmo relativamente simples podem ser implementados em computador e tem-se a garantia matemática de que elas podem aprender a construir a "linha divisória" separatória das classes. Perceptrons são redes neurais que resolvem problemas de classificação de separabilidade linear ([BEA91]). Já os perceptrons multicamadas são redes neurais que operam com bom desempenho tanto em problemas de natureza de separabilidade linear como não-linear.

2.3.2 - Abordagem Sintática ou Estrutural

Para descrever a configuração básica de um *problema de reconhecimento de padrão* através da abordagem de *sintática*, FU ([FU82]) apresenta o seguinte diagrama de blocos:

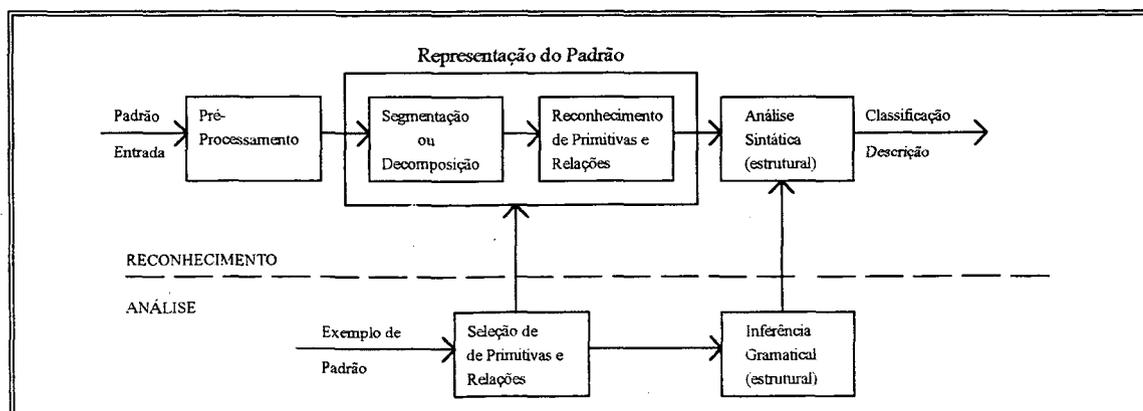


Fig. 2.10 - Diagrama de blocos para um sistema de reconhecimento de padrões - abordagem sintática (estrutural) - Fonte: [FU82]

A idéia nesta abordagem é representar o padrão através de uma estrutura que se caracterize por ser uma *sentença* de alguma linguagem. Por sentença entende-se uma expressão linguística que veicula um pensamento completo. Por exemplo, na língua alemã, "Die Gedanken sind frei" traz em sua estrutura um pensamento completo. É portanto uma sentença na Língua Alemã, mas não o é na Língua Portuguesa, Língua Italiana, Língua Inglesa entre outras.

A sentença que representará o padrão deverá estar de conformidade com uma *linguagem* que permita sua geração. Neto ([NET87]) apresenta definições para os termos "linguagem", "sentença" e outros correlatos. Neto coloca que *linguagem* é uma coleção de cadeias de símbolos de comprimento finito. Essas cadeias são denominadas *sentenças* da linguagem e são formadas pela justaposição de elementos individuais, os *símbolos* ou *átomos* da linguagem. Uma linguagem pode ser representada através de um conjunto de leis de formação de cadeias. A esse conjunto de leis de formação das sentenças dá-se o nome de *gramática*. A linguagem pode também ser representada através de regras de aceitação de cadeias, cujo conjunto denomina-se *reconhecedor*.

Formalmente as gramáticas, dispositivos de geração de sentenças das linguagens que definem, podem ser caracterizadas pela quádrupla ordenada

$$G = (V_n, V_t, P, S) \quad \text{onde,}$$

$V_n =$	Representa o <i>vocabulário não-terminal</i> da gramática, que é utilizado para definir construções auxiliares ou intermediárias na formação das sentenças.
---------	---

$V_t =$	Representa o <i>vocabulário terminal</i> , que são exatamente os símbolos ou átomos dos quais a gramática se vale para definir as leis de formação das sentenças da linguagem.
$P =$	Representa o conjunto de todas as leis de formação utilizadas pela gramática para definir a linguagem. A cada uma destas leis (regras) de formação dá-se o nome de <i>produção</i> da gramática
$S =$	É um elemento de V_n cuja propriedade é ser o não-terminal que dá início ao processo de geração de sentenças. É o <i>símbolo inicial</i> da gramática.

Gramática $G1 = (V1_n, V1_t, P1, S1)$	
onde $V1_n = \{ A, B \}$	
$V1_t = \{ 0, 1 \}$	
$P1 = \{ A \rightarrow 0A, A \rightarrow B, B \rightarrow 1B, B \rightarrow \epsilon \}$	
$S1 = A$	
Obtenção de sentença	aplicando-se
(1) $A \Rightarrow 0A$	$A \rightarrow 0A$
(2) $0A \Rightarrow 00A$	$A \rightarrow 0A$
(3) $00A \Rightarrow 00B$	$A \rightarrow B$
(4) $00B \Rightarrow 001B$	$B \rightarrow 1B$
(5) $001B \Rightarrow 0011B$	$B \rightarrow 1B$
(6) $0011B \Rightarrow 0011$	$B \rightarrow \epsilon$

Fig. 2.11 - Um exemplo de gramática - Fonte: [NET87]

Em processamento de padrões cuja natureza seja pictorial é possível utilizar a abordagem sintática. Gramáticas são construídas para os mais diversos fins em reconhecimento de padrão pictorial. Fu ([FU82]) cita aplicações de construção de gramáticas para cromossomos humanos, impressão digital, dígitos/letras escritos à mão, imagens geradas por satélite, textura de pele de répteis, formatos de raios-X e eletroencefalograma, etc.

Para ilustrar um uso de gramática, a figura 2.12 apresenta o exemplo de gramática para cromossomos apresentado em [HAM92].

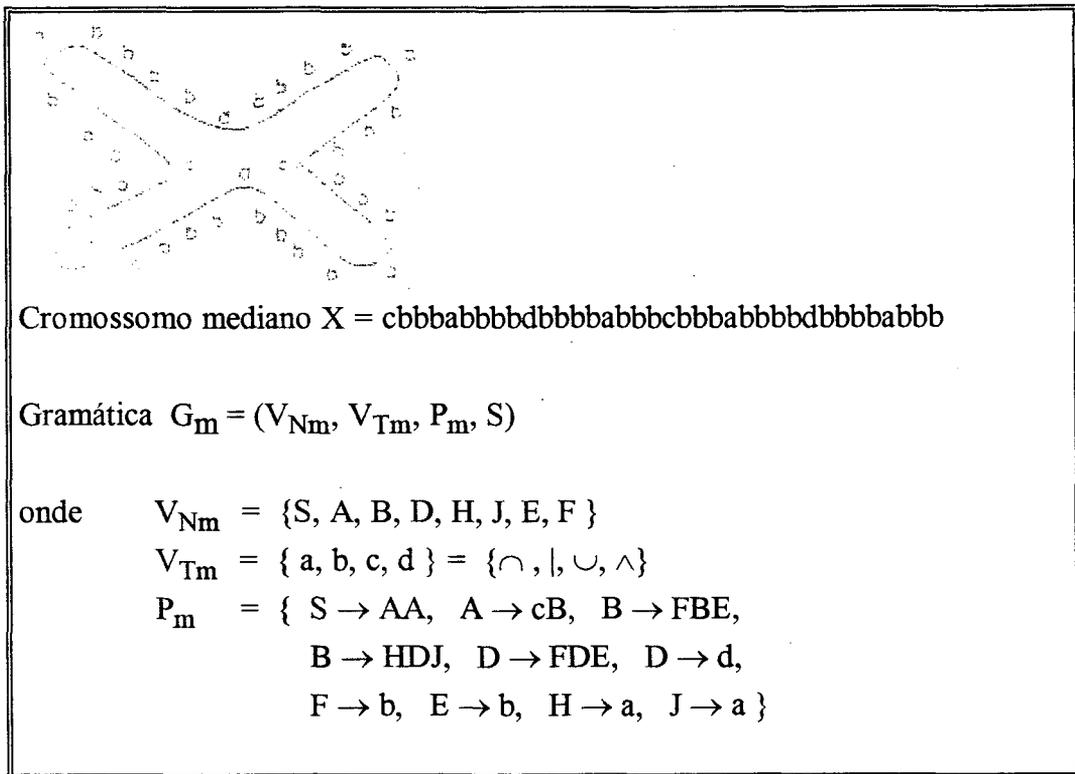


Fig. 2.12 - Um exemplo de gramática para cromossomos - Adaptado de [HAM92]

O vocabulário terminal da gramática, como dito anteriormente, contém os átomos que constituem as sentenças da linguagem. Durante a etapa de análise (ver figura 2.10), a fase de seleção de primitivas corresponderia à identificação dos átomos que compõem o conjunto do vocabulário de terminais.

No exemplo mostrado na figura 2.12, tem-se 4 primitivas, a, b, c, d representando respectivamente os pedaços de contorno com as formas \cap , \cup , $|$, \wedge . É com estas primitivas que serão descritas as "infinitas" sentenças possíveis que representam o formato de cromossomos. Fu ([FU82]) relata a importância da etapa de seleção de primitivas. As primitivas corresponderiam, por analogia, ao vetor de características Y utilizado na abordagem de decisão-teórica. Por isso deve-se escolher primitivas marcantes, que concatenadas umas às outras permitam deixar explícitas as estruturas caracterizadoras do padrão em questão.

As primitivas devem ter como característica a facilidade de prover uma compacta porém adequada descrição dos padrões, e devem ser facilmente extraídas ou reconhecidas pelos mecanismos do tipo função discriminante (abordagem de decisão-teórica). As primitivas escolhidas são altamente dependentes do domínio do problema: assim, em processamento de fala por exemplo, fonemas são

ótimas primitivas. Para reconhecimento de dígitos/letras escritos à mão *strokes* (pequenos pedaços de traços contínuos com uma determinada direção) são mais adequados ([FU82]).

Para padrões pictórias em geral, Fu afirma não existir uma "primitiva universal". Cita por exemplo que figuras podem ser aproximadas por polígonos arbitrários, que por sua vez podem ser expressos como sendo a união de um número finito de polígonos convexos.

Para se construir uma gramática existem duas formas: ou se constrói "a mão", ou se infere através de exemplos do padrão uma possível gramática. O conhecimento *a priori* que o projetista tenha sobre o padrão em questão permite a construção "a mão", direcionando-se para a obtenção de uma gramática mais "enxuta" e eficiente do que uma gramática gerada por inferência pela máquina. A gramática gerada por inferência, baseada em exemplos de padrões tem a desvantagem de não contar com o conhecimento *a priori*, e pode, dependendo dos padrões utilizados, gerar uma gramática que funcione, mas que seja mais complexa e com possíveis redundâncias em sua composição.

Existem vários tipos de gramáticas que se adequam melhor para cada problema particular. Por exemplo, numa aplicação para reconhecimento de impressões digitais e representação de caracteres chineses Fu cita o uso de uma gramática com estrutura de árvore para tal tarefa. Gramáticas com estrutura de grafos servem para representar cenários.

Uma vez que se tenha uma gramática definida para determinada aplicação, como por exemplo o reconhecimento de cromossomos humanos, pode-se então pensar na etapa de reconhecimento propriamente dito (figura 2.10). Inicialmente toma-se um padrão desconhecido e efetua-se o pré-processamento. A função da atividade de pré-processamento é semelhante à descrita para a abordagem de discriminante: sobre o padrão em sua forma "bruta", efetuam-se "lapidações" convenientes procurando deixá-lo o mais enxuto possível sem contudo perder seu teor informativo. As atividades de codificação, filtragem, restauração e realce são efetuadas.

Em seguida o padrão sofre os processos de segmentação e de busca e explicitação de suas primitivas. Fu ([FU82]) cita que a atividade de segmentação pode ser categorizada em três classes: segmentação por clusterização, por detecção de contornos (*edge detection*) ou por extração de região.

Destaca ainda que qualquer que seja a técnica de segmentação utilizada (através de algoritmos matemáticos) será necessária suplementação heurística envolvendo semântica, conforme a natureza do padrão envolvida. Em outras palavras, segmentação é um processo que apresenta dependência do padrão.

A idéia da técnica de detecção de contornos (*edge detection*) é baseada na detecção de descontinuidade. Um contorno ou delimitador (*boundary*) é um local onde ocorre uma mudança mais ou menos abrupta da tonalidade de cinza na imagem. Técnicas de detecção de contorno se baseiam em aspectos matemáticos tais como operadores gradiente, aproximações funcionais, pesquisas heurísticas com programação dinâmica, relaxamento, ajuste de curvas ([FU82]).

A terceira técnica de segmentação, de extração de região, baseia-se no princípio de efetuar contínuas junções ou separações de regiões adjacentes conforme suas peculiaridades de tonalidades de cinza: Se um grupo de pixels de uma imagem, geograficamente próximos tem tonalidade aproximadamente igual, então a região é "unificada" (junção). Se por outro lado estes mesmos pixels apresentam tonalidades bastante discrepantes entre si, então a região é dividida, sendo separada. Horowitz e Pavlidis ([HOR76]) propõem um algoritmo denominado *split-and-merge* que opera desta forma.

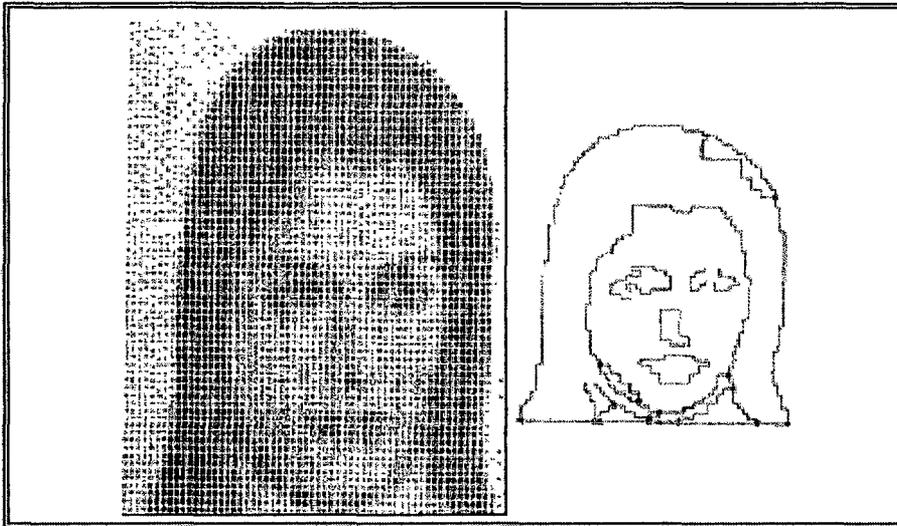


Fig. 2.13 - O processo de segmentação - Técnica de extração de região - Resultado obtido através do algoritmo *split-and-merge* - Fonte: [HOR76]

Uma vez que se tenha o padrão segmentado, ou seja, decomposto até um certo nível, espera-se ter ao alcance as primitivas componentes do padrão explicitadas, de modo que seja fácil manipulá-las. Faz-se necessário agora estabelecer as *relações* entre as primitivas, pois daí tem-se a caracterização do padrão numa forma (normalmente uma *string*) em que a gramática do padrão pode produzir.

Procede-se então à tarefa de análise sintática: um reconhecedor, também denominado *parser*, é construído com base na gramática, para receber strings representativas do padrão e aceitá-las ou não como pertencentes à linguagem descritora de tal padrão. Duas estratégias podem ser aplicadas para se construir um reconhecedor: a estratégia descendente (*top-down*) e a estratégia ascendente (*bottom-up*).

Na estratégia descendente (*top-down*) parte-se do símbolo inicial da gramática (S) e tenta-se, aplicando-se sucessivamente as regras de produção da gramática, gerar a *string* representativa do padrão em questão. Na estratégia ascendente (*bottom-up*) parte-se da *string* representativa do padrão que se quer classificar e tenta-se chegar ao símbolo inicial da gramática (S).

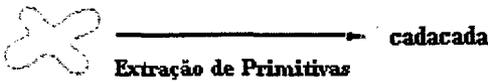
Padrão desconhecido			
			
<i>Parser String</i>	Regra	Unificação	Padrão
S	AA	-	cadacada
AA	$A \rightarrow cB$	c	cadacada
BA	$B \rightarrow HDJ$	-	adacada
HDJA	$H \rightarrow a$	a	adacada
DJA	$D \rightarrow d$	d	dacada
JA	$J \rightarrow a$	a	acada
A	$A \rightarrow cB$	c	cada
B	$B \rightarrow HDJ$	-	ada
HDJ	$H \rightarrow a$	a	ada
DJ	$D \rightarrow d$	d	da
J	$J \rightarrow a$	a	a
(vazio)			(vazio)
Padrão Reconhecido			

Fig. 2.14 - Um exemplo de reconhecedor top-down para cromossomos representados pela gramática apresentada na figura 2.12 - Fonte: [HAM92]

Sintaxe é uma palavra de origem grega que significa "ordenar". A abordagem sintática portanto, tem esta denominação devido à importância que se dá à ordenação, a estruturação em que se apresenta o padrão. As relações (posicionamento por exemplo) entre as primitivas do padrão são levadas em consideração durante o processo de identificação do padrão. Sumarizando o processo de reconhecimento "gramatical" tem-se as seguintes fases:

- Conhecimento do domínio do padrão para construção de uma gramática.
- Tendo-se a gramática, o sistema de reconhecimento "gramatical" é apresentado a um padrão que pode ou não ser uma sentença da linguagem que tal gramática gera.
- Codifica-se convenientemente o padrão. A *codificação* é um processo de estruturação que procura deixar explícitas as primitivas do padrão e suas relações.
- Utiliza-se, na etapa de identificação propriamente dita um reconhecedor que irá ou não aceitar o padrão *codificado* como pertencente à linguagem.

2.3.3 - *Template Matching*

Outra forma de reconhecimento do padrão, já codificado e que não use um reconhecedor é através da técnica denominada *template matching*, traduzindo-se em algo como "correspondência aos modelo". Esta técnica pode ser vista como um caso especial da proposta de abordagem sintática.

Uhr cita a técnica de *template matching* como uma abordagem bastante utilizada na área de reconhecimento de padrões. *Templates* parece significar o que filósofos e psicólogos em geral denominam por "idéias" ou "imagens" ([UHR73]).

Os programas de reconhecimento de padrões possuem um conjunto de *templates* já codificados e sendo representantes típicos das diversas classes que o padrão de interesse apresenta. O padrão que se deseja classificar é então comparado com os diversos templates do programa. Caso ocorra uma correspondência do padrão com algum dos *templates* do programa, então o reconhecimento é finalizado com resultado satisfatório em termos de reconhecimento.

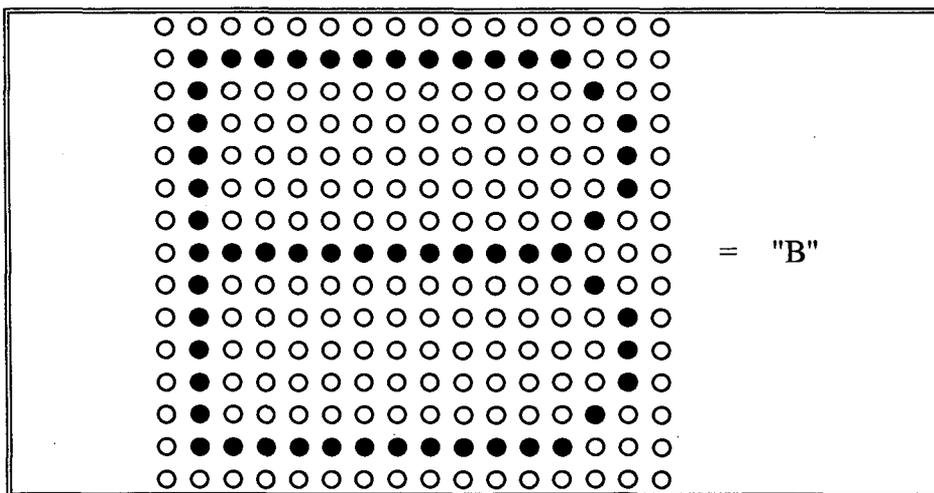


Fig. 2.15 - Um exemplo de *template* para a letra "B". Matriz de bits 15 x 16
- Adaptado de [UHR73]

Suponha que tem-se um *template* para uma letra "B", armazenado numa matriz de bits de dimensão 15 x 16. O processo de *template matching*, deve então apresentar vários dispositivos que permitam de fato efetuar o reconhecimento, superando restrições tais como:

- Reconhecer um padrão de "B" em tamanho reduzido ou ampliado.
- Reconhecer um padrão de "B" com alguma rotação de $(0 < \theta < 360)$.
- Reconhecer um padrão de "B" que apresente algum "ruído" ou esteja incompleto.

Existem propostas de reconhecimento de padrão baseadas na técnica de *template matching*. Para todas elas é crucial uma definição precisa do *modelo de codificação* utilizado para o padrão. Para padrões de imagem em geral, é bastante popular o uso do código octal proposto por Freeman ([FRE61]), principalmente para descrever contornos de objetos.

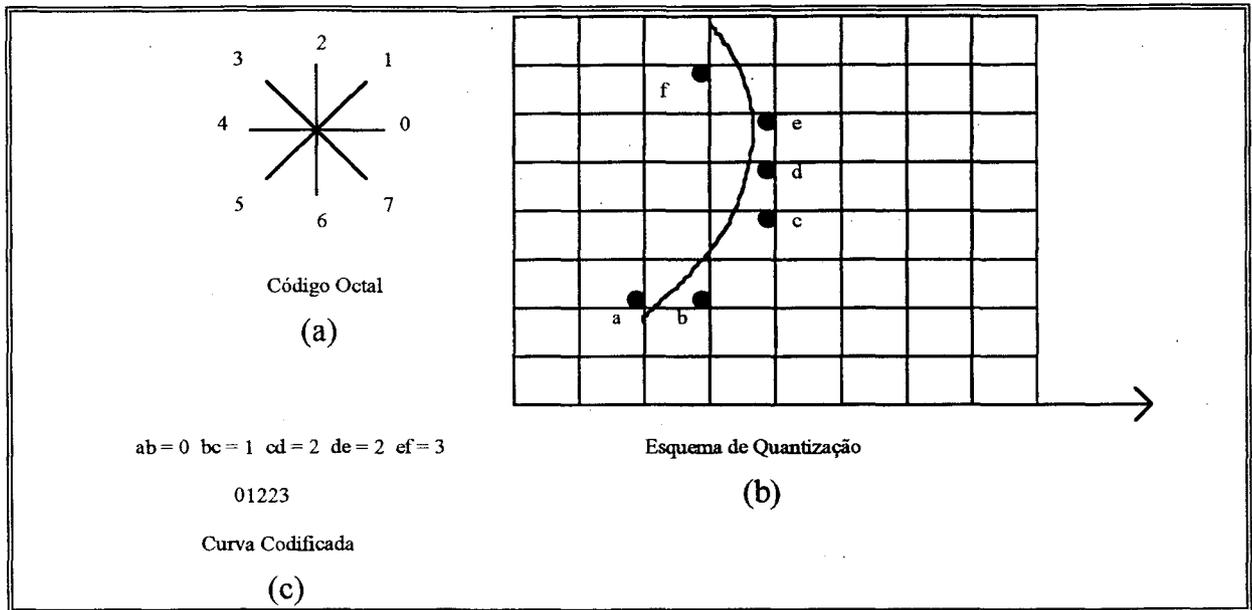


Fig. 2.16 - (a) Código de Freeman (Código Octal) (b) Uma curva em R^2 (c) Codificação da curva conforme o código de Freeman - Adaptado de [FRE64]

Uma vez que se tenha tanto o *template* T como o padrão P que se deseja reconhecer devidamente codificados (com o código de Freeman por exemplo), deve-se efetuar uma comparação entre T e P e efetuar algum tipo de cálculo que indique a "distância" que separa ambos. Em função do valor desta "distância" aceita-se ou rejeita-se P como sendo uma instância do *template* T. E esta é a proposta geral da técnica de *template matching*: calcular distância entre T e P. Esta distância, de modo geral, relaciona-se com a aferição de *medidas de similaridade*.

Uma aplicação apresentada por Freeman ([FRE64]) que utiliza o código octal, direcionada ao reconhecimento de padrões é o jogo de "quebra-cabeça". Uma figura, normalmente contendo um cenário, é "recortada" em vários pedaços irregulares e separada e embaralhada. O jogador deve tentar encaixar corretamente os pedaços adjacentes, mediante análise de seus contornos. Freeman aponta para as dificuldades de descrição adequada dos pedaços, a manipulação destes levando-se em conta operações de rotação e correspondência (*matching*) das partes, e finalmente avaliação sobre o encaixe entre 2 pedaços, se de fato é um encaixe correto.

Feder e Freeman ([FED66]), em outra aplicação, utilizam o código de Freeman e apresentam uma medida de similaridade entre pequenos fragmentos de curvas e porções de curva. O algoritmo de

correlação de cadeias de códigos de Freeman, apresentado pelos autores para efetuar o cálculo da similaridade (distância) tem a limitação de ser altamente sensível à rotação e escala. Isto é, não terá boa performance se, por exemplo, o padrão P for apresentado com uma inclinação de, digamos, 20° em relação ao *template* T, ou mesmo se a escala de P for menor ou maior que a escala de T.

Wagner ([WAG74]), LU e FU ([LU78]) apresentam trabalhos em que se utiliza um cálculo de distância entre 2 *strings* representativas de algum padrão numa determinada codificação. A ideia em linhas gerais é a seguinte:

Dadas 2 *strings* A e B, a distância entre A e B é dada pela sequência de custo mínimo de operações de edição necessárias para transformar uma das strings (A por exemplo) na outra. As operações de edição são, *inclusão*, *deleção* e *substituição* cada uma com um custo associado.

Supondo que as strings A e B são representações de cromossomos:

$$A = \text{cbabdbb} \quad \text{e} \quad B = \text{cbbabdb}$$

Então as seguintes operações em A conduzem até B:

Operação	Caracter de Entrada	Caracter de Saída	Configuração da String
Substituição	b	d (cbab <u>d</u> bb)	A = cbab <u>b</u> bb
Substituição	d	b (cbab <u>b</u> bb)	A = cbab <u>b</u> db
Inclusão	b (cbab <u>b</u> db)	-	A = cbbab <u>b</u> db = B

Supondo ainda que o custo de inserção, deleção e substituição seja 1 para os três casos, o custo mínimo para "levar" A até B é 3.

Sanfeliu e Fu ([SAN83]) utilizam estrutura de grafo para representação do conhecimento. Assim, uma imagem será descrita em termos de estrutura de grafo. Aqui, os grafos são caracterizados através de uma gramática descritiva de grafos (DGG - Descriptive Graph Grammar). Para dois padrões A e B, convenientemente caracterizados por grafos gerados por uma DGG, é efetuado um cálculo de distância entre os grafos que se caracteriza por 2 cálculos:

- O cálculo do número mínimo de transformações (operações de inclusão, deleção e substituição) necessários para transformar um grafo (A por exemplo) no outro, e
- O cálculo do custo de reconhecimento dos nodos do grafo.

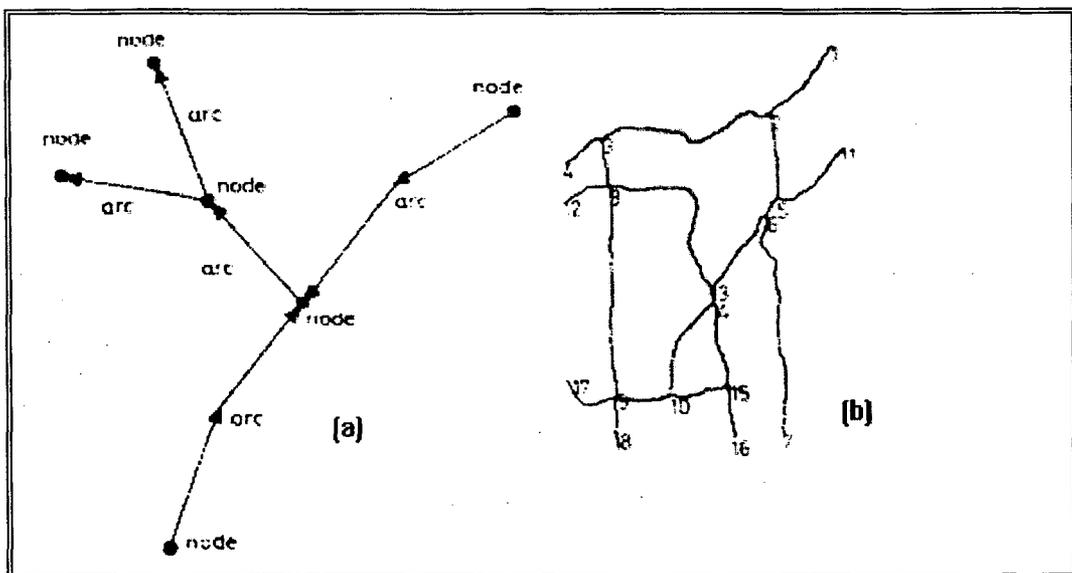
Para este último cálculo, cada nodo do grafo apresenta como característica uma ou mais funções de custo usadas para aferir medidas de similaridade entre nodos de entrada e de comparação.

Eshera e Fu ([ESH84]) trabalhando também com grafos como estrutura de representação, apresentam uma distância, também baseada em custo mínimo de transformações, para os denominados grafos de atributos relacionais (ARG - Attributed Relational Graphs).

Uma das propostas para contornar o problema do *template matching* "rígido" é a utilização de técnicas de "relaxamento" discreto para estruturas relacionais. Kitchen e Rosenfeld apresentam trabalhos neste sentido ([KIT79]), ([KIT80]). Nesta abordagem, o elemento de partida da teoria é a *estrutura relacional*, que se caracteriza por uma tripla $\langle X, \Pi, \phi \rangle$, onde X é um conjunto finito de nodos, Π é um conjunto finito de predicados, cada um com uma ordem associada, e ϕ é uma função que mapeia cada predicado P para uma relação n -ária sobre X .

Em função dessas estruturas relacionais, cria-se em geral, duas estruturas, M e W , representando respectivamente um modelo (ou parte) e um universo, no qual procura-se uma instância de M . A denominação "relaxamento" é usada para caracterizar o método, segundo o qual inicia-se com a atribuição para cada "ponto" da estrutura relacional M um valor que gradativamente vai "convergindo" ao seu valor real através da análise dos valores dos "pontos" vizinhos na estrutura. Conjuntos difusos ([ZAD65]) são empregados neste método.

Fukunaga, Murata, Asano, Izumi ([FUK91]) apresentam uma aplicação usando grafos para representar imagens de mapas. Utilizam o código de Freeman e expressões algébricas para representar "linhas" do mapa, conseguindo com isto representar as estruturas do mapa de forma mais simples. Além disso o modelo é não-sensitivo quanto à escala, rotação e modificação de tonalidade dos níveis de cinza dos pixels da imagem. Para o processo de *matching*, é utilizado uma grau de similaridade entre os nodos do grafo que representa a imagem.



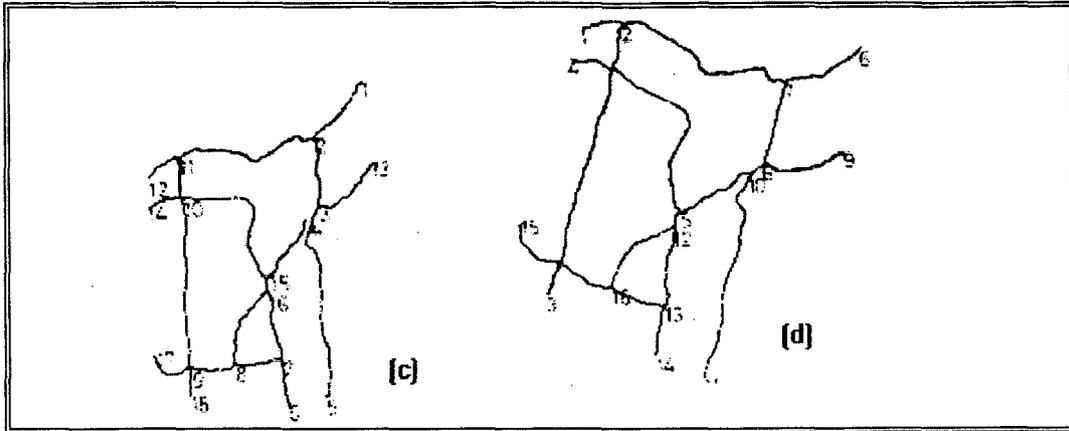


Fig. 2.17 - (a) Um grafo de uma imagem mostrando nodos e arcos (b) Uma mapa *template* (c) O mesmo mapa em diferente escala (d) O mesmo mapa com pequena rotação - Fonte: [FUK91]

Shaout e Suk ([SHA92]) apresentam um tipo especial de *grafo de atributos difusos* (AFG - *Attributed Fuzzy Graph*), o assim denominado *grafo de atributos difusos concorrentes* (AFT - *Attributed Fuzzy Tournament*), para representar incertezas presentes em cenários. A análise/reconhecimento de padrões em geral dá-se através da técnica de *template matching*, mediante a definição de uma distância entre dois AFT's. Em linhas gerais temos que um AFT é uma sêxtupla $\langle V, A, B, E, X, C \rangle$ onde: V é o conjunto de nodos do grafo, A é o conjunto de arcos direcionados do grafo, B é o conjunto de atributos possíveis para os nodos do grafo, E é o conjunto de atributos dos arcos do grafo, X é o conjunto de vetores de graus de pertinência para cada atributo de nodo para cada nodo do grafo e C é o conjunto de vetores de graus de pertinência para cada atributo de arco para cada arco do grafo. Devido à importância desta estrutura para o modelo de identificação de nematóides, uma seção abordando tal estrutura é apresentada adiante neste capítulo.

Tsang e Bilings ([TSA92]), ([TSA94]) comentam sobre análise e reconhecimento de formas curvas dos contornos de objetos. Citam o modelo de análise e reconhecimento de contornos para formatos bidimensionais, proposto por Kashyap e Chellapa ([KAS81]) e Dubois e Glanz ([DUB80]). O modelo é denominado "autoregressivo", e em essência, a proposta é aproximar o contorno de determinado objeto através de vetores da taxa de angularidade equiespacejada entre o contorno e o centróide do objeto.

O *matching* neste modelo dá-se através dos parâmetros do modelo autoregressivo (AR model). Sekita, Kurita e Otsu ([SEK92]) propuseram uma abordagem de reconhecimento para o modelo AR, em que se reconhece justamente através de seus parâmetros. Contudo, o sucesso desta técnica de *matching* é altamente dependente do número de pontos usados na especificação do contorno. Tsang e Billings propõem um algoritmo denominado de *algoritmo de estimação ortogonal*, que procura contornar este tipo de problema. Segundo os autores este algoritmo irá basear o *matching* em

parâmetros ortogonais, que é mais vantajoso em relação ao *matching* sobre os parâmetros do modelo AR. Além disso os parâmetros ortogonais são invariantes ao tamanho, rotação, translação e ponto inicial de delimitação do contorno do padrão em questão.

2.3.4 - Comparação entre as Abordagens Discriminante e Sintática

Em muitos casos é difícil distinguir diferenças entre as abordagens discriminante e sintática. Em alguns casos ocorre uma complementaridade na aplicação de reconhecimento que se está trabalhando.

A fase de seleção das características em ambas abordagens são similares no tocante ao fato de não apresentarem um mecanismo universal de seleção: a seleção dos aspectos decisivamente caracterizadores é completamente dependente da natureza do padrão de interesse, requerendo conhecimento *a priori* para uma boa escolha.

Contudo, os aspectos de destaque selecionados na abordagem sintática representam subpadrões, ou primitivas, enquanto que na abordagem discriminante os aspectos podem ser qualquer conjunto de natureza numérica. Quando se tem a preocupação apenas com classificação de padrões, muito provavelmente a abordagem discriminante será adequada. Quando porém além de classificar, é necessário efetuar descrição do padrão, a abordagem sintática parece ser a melhor opção ([FU82]).

Nota-se também que a etapa de seleção de primitivas para a abordagem sintática é efetuada por métodos discriminantes. Assim a abordagem discriminante é utilizada como parte integrante do sistema de reconhecimento de padrões sintático.

Métodos sintáticos são preferíveis aos métodos discriminantes quando os padrões são altamente complexos e estruturados (por exemplo, em descrição de cenários). Porém, se os padrões de interesse apresentam alta taxa de "ruídos", a gramática e o reconhecedor para estes padrões se tornarão mais complexos. É preferível sob este contexto, utilizar algum método discriminante.

A partir de um conjunto relativamente pequeno de primitivas, e um conjunto também relativamente pequeno de regras de produção, uma gramática permite descrever infinitas instâncias do padrão para o qual ela foi projetada, graças à natureza recursiva de suas regras de produção. A desvantagem desta característica é que o padrão pode na prática apresentar um conjunto finito de instâncias, e a gramática permitir gerar infinitas instâncias, podendo portanto gerar exemplares válidos gramaticalmente mas sem nenhuma vinculação com a realidade do padrão.

2.4 - Processamento de Imagens

Processamento de imagens refere-se a operações que transformam imagens em outras imagens. Os estudos iniciais sobre reconhecimento de padrões eram direcionados em sua quase totalidade para padrões pictoriais. Por isso, o processamento de imagens distinguia-se da tarefa de reconhecimento de padrões (pictoriais) propriamente dita. Em processamento de imagens, "preparava-se" a imagem para a posterior fase de *reconhecimento da imagem*. Processamento de imagens está ligado basicamente a duas etapas: codificação de imagem e realce da imagem.

Para codificação de imagem, o método mais utilizado é o de transformadas. A idéia é aplicar um transformada inversível sobre a imagem original, aproximar esta imagem e daí então construir uma nova imagem através da reversão da transformada aplicada.

Para se obter o realce da imagem, objetivando destacar com mais nitidez o "teor informativo" embutido na figura, utiliza-se não raramente o método de análise da tonalidade de cinza, onde se procura destacar aspectos de relações entre pixels de uma determinada vizinhança e aumentar ou diminuir conforme o caso suas intensidades, a fim de se destacar contornos em geral.

Como se observa do acima exposto, processamento de imagem é uma atividade inicial de pré-requisito para o processo completo de reconhecimento de padrões. Equivale à etapa de pré-processamento destacada nas figuras 2.3 e 2.10 que demonstram o esquema de funcionamento de um sistema de reconhecimento de padrões nas abordagens discriminante e sintática respectivamente. Ao se realizar a fase de segmentação de uma imagem sob a abordagem sintática, pode-se também se considerar que está ocorrendo processamento de imagem, pois afinal, aspectos inerentes a constituição da imagem (intensidade de píxeis) estão sendo levados em conta para se obter a segmentação.

Young e Roos ([YOU88]) apresentam uma figura ilustrativa da diferença entre *processamento de imagem* e *análise de imagem*. Para Young e Roos processamento de imagem é a manipulação de uma imagem para produção de outra imagem, enquanto análise de imagem está ligada com a extração de informação (não necessariamente pictorial) da imagem. Observando a figura 2.10, a análise da imagem estaria ligada à fase de análise sintática, dentro da etapa de reconhecimento.

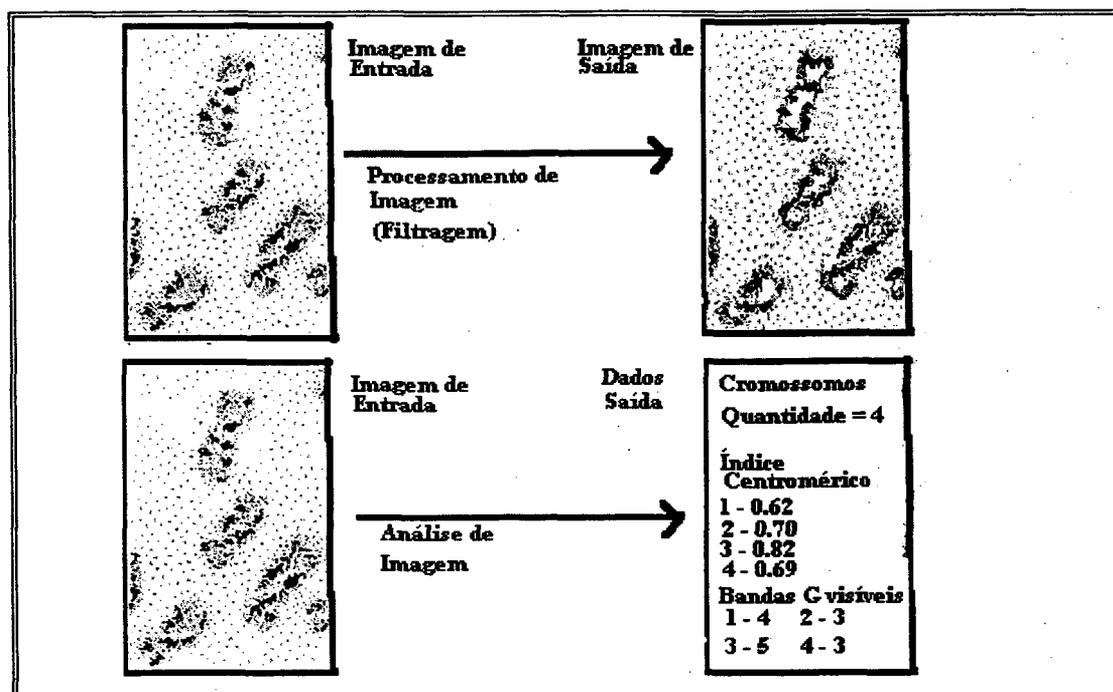


Fig. 2.18 - Diferença entre *processamento de imagem* e *análise de imagem* - Adaptado de [YOU88]

2.5 - Conjuntos Difusos

2.5.1 - Conceitos Introdutórios de Lógica

As raízes da lógica podem ser atribuídas a Aristóteles. A lógica moderna provém, em grande parte, do trabalho do filósofo alemão Gottlob Frege no século XIX. ([NOL91]).

Gomes ([GOM83] e [GOM92]) coloca que a moderna lógica matemática vê a sua própria pedra fundamental na *sentença*.

Definição - Sentença

Sentença é uma expressão lingüística que veicula um pensamento completo.

Considerando-se as duas expressões abaixo,

Paris	(2.4)
Paris é a capital da França	(2.5)

Percebe-se, conforme a definição de sentença, que a expressão (2.4) não é uma sentença, pois não expressa um pensamento completo, enquanto que a expressão (2.5) veicula um pensamento completo, sendo portanto, uma sentença.

Existem sentenças de tipo declarativa, interrogativa e imperativa, sendo que, para a lógica, interessam as sentenças declarativas pois são elas as únicas das quais pode-se dizer serem verdadeiras ou falsas. O que é verdade e o que é falsidade não se discute na lógica, mas sim na teoria do conhecimento ([GOM92]).

"Paris é a capital da França"	Sentença Declarativa	(2.6)
"Que dia é hoje ?"	Sentença Interrogativa	(2.7)
"Leia o livro !"	Sentença Imperativa	(2.8)

Uma sentença declarativa apresenta, um certo "teor informativo", que as sentenças do tipo interrogativa e imperativa não apresentam. Este "teor informativo" da sentença, refere-se à sua *proposição*. Uma proposição é uma entidade abstrata da sentença e refere-se aos valores de verdade (ou valores lógicos [FIL90]) que a sentença pode assumir.

A assim chamada Lógica Clássica tem suas bases em três postulados que em última instância expressam a validade de apenas dois valores de verdade: *verdadeiro* e *falso*. São os seguintes, os postulados da Lógica Clássica ([GOM92]):

Princípio da Não-Contradição: Dadas uma proposição (de uma sentença) e sua negação, não é o caso que ambas sejam ao mesmo tempo verdadeiras ou que ambas sejam ao mesmo tempo falsas.

Princípio do Terceiro Excluído ("Tertium non Datur"): Dadas uma proposição (de uma sentença) e a sua negação, uma delas tem valor de verdade verdadeiro, e a outra delas tem valor de verdade falso (e nada além disso).

Princípio da Identidade: Para toda entidade X vale: X é idêntica a X

Os princípios da Não-Contradição e Terceiro Excluído permitem estabelecer, de imediato, o seguinte corolário:

Princípio da Bivalência: Cada proposição tem um e apenas um valor de verdade: Verdadeiro (V) ou Falso (F).

Por virtude deste corolário, diz-se também que a Lógica Clássica, também designada por Lógica Matemática, é uma *lógica bivalente*, pois a proposição de uma sentença ou tem valor de verdade verdadeiro, ou tem valor de verdade falso.

As sentenças declarativas, que são objeto de estudo da lógica, podem ser representadas através de conjuntos, e daí o interesse em Teoria de Conjuntos como ferramenta auxiliar da lógica para construção de sentenças (que veiculam alguma idéia). Uma revisão razoável da Teoria de Conjuntos é apresentada em Lipschutz ([LIP72]).

2.5.2 - Sentenças e Conjuntos

Observando-se sentenças declarativas tais como

"Maria é brasileira"	(2.9)
"Os brasileiros são latinos"	(2.10)
"Paris é a capital da França"	(2.11)

nota-se que o verbo "ser", empregado nas três sentenças, representa, em termos semânticos, respectivamente a "pertencer a" em (2.9), "estar contido em" em (2.10) e "ser idêntico a" em (2.11). Ou seja, pode-se dizer que "Maria pertence ao conjunto dos brasileiros", "o conjunto dos brasileiros está contido no conjunto dos latinos" e "Paris é idêntico a (o mesmo que) a capital da França".

Através das relações existentes entre elemento e conjunto e entre conjunto e conjunto, pode-se expressar sentenças declarativas. Eis alguns exemplos:

a) "Sócrates é homem"

É uma relação entre elemento e conjunto. Seja $x =$ Sócrates o elemento e $H =$ conjunto dos homens. Então em termos conjuntistas tem-se:

$$x \in H \quad (\text{"Sócrates pertence ao conjunto dos homens"}).$$

b) "Todo homem é mortal"

É uma relação entre dois conjuntos. Seja H = conjunto dos homens e M = conjunto dos mortais. Tem-se então:

$$H \subset M \quad (\text{"O conjunto dos homens está contido no conjunto dos mortais"}).$$

c) "Ontem comprei pão na padaria da esquina"

É uma relação entre elemento e conjunto. Seja x = pão e A = conjunto das coisas que comprei ontem na padaria da esquina. Então:

$$x \in A \quad (\text{"Pão pertence ao conjunto das coisas que comprei ontem na padaria da esquina."}).$$

d) "Nenhum triângulo é quadrado"

É uma relação entre dois conjuntos. Seja T = conjunto dos triângulos e Q = conjunto dos quadrados. Então:

$$T \not\subset Q \quad (\text{"O conjunto dos triângulos não está contido no conjunto dos quadrados"}).$$

e) "Paris é a capital da França"

É uma relação de identidade entre dois elementos. Seja x = Paris e y = capital da França. Então

$$x = y \quad (\text{"Paris é o mesmo que a capital da França"}).$$

Ressalta-se, novamente, que para a Lógica Clássica, cada uma das proposições das sentenças acima ou são consideradas como sendo verdadeiras ou então como falsas. Uma sentença do tipo

$$\boxed{\text{"João é alto"}} \quad (2.12)$$

pode apresentar uma interpretação conjuntista para a Lógica Clássica da seguinte forma: j é o elemento "João" e A é o conjunto dos objetos altos. A proposição apresentará valor de verdade ou *verdadeiro* ou *falso*. Ou $j \in A$ ou $j \notin A$. Não há meio termo.

Zimmerman ([ZIM91]) refere-se a este tipo de sentença, uma sentença *crisp*. Por *crisp* entende-se dicotômica, do tipo sim-ou-não ao invés do tipo mais-ou-menos. Ainda segundo Zimmermann, em situações reais a proposição das sentenças não apresentam este caráter estritamente dicotômico (verdadeiro ou falso). Cita ainda uma colocação do filósofo Bertrand Russell a respeito da visão dicotômica que está naturalmente embutida na Lógica Clássica:

"All traditional logic habitually assumes that precise symbols are being employed. It is therefore not applicable to this terrestrial life but only to an imaged celestial existence". ([ZIM91]).

No caso da sentença (2.12) pode-se perguntar: o que é algo considerado alto ? Será que João não poderia ser considerado "mais ou menos" alto, ou "um pouco" alto, ou "um pouquinho" alto ? Para a Lógica Clássica não. A sentença (2.12) ou é verdadeira ou é falsa.

2.5.3 - Lógica Difusa

Considerando ainda a sentença (2.12) e recolocando as perguntas sobre o fato de João poder ser considerado "mais ou menos" alto, ou "um pouco" alto, ou "um pouquinho" alto, percebe-se, à luz das próprias perguntas, que existem *incertezas* no contexto em que se considera o fato de João ser alto. As incertezas concernem em relação ao *grau de verdade* da proposição da sentença "João é alto". Enfatizando-se mais uma vez, para a Lógica Clássica, uma proposição ou é verdadeira ou é falsa. A modelagem do mundo real através da Lógica Clássica, desconsidera a incerteza.

Chorafas ([CHO90]) coloca que Lógica Difusa é a lógica que trabalha com incertezas. Quer dizer, o contexto ao qual se refere a sentença, não necessariamente precisa estar completamente conhecido.

Zimmermann ([ZIM91]) por sua vez chama a atenção para dois tipos de incerteza. Um tipo de incerteza, é a *incerteza estocástica*, tratada apropriadamente pela teoria de probabilidades e estatística. Neste caso, as sentenças são "bem definidas". Uma sentença do tipo "Hoje é segunda-feira" pode, dependendo do contexto, carregar uma incerteza de natureza estocástica. Supondo que o locutor da frase não saiba o dia da semana, uma coisa é certa: não há dúvidas de que são sete os dias da semana e que portanto é possível obter uma *probabilidade* da sentença ser verdadeira ou não.

O segundo tipo de incerteza comentada por Zimmermann refere-se ao subjetivismo de um termo que pode ser um *termo vago*. Neste caso, o contexto da sentença depende da interpretação dada ao termo. Na sentença (2.12) "alto" é um termo vago, pois depende da concepção do termo "alto" para o locutor/receptor da sentença.

Uma das vantagens que a Lógica Difusa apresenta em relação à Lógica Clássica é o fato de que, na Lógica Difusa, a proposição da sentença pode apresentar *infinitos* valores de verdade (graus de verdade), em contraposição aos dois únicos valores de verdade (verdadeiro/falso) da Lógica Clássica.

Assim como a Lógica Clássica se apóia na Teoria de Conjuntos "tradicional", a Lógica Difusa se apóia na teoria de *conjuntos difusos* formalizada na década de 60 por Zadeh ([ZAD65]).

"The notion of a fuzzy set provides a convenient point of departure for the construction of a conceptual framework which parallels in many respects the framework used in the case of ordinary sets, but is more general than the latter and, potentially, may prove to have a much wider scope of applicability, particularly in the fields of pattern classification and information processing. Essentially, such a framework provides a natural way of dealing with problems in which the source of imprecision is the absence of sharply defined criteria of class membership rather than presence of random variables" ([ZAD65])

2.5.4 - Sentenças e Variáveis Linguísticas

Segundo Zimmerman ([ZIM91]) a noção de variável linguística é fundamental para a Lógica Difusa. Foi definida em 1973 por Zadeh e é caracterizada por uma quintupla $(x, T(x), U, G, \tilde{A})$ onde:

- x é o nome da variável.
- $T(X)$ (ou simplesmente T) denota o conjunto de termos de x , cujo valor é uma variável difusa.
- U é o universo de discurso o qual o valor da variável difusa dos termos de x pode variar.
- G é uma regra sintática para gerar o nome, X de valores de x .
- \tilde{A} é uma regra semântica que associa cada X a seu significado $\tilde{A}(X)$ o qual é um subconjunto difuso de U . Um determinado X , que é um nome gerado por G , é designado por *termo*.

Exemplificando: Seja X uma variável linguística cujo rótulo seja "altura" (o nome desta variável é "altura" e seus valores também serão designados por "altura"). $U = [0, 2.00]$. Os termos desta variável linguística, que também são conjuntos difusos, poderiam ser, "alto", "muito alto", "um pouco alto", "mais ou menos alto", "baixo", "quase baixo", etc. A variável base u é a altura em metros e centímetros. $\tilde{A}(X)$ é a regra que associa um significado (um conjunto difuso) aos termos. No caso poder-se-ia ter por exemplo

$$\tilde{A}(\text{alto}) = \{ (u, \tilde{A}_{\text{alto}}(u)) \mid u \in [0, 2.00] \}$$

onde

$\tilde{A}_{\text{alto}}(u) =$	0,	$u \in [0, 1.70]$	(2.13)
	$(1/15)u - (1.7/15),$	$u \in [1.70, 1.85]$	
	1,	$u \in [1.85, 2.00]$	

De maneira análoga se definiriam as regras \tilde{A} (muito alto), \tilde{A} (um pouco alto), \tilde{A} (mais ou menos alto), \tilde{A} (baixo), \tilde{A} (quase baixo), etc, definindo-se adequadamente sua função de pertinência dentro do universo de discurso U.

$T(x)$ define o conjunto de termos da variável x , que no caso seria $T(\text{altura}) = \{\text{alto, muito alto, um pouco alto, baixo, quase baixo, etc}\}$. $G(x)$ é a regra que gera os termos do conjunto de termos.

Uma vez que se tenha definido o conceito de variáveis linguísticas, as sentenças declarativas que na Lógica Clássica são tratadas com a representação de conjuntos ordinários, passam a ser tratadas pela Lógica Difusa através da representação de conjuntos difusos, permitindo assim manipulação de infinitos graus de verdade embutidos na proposição das sentenças.

Na sentença (2.12), supondo que a altura de João seja 1.77 m, então calculando-se o grau de pertinência (grau de verdade) desta proposição dentro deste contexto conforme a função em (2.13), ter-se-ia que a sentença (2.12) apresenta um grau de pertinência (grau de compatibilidade ou grau de verdade) de 0.23. Se a sentença (2.12) fosse "João é quase baixo", então, tendo-se a regra \tilde{A} (quase baixo) poder-se-ia calcular seu grau de verdade.

2.6 - O Modelo AFT ([SHA92])

Shaout e Suk ([SHA92]) apresentam um tipo especial de *grafo de atributos difusos* (AFG - *Attributed Fuzzy Graph*), o assim denominado grafo de *atributos difusos concorrentes* (AFT - *Attributed Fuzzy Tournament*), para representar incertezas em análise de cenários. A análise/reconhecimento de padrões em geral dá-se através da técnica de *template matching*, mediante a definição de uma distância entre dois AFT's.

A seguir são apresentados conceitos relativos ao modelo AFT. Maiores detalhes são encontrados em ([SHA92]).

2.6.1 - Descrição do Modelo AFT

Descrição 1

Seja $V = \{ v_1, v_2, \dots, v_n \}$ um conjunto de n nodos de um grafo

Seja $A = \{ a_1, a_2, \dots, a_r \}$ um conjunto de r arcos orientados

Um grafo direcionado $T(n) = (V, A)$ é dito um *tournament* se:

→ $\forall v_i \in V, \forall v_k \in V, v_i \neq v_k$, estão concatenados por um e somente um arco orientado: $v_i \rightarrow v_k$ (lê-se " v_i domina v_k ") ou $v_i \leftarrow v_k$ (lê-se " v_k domina v_i "). v_i e v_k apresentam uma relação de dominância *completa, irreflexiva e antisimétrica*.

- S_i é dito o *score* de v_i e representa o número de nodos que v_i domina.

Definição 1

Um *attributed tournament* $AT(n)$ é uma quádrupla $AT(n) = (V, A, B, E)$ onde:

- n é o número de nodos em $AT(n)$,
- $V = \{ v_1, v_2, \dots, v_n \}$ é um conjunto de nodos,
- $A = \{ a_1, a_2, \dots, a_r \}$ é o conjunto de arcos direcionados,
- $B = \{ b_1, b_2, \dots, b_k \}$ é o conjunto de atributos dos nodos,
- $E = \{ e_1, e_2, \dots, e_s \}$ é o conjunto de atributos dos arcos.

Definição 2

Um *attributed fuzzy tournament* $AFT(n)$ é uma sêxtupla $AFT(n) = (V, A, B, E, X, C)$ onde:

- n é o número de nodos em $AFT(n)$,
- $V = \{ v_1, v_2, \dots, v_n \}$ é o conjunto de nodos,
- $A = \{ a_1, a_2, \dots, a_r \}$ é o conjunto de arcos direcionados,
- $B = \{ b_1, b_2, \dots, b_k \}$ é o conjunto de atributos dos nodos,
- $E = \{ e_1, e_2, \dots, e_s \}$ é o conjunto de atributos dos arcos,

- $X = \{ \mu_{b_1}(v), \mu_{b_2}(v), \dots, \mu_{b_k}(v) \}, \forall v \in V$, é o vetor de graus de pertinência dos possíveis valores (atributos) que os nodos de V podem ter,
- $C = \{ \mu_{e_1}(a), \mu_{e_2}(a), \dots, \mu_{e_s}(a) \}, \forall a \in A$, é o vetor de graus de pertinência dos possíveis valores (atributos) que os arcos de A podem ter.

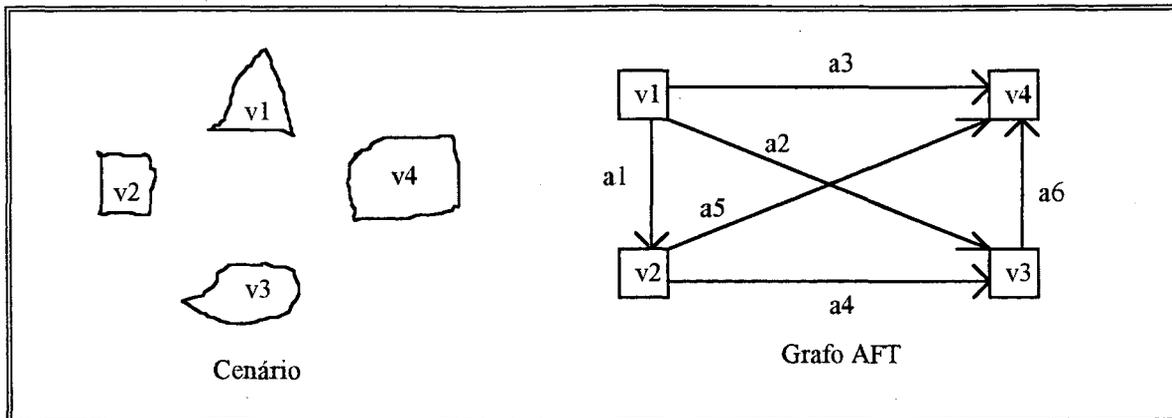


Fig. 2.19 - Um exemplo de cenário com 4 objetos. Os objetos v_1, v_2, v_3, v_4 equivalem aos nodos do grafo AFT.

Para o cenário da figura 2.19 o AFT poderia ser $AFT(n) = (V, A, B, E, X, C)$ onde:

- $n = 4$,
- $V = \{ v_1, v_2, v_3, v_4 \}$,
- $A = \{ a_1, a_2, a_3, a_4, a_5, a_6 \}$,
- $B = \{ \text{Triângulo, Quadrado, Esfera, Retângulo, Trapézio} \}$,
- $E = \{ \text{acima de, abaixo de, à esquerda de, à direita de, em frente de} \}$,
- $X = \{ \mu_{b_1}(v), \mu_{b_2}(v), \dots, \mu_{b_k}(v) \}, \forall v \in V$, é o vetor de graus de pertinência dos possíveis valores (atributos) que os nodos de V podem ter,

Nodo	Atributo do nodo				
	Triângulo	Quadrado	Esfera	Retângulo	Trapézio
v1	.7	.0	.1	.2	.0
v2	.0	.7	.1	.1	.1
v3	.1	.2	.5	.1	.1
v4	.0	.2	.0	.3	.5

- $C = \{ \mu_{e_1}(a), \mu_{e_2}(a), \dots, \mu_{e_s}(a) \}, \forall a \in A$, é o vetor de graus de pertinência dos possíveis valores (atributos) que os arcos de A podem ter.

Arco	Atributo do arco				
	Acima de	Abaixo de	À esquerda de	À direita de	em frente de
a1	.7	.0	.0	.3	.1
a2	.9	.0	.1	.0	.9
a3	.6	.0	.4	.0	.2
a4	.5	.0	.5	.0	.3
a5	.0	.05	.95	.0	1.
a6	.0	.8	.2	.0	.2

No exemplo da figura 2.19, supõe-se a seguinte caracterização semântica para o cenário: os atributos dos nodos formam um único "grupo", no sentido de que a soma de seus graus de pertinência são complementares, atingindo o valor máximo de 1. Os atributos dos arcos estão em dois "grupos". O primeiro grupo que se complementa é formado por { *acima de*, *abaixo de*, *à esquerda de*, *à direita de* } e o outro grupo é formado por { *em frente de* }.

A função de pertinência para { *acima de*, *abaixo de*, *à esquerda de*, *à direita de* } baseia-se no ângulo ϕ da reta que une dois nodos do cenário com o eixo das abcissas no plano cartesiano. Para o grupo { *em frente de* } define-se, arbitrariamente, que a pertinência é uma função proporcional ao ângulo ϕ (vale 0 se $\phi = 45^\circ, 135^\circ, 225^\circ$ ou 315° , e vale 1 se $\phi = 0^\circ, 90^\circ, 180^\circ$ ou 270°).

Definição 3

Um *attributed fuzzy tournament* AFT(n) é transitivo se

$$\forall v_i \in V, \forall v_k \in V, \forall v_m \in V, v_i \neq v_k \neq v_m,$$

$$\text{se } v_i \rightarrow v_k \text{ e } v_k \rightarrow v_m$$

$$\text{então } v_i \rightarrow v_m.$$

Descrição 2

Cálculo da medida de distância entre 2 AFT's

Sejam dois AFT's

$$T_1(n) = (V_1, A_1, B, E, X_1, C_1) \text{ e}$$

$$T_2(m) = (V_2, A_2, B, E, X_2, C_2)$$

onde $n \geq m$, então:

Distância entre 2 nodos $v \in V_1$ e $u \in V_2$, $D_n(v,u)$:

$$D_n(v,u) = d_n [X_1(v), X_2(u)]$$

onde d_n pode ser, por exemplo, o operador *max* ou a distância de Hamming.

Distância entre 2 arcos $p \in A_1$ e $q \in A_2$, $D_a(p,q)$:

$$D_a(p,q) = d_a [C_1(p), C_2(q)]$$

onde d_a pode ser, por exemplo, o operador *max* ou a distância de Hamming.

Unidade é um arco com dois nodos em suas extremidades e é independente de direção.

$$\text{Assim } U(v,p,w) = U(w,p,v)$$

onde $v \in V_1$ (ou $v \in V_2$), $w \in V_1$ (ou $w \in V_2$), $p \in A_1$ (ou $p \in A_2$).

Complexo é uma unidade cuja direção é considerada.

Assim $C(v,p,w) \neq C(w,p,v)$ porém $C(v,p,w) = C(w,p',v)$.

Neste caso $v \in V_1$ (ou $v \in V_2$), $w \in V_1$ (ou $w \in V_2$), $p \in A_1$ (ou $p \in A_2$), p' é o inverso de p .

$$C: \quad v \xrightarrow[p]{\quad} w \quad \quad v \xleftarrow[p']{\quad} w$$

Distância entre dois complexos $D_c [C(v,p,w), C(u,q,z)] :$

$$D_c [C(v,p,w), C(u,q,z)] = f_c[d_n(v,u), d_a(p,q), d_n(w,z)],$$

onde $v, w \in V_1$, $u, z \in V_2$, $p \in A_1$, $q \in A_2$, f_c pode ser, por exemplo, o operador *max*.

A medida de distância deve levar em conta a *unidade*, e não o *complexo*, a fim de se poder calcular independentemente da direção do arco.

Distância entre duas unidades $D_u [U(v,p,w), U(u,q,z)] :$

$$U(v,p,w) \in T_1 \quad \text{e} \quad U(u,q,z) \in T_2:$$

$$D_u [U(v,p,w), U(u,q,z)] = \begin{aligned} & f_u[D_c [C(v,p,w), C(u,q,z)], \\ & \quad D_c [C(v,p,w), C(z,q',u)], \\ & \quad D_c [C(w,p',v), C(u,q,z)], \\ & \quad D_c [C(w,p',v), C(z,q',u)]] \end{aligned}$$

onde $v, w \in V_1$, $u, z \in V_2$, $p \in A_1$, $q \in A_2$, f_u pode ser, por exemplo, o operador *min*.

Medida de Distância D entre dois AFT's:

$$D(T_1, T_2) = f_1 \text{ p } \in A_1 \text{ f}_2 \text{ q } \in A_2 \text{ D}_u [U(v,p,w), U(u,q,z)]$$

Supondo $f_1 = \max$ e $f_2 = \min$, a interpretação da medida da distância seria:

- 1) Para cada unidade de T_1 , o mínimo de T_1 sobre as unidades de T_2 , encontra a distância mais próxima da unidade T_1 em relação à qualquer unidade de T_2 .
- 2) O máximo sobre os mínimos das unidades de T_1 encontrará o "pior" erro entre os pares de unidades mais próximos.

Descrição 3

Propriedades da Medida de Distância D

- a) Dois AFT's T_1 e T_2 são considerados "concordantes" (*exact match*) se

$$D(T_1, T_2) = 0$$

- b) Dois AFT's T_1 e T_2 são considerados maximalmente diferentes se

$$D(T_1, T_2) = 1$$

- c) Sejam $T_1(n)$ e $T_2(m)$ dois AFT's com n e m nodos respectivamente, onde $n \geq m$.
Então

$$D[T_1(n), T_2(m)] = 0 \text{ se } T_2(m) \subset T_1(n), \text{ e } n \geq m$$

- d) D é uma *métrica* se:

$$f_1 = \max$$

$$f_2 = \min$$

$$f_c = \max$$

d_n = distância de Hamming

d_a = distância de Hamming

Isto significa que para 3 AFT's quaisquer T_1, T_2, T_3 vale

$D(T_1, T_2) = 0$ se houver *matching* exato

$D(T_1, T_2) = D(T_2, T_1)$

$D(T_1, T_2) \leq D(T_1, T_3) + D(T_3, T_2)$

2.6.2 - Sensibilidade à Rotação do Modelo AFT

Uma característica do modelo AFT é o fato de ser um modelo de natureza sensível à rotação dos padrões considerados. Sejam os padrões apresentados na figura 2.20:

Em situações como esta, para reconhecimento de caracteres escritos à mão, é adequado que haja sensibilidade à rotação. No caso da figura 2.20 os padrões devem apresentar distância $D = 1$, pois devem estar representando as letras "E" e "M", que são padrões diferentes. Quer dizer, o modelo caracteriza o padrão em relação a um determinado referencial "externo" observado no eixo cartesiano no plano R^2 .

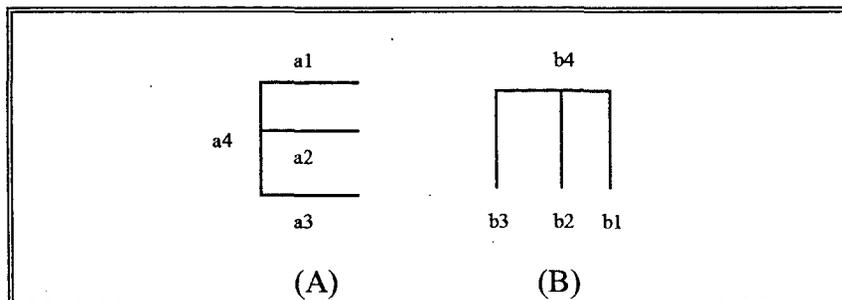


Fig. 2.20 - Dois padrões que podem ou não ser diferentes dependendo do contexto. Podem representar uma letra "E" e um "M" ou uma letra "E" em posição normal e uma letra "E" "deitada".

2.6.3 - Exemplos dos Cálculos envolvidos na Distância D Para Grafos AFT's

Seguem abaixo exemplos dos cálculos necessários para a distância D entre dois grafos AFT. No Anexo II encontra-se também outro exemplo, considerando-se as características usadas no modelo proposto neste trabalho, para estilete de nematóides.

Sejam os padrões P1 e P2 dois cenários, apresentados abaixo, onde se deseja calcular um grau de similaridade entre ambos através do modelo AFT..

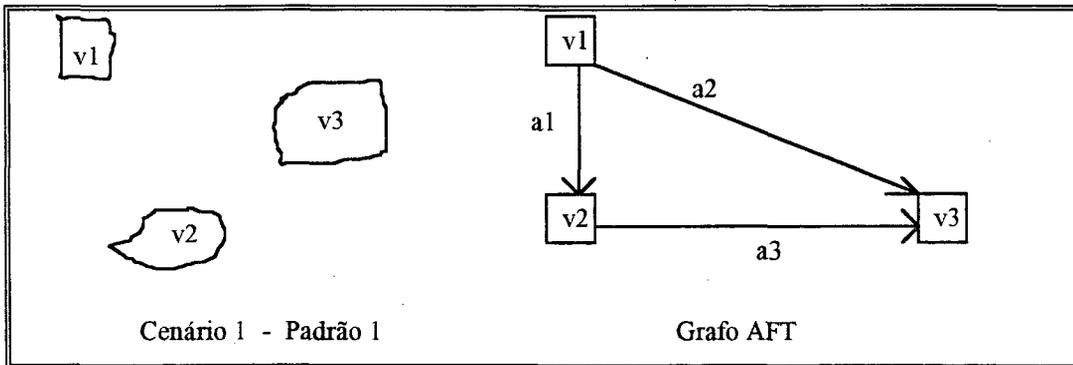


Fig. 2.21 - Cenário 1 (padrão P1)

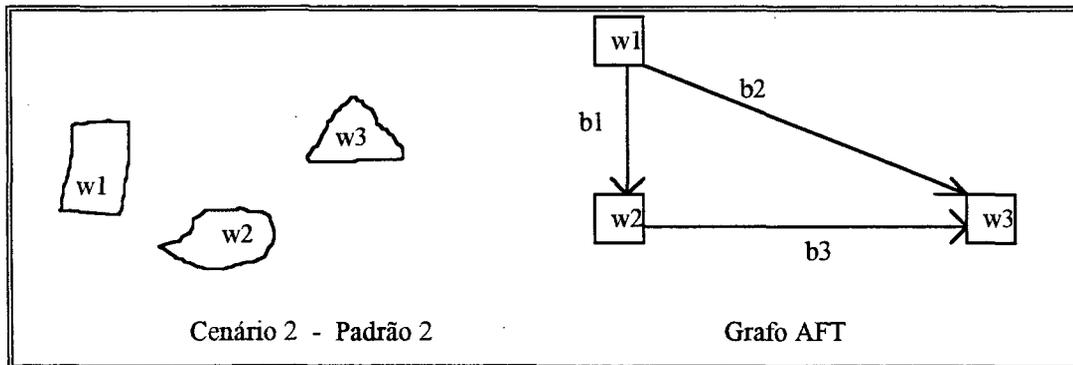


Fig. 2.22 - Cenário 2 (padrão P2)

2.6.3.1 - Definição dos Grafos AFT's dos Padrões P1 e P2

Seja $AFT(n) = (V, A, B, E, X, C)$ onde

$B = \{ \text{Triângulo, Quadrado, Esfera, Retângulo} \}$,

$E = \{ \text{acima de, abaixo de, à esquerda de, à direita de, em frente de} \} =$

$\{ \text{acima de, abaixo de, à esquerda de, à direita de} \} \cup \{ \text{em frente de} \}$

Grafo AFT para o Padrão P1

$$n = 3,$$

$$V = \{ v1, v2, v3 \},$$

$$A = \{ a1, a2, a3 \},$$

$$X =$$

Nodo	Atributo do nodo			
	Triângulo	Quadrado	Esfera	Retângulo
v1	.1	.7	.0	.2
v2	.0	.2	.4	.4
v3	.0	.2	.1	.7

$$C =$$

Arco	Atributo do arco				
	Acima de	Abaixo de	À esquerda de	À direita de	em frente de
a1	.9	.0	.1	.0	.7
a2	.3	.0	.7	.0	.6
a3	.0	.8	.2	.0	.1

Grafo AFT para o Padrão P2

$$n = 3,$$

$$V = \{ w1, w2, w3 \},$$

$$A = \{ b1, b2, b3 \},$$

X =

Nodo	Atributo do nodo			
	Triângulo	Quadrado	Esfera	Retângulo
w1	.0	.3	.0	.7
w2	.0	.1	.7	.2
w3	.8	.1	.0	.1

C =

Arco	Atributo do arco				
	Acima de	Abaixo de	À esquerda de	À direita de	em frente de
b1	.3	.0	.7	.0	.6
b2	.0	.4	.6	.0	.8
b3	.0	.8	.2	.0	.2

A caracterização semântica para os cenários apresentados nas figuras 2.21 e 2.22 é a mesma que a caracterização feita para o cenário da figura 2.19.

2.6.3.2 - Cálculo da Distância D Entre os Grafos AFT dos Padrões P1 e P2

São quatro os cálculos-chave para se conseguir a distância D.

- O cálculo da distância D_n entre 2 nodos e D_a entre 2 arcos
- O cálculo da distância D_c entre 2 complexos
- O cálculo da distância D_u entre 2 unidades e
- O cálculo da distância D entre 2 AFT's .

a) Cálculo da distância D_n e D_a

Shaout ([SHA92]) propõe a distância de Hamming.

Exemplo: Distância entre o nodo v1 do grafo AFT T1 e o nodo w1 do grafo AFT de T2.

$$\begin{aligned} D_n(v1,w1) &= |.1 - .0| + |.7 - .3| + |.0 - .0| + |.2 - .7| = \\ &= .1 + .4 + .0 + .5 = \\ &= 1 \end{aligned}$$

Normalizando $D_n(v1,w1)$ tem-se

$$D_n(v1,w1) = 1 / 2 = .5$$

Exemplo: Distância entre o nodo v2 do grafo AFT T1 e o nodo w2 do grafo AFT de T2.

$$D_n(v2,w2) = |.0 - .0| + |.2 - .1| + |.4 - .7| + |.4 - .2| = .6 / 2 = .3$$

Exemplo: Distância entre o arco a1 do grafo AFT T1 e o nodo b1 do grafo AFT de T2.

$$\begin{aligned} D_a(a1,b1) &= |.9 - .3| + |.0 - .0| + |.1 - .7| + |.0 - .0| + |.7 - .6| = \\ &= .6 + .0 + .6 + .0 + .1 = \\ &= 1.3 \end{aligned}$$

Normalizando $D_a(a1,b1)$ tem-se

$$D_a(a1,b1) = 1.3 / 3 = .43$$

b) Cálculo da distância D_c entre 2 complexos

Shaout ([SHA92]) propõe o operador *max*.

Exemplo: Distância entre $C1(v1,a1,v2)$ e $C1(w1,b1,w2)$

$$\begin{aligned}
 D_c[C1(v1,a1,v2) , C1(w1,b1,w2)] &= \max [D_n(v1,w1), D_a(a1,b1), D_n(v2,w2)] = \\
 &= \max [.5, .43, .3] = \\
 &= .5
 \end{aligned}$$

Exemplo: Distância entre C2(v2,a1',v1) e C2(w2,b1',w1)

a1' é o inverso de a1, e b1' é o inverso de b1

Arco	Atributo do arco				
	Acima de	Abaixo de	À esquerda de	À direita de	em frente de
a1'	.0	.9	.0	.1	.7
b1'	.0	.3	.0	.7	.6

$$\begin{aligned}
 D_c[C2(v2,a1',v1) , C2(w2,b1',w1)] &= \max [D_n(v2,w2), D_a(a1',b1'), D_n(v1,w1)] = \\
 &= \max [.3, .43, .5] = \\
 &= .5
 \end{aligned}$$

Exemplo: Distância entre C1(v1,a1,v2) e C2(w2,b1',w1)

$$\begin{aligned}
 D_c[C1(v1,a1,v2) , C2(w2,b1',w1)] &= \max [D_n(v1,w2), D_a(a1,b1'), D_n(v2,w1)] = \\
 &= \max [.7, .7, .4] = \\
 &= .7
 \end{aligned}$$

Exemplo: Distância entre C2(v2,a1',v1) e C1(w1,b1,w2)

$$\begin{aligned}
 D_c[C2(v2,a1',v1) , C1(w1,b1,w2)] &= \max [d_n(v2,w1), d_a(a1',b1), d_n(v1,w2)] = \\
 &= \max [.4, .7, .7] = \\
 &= .7
 \end{aligned}$$

c) Cálculo da distância D_u entre 2 unidades

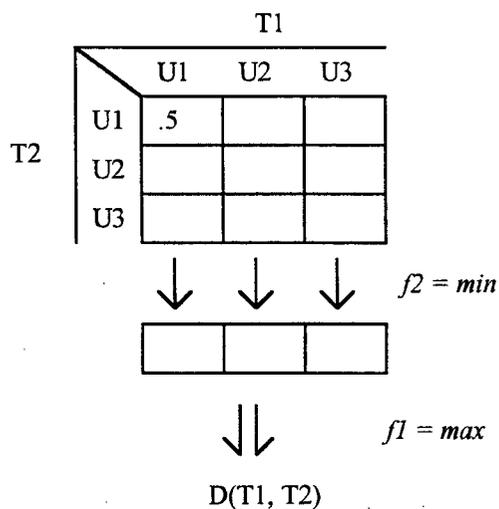
Shaout ([SHA92]) propõe o operador *min*.

Exemplo: Distância entre $U(v1,a1,v2)$ e $U(w1,b1,w2)$

$$\begin{aligned}
 D_u [U(v1,a1,v2), U(w1,b1,w2)] &= \min [D_c [C1(v1,a1,v2), C1(w1,b1,w2)], \\
 & D_c [C1(v1,a1,v2), C2(w2,b1',w1)], \\
 & D_c [C2(v2,a1',v1), C1(w1,b1,w2)], \\
 & D_c [C2(v2,a1',v1), C2(w2,b1',w1)]] = \\
 &= \min [.5, .7, .7, .5] = \\
 &= .5
 \end{aligned}$$

d) Cálculo da distância D entre 2 grafos AFT

Shaout ([SHA92]) propõe efetuar as operações $f2 = \min$ e $f1 = \max$.



CAPÍTULO 3

Caracterização e Identificação de Nematóides

3.1 - Introdução

Este capítulo dedica-se à apresentação geral dos nematóides, animais vermiformes que apresentam importância econômica. Inicia-se o capítulo comentando-se sobre taxionomia. Em seguida são apresentadas caracterizações gerais dos nematóides enfocando suas estruturas básicas, entre as quais está o estilete. Comenta-se ainda sobre a importância econômica desses animais e os aspectos de interesse do presente trabalho ligados à sua identificação.

3.2 - Taxionomia

Taxionomia ou *sistemática*, é o ramo das ciências biológicas relativo ao reconhecimento, descrição, nomenclatura e classificação dos organismos vivos. (...) O homem classifica as coisas porque isso as torna ordenadas e mais fáceis de serem compreendidas. Elimina o aprendizado por repetição. Entretanto, o biólogo se defronta com uma tarefa de fôlego quando se trata de sistematizar o mundo dos organismos vivos. Em grupo algum de objetos existe tanta variedade de indivíduo para indivíduo e de grupo para grupo ([BAK75]).

Propostas para classificação são bastante antigas. Teofrasto (370-285 A.C.) tentou classificar as plantas até então conhecidas com base na forma, período vital e habitat. Muitos outros pesquisadores, trabalhando tanto com vegetais como animais, seguiram este exemplo. Entretanto, a falta de comunicação entre estes naturalistas tornou impossível a adoção de um sistema único de classificação ([BAK75]).

Uma dificuldade que se impõe para uma universalização de um sistema único de classificação refere-se aos *critérios* que devem ser tomados para efetuar tal classificação. Um pesquisador por exemplo, poderia classificar os animais segundo os seus meios de locomoção, enquanto outro pesquisador poderia escolher o habitat (terra, mar ou ar) como critério. Nota-se daí a multiplicidade de sistemas classificadores que podem ser gerados.

Deve-se, porém, antes de se escolher os critérios para classificação, definir o *objetivo* do sistema de classificação. Por exemplo, se o objetivo é classificar plantas para uso medicinal, poder-se-ia talvez

indicar um grupo de vegetais cujo extrato reduzisse a febre, em outro grupo vegetais cujo extrato aliviasse a dor, e assim por diante.

Hoje em dia sabe-se, com base na Teoria Evolucionista, que as muitas variedades de organismos da Terra evoluíram a partir de ancestrais comuns através de sucessivas mutações, e os biólogos concordaram então que o esquema geral de classificação mais útil deveria refletir correlações evolutivas. Em geral estruturas anatômicas parecem refletir consistentemente tais aspectos evolutivos. Isto porém nem sempre é verdade e os biólogos concordam que não é válido um critério único para classificação [BAK75].

Observando-se na figura 3.1 a estrutura anatômica dos animais A e B, tende-se a concluir que o animal A pertence ao grupo dos cães e o animal B pertence ao grupo dos roedores (que inclui esquilos e ratos). Entretanto a fisiologia reprodutiva desses animais revela que não são mamíferos típicos (como o cão e o esquilo), mas sim marsupiais. Apresentam ambos aspectos comuns a marsupiais como o fato de produzirem a cria de forma muito imatura e a carregarem em bolsas próprias durante o tempo necessário para que completem seu desenvolvimento [BAK75].

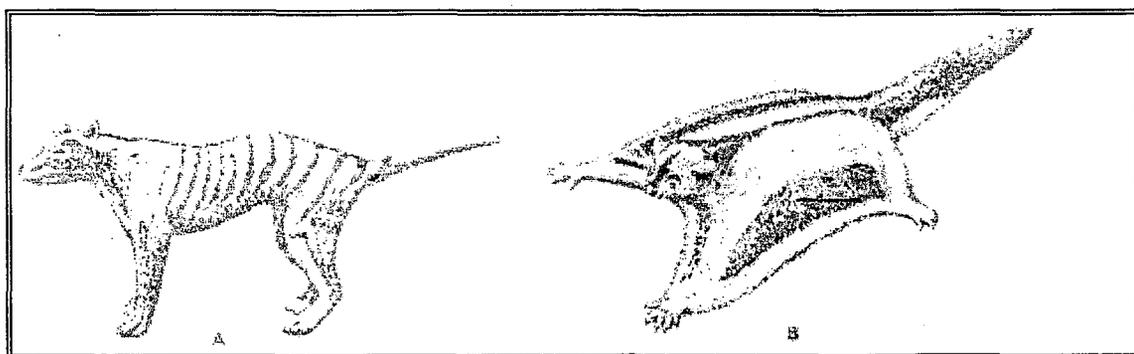


Fig. 3.1 - Exemplos de marsupiais: A e B são marsupiais apesar de parecerem ser tipos de cão e esquilo. - Fonte: [BAK75]

Os critérios que servem de base para classificação são denominados *chave de classificação*. A necessidade de critérios comuns para qualquer chave de classificação é de óbvia importância. Além disso, precisa existir algum plano de nomenclatura e agrupamento, para que acomode não somente todos os animais e vegetais hoje conhecidos, mas também os que forem descobertos futuramente. Em 1735, Carlous Linneus publica a primeira das dez edições de seu trabalho *Systema Naturae*, onde é apresentado um esquema que ainda hoje é utilizado. Consiste em sete agrupamentos assim denominados:

REINO
 FILO (ou RAMO, PHYLUM)
 CLASSE
 ORDEM
 FAMÍLIA
 GÊNERO
 ESPÉCIE

Muitos especialistas acharam que esses sete agrupamentos não eram suficientes. Criaram-se então agrupamentos adicionais na forma de prefixos como *super-* *sub-* e *infra-* e ocasionalmente um grupo lateral denominado *corte* ([BAK75]).

QUADRO 1

Reino	Animalia (Metazoa)
Filo	Chordata
Subfilo	Vertebrata
Superclasse	Tetrapoda
Classe	Mammalia
Subclasse	Theria
Infraclasse	Eutheria
Ordem	Primata
Subordem	Anthropoidea
Superfamília	Hominoidea
Família	Hominidae
Subfamília	Hominidae
Gênero	<i>Homo</i>
Espécie	<i>sapiens</i>
Subespécie	<i>sapiens</i>

Quadro 1 - Classificação taxionômica para o homem (adaptado de [BAK75])

Os grupos constantes dessa chave formam uma *hierarquia*. Cada grupo da hierarquia inclui uma variedade maior de características do que o grupo que lhe é imediatamente inferior. Assim supondo que:

- Um determinado reino X apresente características $\alpha_1, \alpha_2, \alpha_3$
- Um determinado filo Y do reino X apresente características $\beta_1, \beta_2, \beta_3$
- Outro filo Z do reino X apresente características γ_1, γ_2

Admitindo-se que um determinado indivíduo pertence ao filo Y, então automaticamente este indivíduo pertence ao reino X (pois $Y \in X$) e apresenta as características $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3$. Se pertencer ao filo Z então suas características são $\alpha_1, \alpha_2, \alpha_3, \gamma_1, \gamma_2$.

Em taxionomia, a espécie é a unidade básica da classificação. Gêneros, ordens, famílias, classes ou filios existem unicamente como grupos de classificação estabelecidos pelo biólogo. As espécies têm contudo importância real na natureza. Os indivíduos de uma mesma espécie geralmente são reprodutivamente isolados dos indivíduos das demais espécies. Categorias taxionômicas mais elevadas tais como o gênero ou a família têm apenas o que os biólogos consideram realidade histórica ([BAK75]).

Linneus propôs também um sistema de nomenclatura, conhecido como *sistema binomial de nomenclatura* justamente por designar com dois nomes o indivíduo. As normas usadas nesta nomenclatura são:

- Indivíduos são identificados pelo seu nome de gênero e espécie
- Escreve-se o gênero com letra inicial maiúscula e a espécie com letra inicial minúscula
- Os nomes taxionômicos devem ser latinizados e escritos em itálico
- Ocasionalmente quando existir a descrição de subespécie, acrescentá-la ao nome (nomenclatura trinomial)

Exemplos de nomes:

Homo sapiens sapiens

Fasciola hepática

Rattus rattus rattus

Gryllus rubens

3.3 - Nematóides

3.3.1 - Caracterização dos Nematóides

Os nematóides são animais pouco conhecidos do homem, embora ocupem o segundo lugar em espécies, só perdendo para os artrópodes (insetos, piolhos-de-cobra, centopéias, crustáceos, aranhas, ácaros, escorpiões, etc). Sabe-se hoje que, onde possa existir vida, existem nematóides. Nas regiões polares e nos trópicos, nas fontes de águas aquecidas e nas neves vermelhas, nos pântanos e nos desertos, nos picos das mais altas montanhas e nas mais profundas fossas oceânicas, há nematóides ([MON92]).

São animais vermiformes de corpo longo, fino e afilado nas duas extremidades que se encontram no solo e em plantas. Apresentam formato cilíndrico, de modo geral, diferindo assim dos animais do Filo Platyhelminthes. Pertencem ao Filo Nematelminthes (nematos = fio helmintes = verme), embora muitos autores o considerem uma classe do Filo Aschelminthes, a classe nematoda ([AMA79]).

Estruturalmente falando, além do formato tipicamente cilíndrico, apresentam tubo digestivo completo, com boca e ânus, e simetria bilateral. Podem ter vida livre ou ser parasitas de plantas e animais. São em geral animais diócos, isto é, não são hermafroditas. Possuem musculatura apenas longitudinal executando apenas movimentos de flexão do corpo. Os representantes de vida livre têm, geralmente, pequeno tamanho, da ordem de poucos milímetros de comprimento, enquanto que, entre os representantes parasitas encontram-se vermes que podem atingir até alguns metros de comprimento ([AMA79]).

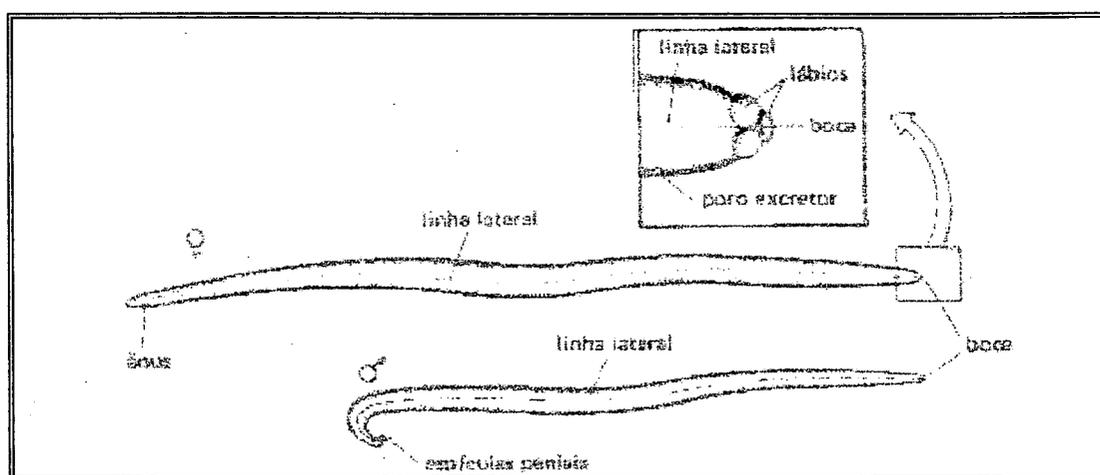


Fig. 3.2 - Aspecto externo de um casal de *Ascaris*. A região anterior foi representada em maior aumento para mostrar os três lábios que circundam o orifício bucal - Fonte: [AMA79]

Os nematóides nutrem-se de outros organismos vivos, tais como bactérias, fungos, e algas, ou mesmo animais microscópicos (protozoários, metazoários, etc). Os nematóides parasitas atacam virtualmente todos os grupos de vegetais e animais. não há planta cultivada ou animal doméstico potencialmente livre de alguma forma nociva de nematóides. O próprio homem é vítima de muitos tipos de nematóides parasitos ([MON92]).

Alguns típicos nematóides parasitas do homem que causam doenças são por exemplo o *Ancylostoma duodenale* e *Necator americanus* que produzem uma doença conhecida como "amarelão". *Wuchereria bancrofti* produz elefantíase, doença que provoca deformações nos membros inferiores do homem. Nematóides do gênero *Meloidogyne* parasitam vegetais, ao ponto de fazer aparecer nestes, nódulos denominados "galhas".

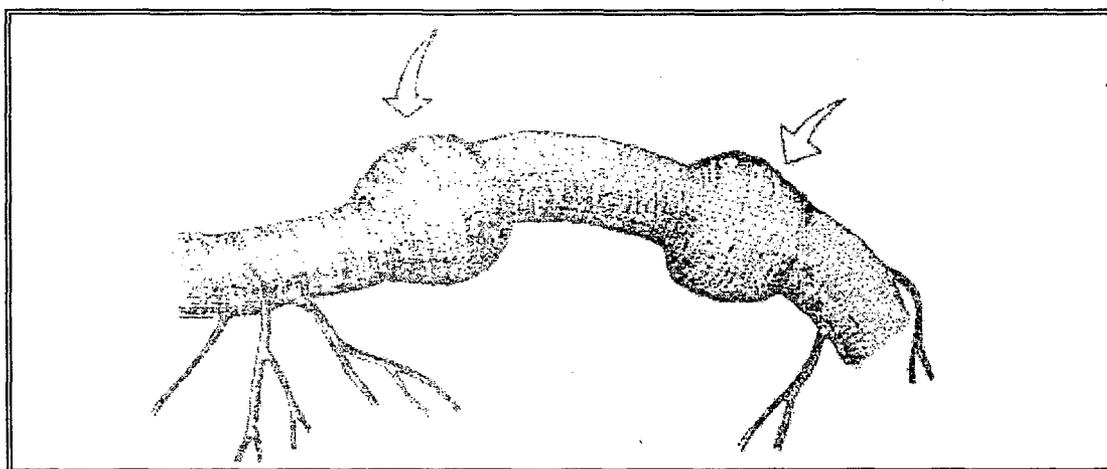


Fig. 3.3 - Os nematodos que vivem em raízes podem provocar nestas o aparecimento de nódulos denominados "galhas". Na figura, o aspecto de uma raiz com duas "galhas" apontadas pelas setas. Fonte: [AMA79]

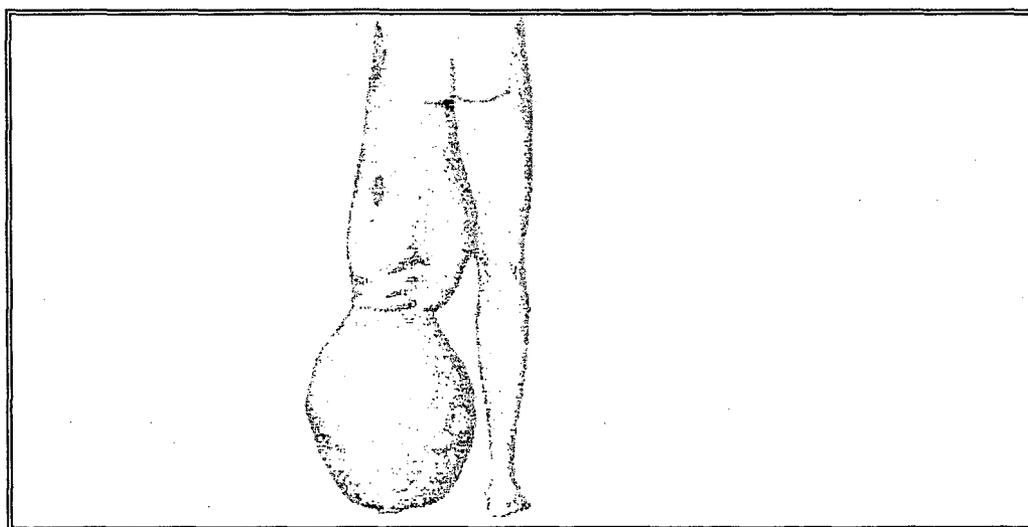


Fig. 3.4 - Vítima de elefantíase causada pela filária, *Wuchereria bancrofti*. Fonte: [AMA79]

3.3.2 - As Estruturas do Nematóide Como Base de Classificação

Mai ([MAI75]) apresenta uma chave de classificação para nematóides "baseada primariamente em caracteres considerados mais facilmente observáveis por estudantes". Segundo Mai, a chave é baseada em características morfológicas e morfométricas de fêmeas adultas, pois são mais adequadas para a identificação taxionômica de modo geral.

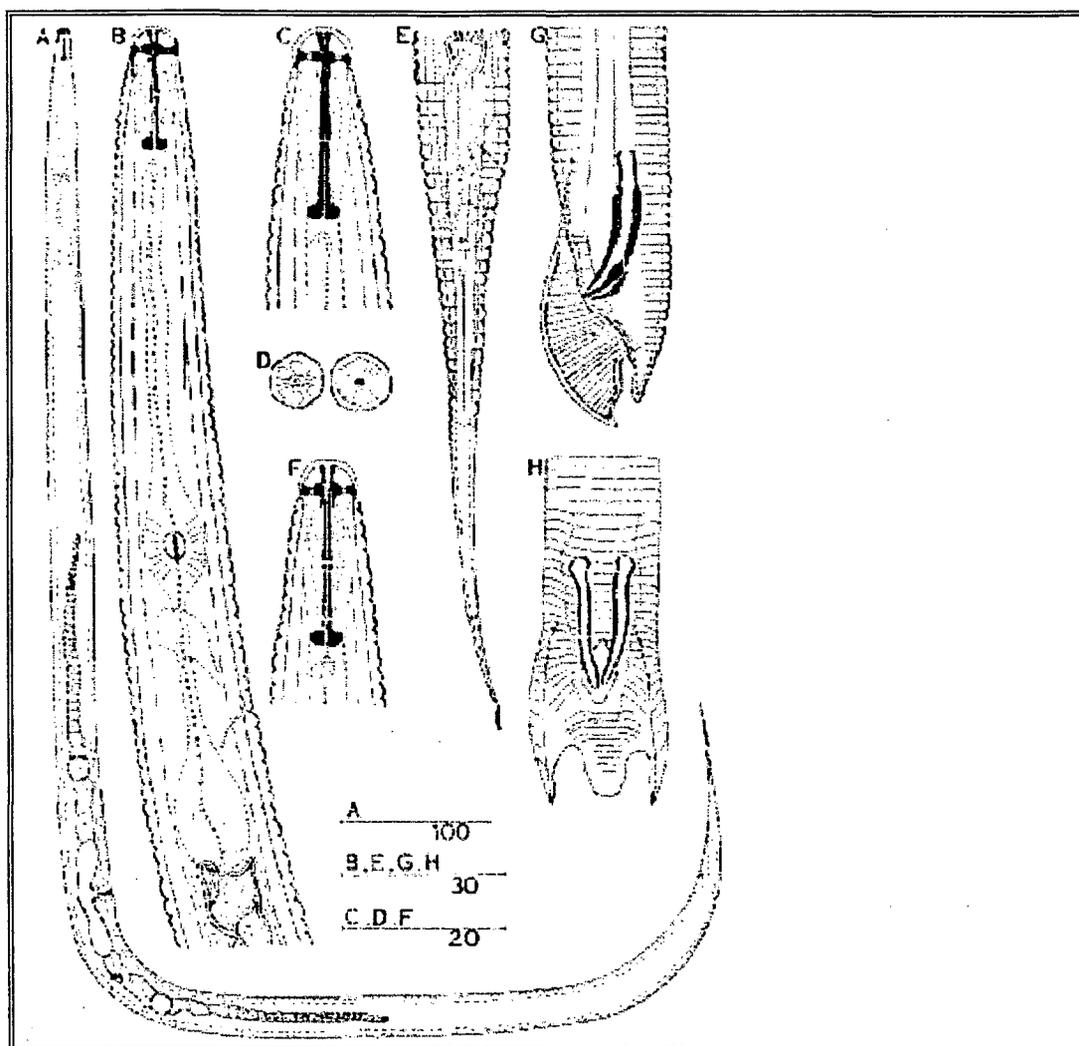


Fig. 3.5 - O nematóide *Brachydorus tenuis*. (Fêmea) A - Corpo do animal B - Porção anterior C - Cabeça D - Secção transversal da região labial no ápice e na base E - vista lateral da porção posterior (Macho) F - Cabeça G - vista lateral da porção posterior H - vista ventral da porção posterior - Fonte: [MAI75]

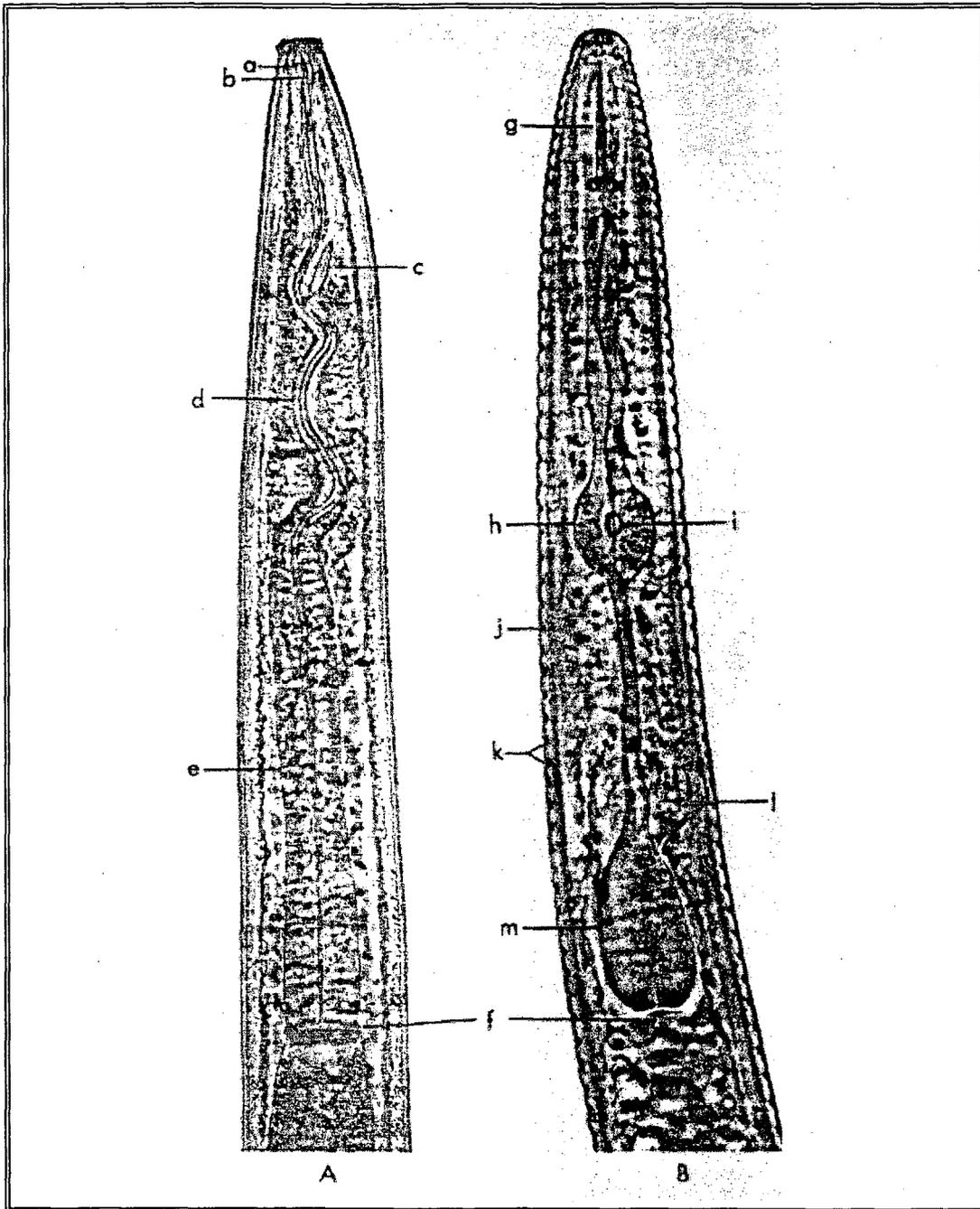


Fig. 3.6 - A - Porção anterior de *Dorylaimus* sp. B - porção anterior de *Tylenchorhynchus* sp. (a) anel direcionador (b) estilete (c) estilete em formação (d) canal esofágico (e) região basal do esôfago (f) válvula intestinal-esofágica (g) estilete com bulbo basal (h) metacorpo (i) válvula do metacorpo (j) "estrias" (k) anulis (l) poro excretor (m) bulbo basal do esôfago - Fonte: [MAI75]

QUADRO 2 - Chave de Classificação

1	- Stylet absent - Stylet present	2
2	- Two-part esophagus, no valvulated apparatus - Three-part esophagus usually with valvulated metacarpus	DORYLAIMIDA 6
3	- Stylet short, curved; body short and thick (0.45 to 1.5 mm long) - Stylet long, straight tapering to a long slender point - Stylet straight, usually not very long	TRICHODORUS 4 A LARGE NUMBER OF GENERA
4	- Stylet extensions with sclerotized basal flanges - Stylet extensions without basal flanges	XIPHINEMA 5
5	- Amphid openings minute - Amphid openings wide	LONGIDORUS PARALONGIDORUS
6	- Dorsal esophaegal gland outlet in metacarpus - Dorsal esophageal gland outlet in procorpus	APHELENCOIDEA 7 TYLENCHOIDEA 9
7	- Vulval flap absent, vagina normal - Vulva with wide overlapping flap, vagina curved	8 RHADINAPHELENCHUS
8	- Tail of female blunt; lateral field with 6 to 14 incisures - Tail of female usually conoid, lateral field with 2 to 14 incisures	APHELENCUS APHELENCHOIDES
9	- Head with setae. No plant parasites - Head without setae Numerous plant parasites	ATYLENCHUS EUTHYLENCHUS 10
10	- Metacarpus absent or reduced - Example of family - Example of family	NEOTYLENCHIDAE NOTHANGUINA NOTHOTYLENCHUS
...		

Quadro 2 - Uma parte da chave de classificação proposta por Mai ([MAI75])

Observando-se o quadro 2, percebe-se que aspectos morfológicos e morfométricos de diversas estruturas (estilete, metacorpo, esôfago, etc) permitirão caracterizar taxionômicamente os nematóides.

3.3.3 - Nematóides Fitoparasitos (Fitonematóides)

3.3.3.1 - Caracterização de Fitonematóides

Os nematóides parasitas de plantas, sua única fonte de alimento, são denominados *fitonematóides*. Organismos patogênicos, vivem no solo atacando as raízes de diversas plantas. Disto resulta a dificuldade de seu reconhecimento pelo produtor e ainda os obstáculos encontrados para o correto diagnóstico dos danos causados por eles os quais são, geralmente, atribuídos a outros fatores envolvidos na produção ([VIL92]).

Os danos causados pelos fitonematóides são reflexos do seu parasitismo na planta. Em geral, os principais sintomas apresentados pelas plantas parasitadas por nematóides caracterizam-se por intensa injúria nas raízes, o que interfere na capacidade de sustentação da planta e na absorção de água e nutrientes. Disso resulta ([LIM92]):

- a) Nanismo e crescimento retardado, além de menor resistência ao frio e à seca.
- b) Necrose, descoloração e enrolamento do caule da planta, causado por nematóides que se alimentam de tecidos do caule.
- c) Manchas foliares nos tecidos do parênquima foliar da planta, por nematóides que se alimentam desse tecido.
- d) Morte dos ponteiros, por nematóides que se alimentam do broto terminal (ponto de crescimento) da planta.
- e) Enrugamento e distorção da folhagem e caule.
- f) Galhas nas folhas e sementes, quando nematóides se alojam no botão floral da planta.
- g) Supressão do crescimento da raiz, quando nematóides se alimentam das pontas de raízes, podendo inclusive ocasionar redução no crescimento de todo o sistema radicular.
- h) Formação de galhas nas raízes.
- i) Bifurcação de raízes, onde a bifurcação da raiz principal dá um aspecto distorcido ao sistema radicular e diminui o valor comercial do produto, como por exemplo, no caso da cenoura.

3.3.3.2 - A Estrutura do Estilete em Fitonematóides

Devido ao processo evolucionário, os fitonematóides, apresentam a união das peças bucais formando uma estrutura rígida, dotada de orifício central e similar a uma agulha. Tal estrutura, denominada de *estilete*, é fundamental para o exercício do parasitismo. Através de músculos ligados a sua base, o estilete se desloca para fora e, em contato com o hospedeiro, é introduzido na região cortical

da raiz . Algumas espécies possuidoras de grandes estiletos podem chegar ao cilindro central da raiz, onde estão os vasos condutores de seiva. Por sucção, a seiva ou os conteúdos celulares fluem através do canalículo central até o esôfago e daí para o intestino ([CAM92]).

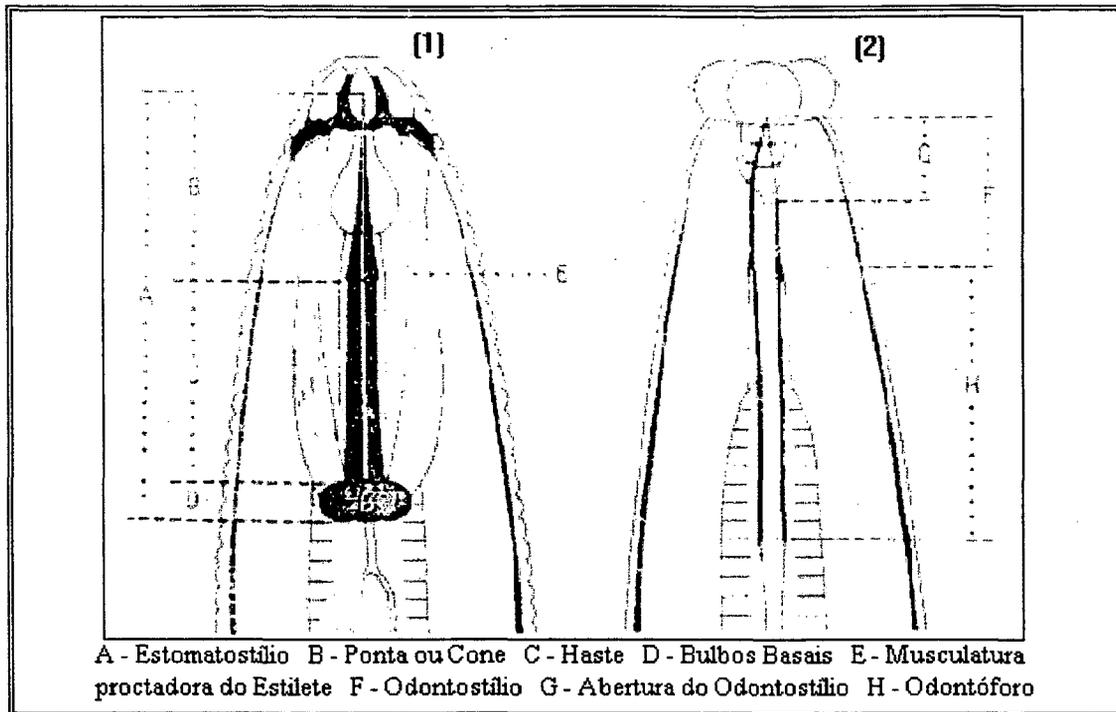
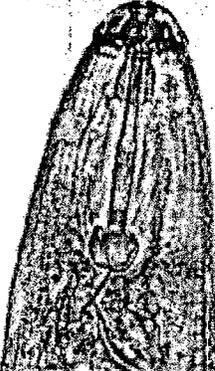
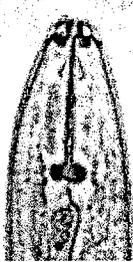
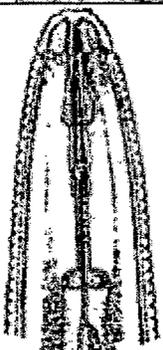
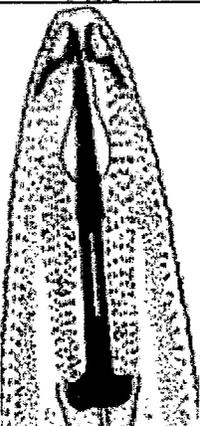
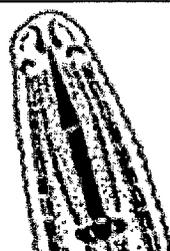
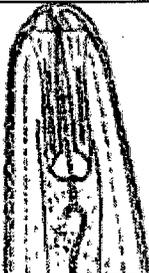
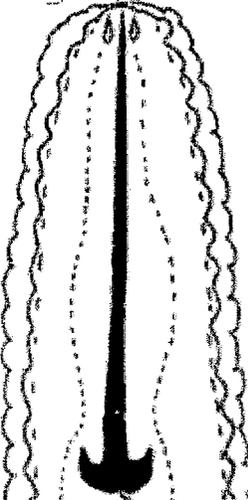
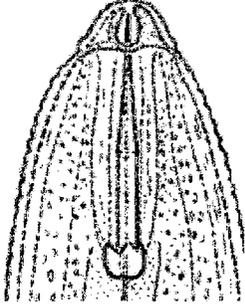
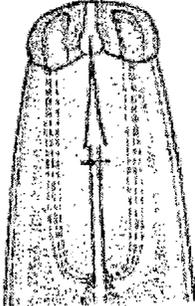
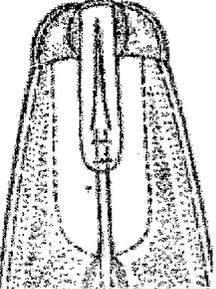
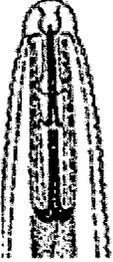
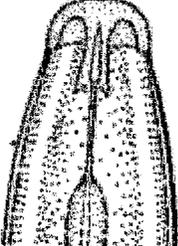
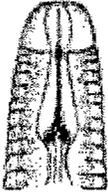


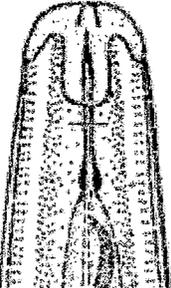
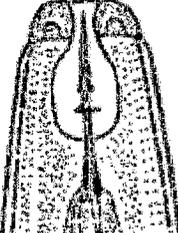
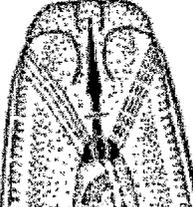
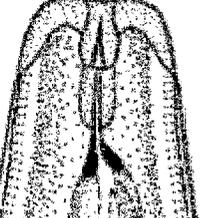
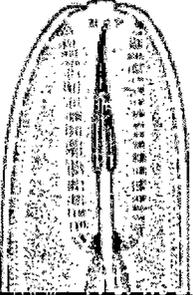
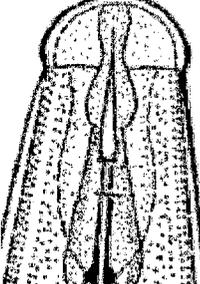
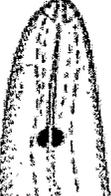
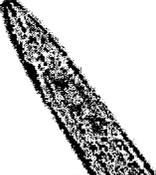
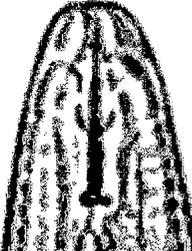
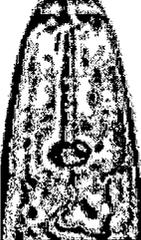
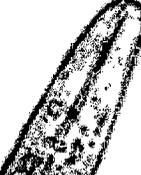
Fig. 3.7 - Exemplos de estilete. Fonte: [MON92]

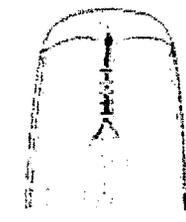
A alimentação de fitonematóides nas plantas não constitui um processo puramente físico. De fato, na parte basal do esôfago existem glândulas que produzem substâncias diversas, incluindo hormônios e enzimas que auxiliam o processo de parasitismo e digestão. O produto secretado caminha através do canalículo localizado na parede do esôfago e deságua em duas regiões distintas desse órgão. Aquele canalículo que libera seus conteúdos próximos do estilete parece conduzir substâncias que auxiliam a penetração do estilete no tecido vegetal ou levam substâncias, as quais, injetadas na região próxima aos vasos condutores de seiva, induzem os tecidos vegetais daquela região à produção de células enormes que constituirão o depósito de alimento para o nematóide durante toda sua vida. Na parte externa da raiz, observa-se a formação de um engrossamento, denominado de "galhas" ([CAM92]).

A seguir é apresentada uma variedade de imagens da parte anterior de nematóides, onde se nota a presença do estilete, e a classificação taxionômica conforme indicação bibliográfica. Percebe-se, ao se observar as imagens, a diversidade de formatos de estilete.

Imagem	Caracterização	Imagem	Caracterização
	Ordem Tylenchida Gênero <i>Hoplolaimus</i> [MAI75] Foto 905 X		Ordem Tylenchida Gênero <i>Meloidodera</i> [MAI75] Foto 826 X
	<i>Rotylenchus robustus</i> [CAS94]		<i>Rotylenchus acuspicaudatus</i> [CAS94]
	<i>Rotylenchus cazorlaensis</i> [CAS94]		<i>Rotylenchus karoensis</i> [CAS94]
	Gênero <i>Criconemoides</i>		Gênero <i>Hirschmanniella</i>

	<p><i>Hemicriconemoides brachyurus</i></p>		<p><i>Helicotylenchus erythrinae</i></p>
	<p>Gênero <i>Hoplolaimus</i></p>		<p><i>Parasitaphelenchus beccus</i> [MAS74]</p>
	<p><i>Parasitaphelenchus gallagheri</i> [MAS74]</p>		<p><i>Rotylenchus alpinus</i> [CAS94]</p>
	<p><i>Cryptaphelenchus ipinius</i> [MAS74]</p>		<p><i>Aglenchus exiguus</i> [MAS74]</p>

	<i>Neoditylenchus corniculatus</i> [MAS74]		<i>Sychnotylenchus mutici</i> [MAS74]
	<i>Deladenus ipini</i> [MAS74]		<i>Anguillonema leperisini</i> [MAS74]
	<i>Robleus cylindricus</i> [MAS74]		<i>Aphelenchoides conophthori</i> [MAS74]
	<i>Trophonema arenarium</i> [MAI75]		Gênero <i>Paratylenchus</i> [MAI75] Foto 770X
	Gênero <i>Tylenchorhynchus</i> [MAI75] Foto 250X		<i>Ditylenchus dispaci</i> [MAI75] Foto 2230 X
	Gênero <i>Paratylenchus</i> [MAI75] Foto 1220 X		Gênero <i>Paratylenchus</i> [MAI75] Foto 710 X

	<p>Gênero <i>Hemicriconematoides</i> [MAI75] Foto 570 X</p>		<p><i>Hirschmanniella</i> <i>gracillis</i> [MAI75] Foto 554 X</p>
	<p><i>Tetylenchus</i> <i>joctus</i> [MAI75] Foto 1520 X</p>		<p><i>Deladenus</i> <i>paradurus</i> [MAS74]</p>

Na chave de classificação proposta por Mai ([MAI75]), aspectos morfológicos do estilete propiciam a seguinte identificação taxionômica, tais como:

- a) Ausência de estilete - A maioria dos nematóides que não possuem estiletos pertencem à superfamília Rhabditoidea.
- b) Presença de estilete - Nematóides com estilete pertencem ou à ordem Tylenchida ou à ordem Dorylaimida.
- c) Estilete curto e curvo além do corpo curto e espesso (0.45 a 1.5 mm) - Pertencem ao gênero *Trichodorus*.
- d) Estilete longo, reto - Podem caracterizar um animal do gênero *Xiphinema*, *Longidorus* ou *Paralongidorus*, dependendo dos aspectos de outras estruturas do nematóide.

- e) Estilete reto, não muito longo - Vários gêneros.
- f) Estilete sem bulbo basal com cauda filiforme - Gênero *Psilenchus*.
- g) Estilete com bulbo basal, com cauda filiforme - Gênero *Brachydorus*.
- h) Estilete longo ($s = 2.5$ ou mais) - Gênero *Tylodorus*.
- i) Estilete curto ($s < 2.5$) - Gênero *Tylenchus*.
- j) Estilete franzino - Gênero *Anguina*, *Subanguina* ou *Ditylenchus*, dependendo de aspectos de outras estruturas do nematóide.
- k) Estilete robusto - Gênero *Rotylenchoides*, *Pratylenchus*, *Radopholoides*, *Acontylus* ou *Hoplotylus*, dependendo de aspectos de outras estruturas do nematóide.
- l) Bulbo basal do estilete arredondado e cutícula com mais de 200 anullis - Gênero *Hemicycliophora*.
- m) Bulbo basal do estilete com formato de âncora - Gênero *Hemicriconemoides*.

3.4 - Importância Econômica dos Nematóides

Fitonematóides são patógenos de difícil controle, fácil disseminação e atacam praticamente todas as culturas de importância econômica, causando prejuízos que vão desde a destruição de mudas até a redução drástica da produção. As consequências desses danos são: redução de cerca de 12% da produtividade e da qualidade dos produtos, que refletem em prejuízos para o produtor e elevação dos preços para o consumidor. O primeiro passo no combate a esses organismos é o reconhecimento adequado de sua existência, ou seja, o diagnóstico correto. Muitas vezes, o produtor confunde, por exemplo, o ataque do nematóide com a deficiência de nutrientes e, com isso, jamais consegue resolver o problema em sua lavoura. ([VIL92]).

O reconhecimento adequado da existência de nematóide implica na sua identificação em termos de classificação taxionômica. Sabendo-se por exemplo o gênero a que pertence o nematóide, é possível analisar históricos da ação parasitária do nematóide de tal gênero, e com isso tomar medidas adequadas para o seu combate.

Por exemplo, Novaretti, Miranda e Alcântara ([NOV82]) efetuando estudos sobre o parasitismo de nematóides em soja numa área bastante infestada de *Meloidogyne javanica*, *Pratylenchus brachyurus* e *Helicotylenchus* sp aplicaram dois nematicidas na lavoura: Temik 10G e Carbofuran 5 G. Como resultado, indicam que o nematicida Carbofuran mostrou-se eficiente no controle de *Pratylenchus brachyurus*, obtendo-se uma redução de até 74% deste espécime na raiz das plantas. Para *Helicotylenchus* sp, também apenas o Carbofuran trouxe algum resultado, embora bem menos efetivo (29%). Para o nematóide *Meloidogyne javanica* ("galhas"), os autores atestam que ambos nematicidas, Temik e Carbofuran, causaram significativa redução desta população.

Zem, Zanon e Lordello ([ZEM82]), também comprovaram experimentalmente a eficácia do nematicida Carbofuran em plantações de batatas no município de Castro-PR, contra os nematóides de "galhas" (*Meloidogyne* sp). Brancalion e Lordello ([BRA82a]) realizaram análises semelhantes dos nematicidas Carbofuran e Aldicarb para controle de *Pratylenchus* sp e *Helicotylenchus* sp na cultura do milho. Também realizaram experimentos com Carbofuran para combate ao nematóide *Rotylenchus reniformis* na cultura de algodão([BRA82b]).

Outros exemplos poderiam ser citados, porém o que se ressalta aqui é a importância da identificação do nematóide, em termos taxionômicos, para que de posse de tal informação seja possível conhecer seu histórico em termos de patogenicidade nas diversas culturas. Deste modo, conforme salienta Vilela ([VIL92]), tem-se buscado a difusão de conhecimentos corretos sobre os danos causados por nematóides, seu reconhecimento e alternativas para amenizar prejuízos através de tecnologias adequadas e que evitem não só o uso de práticas desnecessárias, mas também o excesso de pesticidas, que, além de elevarem os custos da produção, causam danos ao meio ambiente.

3.5 - Fitonematóides das Principais Plantas Cultivadas

Castillo, Vovlas, Barcina e Lamberti ([CAS94]) comentam sobre o nematóide do gênero *Rotylenchus*. Citam, por exemplo, o espécime *Rotylenchus robustus* que é o agente causador de grave doença em cenouras, causando uma severa redução em suas raízes e atrofiando seu crescimento. Outros tipos que danificam as raízes da cenoura são *Rotylenchus buxophilus* e *Rotylenchus laurentinus*.

Mehta ([MEH78]) falando sobre doenças do trigo relata o fato de existirem mais de 10 espécies de nematóides pertencentes a cerca de 8 gêneros, patogênicos ao trigo. Relatam-se casos em que moléstias causadas por nematóides podem resultar em perdas na lavoura superiores a 50%. Nematóides como o *Pratylenchus minyus* e *Pratylenchus neglectus* infeccionam as raízes do trigo causando escurecimento e necrose das mesmas.

Olerícolas tais como quiabo, cenoura, pepino, chuchu, ervilha, feijão, berinjela, tomate, abóbora, batata e batata-doce são altamente suscetíveis ao ataque de nematóides das galhas (gênero *Meloidogyne*).

Em 1979 detectou-se pela primeira vez o nematóide *Ditylenchus dipsaci*, causador da doença denominada "podridão-do-alho", nos principais municípios produtores de alho de Minas Gerais e Santa Catarina. As perdas de plantio podem chegar até a 100%. Quando o ataque é severo, o alto custo de controle leva o agricultor a abandonar o plantio já que o nematóide reproduz-se rapidamente nos tecidos do alho e sobrevive no campo por muito tempo ([HUA92]).

Segundo Novaretti ([NOV92]) a cultura da cana-de-açúcar também sofre com o ataque de nematóides. Prejuízos provenientes desta cultura alcançam a cifra anual de US\$ 16 bilhões com uma estimativa de perda da ordem de 15,3% ao ano. Estudos conduzidos em condições controladas de viveiro mostraram reduções de até 54% do peso seco das raízes de cana, e perdas em torno de 43% na produção quando infectadas pelo nematóide *Meloidogyne javanica*. Na África do Sul os nematóides *Criconeoides sphaerocephalum* e *Pratylenchus zae* causaram redução do peso na ordem de 35 %. Para nematóides do gênero *Tylenchorhynchus* e *Hemicycliophora labiata* houve decréscimo na produção de 51% e 25% respectivamente.

Os nematóides das galhas, gênero *Meloidogyne*, são parasitas obrigatórios e constituem o principal grupo de nematóides fitopatogênicos de importância econômica. Suas espécies estão amplamente distribuídas e atacam quase todas as plantas cultivadas, causando perdas consideráveis na produção ou afetando a qualidade dos produtos ([FER92a]). Colaboradores do Projeto Internacional de *Meloidogyne*, sediado na Universidade da Carolina do Norte, nos Estados Unidos, estimaram o percentual de perdas causadas por "meloidoginose" para algumas culturas no Brasil ([FER92a]).

QUADRO 3

Cultura	(%) Perdas
Banana	8
Feijão-comum	11
Mandioca	3
Citros	5
Café	24
Algodão	17
Videira	6
Mamão	15

Tomate	25
Arroz	4
Soja	23
Cana-de-açúcar	9
Inhame	15

Quadro 3 - Estimativa de Perdas Ocasionadas por *Meloidogyne* spp, para algumas Culturas no Brasil. Fonte: [SIL92]

Ferraz ([FER92b]) em levantamento realizado em 1977 constatou a ocorrência em 23 municípios da Zona da Mata de Minas Gerais das seguintes espécies de nematóides em feijoeiros: *Criconemella onoensis*, *Criconemella ornatus*, *Criconemella sphaerocephalus*, *Helicotylenchus nannus*, *Hemicycliophora lutosa*, *Meloidogyne incognita*, *Meloidogyne javanica*, *Pratylenchus brachyurus*, *Xiphinema elongatum*, *Xiphinema krugi* e *Xiphinema setariae*. O feijão é cultivado em todo o território nacional, porém apresenta baixa produtividade se comparado a outros países. Inúmeros problemas estão ligados a esta baixa produtividade e a presença de nematóides é uma delas.

Em feijoeiros, quando ocorrem infestações muito severas de nematóides das lesões radiculares (*Pratylenchus* spp.) as perdas do rendimento podem ser de 10 a 80 %, ou de 50 a 90 % para os nematóides das galhas (*Meloidogyne* spp.) ([FER92b]).

Para o algodoeiro, as espécies consideradas parasitas são: *Meloidogyne incognita*, *Belonolaimus longicaudatus*, *Rotylenchus reniformis*, *Hoplolaimus galeatus*, *Pratylenchus* spp, *Trichodorus christiei*, *Tylenchorhynchus* spp., *Hoplolaimus columbus*, *Hoplolaimus indicus*. *Meloidogyne incognita* foi detectada em algodoeiros do Paraná e Goiás ([RUA92]).

Aphelenchoides besseyi é um nematóide presente em todas as regiões orizícolas do mundo, causador da doença denominada "ponta branca" do arroz. Já foi detectado no Brasil nos estados do Rio Grande do Sul, São Paulo, Goiás, Mato Grosso, Distrito Federal, Minas Gerais, Santa Catarina, Espírito Santo, Bahia, Alagoas e Maranhão. No Rio Grande do Sul relatam-se perdas entre 30 e 50% na cultivar Stirpe Pelotas em áreas com intenso ataque do nematóide ([SIL92]).

Nematóides dos gêneros *Meloidogyne* spp, *Pratylenchus* spp, e *Rotylenchus reniformis* são os causadores das principais doenças em abacaxizeiros. Podem causar perdas de até 42% da produção só na primeira safra. A estimativa de perda mundial na produção do abacaxi por nematóides está na ordem de 13,7% ([ALM92]). No Brasil encontra-se com frequência as espécies *Pratylenchus brachyurus* e *Meloidogyne incognita* nos abacaxizeiros dos estados da Bahia, Minas Gerais, Paraíba, Pernambuco, Rio de Janeiro e São Paulo.

Tylenchulus semipenetrans é bastante nocivo e altamente distribuído por todas as regiões onde se cultivam citros. É um exemplar típico que ataca plantas cítricas, além da oliveira e da videira. Causa necroses extensas nas raízes deixando-as com coloração escura, fazendo com que se produzam frutos menores e em menor quantidade ([LOR82]).

Meloidogyne ataca a figueira provocando galhas. No Ceará, as perdas da produção de mamão por problemas com nematóides atingem a faixa dos 20%. No morangueiro, *Aphelenchoides besseyi*, o mesmo nematóide que causa a "ponta branca" no arroz, causa uma doença denominada "enfazamento de verão" ([ALM92]).

Moura ([MOU86]) apresenta alguns aspectos econômicos relativos à cultura da banana. Em 1983 a produção mundial de banana atingiu 40,7 milhões de toneladas. O Brasil tem se destacado como o primeiro produtor mundial, responsável por 16% do total produzido no mundo.

Cerca de 38 espécies, pertencentes a 20 diferentes gêneros de nematóides foram relacionados como possíveis parasitas da bananeira. O mais importante nematóide é o *Radophoulus similis*, conhecido como "nematóide cavernícola", embora os mais amplamente disseminados nas bananeiras sejam dos gêneros *Meloidogyne* e *Helicotylenchus*. Em muitos países tropicais têm sido observadas perdas em torno de 12,5 ton/ha, podendo a produção ser reduzida em 50% em bananais com três-quatro anos, devido a presença de nematóides ([LIM86]).

Rotylenchus Reniformis já foi encontrado no Brasil parasitando a soja, tomateiro, citros, algodão e pés de maracujá. No município de Leme (SP) cita-se o caso de uma cultura relativamente extensa tendo sido praticamente dizimada por este nematóide, conduzindo o agricultor a gastos adicionais com adubação que não trouxeram qualquer benefício para recuperação da lavoura ([LOR82]).

O gênero *Ditylenchus* é típico parasita de caules e que causam graves danos à lavoura. O espécime *Ditylenchus dispaci* figura entre os mais nocivos inimigos das plantas, podendo atacar cerca de 350 tipos de plantas ([LOR82]).

Rhadinaphelencus cocophilus causa uma doença denominada "anel vermelho" que ataca coqueiro, dendezeiro e tamareira. A doença inicia com um amarelamento das folhas, continua com uma abundante queda de flores e frutas e aparecimento de lesões escuras sobre o pecíolo do vegetal ([LOR82]).

No Brasil, os nematóides associados ao cafeeiro compreendem um grupo numeroso de espécies, dentre os quais citam-se os gêneros *Meloidogyne*, *Pratylenchus*, *Helicotylenchus*, *Xiphinema* e *Criconemella*. Estima-se que a redução da produção brasileira de café devido à ação dos nematóides seja de 20% ([GON92]).

Complementando esta seção, vale ressaltar a importância do impacto da presença de nematóides na cultura de soja, que contribui de forma significativa para o PIB Brasileiro. Uma apresentação mais

detalhada sobre a presença de nematóides na cultura de soja e suas consequências em termos econômicos encontra-se no Anexo I.

3.6 - Aspectos Relativos à Identificação de Nematóides

Boag, Tophan, Brown e Smith ([BOA88]) justificam o desenvolvimento de sistemas computacionais inteligentes para identificação de nematóides devido o fato de nematóides parasitas de plantas terem importância econômica no sentido de ser prejudicial à lavouras em geral.

Destacam que a taxa de catalogação de novos exemplares tem crescido rapidamente, e isso implica na necessidade de um eficiente método para classificação taxionômica dos nematóides. Citam por exemplo que os gêneros *Helicotylenchus* e *Xiphinema* apresentam aproximadamente 150 espécies.

Barker ([BAR93]) aponta que cursos a nível de graduação enfocando nematologia agrícola contribuíram bastante nas últimas quatro décadas. Muitas das pesquisas pioneiras relativas à genética, ecologia, sistemática e biologia geral de nematóides devem-se aos programas de graduação. Entretanto, ressalta, que o esforço deva ser ampliado, para tentar promover ao máximo a difusão da nematologia agrícola, face à sua importância.

Boag, Tophan, Brown e Smith ([BOA88]) apresentam um histórico de programas de computador para identificação de nematóides: Hall em 1970 escreveu um programa de classificação baseado na chave de classificação apresentada por Lamarck em 1778 enquanto Pankurst em 1975 escreveu um programa exclusivamente restrito a plantas para computadores de grande porte .

Boag e Smith desenvolveram em 1982/1983 um programa para identificação de nematóides do gênero *Helicotylenchus*, que funcionava em minicomputadores. Fortuner em 1983 escreveu um programa denominado NEMAID também para o gênero *Helicotylenchus*, com maior potencial que o de Boag, pois funcionava em mainframe. Boag faz referências também a um programa escrito em linguagem BASIC para PC voltado à identificação do gênero *Longidorus*.

A estratégia geral de identificação dos programas acima mencionados é através da técnica de *template matching*: São designados diferentes aspectos que caracterizam o nematóide, onde cada aspecto tem um determinado peso para a identificação final, e daí se confrontam estes aspectos com um "gabarito" padrão de aspectos representantes daquele grupo de nematóides que se deseja identificar. Coeficientes de similaridade por vezes são usados, como por exemplo o coeficiente de similaridade proposto por Gower ([GOW71]).

Os programas de identificação começam a se popularizar de fato em fins da década de 80, principalmente devido ao surgimento dos microcomputadores de CPU de 16 bits, que levou o microcomputador com algum poder de processamento aos laboratórios.

Peet, Panesar, Sahota e Sutherland ([PEE90]) apresentam um sistema de processamento de imagem digital utilizado para medir várias características físicas de nematóides e para comparação do indivíduo com um determinado grupo de nematóides. Citam trabalhos de Boag, Panesar e Croll que se utilizam de técnicas manuais para efetuar medidas geométricas dos nematóides. A inovação, segundo os autores, é a aquisição dessas mesmas medidas, diretamente através do processamento digital da imagem, embora citem que uma certa interação do operador humano com a imagem seja necessária (por exemplo para marcar *aproximadamente* o local de uma determinada estrutura).

Os programas de identificação portanto, começam a incorporar o processamento da imagem de modo geral, como parte integrante do processo de identificação de nematóides.

Outro trabalho que busca efetuar caracterização de nematóides baseado em processamento de imagem digital é o apresentado por Blanca, Valdivia, Castillo e Barcina ([BLA92]). A estrutura deste trabalho se resume em três etapas: A partir de uma imagem digital do nematóide aplicam-se técnicas para extração dos aspectos geométricos e texturais do nematóide mais marcantes:

- 1) Algoritmos para retirada de "ruídos", e realce da imagem são aplicados.
- 2) Em seguida aplicam-se técnicas de reconhecimento de padrões para se definirem os diferentes formatos geométricos presentes na imagem.
- 3) Finalmente faz-se uso de técnicas estatísticas ou inteligência artificial para identificação do nematóide.

Francl ([FRA92]) cita um glossário de termos referentes a nematóides implementado com tecnologia de hipertexto. No caso foi usado o sistema de hipertexto Black Magic v. 2.0.

Eisenback ([EIS93]) apresenta um exemplo de aplicação de multimídia interativa para o ensino em nematologia. Ressalta o impacto positivo que esta tecnologia traz para as mais diversas áreas de conhecimento, entre as quais a relacionada ao estudo de nematóides.

Bravo e Roca ([BRA94]) desenvolveram um programa para microcomputadores com o software dBASE IV, para caracterização do gênero *Xiphinema*. O programa tem a característica de sistema especialista, pois armazena em banco de dados relacional (estrutura própria do dBASE) aspectos (morfológicos e morfométricos) característicos deste gênero. Contudo não realiza tratamento de imagens.

O mesmo pode se dizer do programa escrito por Robbins e Brown ([ROB94]). É um típico sistema especialista que não realiza tratamento de imagens. Está escrito no software de planilha LOTUS e serve de suporte para diagnóstico e identificação do gênero *Longidorus*.

Cabral ([CAB96]) desenvolve um trabalho de processamento de imagens de nematóides com a finalidade de preparar tais imagens para um reconhecimento de padrões baseado em imagens com

qualidade e disposição espacial favorável: Em essência toma a imagem de um nematóide, normalmente "encaracolado", e a "estica" (denomina este processo de *retificação*), deixando-a em posição vertical (ou horizontal). Isso facilita enormemente um posterior reconhecimento de estruturas do nematóide pois seus órgãos internos não apresentarão mais "deformações" que apresentavam ao acompanhar o encaracolamento inicial na imagem do nematóide.

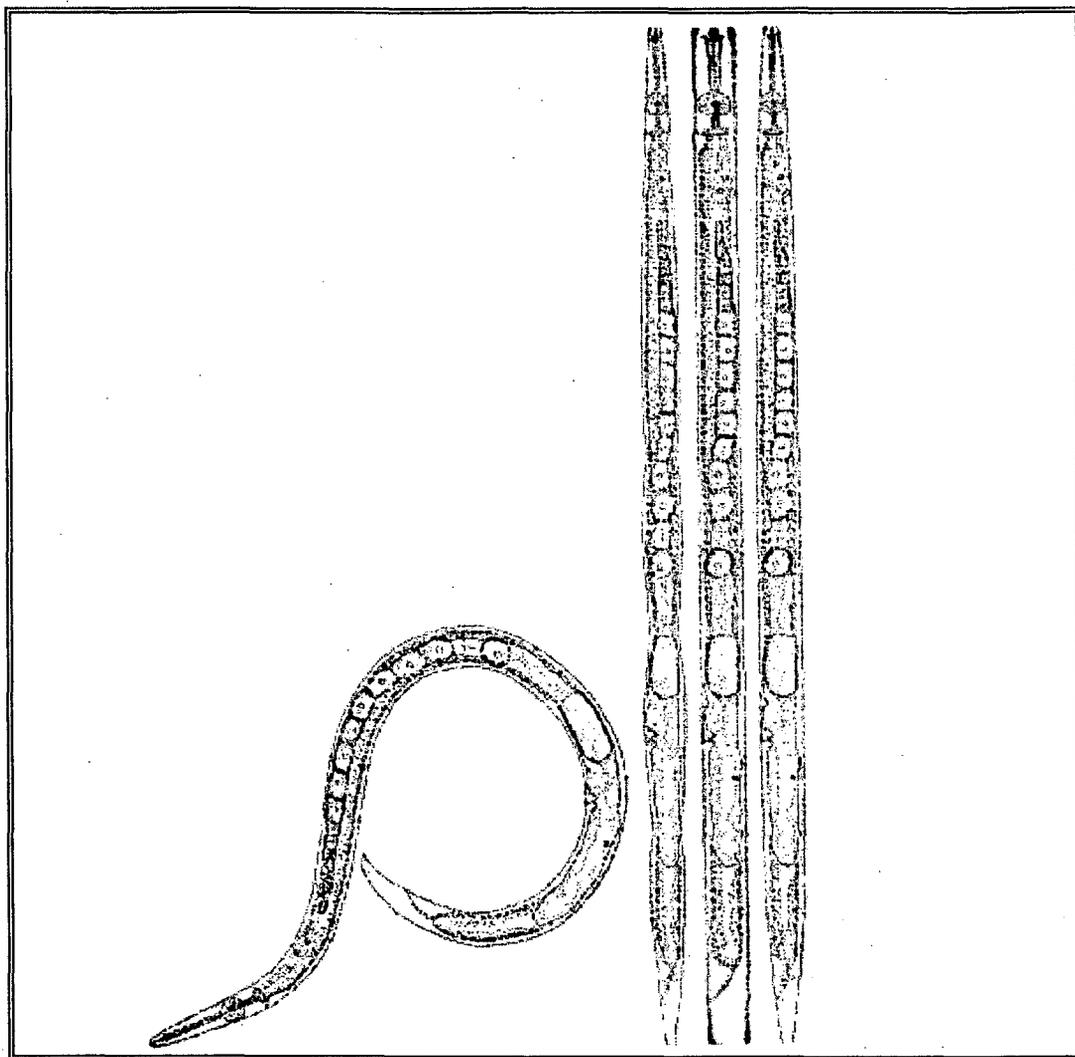


Fig. 3.8 - Pré-processamento da imagem de um nematóide. O nematóide é "desenrolado", ficando retificado. Fonte: [CAB96]

3.7 - Escopo da Aplicação - Identificação de Nematóides Através do Formato do Estilete

O estilete é uma das várias estruturas do nematóide cujos aspectos morfológicos e morfométricos servem de base, conforme apresentado anteriormente, para identificação/classificação

taxionômica, numa determinada chave de classificação. Em alguns casos somente seu formato já levará a uma classificação. Em outros casos, informações adicionais a respeito da morfologia/morfometria de outras estruturas serão necessárias para se chegar a alguma classificação.

O objetivo do presente trabalho visa contudo conseguir efetuar somente um reconhecimento morfológico (e não morfométrico) da estrutura do estilete e, na medida do possível, com base no conhecimento do formato, identificá-lo/classificá-lo. Uma vez que se consiga gerenciar computacionalmente o formato desta estrutura, espera-se que também se consiga o mesmo com o formato de outras estruturas (esôfago, intestino, ovário, etc).

Baldwin e Luc ([BAL95]) alertam para a distinção existente entre *identificação* e *classificação*. Identificação, segundo os autores, é designar um nome, de modo genérico a uma espécie ou população. Taxionomia ou classificação, está ligada ao desenvolvimento de uma hierarquia de categorias (espécie, gênero, família, ordem, etc) baseada em caracteres específicos para cada nível taxionômico. Tais caracteres devem refletir aspectos evolutivos.

Por isso, os elementos usados nos dois casos não apresentam o mesmo teor qualitativo. Os caracteres usados na identificação são mais fáceis de serem percebidos, segundo os autores. Um exemplo de identificação seria a averiguação da estrutura superficial da cutícula em vários grupos da ordem Tylenchida.

Os aspectos levados em consideração para classificação taxionômica, considerando-se o contexto acima exposto, devem ter valor para interpretação evolutiva. Baldwin e Luc apontam aspectos morfológicos e anatômicos sendo válidos para atividades de classificação de uma taxionomia "clássica". Porém, para classificação taxionômica, outros aspectos podem e devem ser levados em conta: ultraestrutura, bioquímica, DNA, cariótipos, modos de reprodução, relação hospedeiro/parasita, etc.

Dentro desta ótica, o que se pretende no presente trabalho é, em suma, proporcionar um modelo de apoio à identificação de nematóides. A identificação aqui pode em certos casos coincidir com alguma classificação taxionômica, pois, conforme salientado, dependendo da chave de classificação que se use, pode-se somente pelo formato do estilete do nematóide, designar sua família, subfamília, gênero e até mesmo espécie.

Neste trabalho utilizam-se contudo os termos *identificação* e *classificação* com o mesmo intuito, para proporcionar algum tipo de identificação, seja esta identificação genérica (que coincide com o termo *identificação* exposto por Baldwin e Luc) ou mesmo complexa envolvendo, se possível, aspectos evolutivos (que coincidiria com a idéia de *classificação taxionômica* de Baldwin e Luc).

CAPÍTULO 4

Modelo para Identificação de Nematóides Através do Estilete

4.1 - Introdução

Neste capítulo é apresentada a proposta do modelo conceitual para identificação de nematóides baseado na característica morfológica do estilete.

É apresentado o esquema geral do sistema de identificação e seus processos componentes: segmentação da imagem, indicação da localização da estrutura do estilete, aproximação poligonal para contorno C do estilete, representação de tal contorno através de um grafo AFT (*Attributed Fuzzy Tournament*) ([SHA92]), digitação das características do estilete (fase de aprendizado) e identificação das características do nematóide de estilete de contorno C (fase de reconhecimento).

4.2 - Esquema Geral do Sistema de Identificação

O sistema para identificação de nematóides através da estrutura do estilete pode ser dividido em duas etapas:

- Aprendizado
- Reconhecimento/Identificação

Na primeira etapa o sistema "aprende" a relacionar um determinado formato de estilete com sua caracterização. É uma etapa em que há a interação homem-máquina, pois o operador do programa, com auxílio do *mouse*, irá apontar para a região da imagem do nematóide correspondente ao estilete, marcando-a convenientemente. Em seguida, o programa gera um polígono representativo do contorno do estilete. Após a geração deste polígono, o programa solicita ao operador para digitar as características gerais do nematóide em questão. As características digitadas e o contorno da região da imagem indicada com o *mouse* são armazenadas convenientemente numa base de conhecimento.

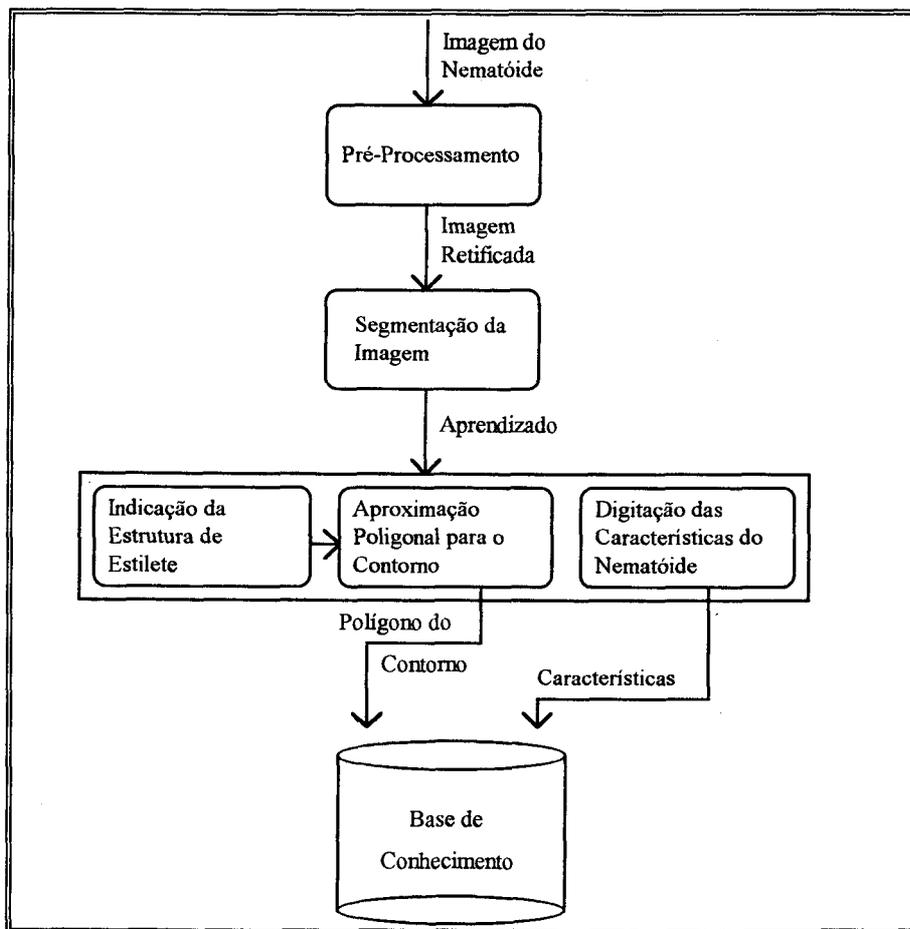


Fig. 4.1 - Esquema para Etapa 1 - Aprendizado do formato do estilete

A segunda etapa refere-se ao reconhecimento/identificação propriamente dito. A partir de uma imagem I de um nematóide não identificado, o sistema

- 1) Interage com o operador do programa, que com o auxílio do *mouse*, irá apontar para a região R da imagem do nematóide correspondente ao estilete, marcando-a convenientemente,
- 2) Com a região R apontada através do *mouse*, o sistema constrói a representação de seu formato, tratando-o como um polígono P (conjunto de vértices e arestas).
- 3) A partir deste polígono P , o sistema:
 - Constrói o grafo AFT $T1$ caracterizador do polígono P .
 - Para cada *template* T existente na base de conhecimento (que armazena o polígono Q caracterizador do *template* T),

- Constrói o grafo AFT T2 caracterizador do polígono Q
 - Efetua *template matching* entre T1 e T2.
- 4) O sistema assume como similar, o template Q cuja distância do grafo AFT T2 em relação ao grafo AFT T1 de P for a menor entre todas as distâncias calculadas.

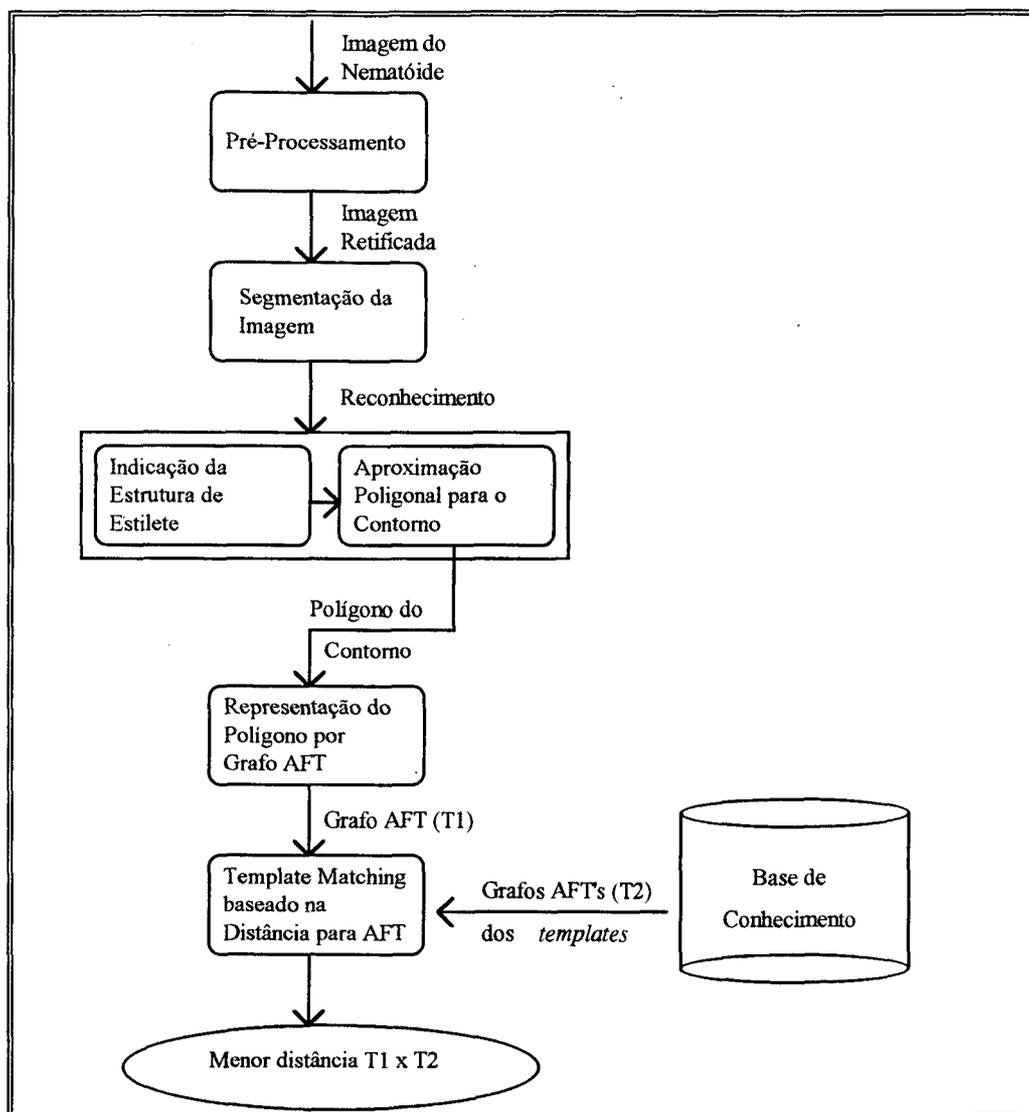


Fig. 4.2 - Esquema para Etapa 2 - Reconhecimento/Identificação do formato do estilete

Nos próximos itens são comentadas cada uma das etapas do modelo, mostradas nas figuras 4.1 e 4.2. Comenta-se também sobre as imagens de nematóides que o sistema se propõe a processar.

4.3 - Considerações Sobre as Imagens de Nematóides Tratadas pelo Modelo

As imagens de nematóides utilizadas pelo programa protótipo estão armazenadas em arquivos do tipo "bitmap" (padrão MS-Windows). São fotos e desenhos, obtidos através do processo de digitalização em scanner.

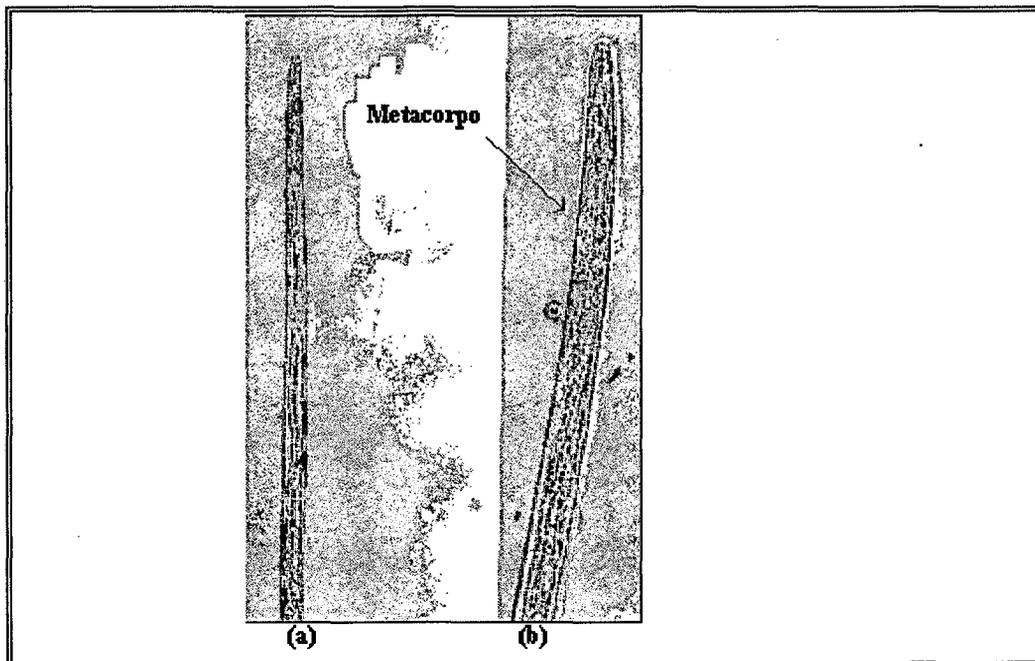


Fig. 4.3 - Foto de um nematóide do gênero *Aphelenchoides*. (a) imagem não apresenta nitidez em nenhuma estrutura. (b) A estrutura do metacorpo está levemente mais nítida, porém o estilete não.

O ideal é que as imagens sejam fotos, obtidas diretamente da lâmina de microscópio, de preferência apresentando o corpo completo do nematóide, com suas estruturas mais nítidas quanto possível. Entretanto, dependendo do equipamento disponível, a foto com o corpo completo poderá não apresentar nitidez nas estruturas, o que inviabilizaria o seu uso pelo programa.

Normalmente as fotos são tiradas da parte de interesse do animal, com maior ampliação possível. Neste caso, a estrutura de interesse, o estilete, apresentará maior nitidez. Fotos contudo são mais difíceis de se obter, e o processo de elaboração por si só requer uma boa habilidade do operador do equipamento fotográfico. Abé, Kito, e Aryuthaka ([ABE93]) comentam problemas correlatos sobre obtenção de imagens de nematóides. Em slides, muitos nematóides longos não "cabem" na imagem, e os autores comentam uma técnica para obtenção da imagem completa em slides.

Quanto à estrutura do estilete, Blanca, Valdivia, Castillo e Barcina ([BLA92]) alertam para o fato de que os estiletos são estruturas tridimensionais, e por isso, é importante garantir que a foto onde se realça a estrutura do estilete seja tomada de um foco que contenha toda sua informação estrutural.

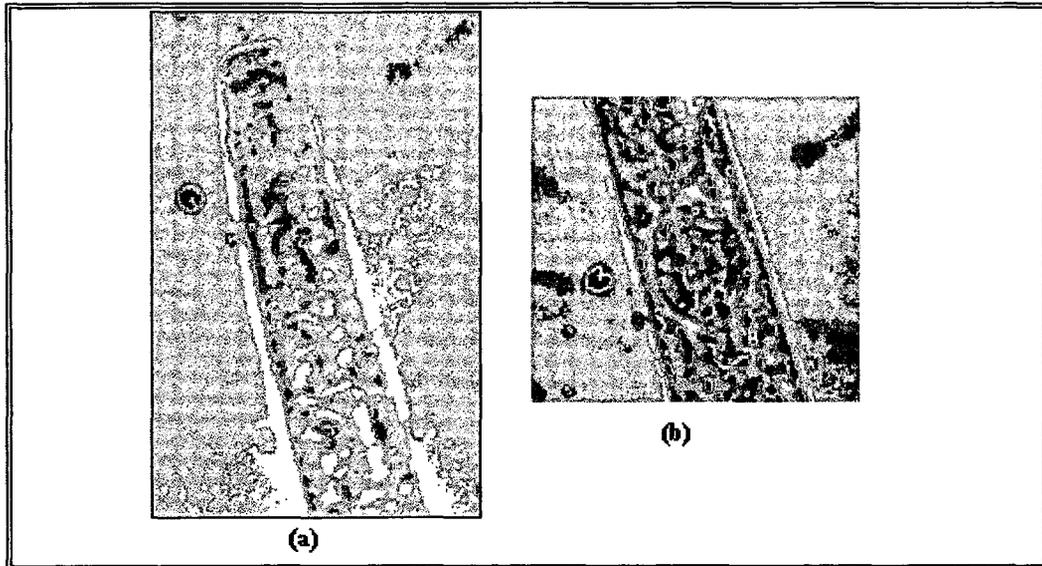


Fig. 4.4 - O mesmo nematóide da figura 4.3. A foto tenta realçar a "cabeça" do nematóide para o estilete, mas ficou fora de foco. O realce da parte mediana do corpo, conseguiu destacar razoavelmente o metacorpo na foto (b).

4.4 - Pré-Processamento

Uma vez que o modelo de grafo AFT é sensível à rotação, é de interesse que seja efetuada retificação da imagem do nematóide, deixando-a na posição vertical (ou horizontal). Conforme citado na seção 3.6 (Aspectos Relativos à Identificação de Nematóides), Cabral ([CAB96]) apresenta um trabalho no sentido de "desencaracolar" e deixar retificadas as imagens de nematóides.

Em vista disso, sendo possível colocar as imagens "retas" e na posição vertical, o problema de sensibilidade à rotação do modelo AFT é resolvido. Além disso, o tempo de processamento para *template matching* reduz-se enormemente pois, supondo que as imagens não estejam retificadas, uma comparação "satisfatória" entre duas imagens implicaria numa rotação "exaustiva" de uma delas para que, a cada rotação efetuada, seja calculada a similaridade entre ambas.

E a etapa de pré-processamento resume-se portanto à submissão da imagem ao processo de retificação.

4.5 - Segmentação da Imagem

Após a retificação da imagem, segue o processo de sua segmentação. A segmentação de imagem é um processo executado tanto na fase de aprendizado como na fase de reconhecimento. É através da segmentação de imagem que o sistema discrimina as várias áreas que a compõe. Cada área, para a imagem de nematóide, representa em princípio uma estrutura (estilete, metacorpo, esôfago, etc).

Para efetuar segmentação da imagem, a proposta é a implementação do algoritmo *split-and-merge* apresentado por Horowitz e Pavlidis ([HOR76]) para segmentação de imagens em tonalidades de cinza.

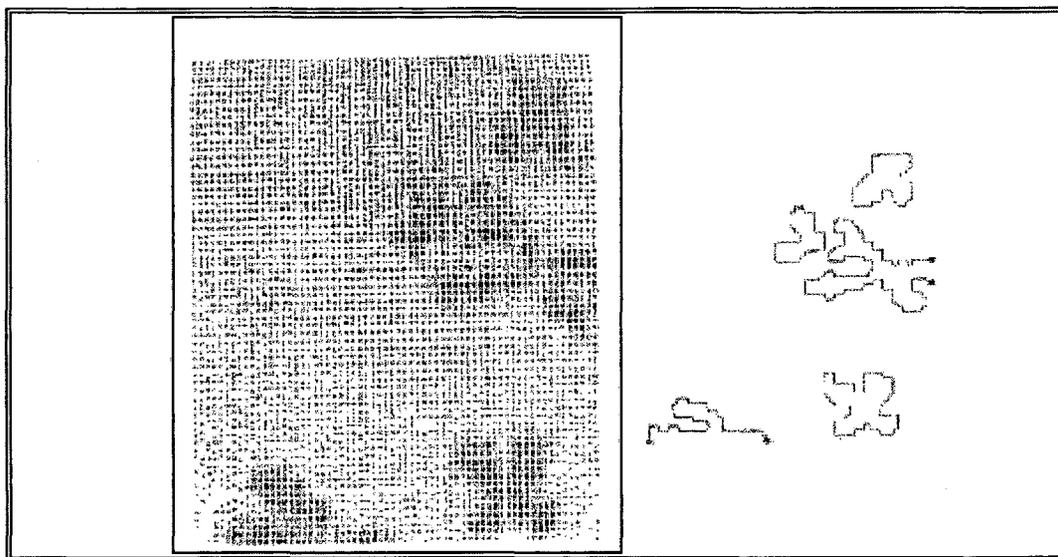


Fig. 4.5 - O processo de segmentação - Técnica de extração de região - Resultado obtido através do algoritmo "split-and-merge" proposto em [HOR76]

Em linhas gerais o algoritmo tem as seguintes características:

- 1) Toma-se uma imagem de dimensão $N \times N$ pixels onde $N = 2^k$, $k > 1$
- 2) Visualiza-se a imagem em *blocos* (quadrados) em vários níveis, cada nível identificado pela largura do *bloco*, caracterizados pela quintupla $\langle x, y, z, M, m \rangle$

$$\text{Bloco}_i = \langle x_i, y_i, z_i, M_i, m_i \rangle, \text{ onde}$$

i	=	i-ésimo bloco da figura
x_i	=	Coluna do pixel superior esquerdo do bloco
y_i	=	Linha do pixel superior esquerdo do bloco
z_i	=	Largura em pixels do bloco
M_i	=	Valor da intensidade máxima entre os pixels do bloco
m_i	=	Valor da intensidade mínima entre os pixels do bloco

- 3) Efetuam-se operações denominadas *merging*: Juntam-se os blocos adjacentes de tamanho z iguais desde que a diferença do maior M entre os blocos com o menor m entre blocos não ultrapasse uma certa tolerância.

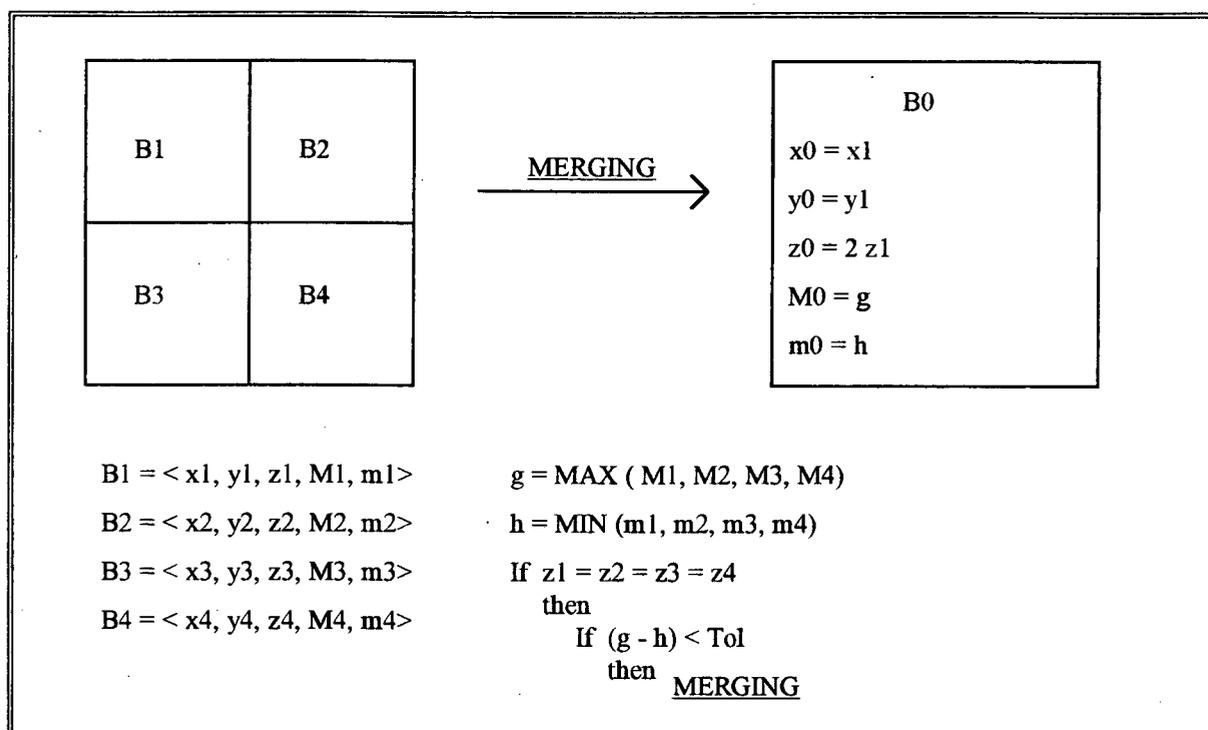


Fig. 4.6 - O processo de segmentação - *Merging*

- 4) Após o *merging* efetua-se a operação denominada *splitting*: Separam-se blocos de tamanho z em 4 blocos distintos de tamanho $z/2$, se a diferença $M - m$ do bloco em questão for maior que uma determinada tolerância.

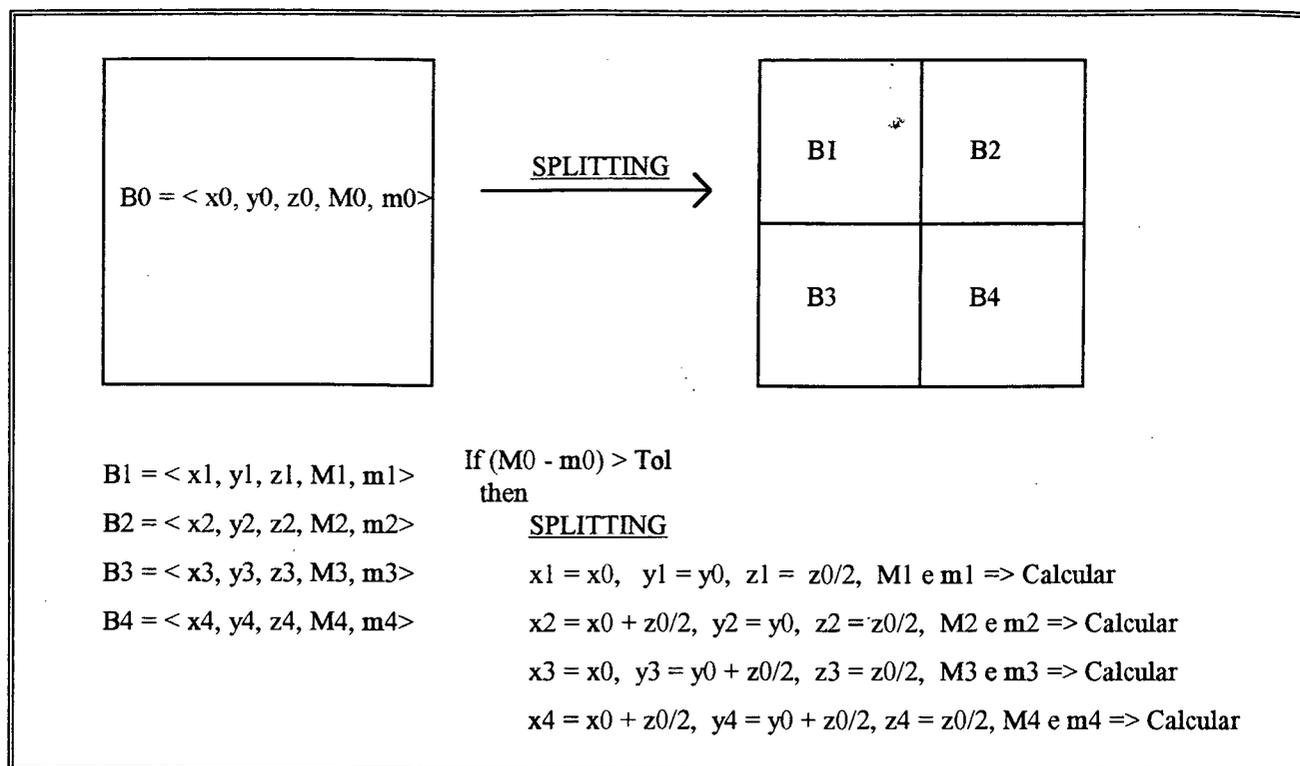


Fig. 4.7 - O processo de segmentação - *Splitting*

- 5) Após as operações de *merging* e *splitting*, efetuam-se as rotinas denominadas *grupamento* e *eliminação*. O grupamento consiste basicamente num *merging* que agrupará regiões de adjacência que atendam ao critério de tolerância estabelecido para o *merging*. Agora as regiões (e não mais blocos) são agrupados independentemente de suas larguras, o que não acontecia durante o *merging* onde só se agrupavam blocos de igual tamanho. A eliminação consiste em "apagar" pequenas regiões que são "gaps" entre duas regiões bem definidas entre si, em termos de suas intensidades médias de pixel.
- 6) Finalmente aplica-se um algoritmo adequado para extração dos contornos das regiões encontradas durante o processo *split-and-merge*.

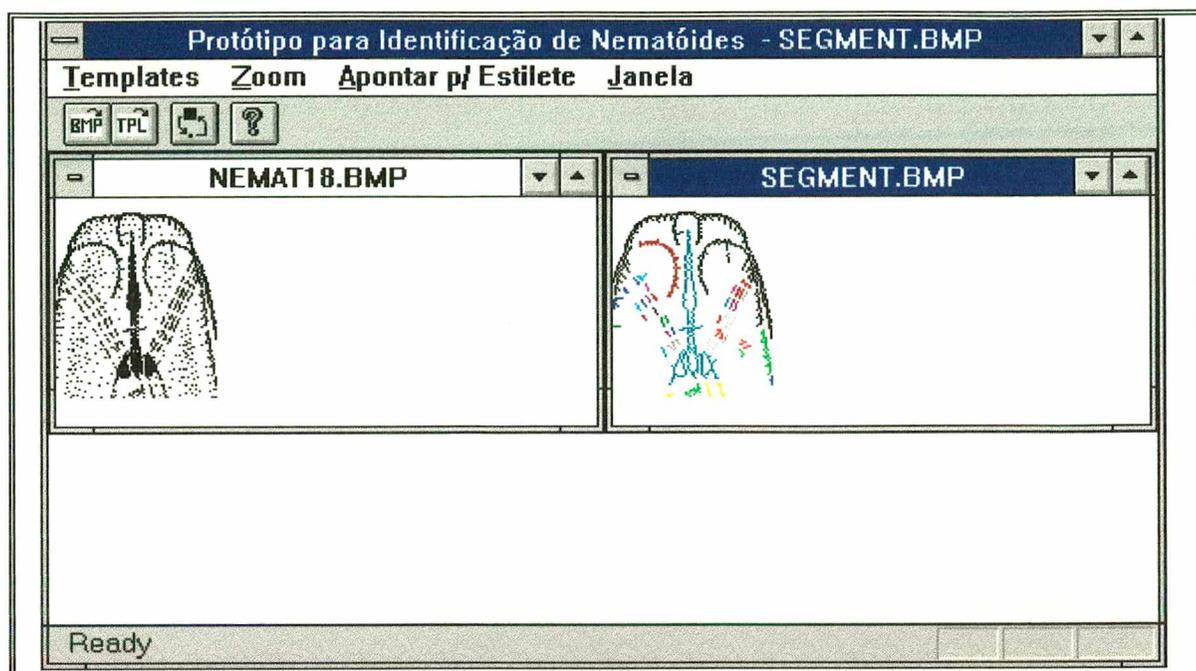


Fig. 4.8 - Segmentação da imagem NEMAT18.BMP. O resultado está na janela SEGMENT.BMP

4.6 - Indicação da Estrutura de Estilete

Após o programa efetuar a segmentação, o operador do programa irá selecionar a região que caracteriza o estilete do nematóide "apontando-a" com o *mouse*. Na figura 4.9, após o processo de segmentação de imagem, gerou-se a janela SEGMENT.BMP. Com o auxílio do *mouse*, o operador apontou para a estrutura do estilete. O programa gerou então a janela ESTILETE.BMP, onde realça a área escolhida.

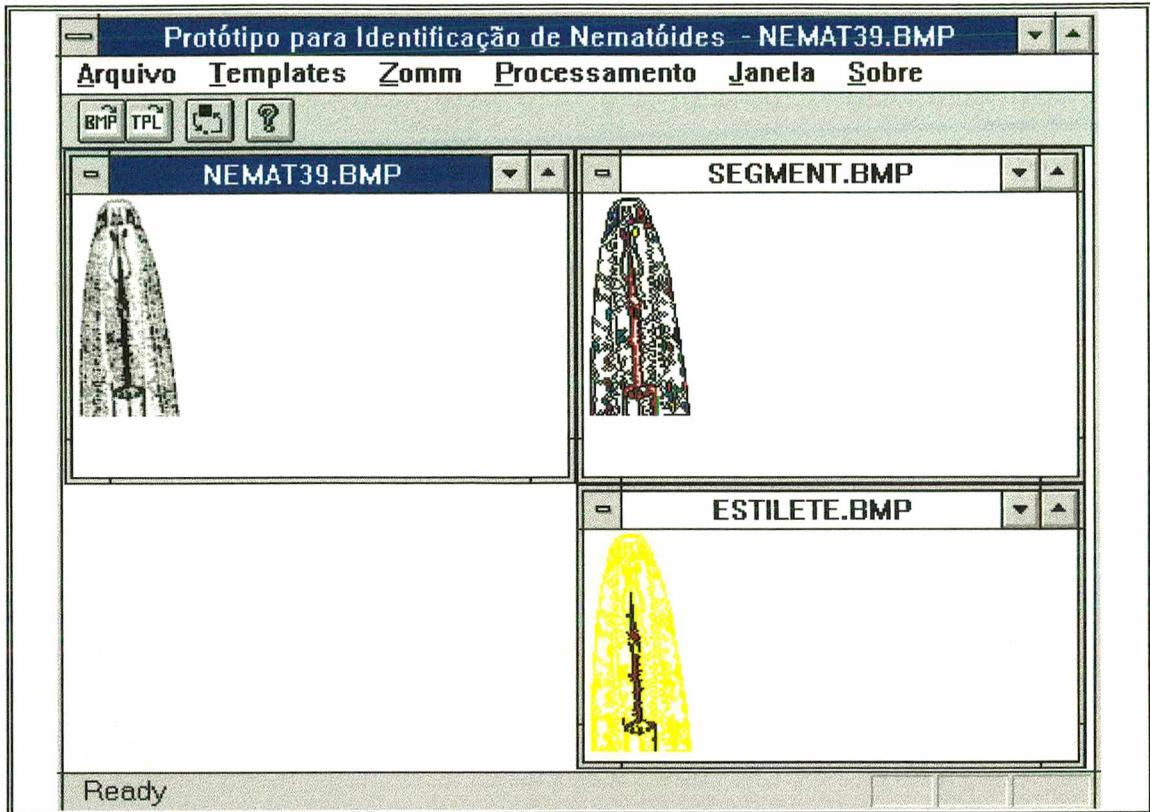


Fig. 4.9 - Indicação da região do estilete por parte do operador.

4.7 - Aproximação Poligonal para o Contorno

Uma vez que se tenha a região escolhida, tem-se automaticamente uma curva em R^2 . A partir daí pode-se aplicar um algoritmo de segmentação para o contorno da região, visando representá-lo através de um polígono.

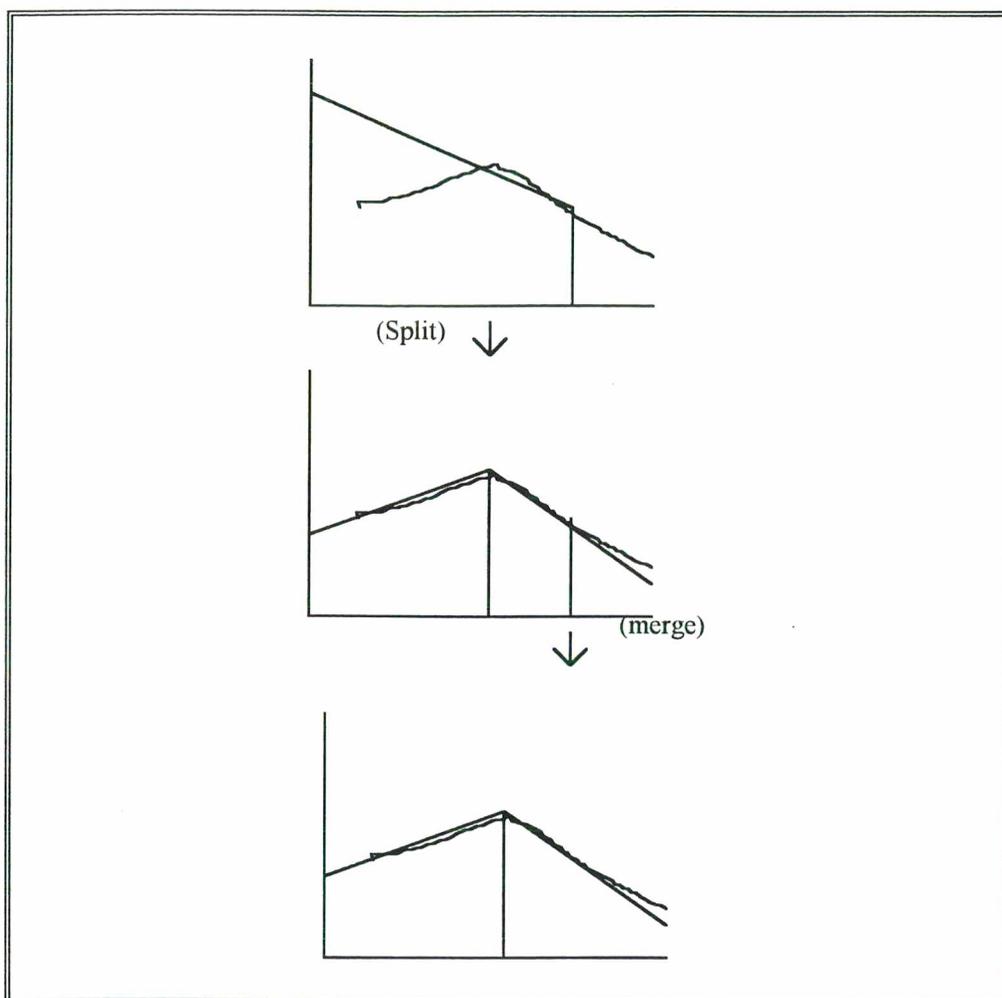


Fig. 4.10 - O processo de *split-and-merge* para aproximação poligonal - Fonte : [PAV74]

Para a segmentação do contorno da região, Pavlidis e Horowitz ([PAV74]) propõem um algoritmo que, igual ao algoritmo de segmentação em regiões, recebe o nome de *split-and-merge*. A idéia deste algoritmo é: partindo de um determinado ponto do contorno achar uma reta que melhor se ajuste nos pontos de contorno seguintes. Esta reta é formada com a incorporação (*merging*) do máximo de pontos possíveis e que não extrapole uma norma de erro para formação de retas. Se esta norma é superada, então o segmento de reta até então montado é separado (*splitting*).

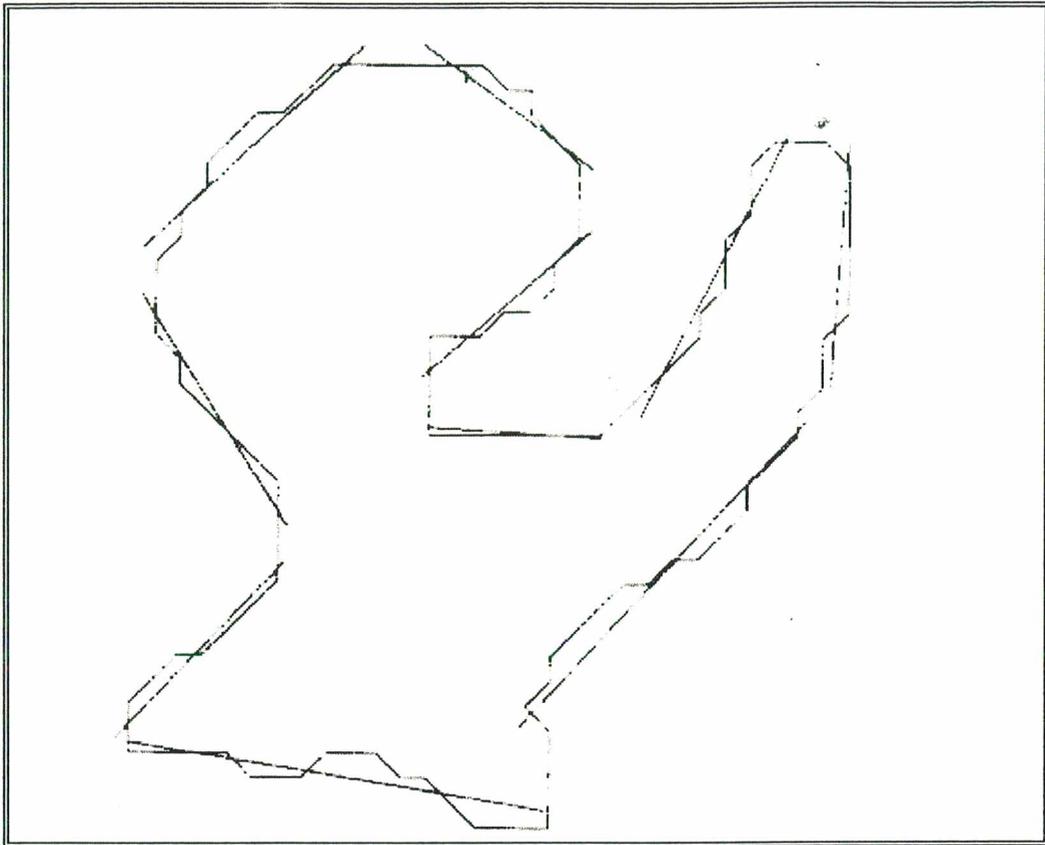


Fig. 4.11 - Exemplo de resultado após aplicação do algoritmo *split-and-merge* - Fonte: [PAV74]

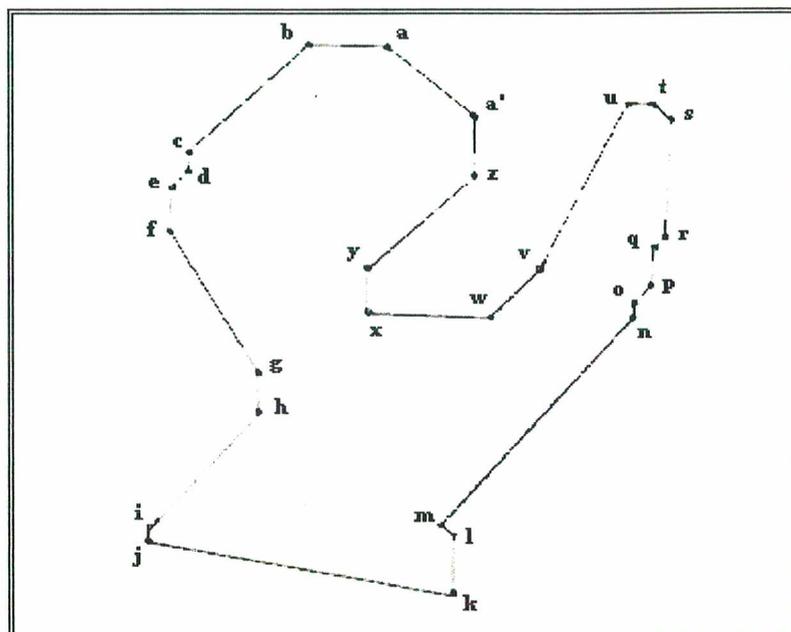


Fig. 4.12 - O polígono representativo do contorno da imagem da figura 4.11

4.8 - Digitação das Características do Nematóide

Operação eetuada apenas durante a fase de aprendizagem. Após o programa gerar o polígono de contorno do estilete, o operador do programa digita as características gerais do nematóide com tal formato de estilete. Em seguida o programa armazena na base de conhecimento, tais características digitadas, além do polígono representativo do contorno do estilete.

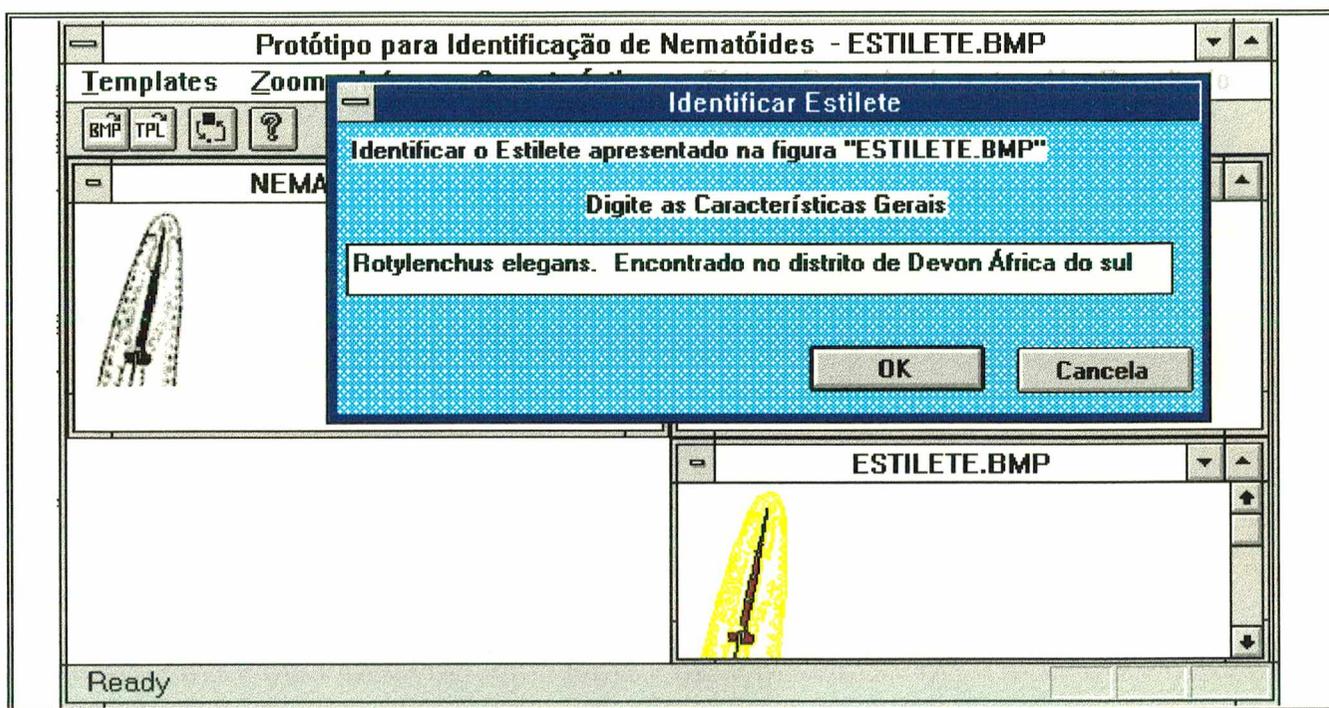


Fig. 4.13 - Digitação das características do nematóide da janela ESTILETE.BMP

4.9 - Representação do Polígono por Grafo AFT

Uma vez que se tenha o contorno do estilete representado por um polígono, é necessário *codificar* esta estrutura de alguma forma. Em outras palavras, é necessário designar alguma representação do polígono de forma a caracterizar sua disposição estrutural. No polígono da figura 4.12, tem-se 27 vértices denominados $a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z$ e a' , e 27 arestas $ab, bc, cd, de, ef, fg, gh, hi, ij, jk, kl, lm, mn, no, op, pq, qr, rs, st, tu, uv, vw, wx, xy, yz, za'$ e $a'a$.

É importante discriminar alguma relação existente entre as arestas do polígono para sua adequada representação. Uma representação do formato do contorno poderia utilizar as relações

(predicados) "estar acima de", "estar abaixo de", "estar à esquerda de", e "estar à direita de". Supondo estes predicados, descreve-se o polígono através das relações entre suas arestas. Assim, algumas das sentenças do conjunto das sentenças que o caracterizariam seriam:

"a aresta ab está acima da aresta yz ",
 "a aresta ab está à direita da aresta bc ",
 "a aresta wx está acima da aresta gh ",
 "a aresta hi está à esquerda da aresta yz ".

O polígono é visto como um cenário onde os objetos são suas arestas e procura-se descrever este cenário através de determinados predicados previamente definidos. Contudo, em cenários existe incerteza. Por exemplo, a sentença "a aresta ab está acima da aresta yz ", apresenta incerteza em sua proposição, pois poder-se-ia dizer também que "a aresta ab está à esquerda de yz ".

Para o presente trabalho, o cenário em questão a ser modelado por grafo AFT ([SHA92]) seria o polígono representativo do contorno do estilete. Para este cenário, os *nodos* $v_k \in V$ do $AFT(n) = (V, A, B, E, X, C)$ que modela o polígono, corresponderiam às suas *arestas*.

Segue a especificação do grafo AFT utilizado para caracterizar o polígono representativo do contorno do estilete.

Grafo AFT para o Polígono do Contorno do Estilete

$AFT(n) = (V, A, B, E, X, C)$ onde

$V = \{ \text{nodos do grafo AFT} \} = \{ \text{arestas do polígono do contorno} \}$

$A = \{ \text{arestas do grafo AFT} \} = \{ \text{arcos das unidades formadas pelos nodos tomados dois a dois} \}$

$B = \{ \text{Horizontal, Vertical, Positivo, Negativo} \} \otimes$
 $\{ \text{Grande, Médio, Pequeno, Muito Pequeno} \} =$
 $\{ \text{Horizontal-Grande, Horizontal-Médio, Horizontal-Pequeno, Horizontal-Muito Pequeno,}$
 $\text{Vertical-Grande, Vertical-Médio, Vertical-Pequeno, Vertical-Muito Pequeno,}$
 $\text{Positivo-Grande, Positivo-Médio, Positivo-Pequeno, Positivo, Muito-Pequeno,}$
 $\text{Negativo-Grande, Negativo-Médio, Negativo-Pequeno, Negativo, Muito-Pequeno} \}$

$E = \{ \textit{Acima de}, \textit{Abaixo de}, \textit{A Esquerda de}, \textit{A Direita de} \}$

$X = \{ \min(\mu_H, \mu_G), \min(\mu_H, \mu_M), \min(\mu_H, \mu_O), \min(\mu_H, \mu_{OO}),$
 $\min(\mu_V, \mu_G), \min(\mu_V, \mu_M), \min(\mu_V, \mu_O), \min(\mu_V, \mu_{POO}),$
 $\min(\mu_P, \mu_G), \min(\mu_P, \mu_M), \min(\mu_P, \mu_O), \min(\mu_P, \mu_{OO}),$
 $\min(\mu_N, \mu_G), \min(\mu_N, \mu_M), \min(\mu_N, \mu_O), \min(\mu_N, \mu_{OO}) \}$

$C = \{ \mu_{Ac}, \mu_{Ab}, \mu_E, \mu_D \}$

Os conjuntos B e X de atributos e graus de pertinência dos nodos do grafo AFT

Dois atributos caracterizam os nodos, a *direção* e o *tamanho* da reta (que é a aresta, ou seja, o nodo do grafo AFT).

Variável linguística = DIREÇÃO

Termos difusos =	Horizontal (H)
	Vertical (V)
	Positivo (P) (coeficiente angular da reta > 0)
	Negativo (N) (coeficiente angular da reta < 0)

Função de pertinência para os termos ([SHA92]) e ([SIY74]):

Horizontal	$\mu_H = 1 - \min \{ \min [\theta , 180 - \theta , 360 - \theta] / 45, 1 \}$
Vertical	$\mu_V = 1 - \min \{ \min [90 - \theta , 270 - \theta] / 45, 1 \}$
Positivo	$\mu_P = 1 - \min \{ \min [45 - \theta , 225 - \theta] / 45, 1 \}$
Negativo	$\mu_N = 1 - \min \{ \min [135 - \theta , 315 - \theta] / 45, 1 \}$

A variável linguística *direção* depende do ângulo θ que a reta forma com o eixo das abcissas.

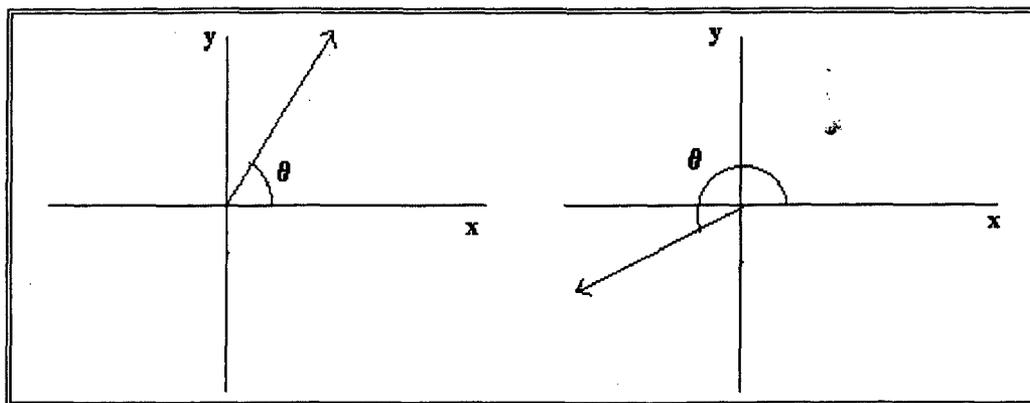


Fig. 4.14 - Determinação do ângulo θ que a reta (nodo) forma com o eixo das abcissas

Para se determinar o ângulo θ considera-se o ponto início da reta como o ponto origem $(0,0)$ e o ponto-fim da reta é calculado relativo a este. Supondo-se uma aresta (nodo) definida pelos ponto-início $(4,7)$ e ponto-fim $(6,2)$, então a reta é caracterizada por $(0,0)$ e $(2,-5)$, encontrando-se no quarto quadrante do eixo das coordenadas cartesianas. Com isso, $270 < \theta < 360$.

Variável linguística = TAMANHO

Termos difusos =	Grande (G)
	Médio (M)
	Pequeno (Q)
	Muito Pequeno (QQ)

Função de pertinência para os termos

Muito Pequeno $\mu_{QQ} =$	1,	se $T < 0.15*N$
	$(-10/N)T + 2.5,$	se $0.15*N \leq T \leq 0.25*N$
	0,	se $T \geq 0.25*N$
Pequeno $\mu_Q =$	0,	se $T < 0.15*N$
	$(10/N)T - 1.5,$	se $0.15*N \leq T \leq 0.25*N$
	1,	se $0.25*N \leq T \leq 0.40*N$
	$(-10/N)T + 5,$	se $0.40*N \leq T \leq 0.50*N$
	0,	se $T \geq 0.50*N$

Médio $\mu_M =$	0,	se $T < 0.40*N$
	$(10/N) T - 4,$	se $0.40*N \leq T \leq 0.50*N$
	1,	se $0.50*N \leq T \leq 0.65*N$
	$(100/(15*N)) T + (80/15),$	se $0.65*N \leq T \leq 0.80*N$
	0,	se $T \geq 0.80*N$

Grande $\mu_G =$	0,	se $T < 0.65*N$
	$(100/(15*N)) T - (65/15),$	se $0.65*N \leq T \leq 0.80*N$
	1,	se $T \geq 0.80*N$

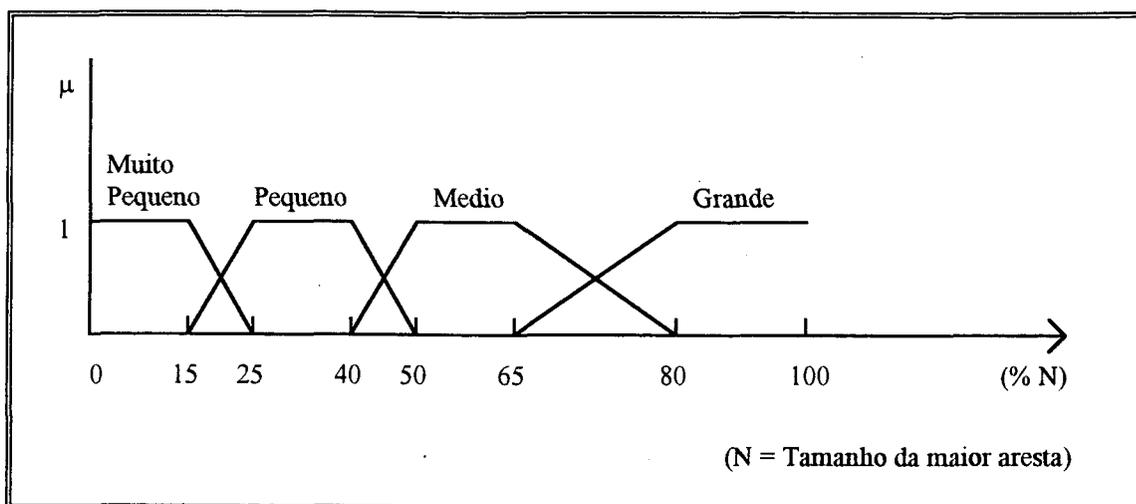


Fig. 4.15 - Função de pertinência para os termos da variável linguística TAMANHO

Para a variável linguística *tamanho*, a estratégia é tomar a maior aresta do conjunto de arestas do polígono é tomar seu comprimento com o valor máximo N . A maior aresta é considerada *grande*. As demais arestas do polígono são classificadas em *grande*, *médio*, *pequeno* ou *muito pequeno* conforme seus tamanhos T em relação à maior aresta de tamanho N .

Os conjuntos A e C de atributos e graus de pertinência dos arcos do grafo AFT

Variável linguística = POSIÇÃO

Termos difusos =	Acima de (Ac)
	Abaixo de (Ab)
	À Esquerda de (E)
	À Direita de (D)

Função de pertinência para os termos ([SHA92]) e ([SIY74]):

Acima de	$\mu_{Ac} = 1 - \min \{ [90 - \phi] / 90, 1 \}$
Abaixo de	$\mu_{Ab} = 1 - \min \{ [270 - \phi] / 90, 1 \}$
À Esquerda de	$\mu_E = 1 - \min \{ [180 - \phi] / 90, 1 \}$
À Direita de	$\mu_D = 1 - \min \{ \min [\phi , 360 - \phi] / 90, 1 \}$

Para caracterizar a relação (arco) entre dois nodos, a estratégia é calcular o ângulo ϕ formado pela linha que une os pontos médios das retas que caracterizam os nodos ([SHA92]). Em função do ângulo ϕ (em relação ao eixo das abcissas), define-se que um nodo "está acima de", ou "está abaixo de", ou "está à esquerda de", ou ainda "está à direita de".

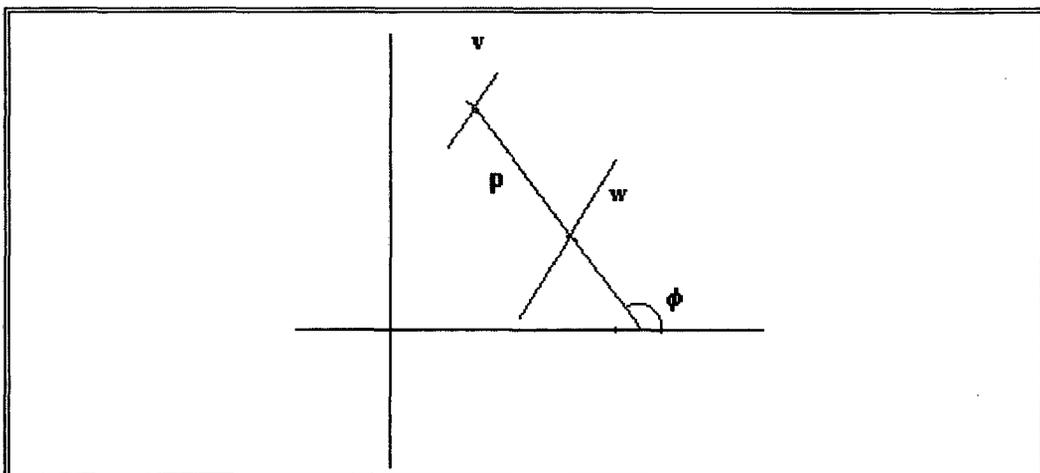


Fig. 4.16 - Determinação do ângulo ϕ caracterizador da relação (arco p) entre os nodos v e w

4.10 - *Template Matching* Baseado na Distância para AFT

Operação efetuada na etapa final da fase de reconhecimento. Tendo o estilete do nematóide a ser identificado devidamente caracterizado por um grafo AFT, então efetua-se uma comparação exaustiva entre esta estrutura e as estruturas de grafos AFT dos *templates* que estão armazenados na base de conhecimento. Efetua-se o cálculo da distância entre dois AFT's (ver Anexo II), escolhendo-se o menor valor gerado por esta comparação.

Ressalta-se porém que em virtude do fato do modelo AFT ser sensível à rotação, a estratégia adequada seria efetuar os cálculos de distância D entre dois padrões na posição original em que se encontram, e também efetuar rotações (arbitrárias) e translações em torno dos eixos dos X e Y sobre o padrão a ser reconhecido. Contudo, se as imagens que geraram os polígonos de contorno/grafos AFT estiverem retificadas, então a rotação e translação sobre o padrão tornam-se desnecessárias. O modelo considera portanto que já houve processo de retificação ([CAB96]) das imagens.

Admite-se então que o nematóide a ser identificado tem as características daquele que está na base de conhecimento e seu grafo AFT apresentou a menor distância.

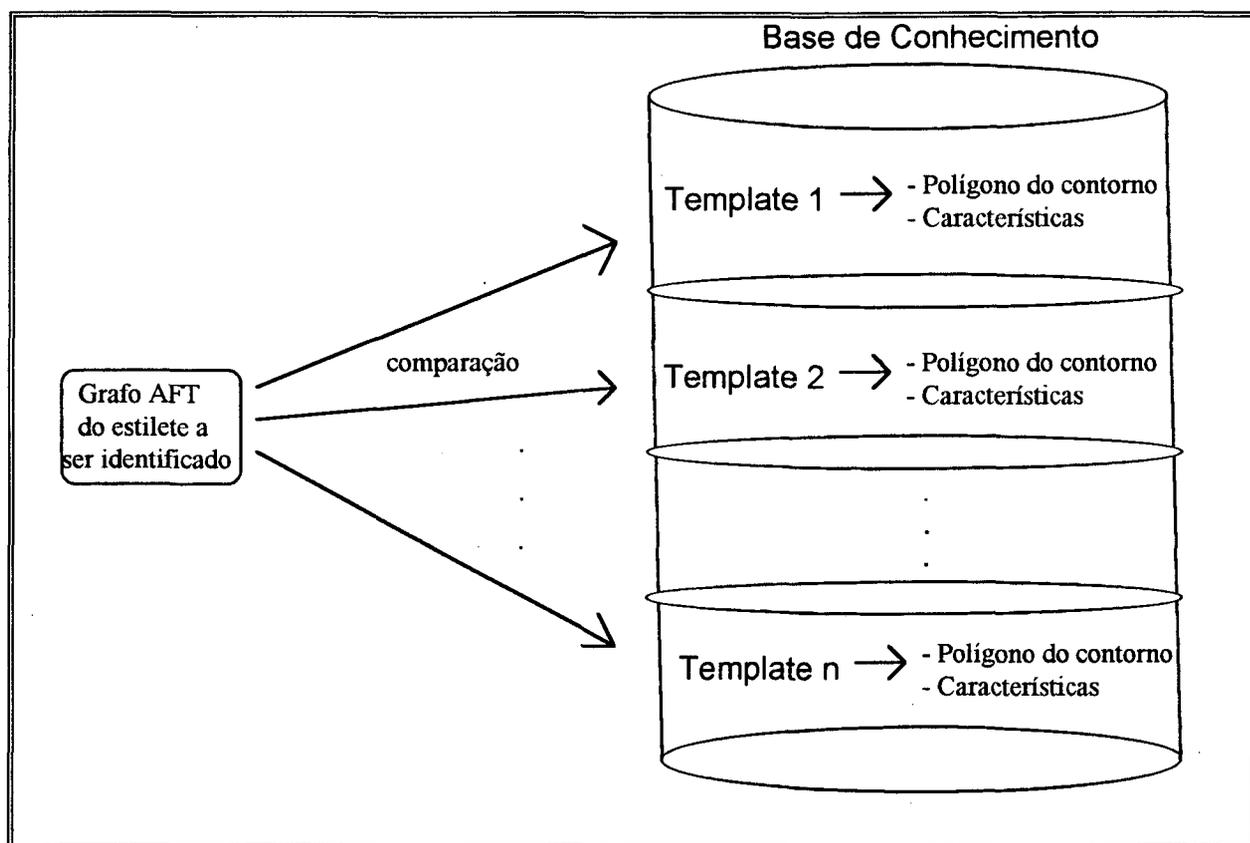


Fig. 4.17 - Esquema para o *template matching*

O grafo AFT apresenta 4 cálculos-chave para se chegar à distância D caracterizadora da similaridade entre dois grafos AFT's:

- Distância entre 2 nodos (D_n) e entre 2 arcos (D_a)
- Distância entre 2 complexos (D_c)
- Distância entre 2 unidades (D_u)
- Distância entre 2 AFT's (D).

Para o presente trabalho foram utilizados os seguintes operadores, apresentados na tabela abaixo, para calculo da distância D entre 2 grafos AFT's.

Sejam T_1 e T_2 dois grafos AFT's onde

- $T_1 = (V_1, A_1, B, E, X_1, C_1)$,
- $T_2 = (V_2, A_2, B, E, X_2, C_2)$,
- $U(v,p,w) \in T_1$,
- $U(u,q,z) \in T_2$

então

		Natureza do operador
a)	$D_n(v,u) = d_n [X_1(v), X_2(u)]$	$d_n =$ Distância de Hamming
	$D_a(p,q) = d_a [C_1(p), C_2(q)]$	$d_a =$ Distância de Hamming
b)	$D_c [C(v,p,w), C(u,q,z)] = f_c [D_n(v,u), D_a(p,q), D_n(w,z)]$	$f_c =$ média aritmética de $D_n(v,u), D_a(p,q), D_n(w,z)$
c)	$D_u [U(v,p,w), U(u,q,z)] = f_u [D_c [C(v,p,w), C(u,q,z)], D_c [C(v,p,w), C(z,q',u)], D_c [C(w,p',v), C(u,q,z)], D_c [C(w,p',v), C(z,q',u)]]$	$f_u = \min$
d)	$D(T_1, T_2) = f_1 \prod_{p \in A_1} f_2 \prod_{q \in A_2} d_u [U(v,p,w), U(u,q,z)]$	$f_1 = \max$ e $f_2 = \min$

CAPÍTULO 5

Implementação dos Algoritmos

5.1 - Introdução

O modelo proposto para identificação do nematóide através da estrutura do estilete apresenta, em termos de algoritmos e estruturas de dados, três algoritmos-chave:

- 1) Algoritmo de segmentação de imagem
- 2) Algoritmo para aproximação poligonal de contornos
- 3) Algoritmo para cálculo da distância entre 2 grafos AFT

Este capítulo apresenta o pseudo-código e as estruturas de dados mais importantes para implementação desses algoritmos. Em seguida, apresenta os resultados e a discussão de um teste efetuado com o programa protótipo.

5.2 - Algoritmo para Segmentação de Imagem ([HOR76])

A imagem a ser tratada deve ter dimensão de $N \times N$ píxeis onde $N = 2^k$, $k > 1$. Se a imagem não tiver estas especificações, o programa, ao tratá-la, fará um "arredondamento" da imagem. Assim, supondo que a imagem tenha largura de 101 píxeis e altura de 143 píxeis, o algoritmo reserva uma área de memória para uma matriz de píxeis de 256×256 ($256 = 2^8$, a menor potência de 2 que "engloba" 143×101 píxeis).

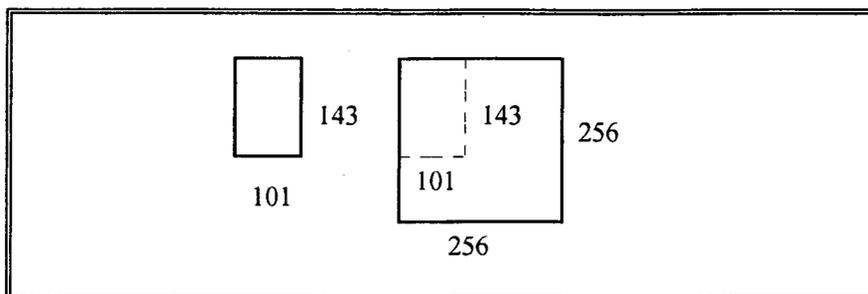


Fig. 5.1 - Imagem de 143 x 101 píxeis. O algoritmo de segmentação aloca memória para uma imagem de lado $N = 2^8 = 256$ píxeis

Formam-se *blocos* (quadrados) em vários *níveis*, cada nível identificado pela largura do bloco, caracterizados pela quintupla $\langle x, y, z, M, m \rangle$

Bloco_{*i*} = $\langle x_i, y_i, z_i, M_i, m_i \rangle$, onde

i = *i*-ésimo bloco da figura

x_i = Coluna do píxel superior esquerdo do bloco

y_i = Linha do píxel superior esquerdo do bloco

z_i = Largura em píxeis do bloco

M_i = Valor da intensidade máxima entre os píxeis do bloco

m_i = Valor da intensidade mínima entre os píxeis do bloco

Para uma imagem de 256 x 256 píxeis tem-se 8 níveis.

Nível L	Quantidade de blocos	Dimensão de cada bloco
0	1	256 x 256 píxeis
1	4	128 x 128 píxeis
2	16	64 x 64 píxeis
3	64	32 x 32 píxeis
4	256	16 x 16 píxeis
5	1024	8 x 8 píxeis
6	4096	4 x 4 píxeis
7	16384	2 x 2 píxeis
8	65536	1 x 1 píxel (píxel da figura)

Designando-se cada nível por L tem-se que:

$$0 \leq L \leq l_n, \quad \text{onde } l_n = \log_2 N$$

Supondo uma imagem de largura $N = 256$ então

$$l_n = \log_2 256 = 8 \quad \text{e} \quad 0 \leq L \leq 8$$

A primeira estrutura de dados a ser utilizada pelo algoritmo *split-and-merge* é o *bloco*, que como foi dito anteriormente é uma quintupla $\langle x_i, y_i, z_i, M_i, m_i \rangle$. Para esta mesma imagem de largura $N = 256$, haveriam 65536 blocos

Imagem de Largura $N = 256$					
(Situação dos blocos supondo início do processamento no Nível 1)					
Bloco Nº	Y	X	Z	M	m
0	0	0	128	r0	s0
1	0	128	128	r1	s1
2	128	0	128	r2	s2
3	128	128	128	r3	s3
4	-	-	-	-	-
.					
.					
.					
65.535	-	-	-	-	-

Fig. 5.2 - Estrutura de dados do *bloco* para o algoritmo: 65.536 blocos para uma imagem de 256 x 256 pixels. Alocação de 5 vetores para a quintupla $\langle x_i, y_i, z_i, M_i, m_i \rangle$. A figura mostra os blocos 0, 1, 2 e 3, os únicos considerados no nível 1.

A imagem passa a ser vista como um conjunto de *blocos*. A referência (índice de vetor) ao bloco dependerá do nível L escolhido para se iniciar o algoritmo *split-and-merge*.

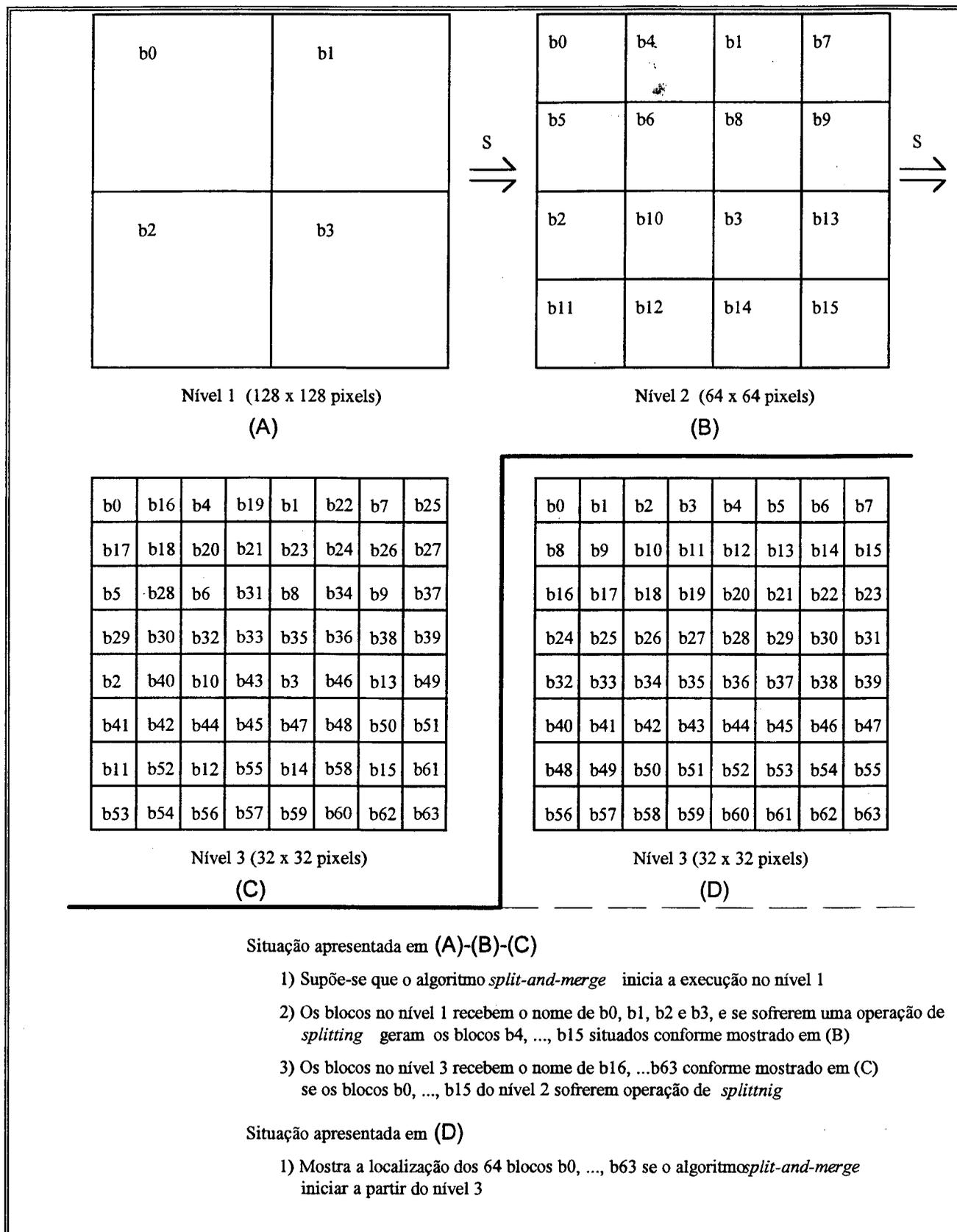


Fig. 5.3 - Uma imagem de 256 x 256 pixels e os blocos no início do algoritmo no nível 1 (A) e no nível 3 (D)

Outro conceito importante para este algoritmo é o de *sucessor*. Cada bloco B_i de um determinado nível L "gera" 4 sucessores no nível seguinte. Seja $B_i = \langle x_i, y_i, z_i, M_i, m_i \rangle$ no nível L_α . Então no nível seguinte $L_{\alpha+1}$, os 4 sucessores B_i de serão:

- Sucessor 1: $x = x_i, y = y_i, z = z_i / 2, M = r1, m = s1,$
 Sucessor 2: $x = x_i + z_i / 2, y = y_i, z = z_i / 2, M = r2, m = s2,$
 Sucessor 3: $x = x_i, y = y_i + z_i / 2, z = z_i / 2, M = r3, m = s3,$
 Sucessor 4: $x = x_i + z_i / 2, y = y_i + z_i / 2, z = z_i / 2, M = r4, m = s4,$

Exemplificando: na figura 5.3-(A), onde se considera que o algoritmo inicia no nível 1, e os quatro blocos b_0, b_1, b_2, b_3 sofreram a operação de *splitting*, então

o bloco b_0 no nível 1 possui 4 sucessores naturais no nível 2:

- Sucessor 1: bloco b_0 , $x = 0, y = 0, z = 64, M = r1', m = s1',$
 Sucessor 2: bloco b_4 , $x = 64, y = 0, z = 64, M = r2', m = s2',$
 Sucessor 3: bloco b_5 , $x = 0, y = 64, z = 64, M = r3', m = s3',$
 Sucessor 4: bloco b_6 , $x = 64, y = 64, z = 64, M = r4', m = s4',$

o bloco b_3 no nível 2 possui 4 sucessores naturais no nível 3:

- Sucessor 1: bloco b_3 , $x = 128, y = 128, z = 32, M = r1'', m = s1'',$
 Sucessor 2: bloco b_{46} , $x = 160, y = 128, z = 32, M = r2'', m = s2'',$
 Sucessor 3: bloco b_{47} , $x = 128, y = 160, z = 32, M = r3'', m = s3'',$
 Sucessor 4: bloco b_{48} , $x = 160, y = 160, z = 32, M = r4'', m = s4'',$

O valor máximo e mínimo entre os pixels de um determinado bloco, qualquer que seja o seu nível L , depende de como o pixel da tela é representado em memória. Um pixel da tela pode estar representado em memória por:

- 1 bit,
- 1 *nibble* (metade de um byte = 4 bits),
- 1 byte (= 8 bits), ou
- 3 bytes.

Quando se representa o píxel da tela por um único bit, então somente 2 cores são possíveis: preto (bit 0) ou branco (bit 1). Se o píxel for representado por um nibble, então são possíveis 16 cores ou intensidades de cinza. Se um byte representa o píxel, então pode-se ter 256 cores ou intensidades de cinza. Se se usar 3 bytes (padrão RGB) então pode-se ter 16.777.216 cores. O programa implementado permite gerenciar representação de píxeis de 1 bit, 1 *nibble* ou 1 byte.

Assim, se o píxel da imagem estiver representado por 1 bit em memória, o valor da intensidade do píxel, para o programa, será ou 0 (preto) ou 1 (branco). Se o píxel da imagem está representado por um nibble o valor da intensidade varia entre 0 e 15. E se um byte em memória representa o píxel, a escala de intensidade do píxel varia entre 0 e 255.

Portanto, quando se considera um bloco B_i qualquer, em qualquer nível, o valor de sua intensidade máxima M_i e intensidade mínima m_i dependerá da forma de representação do píxel.

A idéia do algoritmo é iniciar num nível intermediário e tentar efetuar a operação de *merging*, ou seja, um agrupamento entre blocos. Pode-se iniciar no nível mais alto ou no nível mais baixo, porém a eficiência do algoritmo ficará comprometida. Assim o primeiro passo é determinar o nível intermediário.

Na implementação deste programa tomou-se o nível do meio. Assim, para uma imagem de 256 x 256 píxeis, com 8 níveis, escolhe-se os blocos do nível 4 para iniciar o *merging*. Uma vez que se tenha escolhido o nível de início do programa deve-se inicializar a estrutura de dados de bloco para o nível inicial. O algoritmo de inicialização é o seguinte:

```

N = Largura da imagem;
Estrutura BLOCO = { Arrays X, Y, Z, M, m (dimensão D = N2 elementos) }
L0 = Nível inicial
S0 = N / 2L0; (lado do bloco)
k = 0;
Para I = 1 Até N Passo S0
  Para J = 1 Até N Passo S0
    X(k) = I; Y(k) = J; Z(k) = S0;
    M(k) = MAX(PIXELab , a = X(k),...,X(k)+S0, b = Y(k),...,Y(k)+S0)
    m(k) = MIN(PIXELab , a = X(k),...,X(k)+S0, b = Y(k),...,Y(k)+S0)
    k = k + 1;
  Fim-Para
Fim-Para

```

Tendo inicializado os blocos do nível inicial L_α , começa a operação de *merging*. Em essência, a operação de *merging* tentará agrupar blocos do nível L_α entre si, de modo a "gerar" blocos de nível inferior $L_{\alpha-1}$. Depois tenta-se agrupar blocos do nível $L_{\alpha-1}$ de modo a "gerar" blocos do nível superior $L_{\alpha-2}$, e assim sucessivamente, até a etapa de geração de bloco do nível 0 l_0 .

O critério para o agrupamento na operação de *merging* é o seguinte:

- Agrupam-se apenas *blocos* adjacentes.
- Como o agrupamento deve "gerar" um *bloco* (quadrado) num nível inferior, deve-se então tentar agrupar 4 *blocos* adjacentes.
- Quatro blocos B_1 , B_2 , B_3 e B_4 adjacentes, são agrupados se a diferença entre a maior intensidade M entre os quatro blocos e a menor intensidade m entre os mesmos quatro blocos for menor que uma determinada *tolerância*.

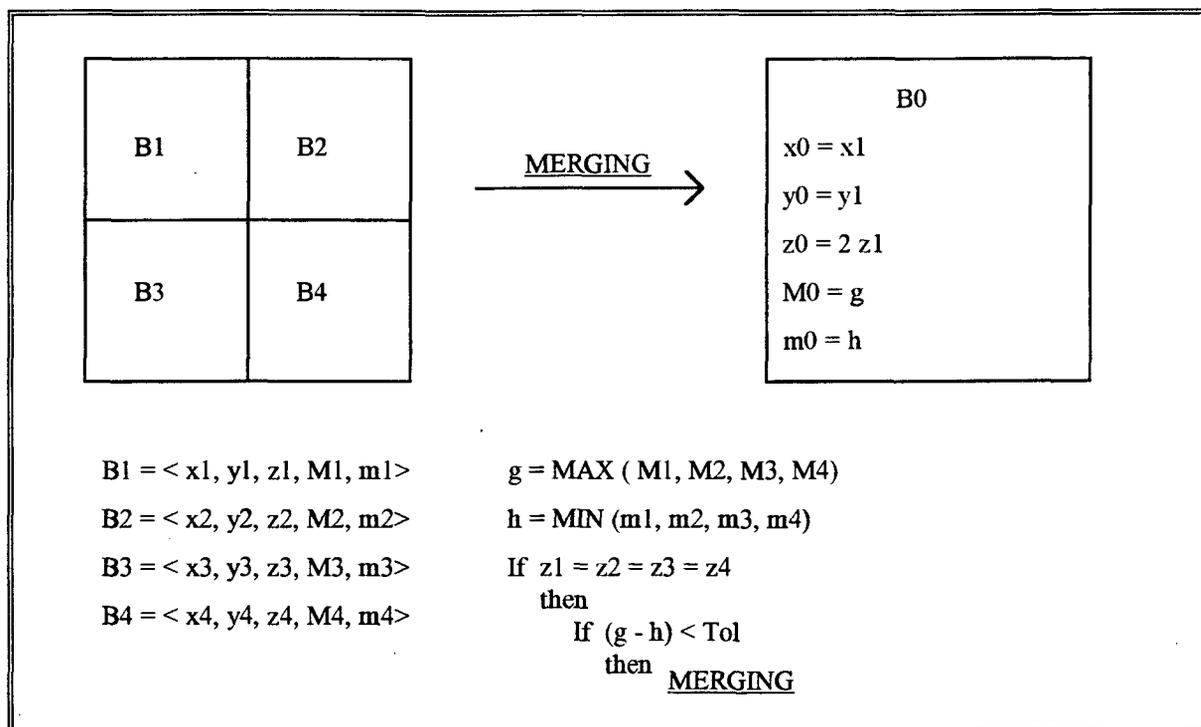


Fig. 5.4 - O processo de segmentação - *Merging*

Na figura 5.5 é apresentado um exemplo de *merging*. Supõe-se que a imagem seja de 256 x 256 píxeis e, o algoritmo *split-and-merge* inicia no nível 2, onde os blocos apresentam tamanho $z = 64$.

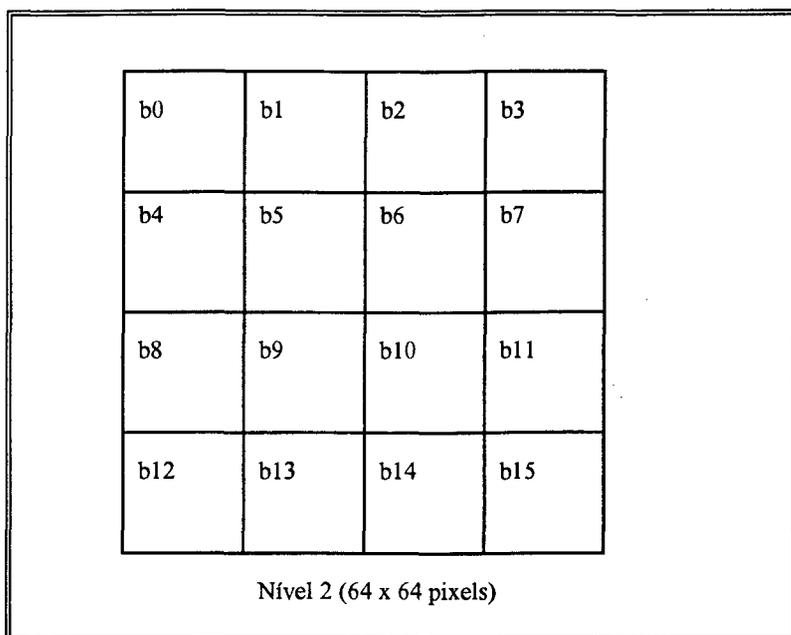


Fig. 5.5 *Merging* para o nível 2 da figura de 256 x 256 pixels

No caso, a operação de *merging* inicialmente tentará agrupar os seguintes blocos:

(b0, b1, b4, b5), (b1, b2, b5, b6), (b2, b3, b6, b7),
 (b4, b5, b8, b9), (b5, b6, b9, b10), (b6, b7, b10, b11),
 (b8, b9, b12, b13), (b9, b10, b13, b14) e (b10, b11, b14, b15).

O algoritmo proposto por Horowitz e Pavlidis ([HOR76]) para a operação de *merging* é o seguinte:

E = Tolerância para *merging*

Estrutura BLOCO = { Arrays X, Y, Z, M, m (dimensão $D = N^2$ elementos) }

L0 = Nível inicial

S0 = $N / 2^{L0}$; (lado do bloco)

L = L0; (nível inicial - Desejável um nível intermediário)

S = S0

Enquanto L > 0 Faça

X(k0) = 0; Y(k0) = 0;

$k1 = 2^{L0} \cdot [X(k0) + (2^{L0} * Y(k0))] / N$;

$k2 = k1 + 2^{L0-L}$;

$k3 = k1 + 2^{2L0-L}$;

$$k_4 = k_3 + 2^{L_0 - L};$$

Para todo $[B_{k_1} \in \text{nível } L]$ Faça

Se $Z(k_1) = Z(k_2) = Z(k_3) = Z(k_4) = S$

Então

$$G = \text{MAX}(M(k_1), M(k_2), M(k_3), M(k_4));$$

$$H = \text{MIN}(m(k_1), m(k_2), m(k_3), m(k_4));$$

Se $(G - H) \leq (2 * E)$

Então

$$Z(k_2) = 0; Z(k_3) = 0; Z(k_4) = 0;$$

$$Z(k_1) = 2 * S; M(k_1) = G; m(k_1) = H;$$

Fim-Se;

Fim-Se;

Fim-Para;

$$L = L - 1;$$

$$S = 2 * S;$$

Fim-Enquanto;

Horowitz e Pavlidis comentando a respeito do valor da tolerância E informam que em suas experiências, obtiveram resultados satisfatórios quando utilizaram uma tolerância $E = 16$ em imagens que apresentavam 128 tonalidades de cinza. Informam ainda que para outras faixas de intensidade de pixel, uma escala linear é adequada ([HOR76]). Assim, segundo Pavlidis

$$E = [\text{Quantidade de tonalidades cinza} * 16] / 128 .$$

No programa em questão, foi dito que os pixels das imagens são representados em 2, 16 ou 256 tonalidades de pixels. Em vista deste fato, tem-se as seguintes tolerâncias E :

2 tonalidades:	$E = 0$
16 tonalidades:	$E = 2$
256 tonalidades:	$E = 32$

Exemplificando: se um pixel apresenta valor de intensidade 187 numa imagem de 256 tonalidades, então, devido à tolerância $E = 32$, tem-se que para efeitos de cor, as intensidades entre 155 e 219 ($= 187 \pm 32$) serão consideradas homogêneas.

Supondo na figura 5.5 que cada píxel seja representado em memória por um byte e que as intensidades máxima e mínima para os blocos b0, b1, b4 e b5 que tentam se agrupar sejam:

$$\begin{aligned} b0 &= \langle 0, 0, 64, 187, 14 \rangle \\ b1 &= \langle 64, 0, 64, 236, 165 \rangle \\ b4 &= \langle 0, 64, 64, 202, 71 \rangle \\ b5 &= \langle 64, 64, 64, 192, 0 \rangle \end{aligned}$$

Então, tem-se que $E = 32$, e durante o algoritmo de *merging* ocorre que

$$G = \text{MAX}(187, 236, 202, 192) = 236$$

$$H = \text{MIN}(14, 165, 71, 0) = 0$$

Se $[(G - H) = 236] \leq [(2 * E) = 64]$ então AGRUPAR.

Como a condição $[(G - H) = 236] \leq [(2 * E) = 64]$ é falsa, no caso não ocorre *merging*.

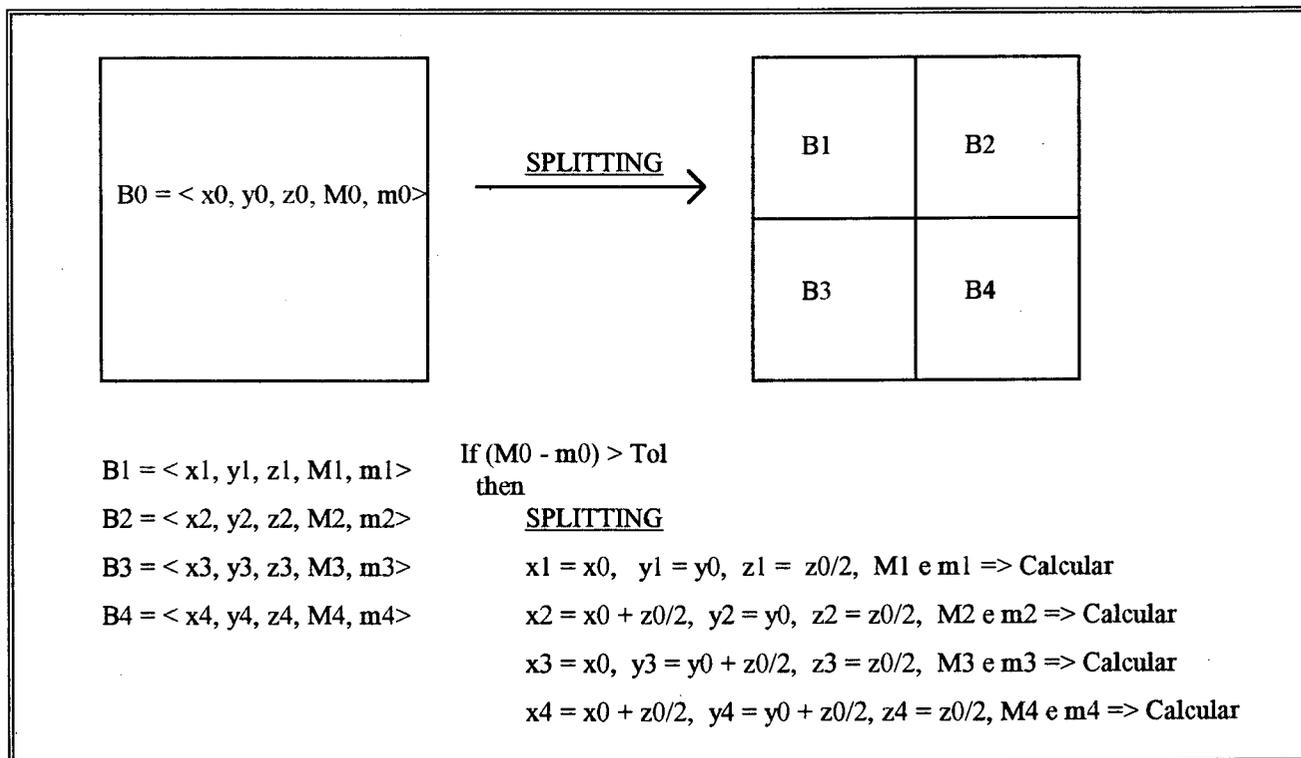


Fig. 5.6 - O processo de segmentação - *Splitting*

Após o *merging* efetua-se a operação denominada *splitting*, que em síntese significa dividir um bloco, originando seus sucessores, pois o bloco apresenta a diferença "interna" ($M - m$) em questão maior que a tolerância E já utilizada na operação de *merging*.

O algoritmo para a operação de *splitting* seria o seguinte:

E = Tolerância para *splitting* (= *merging*)

L_0 = Nível inicial

$S_0 = N / 2^{L_0}$; (lado do bloco)

L_N = Último nível da imagem

Estrutura BLOCO = { Arrays X, Y, Z, M, m (dimensão $D = N^2$ elementos) }

$C = 4^{L_0}$ (número de blocos no nível inicial)

Para $k_1 = 1$ Até C

Se $0 < Z(k_1) \leq S_0$

Então

$L = L_0$; $S = S_0$;

Enquanto $L < L_N$ Faça

Se $(M(k_1) - m(k_1)) > (2 * E)$

Então

$k_2 = C + 1$; $k_3 = C + 2$; $k_4 = C + 3$;

$Z(k_1) = S / 2$; $Z(k_2) = Z(k_1)$; $Z(k_3) = Z(k_1)$;

$X(k_2) = X(k_1) + Z(k_1)$; $Y(k_2) = Y(k_1)$;

$X(k_3) = X(k_1)$; $Y(k_3) = Y(k_1) + Z(k_1)$;

$X(k_4) = X(k_1) + Z(k_1)$; $Y(k_4) = Y(k_1) + Z(k_1)$;

$M(k_1) = \text{MAX}(\text{PIXELab}, a = X(k_1), \dots, X(k_1) + Z(k_1), b = Y(k_1), \dots, Y(k_1) + Z(k_1))$

$M(k_2) = \text{MAX}(\text{PIXELab}, a = X(k_2), \dots, X(k_2) + Z(k_1), b = Y(k_2), \dots, Y(k_2) + Z(k_1))$

$M(k_3) = \text{MAX}(\text{PIXELab}, a = X(k_3), \dots, X(k_3) + Z(k_1), b = Y(k_3), \dots, Y(k_3) + Z(k_1))$

$M(k_4) = \text{MAX}(\text{PIXELab}, a = X(k_4), \dots, X(k_4) + Z(k_1), b = Y(k_4), \dots, Y(k_4) + Z(k_1))$

$m(k_1) = \text{MIN}(\text{PIXELab}, a = X(k_1), \dots, X(k_1) + Z(k_1), b = Y(k_1), \dots, Y(k_1) + Z(k_1))$

$m(k_2) = \text{MIN}(\text{PIXELab}, a = X(k_2), \dots, X(k_2) + Z(k_1), b = Y(k_2), \dots, Y(k_2) + Z(k_1))$

$m(k_3) = \text{MIN}(\text{PIXELab}, a = X(k_3), \dots, X(k_3) + Z(k_1), b = Y(k_3), \dots, Y(k_3) + Z(k_1))$

$m(k_4) = \text{MIN}(\text{PIXELab}, a = X(k_4), \dots, X(k_4) + Z(k_1), b = Y(k_4), \dots, Y(k_4) + Z(k_1))$

Senão

$L = L_N$;

Fim-Se;

$$L = L + 1; S = S/2; C = C + 3;$$

Fim-Enquanto;

Fim-Se;

Fim-Para;

Após as operações de *merging* e *splitting*, efetuam-se as rotinas denominadas *grouping* e *elimination*. O grupamento (*grouping*) consiste basicamente num *merging* que agrupará regiões de adjacência que atendam ao critério de tolerância estabelecido para o *merging*. Agora as regiões (e não mais blocos) são agrupados independentemente de suas larguras, o que não acontecia durante o *merging* onde só se agrupavam blocos de igual tamanho.

Para se efetuar *grouping*, é necessária uma estrutura de dados de matriz. E a matriz de adjacência de blocos em última instância é uma inversão da relação entre bloco e píxel. A matriz informa a que bloco pertence cada píxel da imagem. Admitindo-se a imagem de 256 x 256 píxeis, então tem-se uma matriz de 256 x 256 posições. Para inicializar a matriz de adjacência A procede-se:

C = quantidade total de blocos após *splitting*

Matriz A (dimensão N x N)

Estrutura BLOCO = { Arrays X, Y, Z, M, m (dimensão D = N² elementos) }

Para K = 1 Até C tal que Z(k) ≠ 0

Para I = X(k) Até X(k) + Z(k)

Para J = Y(k) Até Y(k) + Z(k)

A(I,J) = K;

Fim-Para;

Fim-Para;

Fim-Para;

Além da matriz A, o algoritmo para *grouping* exige o uso de uma estrutura de lista S, que será operada na forma de pilha. Um bloco, durante o *grouping*, apresentará um dos seguintes estados:

- Não marcado e não processado (Z(k) > 0)
- Marcado e não processado (Z(k) < 0 e B_k está na pilha S)
- Marcado e já processado (Z(k) < 0 e B_k não está na pilha S)

A terceira estrutura de dados para o algoritmo, é uma estrutura de lista R que representa a lista de *regiões* da imagem. Uma *região* é um conjunto de blocos B_k, que atendem ao critério de tolerância

E de intensidade de píxeis. A diferença em relação à operação de *merging* é que na região serão agrupados blocos de diversos tamanhos. É através da região que se pode "visualizar" formatos "deformados", pois poderão estar juntos, numa mesma região, blocos de tamanho diversos tais como 64, 32, 16, 1, etc. Também ocorre um acréscimo na estrutura de dados de BLOCO. Acrescenta-se, além da quintupla $\langle X, Y, Z, M, m \rangle$, a variável RB, para indicar a que região o bloco pertence. Segue o algoritmo para *grouping*.

```

E = Tolerância para splitting (= merging)
N = 0; (quantidade de regiões)
Matriz A (dimensão N x N)
Estrutura BLOCO = { Arrays X, Y, Z, M, m (dimensão D = N2 elementos) Variável RB }
S = Pilha vazia;
Estrutura REG = { SIZE, SUM, LISTA-DE-BLOCOS }
Para k = 1 Até C (k = Número do bloco)
  Se Z(k) > 0
    Então
      Z(k) = - Z(k);
      "Empilhar" k em S;
      U = M(k); V = m(k); N = N + 1;
      REG(N).SIZE = 0; (qtde. píxeis na região de índice N)
      REG(N).SUM = 0; (soma das intensidades dos píxeis)
      Enquanto S = Pilha-NÃO-Vazia Faça
        BUF-BLOCO = "Desempilhar" k de S;
        REG(N).LISTA-DE_BLOCOS = Anexar BUF-BLOCO;
        REG(N).SIZE = REG(N).SIZE + [Z(k) * Z(k)];
        Para α = X(k) Até [X(k) - Z(k) - 1]
          Para β = Y(k) Até [Y(k) - Z(k) - 1]
            REG(N).SUM = REG(N).SUM + PIXEL(α,β);
          Fim-Para;
        Fim-Para;
      Para todo [BLOCO-J ∈ A] tal que [BLOCO-J @ BUF-BLOCO] (@ = Adjacente)
        G = MAX (u, Mj); H = MIN(v, mj);
        Se (G - H) ≤ (2*E)
          Então

```

"Empilhar" J em S;

$U = G; \quad V = H;$

$Z(J) = -Z(J);$

Fim-Se;

Fim-Para;

Fim-Enquanto;

Fim-Se;

Fim-Para;

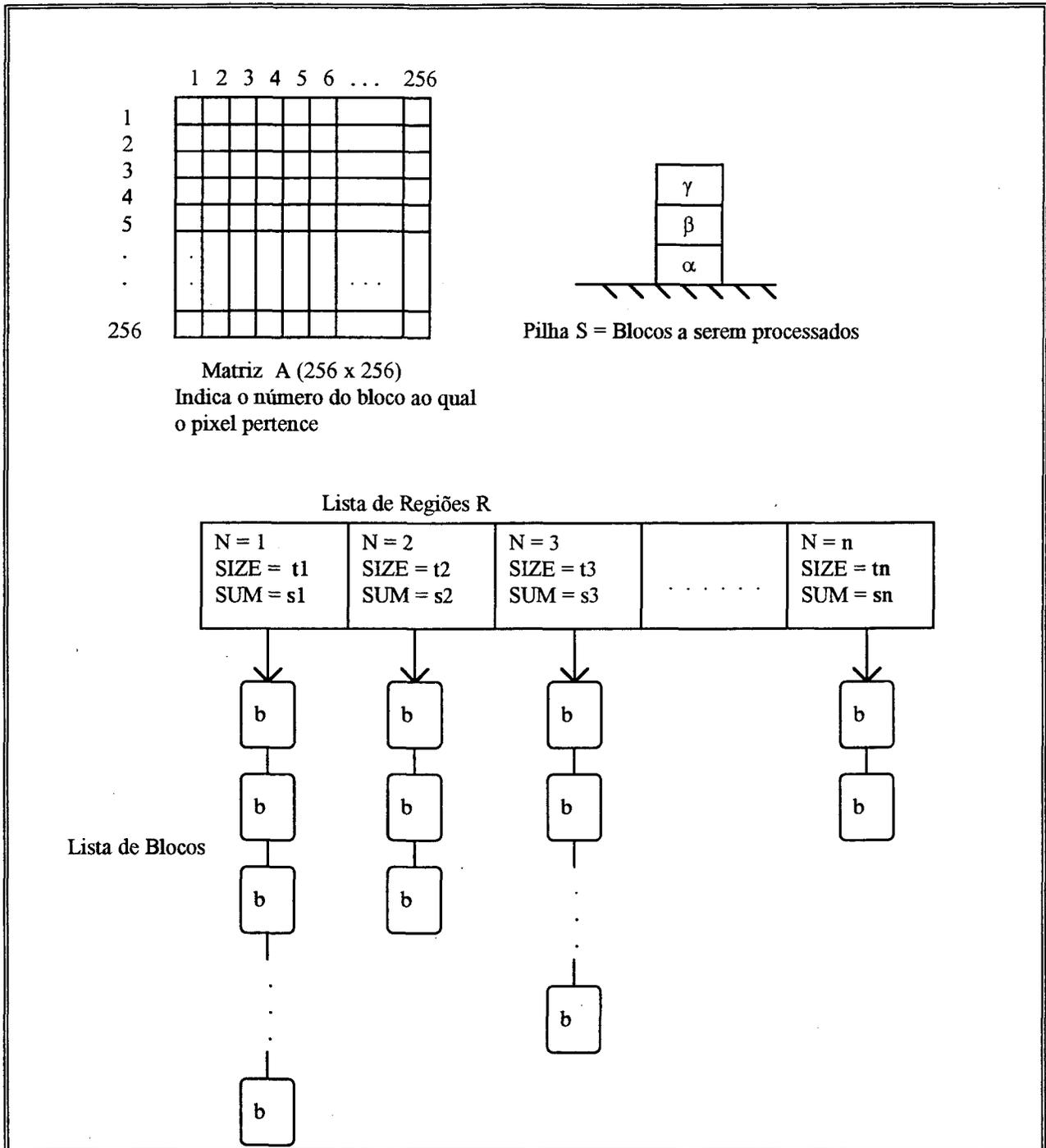


Fig. 5.7 - Estruturas de dados utilizadas no algoritmo de *grouping*

Após a operação de *grouping*, obtém-se as regiões cujos píxeis atendem à tolerância E designada nos algoritmos de *merging* e *splitting*. Porém, existem "gaps" entre algumas regiões. Estes "gaps" correspondem a estreitas regiões que são, em última instância, regiões de transição entre duas "grandes" regiões "bem definidas" (no sentido de apresentarem conteúdo semântico relevante).

Segundo Horowitz e Pavlidis, estes "gaps", na prática, são de tamanho bem pequeno, frequentemente sendo um simples píxel, podendo ser desprezados.

Em vista disso uma rotina de eliminação dos "gaps" é executada. A eliminação consiste em anexar os "gaps" na região adjacente que "melhor" o acomodar. Utiliza-se como critério para escolha da região hospedeira, a sua intensidade de brilho média e um valor máximo de expansão permitido para cada região. Assim, supondo que uma região de "gap" G deva ser eliminada, sendo anexada ou à região α ou à região β (ambas adjacentes em relação a G) e admitindo-se que:

- Píxel da imagem apresenta 256 tonalidades
- Região "gap" G = { Intensidade média = 45.3, Tamanho em píxeis = 33 }
- Região α = { Intensidade média = 54.5, Tamanho em píxeis = 2463 = 2055 originais + 408 de outros "gaps" anteriormente acrescentados }
- Região β = { Intensidade média = 59.7, Tamanho em píxeis = 2102 = 1870 originais + 232 de outros "gaps" anteriormente acrescentados }
- Máximo de expansão para uma região = 20% do tamanho original
- Diferença máxima permitida entre as intensidades médias de brilho = 16

A região α permite um máximo de expansão de 411 píxeis (20% de 2055). Apesar de a diferença de intensidade média de brilho em relação ao "gap" G ser menor que a tolerância de 16 ($59.7 - 45.3 = 9.2 < 16$), α não anexará a região G, pois ultrapassaria sua cota de 20 % em termos de quantidade de píxeis (pode ir até $2055 * 1.2 = 2466$ píxeis, já tem 2463, e teria que anexar mais 33 píxeis de G).

Já a região β permite um máximo de expansão de 374 píxeis (20% de 1870). Sua diferença de intensidade média de brilho em relação ao "gap" G é menor que a tolerância de 16 ($59.7 - 45.3 = 14.4 < 16$), e sua cota de expansão permite anexar até 374 píxeis, tendo já anexado 232 píxeis. Ao se anexarem os 33 píxeis do "gap" G a cota ainda assim não seria ultrapassada.

No exemplo, portanto, a eliminação realizaria a anexação de G pela região β . Abaixo é apresentado o algoritmo de eliminação. Utiliza-se uma estrutura de lista L, que representa as regiões adjacentes candidatas a anexar a região de "gap".

N = Quantidade de regiões da lista de regiões R formada durante o *grouping*

L = Lista Vazia;

E_{size} = Percentual máximo para expansão da região

E_{ave} = Tolerância para a diferença entre as médias de intensidade de brilho entre 2 regiões

Estrutura BLOCO = { Arrays X, Y, Z, M, m (dimensão D = N^2 elementos) Variável RB }

Matriz A (dimensão N x N)

Estrutura-Vetor REG[1..N] = { SIZE, SUM, LISTA-DE-BLOCOS }

Para I = 1 Até N (I = Número da região)

Se REG(I).SIZE > 0

Então

Para todo [BLOCO-K ∈ REG(I)]

Para todo [BLOCO-J ∈ A] tal que [BLOCO-J @ BLOCO-K] (@ = Adjacente)

Se [BLOCO-J.RB ≠ I] E [BLOCO-J ∉ L]

Então

"Empilhar" BLOCO-J.RB em L;

Fim-Se;

Fim-Para;

Repita

AVG = MIN_{a ∈ L} | REG(a).SUM / REG(a).SIZE - REG(I).SUM / REG(I).SIZE |;

M = IND (MIN_{a ∈ L} | REG(a).SUM / REG(a).SIZE - REG(I).SUM / REG(I).SIZE |);

REG(I).SIZE = REG(I).SIZE + REG(M).SIZE;

REG(I).SUM = REG(I).SUM + REG(M).SUM;

REG(I) = REG(I) || REG(M); (|| - operação de concatenação)

REG(M).SIZE = 0;

Para todo [BLOCO-P ∈ REG(M)]

BLOCO-P.RB = I;

Para todo [BLOCO-Q ∈ A] tal que [BLOCO-Q @ BLOCO-P]

Se [BLOCO-Q.RB ≠ I] E [BLOCO-Q.RB ≠ M] E [BLOCO-Q ∉ L]

Então

"Empilhar" BLOCO-Q.RB em L;

Fim-Se;

Fim-Para;

Fim-Para;

Até [REG(I).SIZE > E_{size}] OU [AVG > E_{avg}];

Fim-Repita;

Fim-Para;

Fim-Se;

Fim-Para;

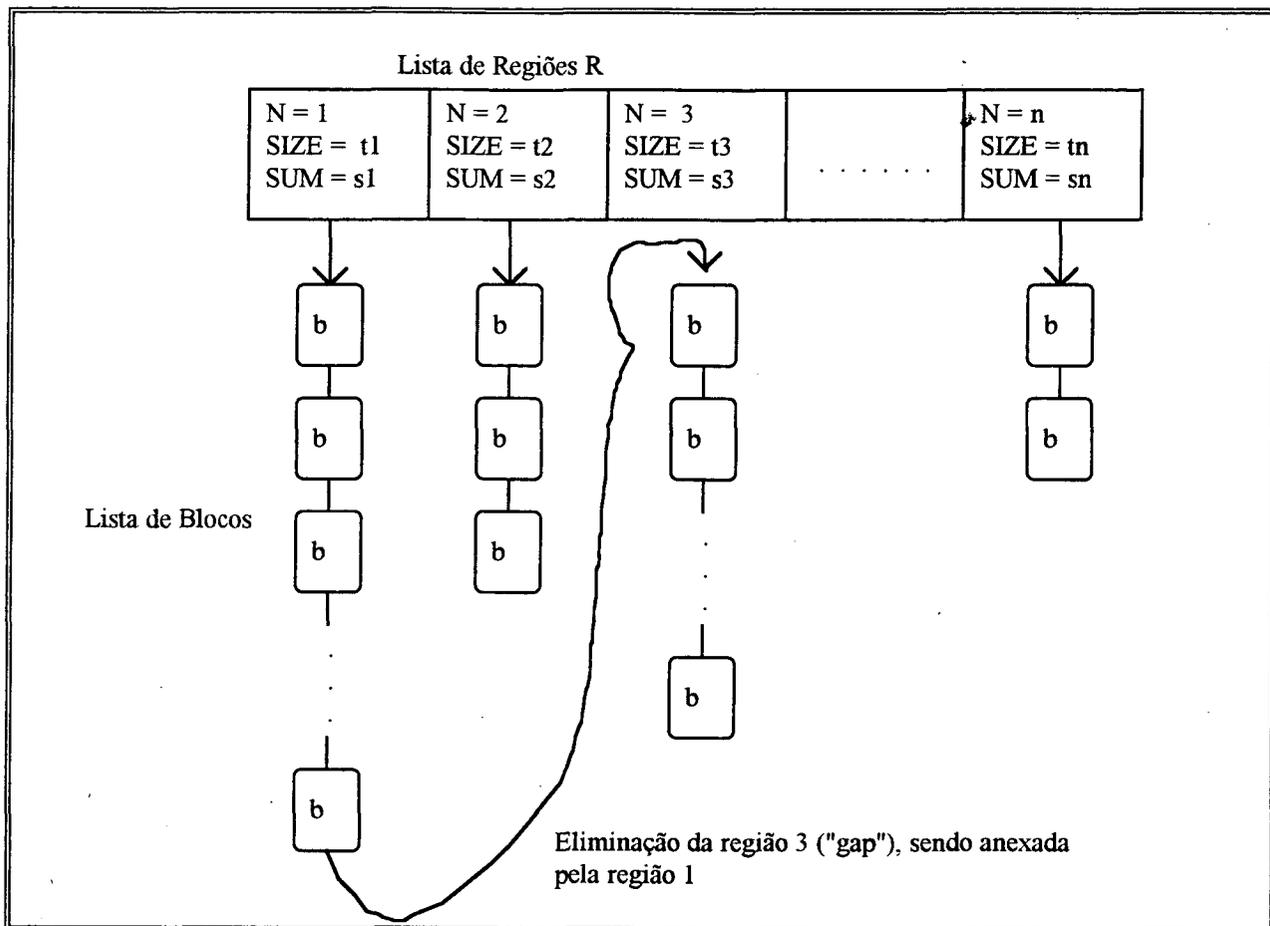


Fig. 5.8 - O processo de eliminação de uma região "gap"

Após o processo de segmentação, deseja-se extrair os contornos das regiões encontradas durante o processo *merging-splitting-grouping-elimination*. Para se proceder à extração dos contornos é necessário que na lista de regiões se especifique o píxel do ponto mais ao alto e mais à direita, ou seja, o píxel do canto superior direito. Portanto à estrutura de dados da lista de regiões REG, adicionam-se as variáveis XD e YD, respectivamente coluna e linha do píxel canto superior direito da região.

A matriz de adjacências A, que relatava a relação de pertinência entre píxel e bloco, agora passará a indicar a relação de pertinência entre píxel e região. Quer dizer, cada entrada na matriz A passa a indicar o número da região a que o píxel pertence. Para se efetuar estas tarefas iniciais procede-se o seguinte algoritmo:

N = Quantidade de regiões da lista de regiões R resultante após a eliminação

Estrutura BLOCO = { Arrays X, Y, Z, M, m (dimensão $D = N \times N$ elementos) Variável RB }

Matriz A (dimensão $N \times N$)

Estrutura-Vetor $REG[1..N] = \{ SIZE, SUM, LISTA-DE-BLOCOS, XD, YD \}$

Para $I = 1$ Até N (I = Número da região)

$XD = 0; YD = 999999;$

Para todo $[BLOCO-K \in REG(I)]$

Para $K = X(k)$ Até $X(k) + Z(k)$

Para $L = Y(k)$ Até $Y(k) + Z(k)$

$A(K,L) = I;$

Se $[YD > K]$ OU $[YD = K \text{ E } XD < L]$

Então

$XD = L; YD = K;$

Fim-Se;

Fim-Para;

Fim-Para;

Fim-Para;

Fim-Para;

Para encontrar os píxeis de contorno da região basta iniciar do ponto (XD, YD) do canto superior direito da região indo em sentido anti-horário e, para checar se o píxel é de contorno, basta conferir apenas 4 píxeis vizinhos do píxel atual. Verificam-se os píxeis a norte, a oeste, ao sul e a leste, para saber quais destes são píxeis de contorno.

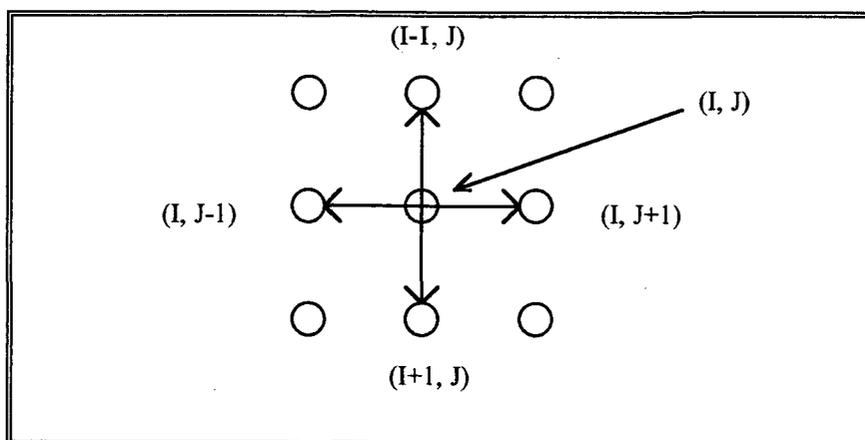


Fig. 5.9 - O píxel (I, J) , pertencente ao contorno, e os píxeis candidatos a serem pontos do contorno

Pode-se desenvolver a rotina de busca de contorno aplicando-se um algoritmo recursivo, ou, para driblar a recursividade, utilizar uma estrutura de pilha "explícita" no corpo do algoritmo. No presente programa, foi utilizada a segunda estratégia.

N = Quantidade de regiões da lista de regiões R resultante após a eliminação

Matriz A (dimensão N x N) (relação entre píxel e região)

Estrutura-Vetor REG[1..N] = { SIZE, SUM, LISTA-DE-BLOCOS, XD, YD }

Para I = 1 Até N (I = Número da região)

P = Pilha Vazia;

Grava (REG(I).XD, REG(I).YD); (Grava o 1º ponto do contorno)

XCOL = REG(I).XD; YLIN = REG(I).YD;

"Empilhar" (XCOL, YLIN);

Enquanto P = Pilha-NÃO-Vazia Faça

"Lê Topo da Pilha" para XT, YT; (não desempilha !!)

Se Já checou todas os sentidos (N, No, O, So, S, Se, L, Ne)

Então

"Desempilhar" (XT, YT) de P;

GoTo L_LerTopo;

Fim-Se;

Se [A(XT, YT-1) = I] E [A(XT, YT-1) = "Não marcado"] (ponto ao norte)

Então

Se { A(XT, YT-2) ou A(XT-1, YT-1) ou A(XT, YT) ou A(XT+1, YT-1) } ≠ I

Então

"Empilhar" (XT, YT-1); // Ponto no sistema cartesiano de coordenadas (X,Y)

"Marcar" A(YT-1, XT); // Matriz A(Linha,Coluna)

"Marcar" Checou Vizinho NORTE de XT, YT;

Grava (XT, YT-1);

GoTo L_LerTopo;

Fim-Se;

Fim-Se;

Se [A(XT-1, YT-1) = I] E [A(XT-1, YT-1) = "Não marcado"] (ponto a noroeste)

Então

Se { A(XT-1, YT-2) ou A(XT-2, YT-1) ou A(XT-1, YT) ou A(XT, YT-1) } ≠ I

Então

"Empilhar" (XT-1, YT-1); // Ponto no sistema cartesiano de coordenadas (X,Y)

"Marcar" A(YT-1, XT-1); // Matriz A(Linha,Coluna)

"Marcar" Checou Vizinho NOROESTE de XT, YT;

Grava (XT-1, YT-1);

GoTo L_LerTopo;

Fim-Se;

Fim-Se;

Se [A(XT-1, YT) = I] E [A(XT-1, YT) = "Não marcado"] (ponto a oeste)

Então

Se { A(XT-1, YT-1) ou A(XT-2, YT) ou A(XT-1, YT+1) ou A(XT, YT) } \neq I

Então

"Empilhar" (XT-1, YT); // Ponto no sistema cartesiano de coordenadas (X,Y)

"Marcar" A(YT, XT-1); // Matriz A(Linha,Coluna)

"Marcar" Checou Vizinho OESTE de XT, YT;

Grava (XT-1, YT);

GoTo L_LerTopo;

Fim-Se;

Fim-Se;

Se [A(XT-1, YT+1) = I] E [A(XT-1, YT+1) = "Não marcado"] (ponto a sudoeste)

Então

Se { A(XT-1, YT) ou A(XT-2, YT+1) ou A(XT-1, YT+2) ou A(XT, YT+1) } \neq I

Então

"Empilhar" (XT-1, YT+1); // Ponto no sistema cartesiano de coordenadas (X,Y)

"Marcar" A(YT+1, XT-1); // Matriz A(Linha,Coluna)

"Marcar" Checou Vizinho SUDOESTE de XT, YT;

Grava (XT-1, YT+1);

GoTo L_LerTopo;

Fim-Se;

Fim-Se;

Se [A(XT, YT+1) = I] E [A(XT, YT+1) = "Não marcado"] (ponto ao sul)

Então

Se { A(XT, YT) ou A(XT-1, YT+1) ou A(XT, YT+2) ou A(XT+1, YT+1) } \neq I

Então

"Empilhar" (XT, YT+1); // Ponto no sistema cartesiano de coordenadas (X,Y)

"Marcar" A(YT+1, XT); // Matriz A(Linha, Coluna)

"Marcar" Checou Vizinho SUL de XT, YT;

Grava (XT, YT+1);

GoTo L_LerTopo;

Fim-Se;Fim-Se;

Se [A(XT+1, YT+1) = I] E [A(XT+1, YT+1) = "Não marcado"] (ponto a sudeste)

Então

Se { A(XT+1, YT) ou A(XT, YT+1) ou A(XT+1, YT+2) ou A(XT+1, YT+2) } \neq I

Então

"Empilhar" (XT+1, YT+1); // Ponto no sistema cartesiano de coordenadas (X,Y)

"Marcar" A(YT+1, XT+1); // Matriz A(Linha, Coluna)

"Marcar" Checou Vizinho SUDESTE de XT, YT;

Grava (XT+1, YT+1);

GoTo L_LerTopo;

Fim-Se;Fim-Se;

Se [A(XT+1, YT) = I] E [A(XT+1, YT) = "Não marcado"] (ponto a leste)

Então

Se { A(XT+1, YT-1) ou A(XT, YT) ou A(XT+1, YT+1) ou A(XT+2, YT) } \neq I

Então

"Empilhar" (XT+1, YT); // Ponto no sistema cartesiano de coordenadas (X,Y)

"Marcar" A(YT, XT+1); // Matriz A(Linha, Coluna)

"Marcar" Checou Vizinho LESTE de XT, YT;

Grava (XT+1, YT);

GoTo L_LerTopo;

Fim-Se;Fim-Se;

Se [A(XT+1, YT-1) = I] E [A(XT+1, YT-1) = "Não marcado"] (ponto a nordeste)

Então

Se { A(XT+1, YT-2) ou A(XT, YT-1) ou A(XT+1, YT) ou A(XT+2, YT-1) } \neq I

Então

"Empilhar" (XT+1, YT-1); // Ponto no sistema cartesiano de coordenadas (X,Y)

"Marcar" A(YT-1, XT+1); // Matriz A(Linha,Coluna)

"Marcar" Checou Vizinho NORDESTE de XT, YT;

Grava (XT+1, YT-1);

GoTo L_LerTopo;

Fim-Se;

Fim-Se;

L_LerTopo :

Fim-Enquanto;

Fim-Para;

A "gravação" do contorno é feita da seguinte forma: tem-se a imagem inicial original que sofreu o processo de segmentação. Durante a execução do algoritmo de extração de contornos, é gerado um arquivo "espelho" do arquivo imagem original. O arquivo "espelho" apresentará as mesmas dimensões, em termos de largura/altura da imagem. Em outras palavras, o arquivo espelho apresentará a mesma quantidade de píxeis que o arquivo original. A diferença é que os píxeis "internos" das regiões apresentam cor branca, enquanto que os píxeis de contorno serão coloridos (o programa utiliza 16 cores). Numa mesma região, os píxeis de contorno terão a mesma cor. As 16 diferentes cores servem para uma melhor distinção visual entre as várias regiões originadas da segmentação.

Ressalta-se ainda que para a "gravação", utiliza-se uma estrutura adequada de lista, que servirá para a etapa de aproximação poligonal do contorno, e por isso, está detalhada a seguir.

5.3 - Algoritmo para Aproximação Poligonal para o Contorno

Para entender esta operação, utiliza-se como exemplo, o padrão mostrado na figura 5.10, no caso, uma letra "a" minúscula escrita à mão, e apresenta-se as etapas para obtenção do polígono representativo de seu contorno.

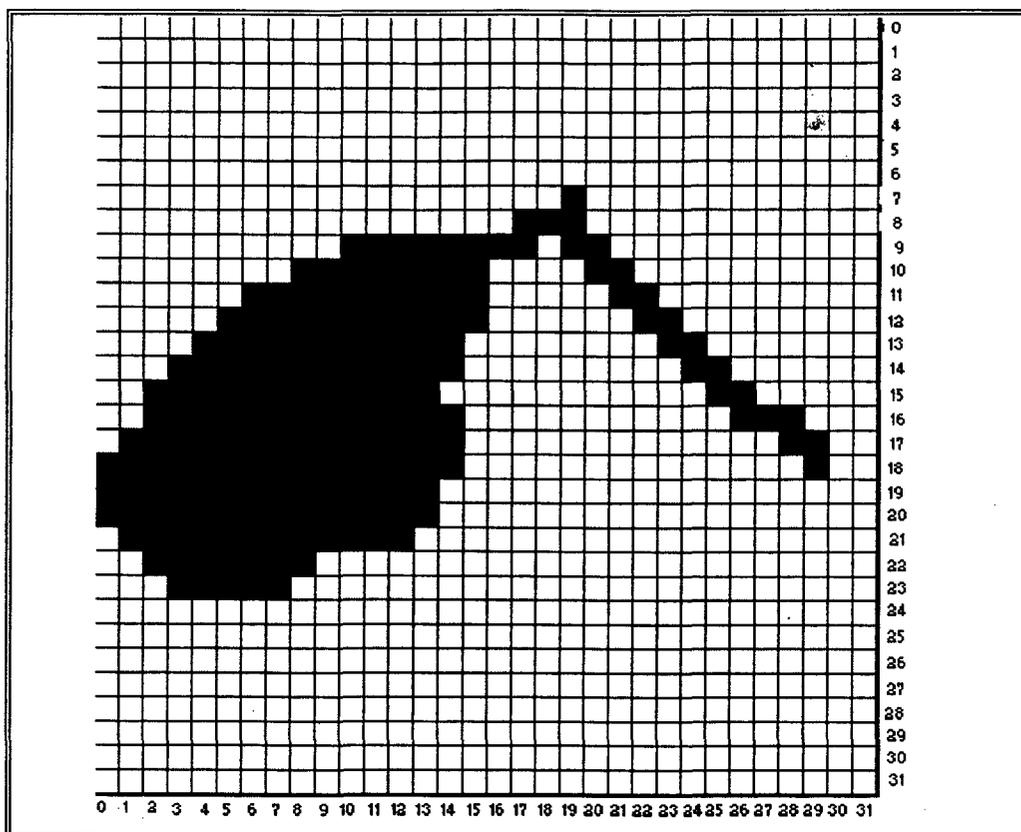


Fig. 5.10 - Um padrão para uma letra "a" minúscula. O algoritmo de segmentação já foi aplicado, e como só existe a letra "a", foi gerada uma única região. O objetivo agora é representar seu contorno.

Foi descrito no algoritmo de extração de contornos, que ocorria "gravação" dos pontos de contorno. A estrutura de dados utilizada para tal gravação é caracterizada por uma estrutura de lista *C* contendo vários *strokes*. Um *stroke* é um conjunto de pontos adjacentes que inicia num determinado ponto e termina ou no "último" ponto do contorno, ou num *ponto de bifurcação*.

Além disso o critério de "gravação" que vai gerar a estrutura de lista *C* é o seguinte:

- 1) Tendo o ponto atual *P* do contorno, procura-se o próximo ponto vizinho em sentido anti-horário e iniciando com o ponto ao norte do ponto atual.
- 2) Segue-se o "caminho" com o primeiro ponto *Q* de contorno encontrado,
- 3) Posteriormente, devido a natureza recursiva do algoritmo, volta-se ao ponto *P* para se checar seus demais vizinhos.

Assim, supondo que um ponto de contorno *P* tenha dois vizinhos que também são pontos de contorno, uma a noroeste, e outro a leste, então, devido a busca em sentido anti-horário, é traçado em primeiro lugar o "caminho" a noroeste. Após todos os caminhos "derivados" do ponto a noroeste serem traçados é que será traçado o "caminho" a leste do ponto *P*.

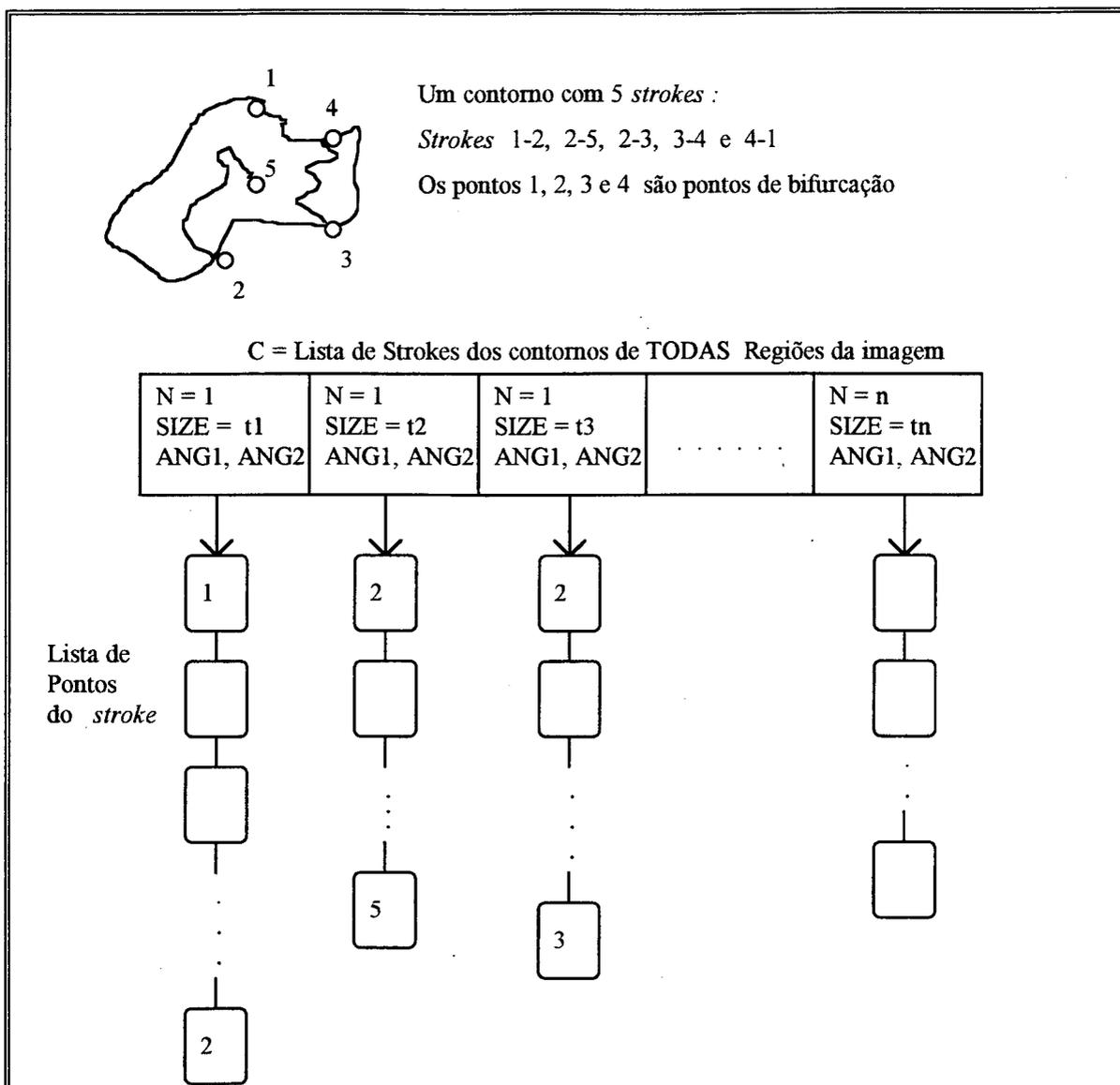


Fig. 5.11 Estrutura de dados para armazenar os contornos. No exemplo é mostrado o armazenamento de 3 dos 5 contornos existentes para a figura: 1-2, 2-5, 2-3. Os *strokes* são da região N = 1.

Considerando a figura 5.10, a lista de *strokes*, após o algoritmo de segmentação em regiões seria (os pontos são mostrados na forma matricial de (Linha, Coluna)):

Região 1 - stroke 1	(7,19), (8,18), (8,17), (9,16), (9,15), (9,14), (9,13), (9,12), (9,11), (9,10), (10,9), (10,8), (11,7), (11,6), (12,5), (13,4), (14,3), (15,2), (16,2), (17,1), (18,0), (19,0), (20,0), (21,1), (22,2), (23,3), (23,4), (23,5), (23,6), (23,7), (22,8), (21,9), (21,10), (21,11), (21,12), (20,13), (19,13), (18,14), (17,14), (16,14), (15,13), (14,14), (14,13), (15,12), (15,11), (15,10)
Região 1 - stroke 2	(9,15)
Região 1 - stroke 3	(9,16), (9,17)
Região 1 - stroke 4	(8,18), (9,19), (8,19), (9,20), (10,20), (11,21), (10,21), (11,22), (12,22), (13,23), (12,23), (13,24), (14,24), (15,25), (14,25), (15,26), (16,26), (16,27), (17,28), (16,28), (17,29), (18,29)

O início da formação dos *strokes* dá-se a partir do ponto superior mais à direita, que foi calculado durante o algoritmo de segmentação, conforme apresentado anteriormente. No exemplo, o ponto de partida é o ponto (7,19) o superior mais à direita.

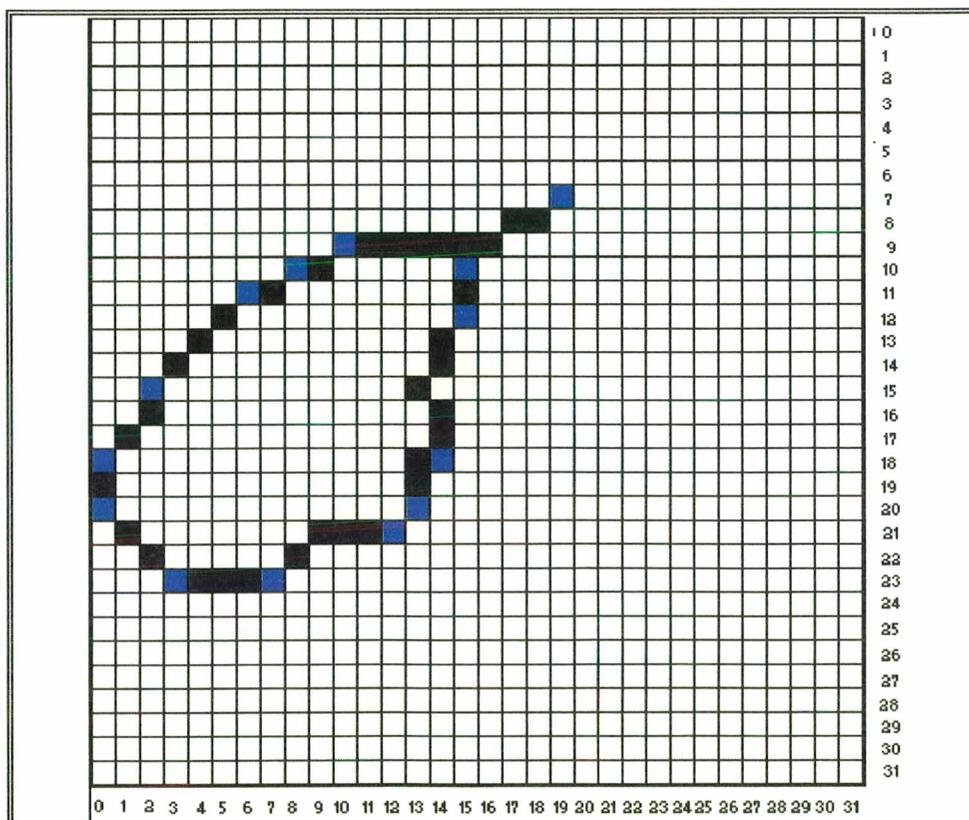


Fig. 5.12 Os melhores pontos para servirem de vértices, e daí representar o *stroke 1* da região 1 por um polígono

Na implementação do programa, considera-se *stroke*, aquele que apresenta 3 ou mais pontos. No exemplo, serão desconsiderados durante o processamento, os *strokes* 2 e 3 da região 1

É necessário agora estabelecer quais pontos serão pontos-base para dividir o contorno e formar um polígono. Ou seja, quer-se determinar, para cada *stroke*, quais os seus pontos-chave que servirão melhor como vértices para formação de um polígono. Por exemplo, para o *stroke* 1 da região 1 da letra "a", os pontos que melhor se adaptariam como pontos-chave (vértices de um polígono) seriam os pontos (7,19), (9,10), (10,8), (11,6), (15,2), (18,0), (20,0), (23,3), (23,7), (21,12), (20,13), (18,14), (15,12) e (15,10).

O conjunto de pontos entre os pontos-vértices adjacentes passam a ser representados por uma reta, que é a aresta do polígono. Assim por exemplo, os pontos (7,19), (8,18), (8,17), (9,16), (9,15), (9,14), (9,13), (9,12), (9,11) e (9,10), que estão entre os vértices (7,19) e (9,10), passam a ser representados pela reta de equação $Y = mX + b$ determinadas justamente pelos pontos-vértices (7,19) e (9,10).

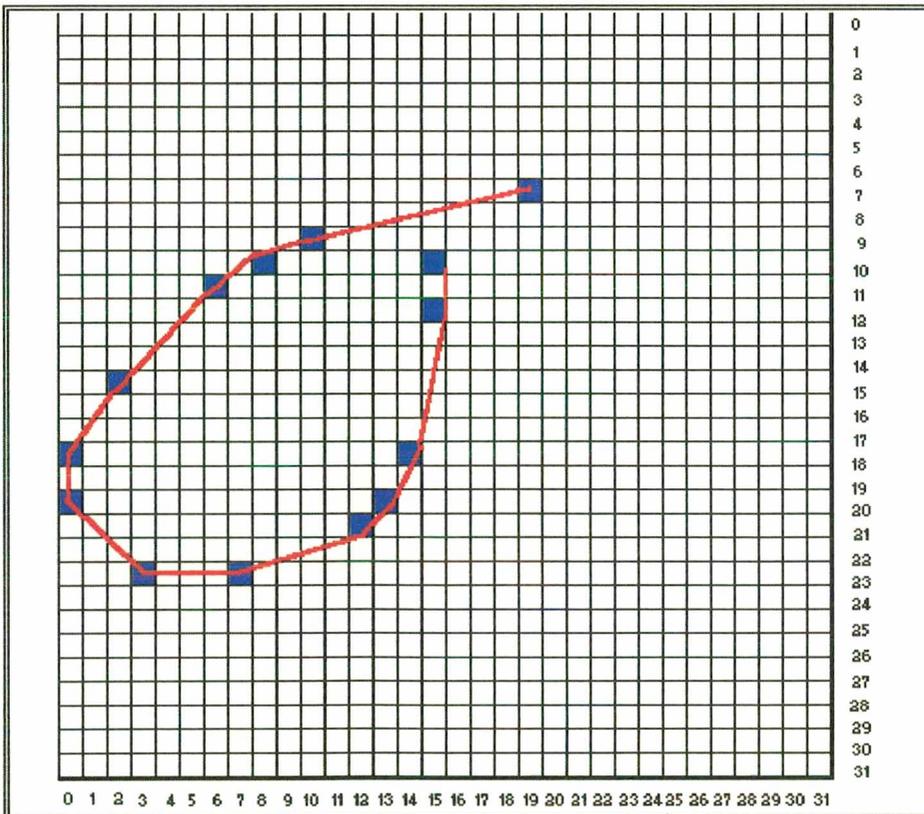


Fig. 5.13 - Representação do *stroke* 1 da região 1 por um polígono. As arestas representam o conjunto de pontos "contidos" entre dois pontos de vértice.

Em vista do que foi apresentado tem-se 2 questões a resolver:

- Como obter os "melhores" pontos de vértice ?
- Como garantir que a reta $Y = mX + b$ que passa exatamente entre dois pontos de vértice adjacentes representa satisfatoriamente o conjunto de pontos contido entre estes mesmos dois vértices ?

5.3.1 - Envelope Convexo (*Convex Hull*)

A primeira questão refere-se à escolha de "bons" pontos para "quebrar" o *stroke*. O "pontapé" inicial pode ser dado através da geração de um envelope convexo (*convex hull*) ([MEH84]) para o *stroke*. Uma região fechada é dita *convexa* se entre todos os seu pontos internos for possível ligar uma reta que "não sai" da região interna da figura. Em caso contrário a região é dita *não convexa*.

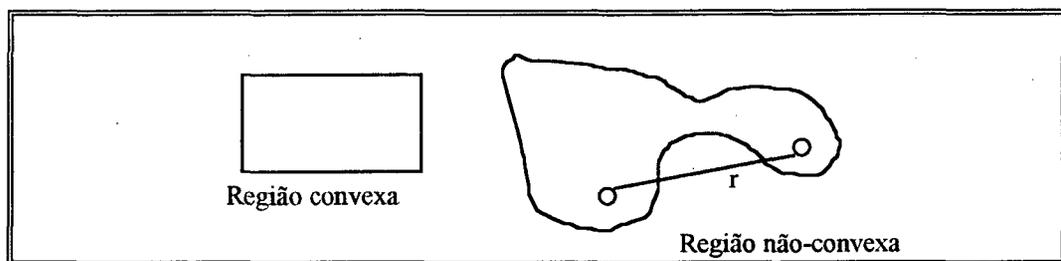


Fig. 5.14 Um exemplo de região convexa e não-convexa. Na região convexa, todos os pontos se ligam sem que a reta saia da área interna. Na região não-convexa, vê-se um exemplo entre 2 pontos da região ligados por uma reta r que apresenta um pedaço fora da área interna da região.

O envelope convexo (*convex hull*) é a menor região convexa que engloba uma determinada região ([MEH84]). A proposta, portanto, para se obter os pontos-vértices ideais é, para cada *stroke*, "visualizá-lo como uma região que será "coberta" por um envelope convexo. A proposta do envelope convexo é justamente tomar os pontos "mais salientes" da região a ser englobada. Mehlhorn ([MEH84]) apresenta um algoritmo para obtenção do envelope convexo. Este algoritmo foi utilizado no programa protótipo e a idéia é a seguinte:

- O polígono S é visto como um conjunto de pontos. Independe se S é ou não convexo. Normalmente será não convexo.
- Dois pontos, P_1 e P_r representam, respectivamente, as coordenadas mínima e máxima do polígono.

- A corda $L(P_1, P_T)$ divide o polígono em duas regiões, denominadas convexo-superior e convexo-inferior.
- Calculam-se inicialmente os pontos para o convexo-superior.
- Calculam-se depois os pontos para o convexo-inferior.

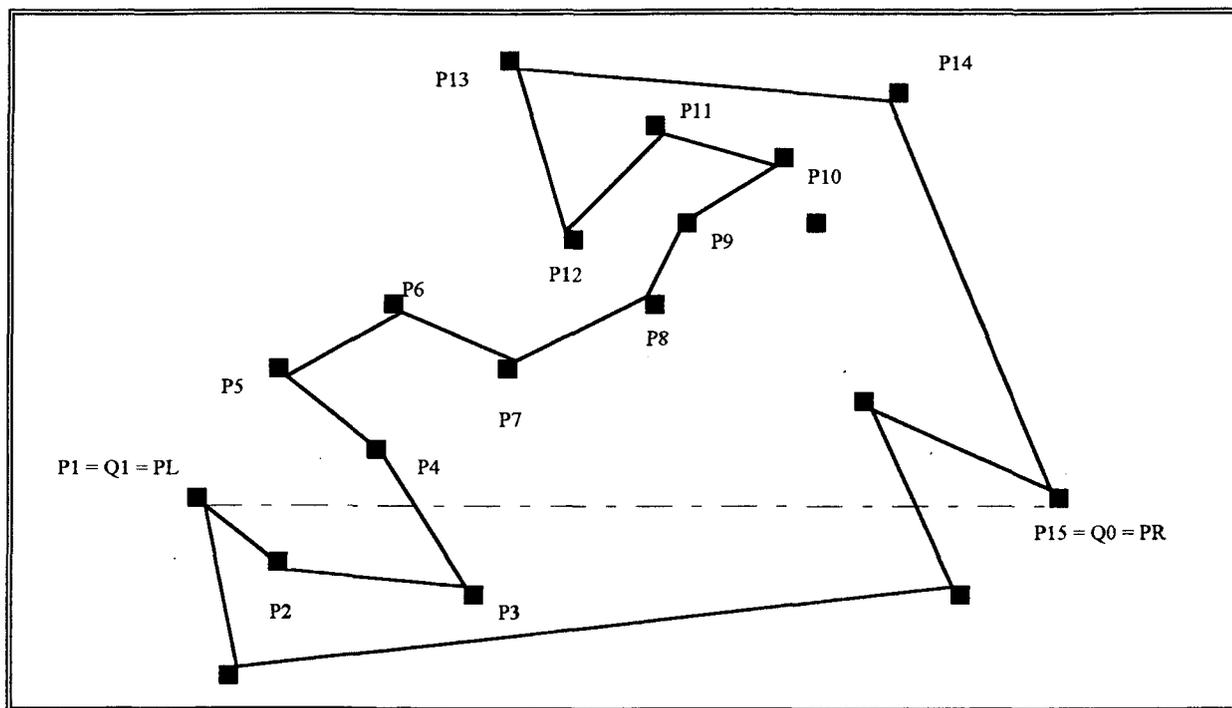


Fig. 5.15 - Um exemplo de polígono cujos pontos do convexo superior são P4, P5, P6, P7, P8, P9, P10, P11, P12, P13 e P14. P1 e P15 são, respectivamente, coordenada mínima e máxima. Adaptado de [MEH84]

Para o algoritmo de cálculo do convexo-superior (ou inferior), utiliza-se uma estrutura de pilha $Q(q_0, q_1, \dots, q_n)$ para armazenar os pontos candidatos a serem pontos "salientes" do envelope convexo. A pilha Q é inicializada com $q_0 = P_T$, $q_1 = P_1$, e q_2 com o primeiro ponto do caminho $P_1, P_2, P_3, P_4, \dots, P_n$ que está acima da corda $L(P_1, P_T)$.

Para calcular q_2 procede-se:

```

S = 2;
Enquanto  $(q_0, q_1, P_S)$  "está a esquerda" Faça
    S = S + 1;
Fim-Enquanto;

```

(q_0, q_1, P_S) significa uma análise do ponto (coordenada) P_S em relação à corda (q_0, q_1) .

Para entender o algoritmo, é preciso notar que o sentido da corda L importa, pois são utilizados os conceitos de "ponto à esquerda da corda" ou "ponto à direita da corda". Assim, por exemplo, na figura 5.15, o ponto P_7 está "à direita" da corda $L(P_6, P_8)$. Porém, P_7 está "à esquerda" da corda $L(P_8, P_6)$.

Uma vez que se tenha a pilha Q com os 3 elementos iniciais $q_0 = P_r$, $q_1 = P_l$, e q_2 o primeiro ponto do convexo-superior procede-se efetivamente o algoritmo de cálculo dos pontos do convexo-superior. É apresentado o algoritmo ([MEH84]) para o convexo-superior. q_t é o ponto do topo da pilha, P_S é o ponto em avaliação. Para entender o algoritmo é interessante observar a figura 5.15.

N = Número de pontos que estão "entre" P_l e P_r .
 S = inicializado com o índice do ponto P que está na pilha Q como Q_2
Enquanto (q_0, q_1, P_S) "está a esquerda" Faça
 Repita
 $S = S + 1$;
 Até P_S ["estar à esquerda de" $L(q_t, q_0)$] OU
 P_S ["estar à esquerda de" $L(q_{t-1}, q_t)$]
 Fim-Repita;
 Enquanto (q_{t-1}, q_t, P_S) NÃO "está à direita" Faça
 "desempilhar" q_t da pilha q ;
 Fim-Enquanto;
 "empilhar" P_S na pilha Q ;
 Fim-Enquanto;
 A sequência de pontos q_0, \dots, q_t é o envelope convexo procurado;

Ao final da execução deste algoritmo, para o convexo-superior da figura 5.15, o envelope convexo será o polígono formado pelos pontos $P_1, P_5, P_{13}, P_{14}, P_{15}$ e P_1 . O algoritmo acima, com as devidas adaptações efetua a construção do convexo-inferior.

O primeiro passo para calcular o envelope convexo de um determinado *stroke* é achar os pontos P_l e P_r . No programa protótipo procede-se da seguinte forma:

- 1) Parte-se do primeiro ponto do *stroke* (ver estrutura de dados na figura 5.11), tomando-o como ponto mínimo P_l .
- 2) Procura-se o ponto máximo P_r de modo que entre P_l e P_r os pontos intermediários formem um convexo superior ou um convexo inferior.
- 3) Efetua-se o algoritmo de envelope convexo para este conjunto.

4) Se o *stroke* ainda tem mais pontos "depois" de P_r , então faz-se $P_1 = P_r$ e volta ao passo 2.

No caso do *stroke* 1 da região 1 da letra "a", mostrado na figura 5.12, encontrar-se-á $P_1 = (7,19)$ e $P_r = (23,3)$ para o primeiro convexo. O segundo e último convexo calculado apresenta $P_1 = (23,3)$ e $P_r = (10,15)$. Aplicando-se o algoritmo proposto por Melhorn ([MEH84]) encontram-se os pontos de vértice mostrados na figura 5.13.

Em adição aos pontos gerados pelo algoritmo do envelope convexo, o programa-protótipo acrescenta os pontos de bifurcação, ou seja, aqueles pontos que "se partem" em duas ou mais direções. Na figura 5.12, o ponto (8,18) é ponto de bifurcação, pois, partindo-se de (7,19) chega-se em (8,18) e daí três caminhos podem ser seguidos: o ponto (8,17) do *stroke* 1 da região 1 ou os pontos (8,19) e (9,19), ambos pertencentes ao *stroke* 2 da região 1.

Por isso, o ponto (8,18) também é inserido como um vértice do polígono, juntamente com os pontos gerados pelo envelope convexo. Ainda na figura 5.12, assim como o ponto (8,18), o ponto (8,17) (que pode seguir por (9,16) ou (9,17)) também é ponto de bifurcação. Nas figuras 5.16, 5.17 e 5.18 mostra-se um exemplo onde os pontos de bifurcação, aliados aos pontos gerados pelo envelope convexo, auxiliarão a garantir um delineamento de contorno mais adequado.

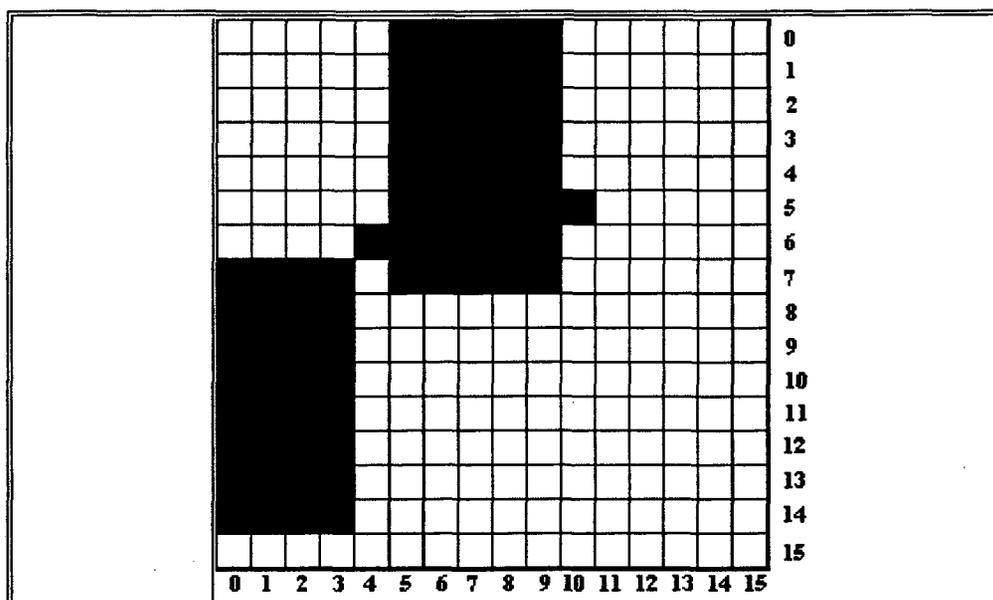


Fig. 5.16 Um exemplo de imagem que terá problemas com o contorno se não se utilizar ponto de bifurcação.

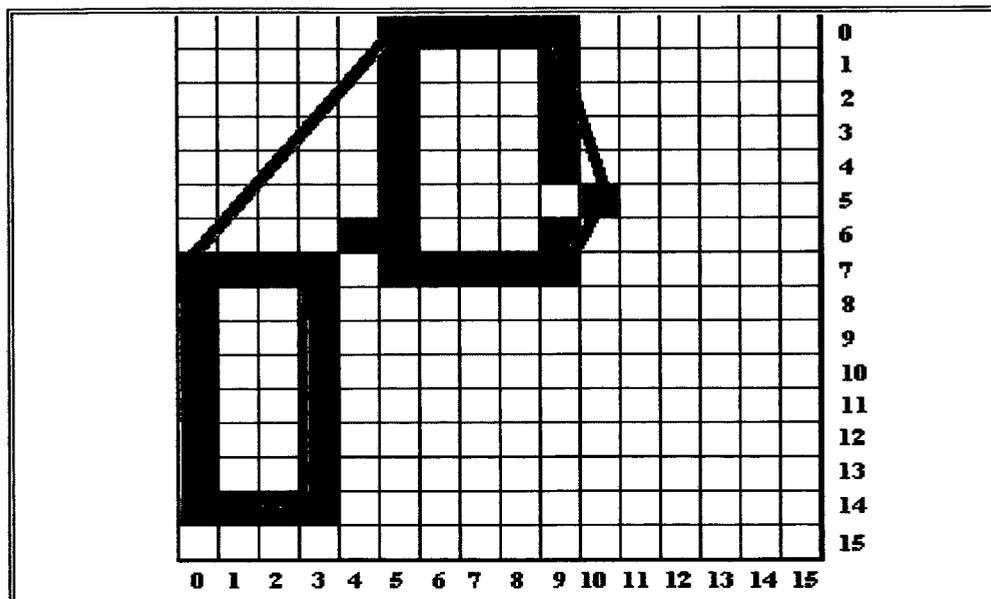


Fig. 5.17 - Após o algoritmo do envelope convexo (*convex hull*), nota-se que a delimitação da linha vermelha do envelope convexo "deformou" o contorno na parte que liga os pontos (0,5) e (7,0).

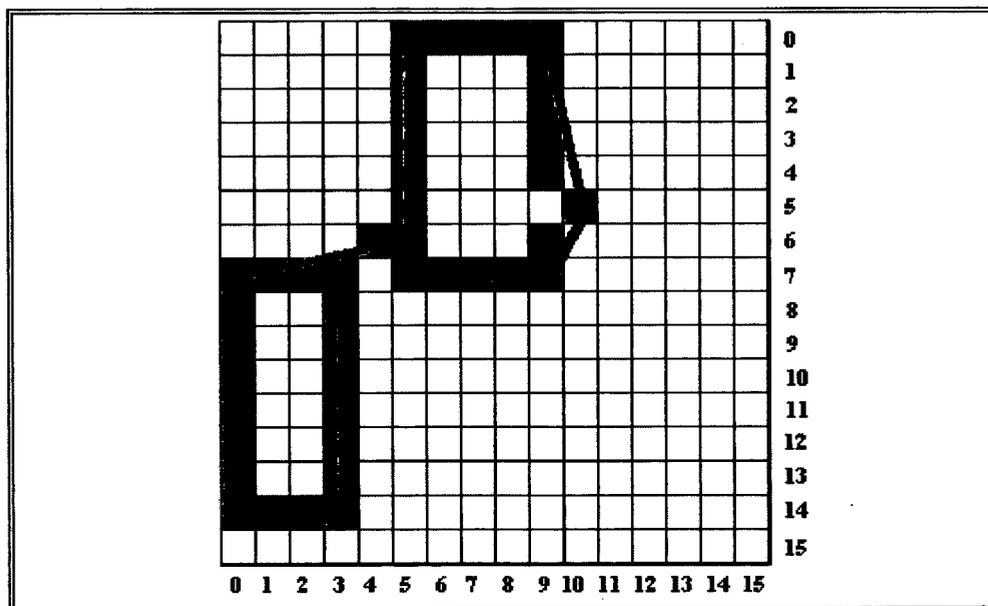


Fig. 5.18 - Se ao envelope convexo, forem anexados os pontos de bifurcação, a "deformação" anteriormente mostrada desaparece. No caso, o ponto de bifurcação (6,5) auxiliou a formação do contorno mais adequada.

5.3.2 - Um Critério Para Definição das Arestas do Polígono

A segunda pergunta a ser analisada, relativa ao polígono representativo do contorno, refere-se à representação de um determinado "pedaço" de pontos do contorno, situados "entre" dois vértices adjacentes por uma reta $Y = mX + b$. A questão é saber se a reta não "deforma" em demasiado a forma do contorno, na parte que ela representa.

A proposta de Pavlidis e Horowitz ([PAV74]) em resumo é: se tal reta não "prejudica" visualmente, então deve-se tentar "agrupar" (*merging*) na reta os pontos do próximo conjunto de pontos. Se ao agrupar tal conjunto a "deformação visual" não for prejudicada, então tenta-se agrupar o seguinte conjunto e assim por diante.

Por outro lado, se a reta que representa um dado conjunto de pontos "deforma" em demasiado o seu visual, então "quebra-se" (*splitting*) a reta ao meio, originando 2 novos segmentos de retas. Com menos pontos, a reta terá uma chance maior de representá-los de forma satisfatória. Se ainda assim não representa, então a proposta é separar o conjunto em 2 metades sucessivamente até a reta representá-los com uma precisão aceitável. A essência destes processos (*merging* e *splitting*) está representada na figura 4.10.

O critério para considerar se uma reta representa "bem" ou não um conjunto de pontos é determinado pelo estabelecimento de uma *norma de erro* ([PAV74]). Para o programa-protótipo implementado o critério adotado foi o seguinte:

- Toma-se o conjunto de pontos "entre" dois vértices (inclusive) fornecidos pelo algoritmo do envelope convexo (*convex hull*).
- Aplica-se a rotina de regressão linear sobre este conjunto. A reta gerada $Y_r = m_r X_r + b_r$, representativa de tal conjunto é tomada como o ponto de referência para estabelecimento de uma norma de erro.
- Considera-se a reta $Y = mX + b$, que é determinada pelos dois pontos de vértice.
- Adota-se uma norma de erro, baseada nos ângulos θ e θ_r que as retas Y e Y_r apresentam em relação ao eixo das abcissas. O erro máximo adotado permitido entre θ e θ_r é de 4% .

Se $(|\theta - \theta_r| / \theta_r) > 0.04$, então considera-se que a reta Y não representa de modo satisfatório o conjunto de pontos entre 2 vértices.

Chama-se a atenção sobre a natureza das duas retas Y e Y_r . Y_r , a reta de regressão, não necessariamente contera os pontos de vértices que "delimitam" o conjunto. A reta Y por sua vez sempre os contém, pois ela é gerada justamente por ambos.

Também quanto ao ângulo que a reta forma com o eixo das abcissas, é importante ressaltar que o sentido em que se "caminha" na reta é levado em conta. O ângulo θ que a reta forma com o eixo das abcissas é o ângulo "à direita" do ponto de origem da reta.

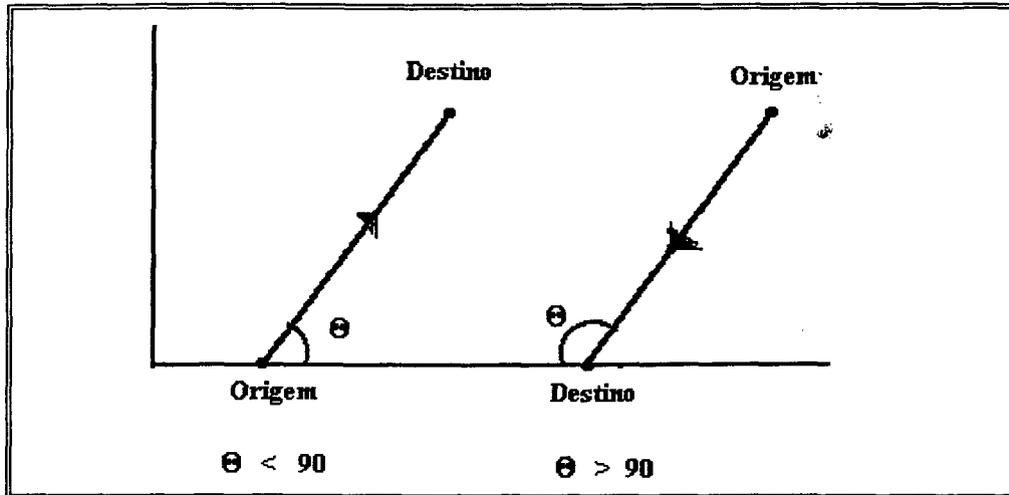


Fig. 5.19 - Para o programa protótipo o ângulo θ que a reta faz com o eixo das abscissas é importante. Será o ângulo "à direita" do ponto que origina a reta

Forsythe ([FOR72]) apresenta, de forma clara e sucinta, os conceitos teóricos matemáticos necessários ao entendimento do algoritmo de regressão linear.

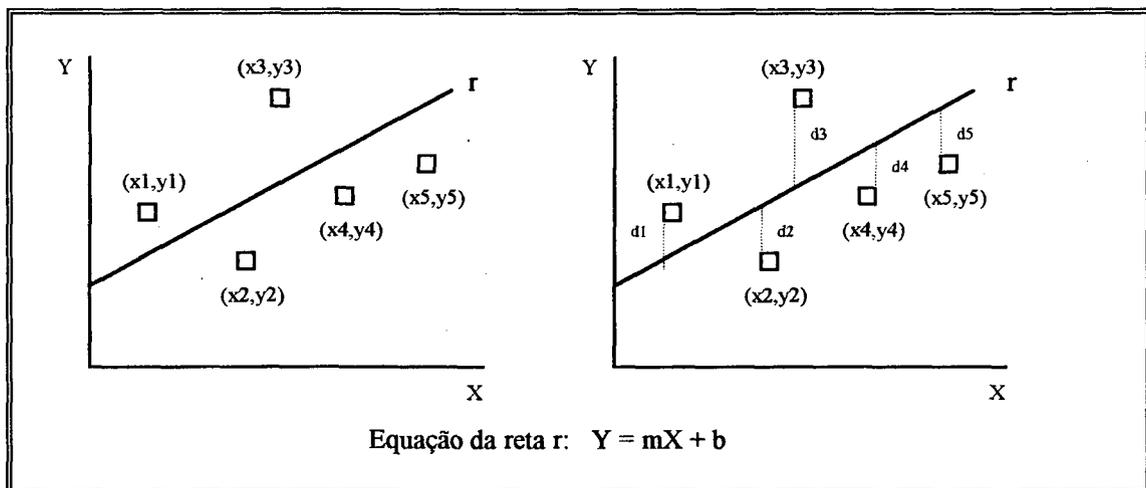


Fig. 5.20 - Regressão Linear para os pontos de coordenadas (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) , (x_5, y_5) . Por regressão linear encontra-se a reta $Y = mX + b$, que apresenta a "menor" distância média entre o conjunto dos 5 pontos.

Um dos resultados apresentados por Forsythe é o fato de a reta de regressão passar pelo ponto cujas coordenadas equivalem à média das coordenadas dos pontos do conjunto. Abaixo segue o algoritmo de regressão linear ([FOR72]) :

```

N = número total de pontos (inclusive os vértices)
SOMA_X = 0; SOMA_Y = 0; SOMA_X2 = 0; SOMA_XY = 0;
I = 1;
Enquanto I ≤ N Faça
  Ler (X,Y); { Ler um ponto do conjunto }
  SOMA_X = SOMA_X + X; SOMA_Y = SOMA_Y + Y;
  SOMA_X2 = SOMA_X2 + X2;
  SOMA_XY = SOMA_XY + (X*Y);
  N = N + 1;
Fim-Enquanto;
XM = SOMA_X / N;
YM = SOMA_Y / N;
SIGX2 = SOMA_X2 - (N * XM2);
SIGXY = SOMA_XY - (N * XM * YM);
M = SIGXY / SIGX2; { M = coeficiente angular de y = Mx + B }

```

Para exemplificar, considera-se novamente a letra "a" citada anteriormente. A figura 5.12, mostra os vértices-chave gerados pelo algoritmo do envelope convexo e fixação dos pontos de bifurcação. Um conjunto de pontos submetido ao cálculo de regressão linear seria o conjunto de pontos "delimitados" pelos vértices (8,17), que é ponto de bifurcação, e (9,10). São os pontos (8,17), (9,16), (9,15), (9,14), (9,13), (9,12), (9,11) e (9,10). O resultado obtido no programa foi o seguinte:

- Coeficiente angular m_r da reta $Y_r = m_r X_r + b_r$: $m_r = 0.08333$
- Como $\theta_r = \text{arc tg } m_r$, $\theta_r = 4.76^\circ$. O sentido da reta porém é sentido sudoeste. Então o ângulo "à direita" do ponto origem da reta é $180 - \theta_r$. Faz-se então $\theta_r = 175.24^\circ$.
- O coeficiente angular m da reta $Y = mX + b$, que é determinada exatamente pelos dois pontos de vértices é $m = 0.14286$
- Como $\theta = \text{arc tg } m$, $\theta = 8.13^\circ$. O sentido da reta porém é sentido sudoeste. Então o ângulo "à direita" do ponto origem da reta é $180 - \theta$. Faz-se então $\theta = 171.87^\circ$.

A norma de erro considerada é $(|\theta - \theta_r| / \theta_r) = (|171.87 - 175.24| / 175.24) = 0.0192 \cong 1.92\% < 4\%$. Como a norma de erro não foi desrespeitada, aceita-se a representação da reta para este conjunto de pontos. Durante o processo de *merging-e-splitting*, este conjunto não será fracionado, podendo, isto sim, ser talvez agrupado com outro conjunto de pontos, o que dependerá, naturalmente do fato do novo agrupamento não infringir a norma de erro.

O algoritmo *split-and-merge* para segmentação de contornos proposto por Pavlidis ([PAV74]) e implementado com certas adaptações no programa-protótipo é o seguinte (para melhor compreensão ver figura 5.11 mostrando a lista C de *strokes* das regiões) :

CONT é o nome da lista de *strokes*. Nesta estrutura, ANG1 e ANG2 representam, respectivamente, o ângulo da reta com o eixo das abcissas calculado por regressão linear e o ângulo da reta com o eixo das abcissas, determinada pelos pontos de vértice (o primeiro e o último ponto de LISTA-PONTOS-DE-STROKE).

```

TOT = Total de elementos na estrutura CONT;
Estrutura-Vetor CONT[1..TOT] = { N, SIZE, ANG1, ANG2, LISTA-PONTOS-DO-STROKE }
Estrutura AUX = { N, SIZE, ANG1, ANG2, LISTA-PONTOS-DO-STROKE }
BOOL = TRUE;
Enquanto BOOL = TRUE Faça
    BOOL = FALSE;
    { =====> SPLITTING <===== }
    I = 1;
    Enquanto I ≤ TOT Faça
        Conjunto-completo (Y,X) ∈ CONT(I).LISTA-PONTOS-DO-STROKE
        CONT(I).ANG1 = RegressãoLinear em {CONT(I).LISTA-PONTOS-DO-STROKE }
        CONT(I).ANG2 = RetaPontoInicioEPontoFim;
        Fim-Conjunto-completo;
        Se ( | CONT(I).ANG2 - CONT(I).ANG1 | / CONT(I).ANG1 ) > 0.04
            Então { efetuar split }
                BOOL = TRUE;
                MEIO = INDICE(CONT(I).LISTA-PONTOS-DO-STROKE.PontoDoMeio);
                ULTIMO = INDICE(CONT(I).LISTA-PONTOS-DO-STROKE.UltimoPonto);
                CONT(I).LISTA-PONTOS-DO-STROKE.UltimoPonto =
                    CONT(I).LISTA-PONTOS-DO-STROKE(MEIO);
                TOT = TOT + 1;
                CONT(TOT).N = CONT(I).N;
                K = 1;
                Para J = MEIO Até ULTIMO
                    CONT(TOT).LISTA-PONTOS-DO-STROKE(K) =
                        CONT(I).LISTA-PONTOS-DO-STROKE(J);
                    K = K + 1;
                Fim-Para;
            Fim-Se;
        Fim-Enquanto;
    Fim-Enquanto;

```

$I = I + 1;$

Fim-Enquanto;

{ \Longrightarrow MERGING \Longleftarrow }

$I = 1;$

Enquanto $I < TOT$ Faça

Conjunto-completo $(Y,X) \in \text{CONT}(I).\text{LISTA-PONTOS-DO-STROKE}$ E

Conjunto-completo $(Y,X) \in \text{CONT}(I+1).\text{LISTA-PONTOS-DO-STROKE}$

$\text{AUX.N} = \text{CONT}(I).\text{N};$

$\text{AUX.SIZE} = \text{CONT}(I).\text{SIZE} + \text{CONT}(I+1).\text{SIZE};$

$\text{AUX.LISTA-PONTOS-DO-STROKE} =$
 $\text{CONT}(I).\text{LISTA-PONTOS-DO-STROKE} \cup$
 $\text{CONT}(I+1).\text{LISTA-PONTOS-DO-STROKE};$

$\text{AUX.LISTA-PONTOS-DO-STROKE.PrmeiroPonto} =$
 $\text{CONT}(I).\text{LISTA-PONTOS-DO-STROKE}(\text{PrimeiroPonto});$

$\text{AUX.LISTA-PONTOS-DO-STROKE.UltimoPonto} =$
 $\text{CONT}(I+1).\text{LISTA-PONTOS-DO-STROKE}(\text{UltimoPonto});$

$\text{AUX.ANG1} = \text{RegressãoLinear em } \{ \text{AUX.LISTA-PONTOS-DO-STROKE} \}$

$\text{AUX.ANG2} = \text{RetaPontoInicioEPontoFim};$

Fim-Conjunto-completo;

Se $(| \text{AUX.ANG2} - \text{AUX.ANG1} | / \text{AUX.ANG1}) \leq 0.04$

Então { efetuar merge }

$\text{BOOL} = \text{TRUE};$

$\text{CONT}(I).\text{LISTA-PONTOS-DO-STROKE.UltimoPonto} =$
 $\text{AUX.LISTA-PONTOS-DO-STROKE.UltimoPonto};$

$\text{CONT}(I).\text{LISTA-PONTOS-DO-STROKE} =$
 $\text{AUX.LISTA-PONTOS-DO-STROKE};$

Deletar $\text{CONT}(I+1);$

$\text{TOT} = \text{TOT} - 1;$

Fim-Se;

$I = I + 1;$

Fim-Enquanto;

Fim-Enquanto;

Após este processo, obtém-se finalmente um polígono representativo para o contorno da região em questão, e que no presente trabalho refere-se à estrutura do estilete do nematóide. O ideal é que se obtenha um polígono com o mínimo de arestas possível, porém mantendo o teor informativo da estrutura. Quanto menos arestas, menos "denso" é o polígono do contorno e, assim, no processo de *template matching* (comparação entre 2 formatos), menos estruturas terão que "combinar", e com isso

aumenta-se a chance de sucesso na comparação ao mesmo tempo que diminui-se o tempo de processamento.

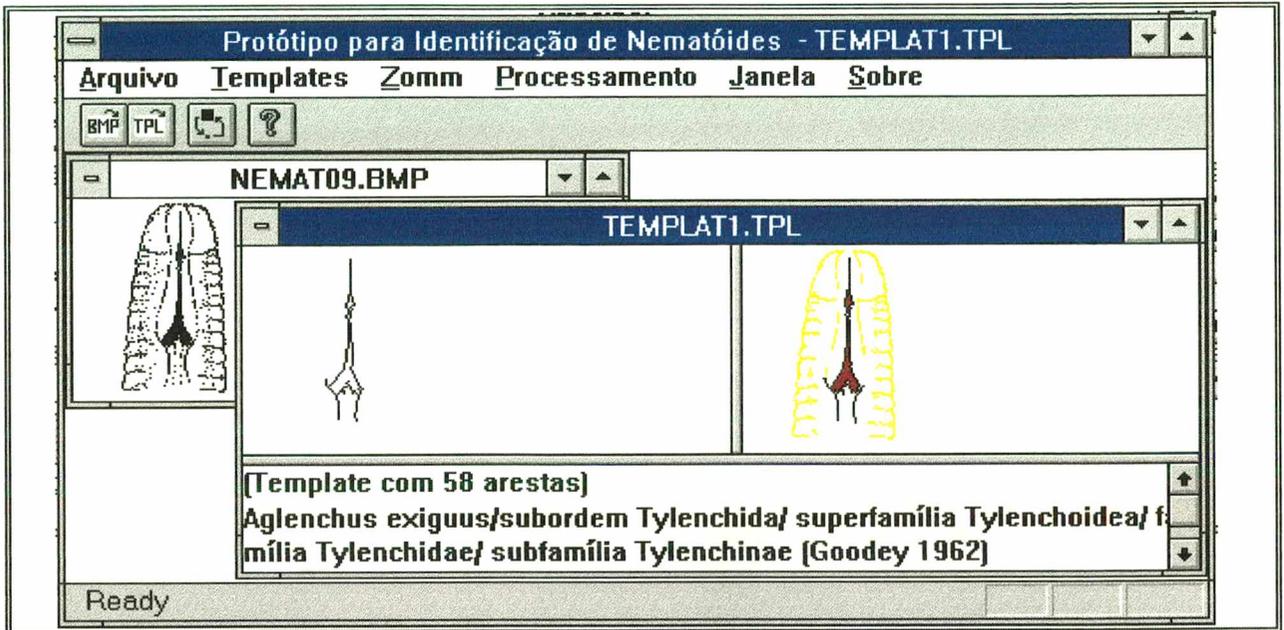


Fig. 5.21 - O contorno do estilete para a imagem de NEMAT09.BMP. Polígono com 58 arestas executado com norma de erro de 4%.

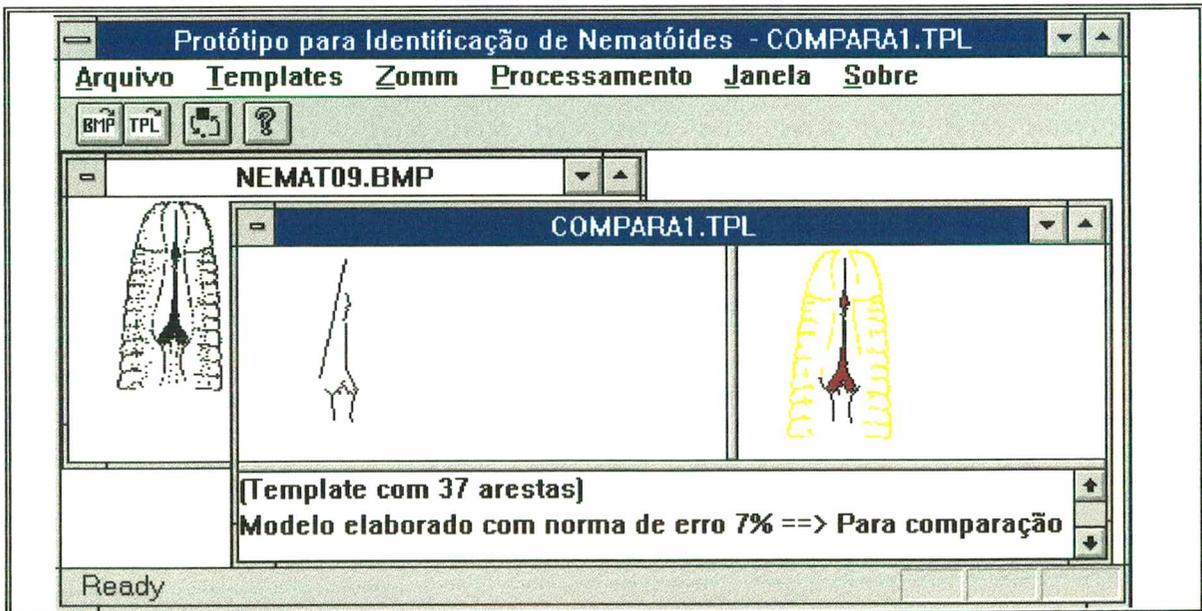


Fig. 5.22 - A mesma imagem da figura 5.21 gera um polígono para contorno com 37 arestas. erro é de 7%.

No exemplo da figura 5.21, a imagem NEMAT09.BMP "gerou" o polígono do contorno do estilete com um total de 58 arestas. Este processo foi elaborado com a norma de erro de 4%.

Na figura 5.22, observa-se que a mesma imagem NEMAT09.BMP da figura 5.21 gera um contorno com uma considerável deformação visual. Isto porque a norma de erro para formação de retas de representação das arestas é de 7% contra os 4% estabelecidos para o programa-protótipo. Apresenta 37 arestas, 21 a menos que o gerado no processo sob margem de erro 4%.

5.4 - Algoritmo para Calcular a Distância entre 2 AFT's

No processo de aproximação poligonal do contorno do estilete obtém-se os vértices e arestas do polígono representativo do contorno do estilete. A estrutura onde se armazenam tais informações é uma estrutura de lista, que na etapa de cálculo da distância entre 2 AFT's serve de subsídio para o cálculo discriminador das diversas unidades que compõem o grafo AFT de tal polígono.

Lista de Arestas A (polígono do contorno)

(x1,y1) início	(x2,y2) início	(x3,y3) início		(xn,yn) início
(x1,y1) fim	(x2,y2) fim	(x3,y3) fim	(xn,yn) fim
COMPR = c1	COMPR = c2	COMPR = c3		COMPR = cn

Lista de Unidades U (grafo AFT)

Vetor DT1i	Vetor DT2i	Vetor DT3i	Vetor DT4i		Vetor DTmi
Vetor P1	Vetor P2	Vetor P3	Vetor P4	
Vetor DT1f	Vetor DT2f	Vetor DT3f	Vetor DT4f		Vetor Pmf

2 Vetores DT = Direção-Tamanho	1 Vetor P = Posição
Horizontal-Grande	Acima de
Horizontal-Médio	Abaixo de
Horizontal-Pequeno	À Esquerda de
Horizontal- Muito Pequeno	À Direita de
Vertical-Grande	
Vertical-Médio	
Vertical-Pequeno	
Vertical-Muito Pequeno	
Positivo-Grande	
Positivo-Médio	
Positivo-Pequeno	
Positivo-Muito Pequeno	
Negativo-Grande	
Negativo-Médio	
Negativo-Pequeno	
Negativo-Muito Pequeno	

Fig. 5.23 - Estrutura de dados para armazenar arestas do polígono e unidades do grafo AFT

O algoritmo que efetua os cálculos para a distância D entre 2 AFT's é o seguinte (não está implementada a lógica para rotação/translação da imagem):

N = Número de arestas que representa o polígono P do padrão a ser reconhecido
M = (N * (N-1)) / 2; // C_{N,2} = número de unidades do grafo AFT de P
Estrutura-Vetor ARESTA [1..N] = { XI, YI, XF, YF }
Vetor DT-PAD-1[16], DT-PAD-2[16] ;

```

Vetor P-PAD[4];
Estrutura-Vetor ARESTA-TPL [1.. $\alpha$ ] = { XI, YI, XF, YF}
Estrutura-Vetor UNID-TPL [1.. $\beta$ ] = { Vetor DTI[16], Vetor P[4], Vetor DTF[16] }

DIST-F1 = 0; // Conterá o valor da distância entre 2 AFT's
Enquanto tem-arquivo-TPL Faça
  Ler template para ARESTA-TPL e UNID-TPL;
  Para I = 1 Até N
    Para J = I+1 Até N
      // ==> Formar uma Unidade do padrão a ser reconhecido <==== //
      "Calcular" DT-PAD-1 em função de ARESTA[I].{(XI, YI), (XF, YF)};
      "Calcular" DT-PAD-2 em função de ARESTA[J].{(XI, YI), (XF, YF)};
      "Calcular" P-PAD em função de DT-PAD-1 e DT-PAD-2;
      // DIST-F2 conterá o mínimo entre { PAD x Unidades do Template }
      DIST-F2 = 9999;
      // ==> Comparar esta unidade com todas as unidades do template
      Para K = 1 Até  $\beta$ 
        SIMILAR =  $d_{11}$  em função de { DT-PAD-1/DT-PAD-2/P-PAD } e
          { UNID-TPL[K].DTI, UNID-TPL[K].DTF, UNID-TPL[K].P }
        Se SIMILAR < DIST-F2
          Então
            DIST-F2 = SIMILAR;
          Fim-se;
        Fim-Para;
      Se DIST-F2 > DIST-F1
        Então
          DIST-F1 = DIST-F2;
        Fim-se;
      Fim-Para
    Fim-Para
  Fim-Enquanto;

```

5.5 - Um Teste do Programa-Protótipo

Para exemplificar a dinâmica de funcionamento do modelo proposto neste trabalho, é apresentado no que se segue, um teste para identificação de um determinado padrão, em confronto com três *templates*.

Padrão a ser reconhecido

Escolheu-se para padrão a ser identificado, o nematóide da imagem PADREC.BMP, apresentada na figura 5.24.



Fig. 5.24 - Nematóide representando o padrão a ser identificado

Base de conhecimento

Para o teste, utilizou-se uma base de conhecimento contendo três *templates* cujas imagens denominam-se TEMPL01.TPL, TEMPL02.TPL e TEMPL03.TPL., apresentados juntamente com suas representações poligonais nas figuras 5.25, 5.26 e 5.27 respectivamente.

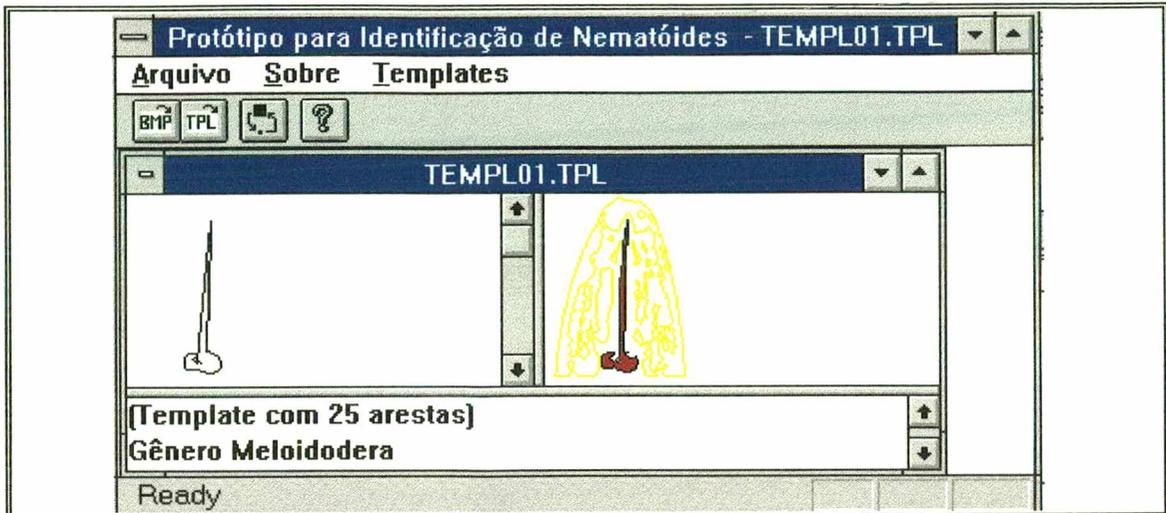


Fig. 5.25 - O primeiro *template* da base de conhecimento

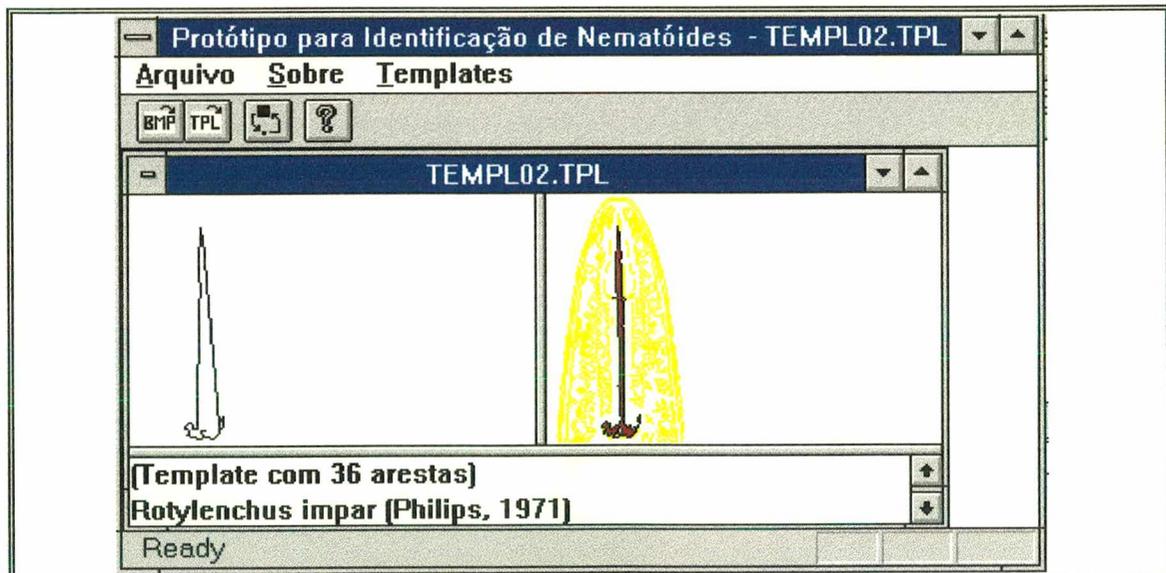


Fig. 5.26 - O segundo *template* da base de conhecimento

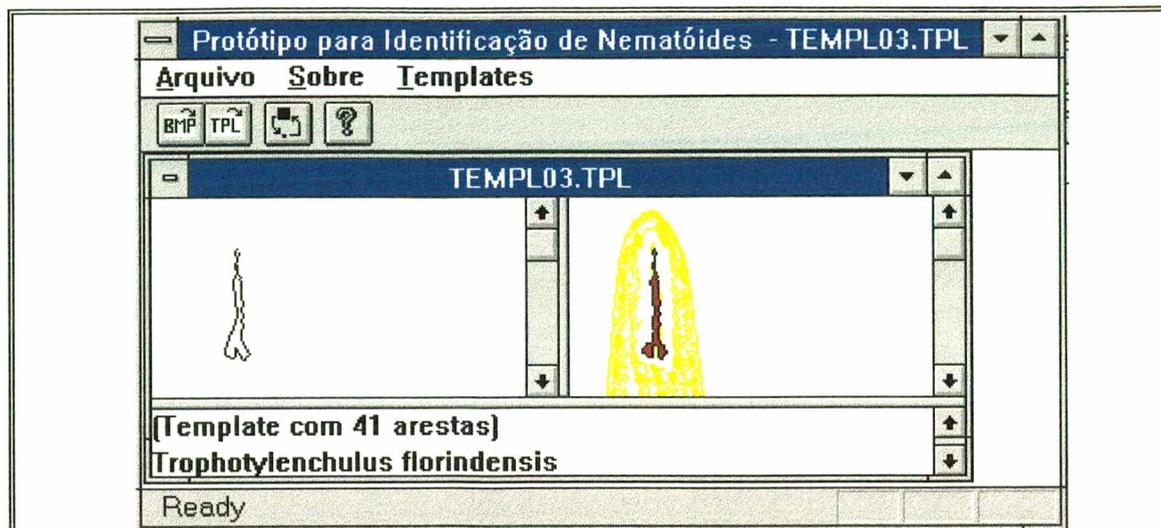


Fig. 5.27 - O terceiro *template* da base de conhecimento

Pode-se notar que há deformação no polígono representativo dos *templates*. Isto deve-se à norma de erro mais tolerante aplicada durante o processo de aproximação poligonal. Se fosse de interesse, poder-se-ia aplicar uma norma mais rigorosa (percentual de erro menor) e o polígono tomaria uma forma mais delineada. Contudo, o número de arestas do polígono aumentaria, o que faria também aumentar o tempo de processamento para o *template matching*.

O Resultado após o *template matching*

Uma vez apresentado o padrão PADREC.BMP e sendo solicitada ao programa sua identificação, após o *template matching* a resposta foi a seguinte, apresentada na figura 5.28:

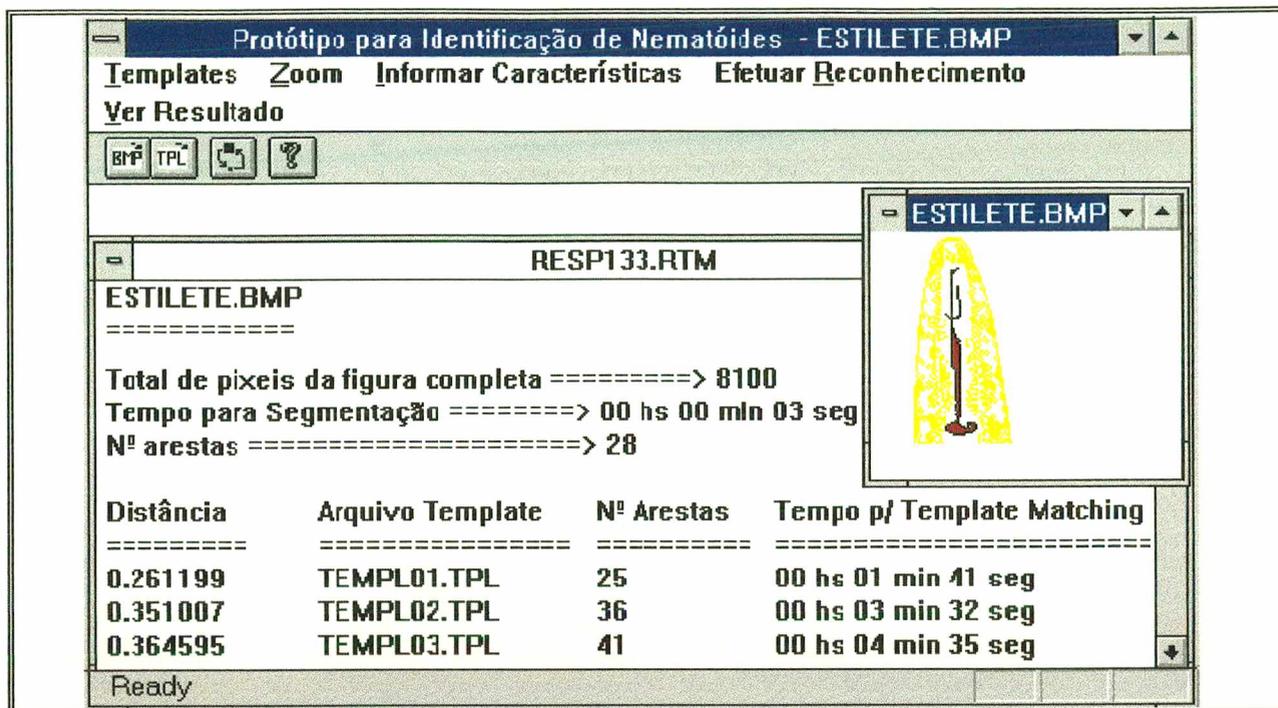


Fig. 5.28 - O resultado após *template matching*

PADEREC está representado na janela ESTILETE.BMP e conforme se observa, a similaridade de PADREC.BMP é maior com o *template* TEMPL01.TPL (0.261199), em seguida é mais parecido com o *template* TEMPL02.TPL (0.351007), e finalmente está mais "distante" do *template* TEMPL03.TPL (0.364595). O teste foi executado num microcomputador PC-Pentium com *clock* de 133 Mhz.

Interpretação do resultado

O resultado reflete numericamente uma questão típica de raciocínio aproximado. Para formular tal questão coloca-se o polígono do contorno do padrão a ser reconhecido "em frente" aos polígonos dos contornos dos *templates* como mostra a figura 5.29.

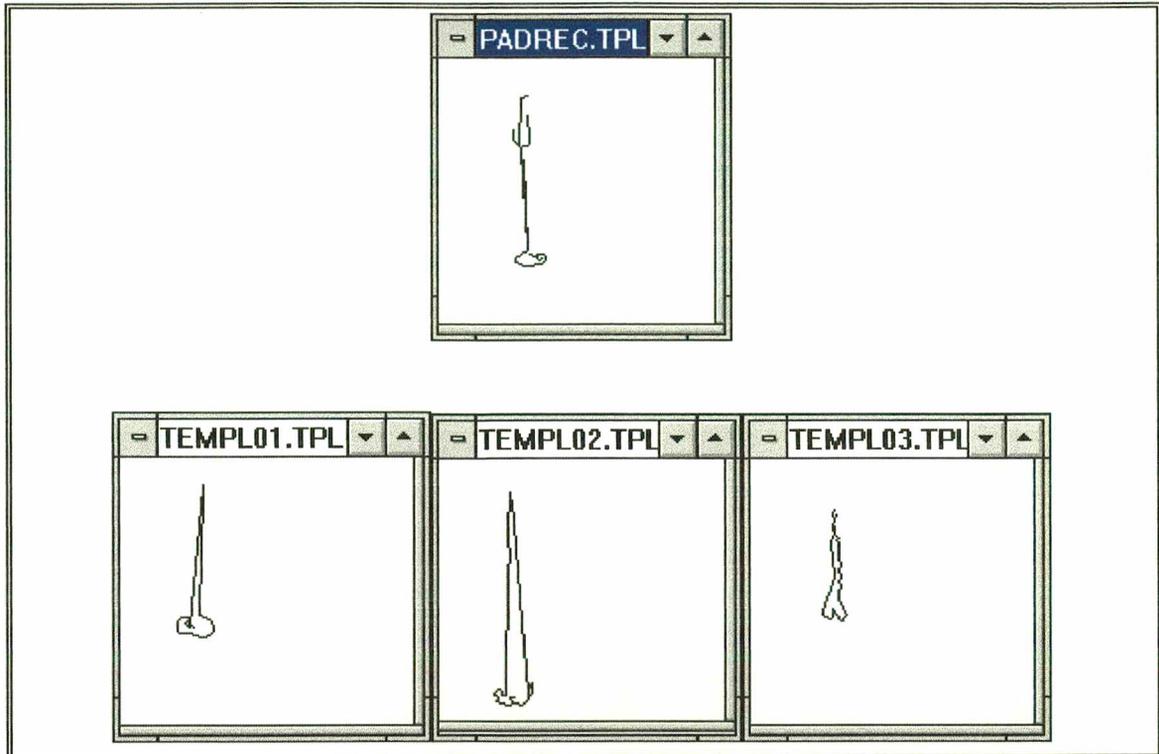


Fig. 5.29 - Raciocínio aproximado, para julgar a semelhança do padrão PADREC.BMP em relação a TEMPL01.TPL, TEMPL02.TPL e TEMPL03.TPL.

Coloca-se então a pergunta:

Qual das três figuras (TEMPL01, TEMPL02, TEMPL03) é mais parecida com PADREC ?

Intuitivamente uma pessoa utilizando seu próprio "raciocínio aproximado" (algum algoritmo de clusterização por exemplo) muito provavelmente responderá que é o *template* TEMPL01.

E foi exatamente o que o programa "respondeu". A resposta, não se deu porque TEMPL01 e PADREC apresentam quase o mesmo número de arestas, mas sim porquê as *relações* entre as 28 arestas do padrão "concordam" (*matching*) mais com as relações entre as arestas de TEMPL01 do que as relações entre as arestas de TEMPL02 e TEMPL03.

O sistema, como se observa na análise do teste, de fato age como um dispositivo inteligente, no sentido de analisar um determinado cenário e apresentar determinado "comportamento" perante o mesmo.

CAPÍTULO 6

Conclusões e Recomendações

6.1 - Conclusões

A caracterização do estilete do nematóide, como um cenário onde existem incertezas mostrou ser uma proposta atraente, uma vez que:

- 1) O estilete apresenta um formato anatômico relativamente "uniforme", ou seja, cujo contorno tenha a possibilidade de ser representado de maneira satisfatória por um polígono.
- 2) O polígono caracterizador do contorno do estilete necessita de uma estrutura de dados simples, no caso uma estrutura de lista duplamente encadeada, para armazenar os pontos-vértices de suas arestas.
- 3) A estrutura de lista para as arestas do polígono é a única estrutura que realmente necessita estar armazenada na base de conhecimento, pois é a partir dela que se constrói o grafo AFT caracterizador do contorno.
- 4) O grafo AFT, que representa o cenário juntamente com suas incertezas é uma estrutura bastante simples facilmente implementada em termos computacionais, e com alto poder de representação de cenários.

Este modelo também é passível de aplicação a outros órgãos do nematóide, que apresentem, a exemplo do estilete, um formato anatômico relativamente "uniforme". Assim, órgãos como o metacorpo ou esôfago, por exemplo, podem de maneira análoga ao estilete ser representados por grafos AFT. Com isso aumenta-se desde já o escopo da proposta do modelo, ampliando-se conseqüentemente seu poder discriminador em termos de identificação/classificação de nematóides.

6.2 - Recomendações

A proposta de se estabelecer um modelo de identificação de nematóides através de suas imagens digitalizadas, está embasada em três quesitos chave:

- 1) Um processo de tratamento da imagem para extração da região de interesse
- 2) A estrutura a ser usada para representar tal região de interesse
- 3) O método a ser utilizado no processamento da estrutura(que é vista como dado), para se obter *informação*.

No modelo apresentado, o processo de tratamento da imagem se caracteriza pelo uso do algoritmo *split-and-merge* ([HOR76]). É um algoritmo consistente, típico extrator de regiões, conciso e fácil de se implementar computacionalmente, embora um pouco trabalhoso. Tem a vantagem de já chegar "quase" à região desejada.

Para o modelo proposto apresentou um desempenho satisfatório, em termos de eficiência e eficácia. Contudo é um algoritmo cujo parâmetro de avaliação é apenas a intensidade luminosa do pixel, não levando em conta aspectos semânticos da imagem.

Para complementar esta abordagem, e melhorar a eficácia de seus resultados de extração de região, o algoritmo poderia ser complementado de modo a tratar de alguma forma a imagem semanticamente, por exemplo, por modelos da linha das redes neurais/raciocínio aproximado (conjuntos difusos).

Quanto à estrutura de representação da informação que se trata (no caso o estilete dos nematóides), a representação através de polígono é satisfatória. Contudo os critérios que se utilizam para gerar o polígono caracterizador do contorno são insuficientes. No modelo proposto, a etapa de aproximação poligonal para o contorno do estilete se baseia nos pontos de contorno pertencentes a seu envelope convexo (*convex hull*) e pontos de bifurcação do contorno. O polígono caracterizador do contorno, gerado apenas por estes critérios ou terá um número excessivo de arestas, o que influi negativamente em termos de tempo de processamento, ou terá um número pequeno de arestas que porém não representam adequadamente o contorno em termos de "visual".

Portanto, é de interesse que se estudem métodos (de morfologia matemática por exemplo) a serem implementados de modo a dotar a etapa de aproximação poligonal de mecanismos que gerem polígonos com poucas arestas, porém com alta representatividade do contorno.

Quanto ao método de processamento dos dados, utilizou-se no modelo o grafo AFT, que de fato, representa de modo satisfatório um cenário e suas incertezas. Contudo uma limitação desta estrutura para o modelo identificador de nematóides é o fato de ser sensível à rotação. Em virtude disso sugere-se que outros modelos, que sejam não-sensíveis à rotação sejam estudados e comparados.

Para o cálculo da distância entre dois grafos AFT's está sendo empregada a "força bruta": leva-se em conta a similaridade entre *todas* as unidades componentes dos grafos AFT em comparação, para se chegar ao valor final da distância.

Assim, por exemplo, se um polígono P1 representativo de um determinado contorno tem 20 arestas e outro polígono P2 representativo de outro contorno apresenta 25 arestas, então estão em confronto $C_{20,2} = 190$ unidades contra $C_{25,2} = 300$ unidades. Confrontadas todas as 190 unidades do grafo AFT de P1 com todas as 300 unidades do grafo AFT de P2 tem-se $190 \times 300 = 57000$ comparações. Para reduzir este montante de comparações pode-se, por exemplo, clusterizar os padrões P1 e P2 e efetuar cálculos entre as unidades dos *clusters* C_1, C_2, \dots, C_n de P1 e as unidades dos *clusters* K_1, K_2, \dots, K_m de P2.

Resta ainda comentar sobre a caracterização, em termos semânticos, do grafo AFT proposto no modelo deste trabalho. A natureza do grafo AFT permite um significativo poder de expressão em termos semânticos, quando se caracteriza um nodo do grafo (aresta do polígono) em termos de sua *direção* e *tamanho*. A relação (arco) entre dois nodos também fica bem caracterizada em função da *posição* relativa entre ambos.

Com isso, o grafo AFT torna-se de fácil interpretação por parte de um observador (uma pessoa entenderá, por exemplo, quando se diz que "uma aresta horizontal-grande está à esquerda de uma aresta vertical-média com grau de pertinência α "). Para se conseguir este melhor poder de expressão em termos semânticos utilizou-se duas variáveis linguísticas (*direção* e *tamanho*) para caracterização dos nodos e uma variável linguística (*posição*) para caracterização dos arcos do grafo AFT, que levam à existência de 16 e 4 termos (conjuntos difusos) respectivamente, somando um total de 20 termos.

Esta proposta de caracterização utilizando 20 termos linguísticos para o grafo AFT tem como objetivo embasar o cálculo da distância, ou seja, "gerar" um número representativo da similaridade entre dois padrões. Porém, esta caracterização, que está bem delineada semânticamente fica restrita apenas "internamente", no sentido de se limitar ao cálculo da similaridade, não sendo utilizada "externamente" (por exemplo, para "explicar" a um observador como se chegou a tal valor de similaridade). Com isso, a caracterização dos nodos e arcos poderiam ser simplificados em termos de representação, para duas variáveis (não linguísticas) para nodos e uma variável (não linguística) para arcos. Os nodos poderiam ser caracterizados pelo ângulo θ que formam com o eixo dos X e seu tamanho relativo (em percentual). Os arcos seriam caracterizados unicamente pelo ângulo ϕ que formam com o eixo dos X.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ABE93] ABÉ, H.; KITO, K.; ARYUTHAKA, C.; TAMURA, H.; *Use of a Photographic Slide Mount for Nematode Preparations*, Nematologica - International Journal of Nematological Research, Vol. 39, N° 1, Jan 1993
- [ALM92] ALMEIDA, V.F.; *Nematóides em Fruteiras*, in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 65-72, 1992
- [AMA79] AMABIS, J.M.; MARTHO, G.R.; MIZUGUCHI, Y.; *Biologia - Os Seres Vivos Vol. 2*, Ed. Moderna LTDA. 1979
- [AND72] ANDREWS, H.C.; *Introduction to Mathematical Techniques in Pattern Recognition*, John Wiley & Sons, Inc., 1972
- [ANT92] ANTÔNIO, H.; *Fitonematóides na Cultura da Soja* in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 60-65, 1992
- [BAK75] BAKER, J.J.; ALLEN, G.E.; *Estudo da Biologia Volume 2*, Ed. Edgard Blucher Ltda., 1975
- [BAL95] BALDWIN J.G.; LUC, M.; *Current Problems in Taxonomy*, Nematologica - International Journal of Nematological Research, Vol. 41, N° 3, Maio 1995
- [BAR93] BARKER, K.R.; *The Future of Nematology: Introduction*, Journal of Nematology, Vol. 25, N° 3, p. 367-373, setembro/1993
- [BEA91] BEALE, R.; JACKSON, T.; *Neural Computing: An Introduction*. Philadelphia, PA/USA, Adam Higler, 1991
- [BEZ81] BEZDEK, J.; *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York, Plenum Press, 1981.

- [BLA92] BLANCA, N.P.; VALDIVIA, J.F.; CASTILLO, P.; BARCINA, A.G.; Detecting Nematode features from Digital Images, Journal of Nematology Vol 24 N° 2, 1992, p. 289-297
- [BOA88] BOAG, B.; TOPHAM, P.B.; BROWN, D.J.F.; SMITH, P.; The Use of microcomputers for Nematode Identification, in Nematode Identification and Expert System Technology. New York, Plenum Press, 1988
- [BRA82a] BRANCALION, A.M.; LORDELLO, L.G.E.; Emprego de ALDICARB e CARBOFURAN no Controle de Nematóides em Cultura de Milho, Publicação N° 5, Piracicaba-SP, 1982
- [BRA82b] BRANCALION, A.M.; LORDELLO, L.G.E.; Controle Químico de Rotylenchus reniformis em Algodoeiro, Publicação N° 5, Piracicaba-SP, 1982
- [BRA94] BRAVO, M.A.; ROCA, F.; Using dBASE for Identification of Species of the Genus Xiphinema, 22nd International Symposium Proceedings of the European Society of Nematologists, Ghent, Belgium, 7-12 August, 1994. in Journal of Nematology, Vol. 41, N° 3, may, 1995
- [CAB96] CABRAL, R.B.; Retificação e Análise de Características de Imagens de Nematóides, Dissertação de Mestrado, Depto. de Engenharia de Produção e Sistemas, UFSC - Universidade Federal de Santa Catarina, 1996
- [CAM92] CAMPOS, V.P.; Danos e Prejuízos Causados por Fitonematóides, in Informe Agropecuário, Belo Horizonte, Vol. 16 N° 172 p. 14-15, 1992
- [CAS94] CASTILLO, P.; VOVLAS, N.; BARCINA, A.G.; LAMBERTI, F.; The Plant Parasitic Nematode Rothylenchus (A monograph), Instituto di Agricultura Sostenibile, C.S.I.C. Córdoba Spain, Instituto di Namatologia Agraria, C.N.R., Bari Italy, Centro de Investigación y Desarrollo Agrario, C.I.D.A., Granada Spain, 1994
- [CHO90] CHORAFAS, D.N.; Knowledge Engineering. Ed. Van Nostrand Reinhold, 1990.

- [DUB80] DUBOIS, S.R.; GLANZ, F.H.; *An Autoregressive Model Approach to Two-Dimensional Shape Classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, N° 8, p. 55-66, 1980
- [EIS93] EISENBACK, J.D.; *Interactive Multimedia for Teaching Nematology*, Journal of Nematology, Vol. 25, N° 3, p. 367-373, september, 1993
- [ESE84] ESEHERA, M.A.; FU, K.S.; *A Graph Distance Measure for Image Analysis*, IEEE Transactions on Systems, Man and Cybernetics vol 14 N° 3, pp. 398-408, Maio/Junho 1984
- [FED66] FEDER, J.; FREEMAN, H.; *Digital Curve Matching Using a Contour Correlation Algorithm*, IEEE Int. Conv. Rec. Pt. 3, p.69-85, 1966
- [FER92a] FERRAZ, S.; MENDES, M.L.; *O Nematóide das Galhas*, in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 42-45, 1992
- [FER92b] FERRAZ, S.; MENDES, M.L.; *Nematóides do Feijoeiro*, in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 46-47, 1992
- [FIL90] FILHO, E.A.; *Iniciação à Lógica Matemática*, Ed. Nobel, São Paulo 16ª edição, 1986, 4ª reimpressão, 1990
- [FOR72] FORSYTHE, A.; *Ciência de Computadores - Volume 2*, Ao Livro Técnico S.A., 1972
- [FRA92] FRANCL, L.J.; *A Hypertext Glossary of Nematological Terms*, Journal of Nematology, Vol. 24, N° 4, december, 1992
- [FRE61] FREEMAN, H.; *On The Encoding of Arbitrary Geometric Configurations*, IEEE Transactions on Electronic Computers, EC-10, p. 260-268, 1961
- [FRE64] FREEMAN, H.; *Apictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern recognition*, IEEE Transactions on Electronic Computers, EC-13, p. 118-127, 1964

- [FU76] FU, K.S.; ROSENFELD, A.; *Pattern Recognition and Image Processing*, IEEE Transactions on Computers, Vol. C-25 N° 12, p. 1336-1345 (Dez/76)
- [FU82] FU, K.S.; *Syntactic Pattern Recognition and Applications*, Englewoods Cliffs, New Jersey: Prentice-Hall, 1982
- [FUK91] FUKUNAGA, K.; MURATA, H.; ASANO, T.; IZUMI, M.; *Image Registration Using an Image Graph and its Application to Map Matching*, IEE Proceedings-E, Vol. 138, N° 2, pp. 79-84, Março/1991
- [GOM83] GOMES, N.G.; *Introdução à Lógica Matemática*; apostila utilizada no curso de Lógica da Universidade de Brasília, 2° semestre de 1983
- [GOM92] GOMES, N.G.; apostila utilizada no curso de Lógica da Universidade de Brasília, 2° semestre de 1992
- [GON92] GONÇALVES, W; *Melhoramento do Cafeeiro Visando a Resistência a Nematóides in Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 72-77, 1992
- [GOW71] GOWER, J.C.; *A General Coefficient of Similarity and Some of its Properties*, Biometrics, N° 27 p. 857-871, 1971
- [GRI94] GRIMSON, W.E.L.; MUNDY, J.L.; *Computer Vision Applications*, Communications of the ACM, Vol 37 N° 3, Março/1994, p.45-51
- [GUP85] GUPTA, M.M.; *Aproximate Reasoning In Expert Systems*. New York, Elsevier Science Publishing Company Inc, 1985
- [HAM92] HAMILTON, R.J.; PRINGLE, R.D.; GRANT, P.M.; *Syntactic techniques for pattern recognition on sampled data signals*, IEE Proceedings-E, Vol. 139 N° 2, p.156-164, Março/1992
- [HOR76] HOROWITZ, S.L.; PAVLIDIS, T.; *Picture segmentation by a Tree Traversal Algorithm*, Journal of the Association for Computing Machinery, Vol. 23, N° 2, Abril/1976, p.368-388

- [HUA92] HUANG, S.P.; *Nematóides que Atacam Olerícolas e seu Controle*, in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 31-36, 1992
- [JAI94] JAIN, A.K.; MAO, J.; *Neural Networks and Pattern Recognition*, p. 194-212, in *Computational Intelligence - Imitating Life*, Piscataway, New Jersey/USA, IEEE Press, 1994
- [KIT79] KITCHEN, L.; ROSENFELD, A.; *Discrete Relaxation for Matching Relational Structures*, IEEE Transactions on Systems, Man and Cybernetics vol 9 N° 12, p.869-874, Dezembro1979
- [KIT80] KITCHEN, L.; *Relaxation Applied to Matching Quantitative Relational Structures*, IEEE Transactions on Systems, Man and Cybernetics vol 10 N° 2, 96-101, Fevereiro1980
- [LIM86] LIMA, R.D.; FERRAZ, S.; SANTOS, J.M.; *Problemas Causados por Fitonematóides em Bananeira*, in *Informe Agropecuário*, Vol. 12 N° 133 p. 55-60, 1986
- [LIM92] LIMA, R.D., *Sintomatologia e Diagnose de Nematóides*, in *Informe Agropecuário*, Vol. 16 N° 172 p. 17-22, 1992
- [LIP72] LIPSCHUTZ, S.; *Teoria dos Conjuntos*, São Paulo, McGraw-Hill (coleção Schaum), 1972
- [LOR82] LORDELLO, L.G.E., *Nematóides das Plantas Cultivadas*, 7ª Ed., São Paulo, Ed. Nobel, 1982
- [LU78] LU, S-Y.; FU, K.S.; *A Sentence-to-Sentence Clustering Procedure for Pattern Analysis*, IEEE Transactions on Systems, Man and Cybernetics vol 8 N° 5, p.381-3891978
- [MAI75] MAI, W.F.; *Pictorial Key to Genera of Plant-Parasitic Nematodes*, 4ª ed., Cornell University Press, 1975

- [MAS74] MASSEY, C.L.; *Biology and Taxionomy of Nematode Parasites and Associates od Bark Beetles in the United States*, Agriculture Handbook No. 446, Forest Service, United States Departament of Agriculture, March 1974
- [MEH78] MEHTA, Y.R.; *Doenças do Trigo e seu Controle*, Ed. Agronômica ceres LTDA, São Paulo, 1978
- [MEH84] MEHLHORN, K.; *Multi-dimensional Searching and Computational Geometry*, Berlin-Heidelberg, Springer Verlag, 1984.
- [MEN92] MENDES, M.L.; *O Nematóide do Cisto de Soja*, p.399-416, in *Simpósio sobre Cultura de Soja nos Cerrados*, Uberaba, 1992.
- [MON92] MONTEIRO, A.R.; *Características Gerais dos Nematóides in Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 5-13, 1992
- [MOR91] MORAIS, R; *Educação em Tempos Obscuros*, São Paulo, Coleção Polêmicas do Nosso Tempo, N° 41, Ed. Cortez, 1991
- [MOU86] MOURA, P.A.M.; *Aspectos Econômicos da Cultura da Bananeira*, in *Informe Agropecuário*, Belo Horizonte, Vol. 12 N° 133 p. 3-7, 1986
- [NET87] NETO, J. J.; *Introdução à Compilação*. Rio de Janeiro, LTC - Livros Técnico e Científicos, 1987
- [NOL91] NOLT, J.; ROHATYN, D.; *Lógica*, São Paulo, McGraw-Hill (coleção Schaum), 1991
- [NOV82] NOVARETTI, W.R.T.; MIRANDA, M.A.C.; ALCÂNTARA, V.S.B.; *Tratamento Químico Visando o Controle do Nematóide de Soja* in *V Reunião Brasileira de Nematologia*, Publicação N° 5, Piracicaba-SP, 1982
- [NOV92] NOVARETTI, W.R.T.; *Nematóides em Cana-de-Açúcar e seu Controle*, in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 37-42, 1992

- [PAV74] PAVLIDIS, T., HOROWITZ, S.L., *Segmentation of Plane Curves*, IEEE Transactions on Electronic Computers, Vol. C-23, N° 8, p. 860-870, Agosto/1974
- [PEE90] PEET, F.G.; PANESAR, T.S.; SAHOTA, T.S.; SUTHERLAND, J.R.; *A Digital Image Analysis System for Comparing Groups of Small Nematodes*, Journal of Nematology Vol 22 N° 3, Julho/1990
- [ROB94] ROBBINS, R.T.; BROWN, D.J.F.; *A Computer Program "LONG-SORT" to Assist with the Diagnosis and Identification of longidorus species*, 22nd International Symposium Proceedings of the European Society of Nematologists, Ghent, Belgium, 7-12 August, 1994. in Journal of Nematology, Vol. 41, N° 3, maio/1995
- [ROE92] ROESSING, A.C.; GUEDES, L.C.A.; *Aspectos Econômicos do complexo Soja: Sua Participação na Economia Brasileira e Evolução na Região do Brasil Central*, p. 1-51 in *Simpósio sobre Cultura de Soja nos Cerrados*, Uberaba, 1992.
- [RUA92] RUANO, O.; CARNEIRO, R.G.; BRITO, J.A.; SILVA, J.F.V.; *Nematóides na Cultura do Algodoeiro*, in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 48-56, 1992
- [SAN83] SANFELIU, A.; FU, K.S.; *A Distance Measure Between Attributed Relational Graphs for Pattern Recognition*, IEEE Transactions on Systems, Man and Cybernetics vol 13 N° 3, p. 353-362, Maio/Junho 1983
- [SEK92] SEKITA, I.; KURITA, T.; OTSU, N.; *Complex Autoregressive Model for Shape Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, N° 14, p. 489-496, 1992
- [SHA92] SHAOUT, A.; SUK, M; *Distance Meausure for Attributed Fuzzy Tournaments*, IEE Proceedings-E, Vol. 139, N° 5, pp. 373-378, Setembro/1992
- [SIL92] SILVA, G.S.; *A Ponta Branca e a Rizicultura Nacional*, in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 57-59, 1992

- [SIY74] SIY,P.; CHEN, C.S.; *Fuzzy Logic for Handwritten numerical Character Recognition*, IEEE Transactions on Systems, Man and Cybernetics N° 4, p. 570-575, Nov/1974
- [TSA92] TSANG, K.M., BILLINGS, S.A., *Orthogonal Estimation Algorithm for Complex Number Systems*, International Journal Systems and Science, N° 23, p. 1011-1018, 1992
- [TSA94] TSANG, K.M.; BILLINGS, S.A.; *Two-Dimensional Pattern Analysis and Classification using a Complex Orthogonal Estimation Algorithm*, IEE Proceedings - Visual Image Signal Processing Vol. 141, N° 5, pp. 339-347, Outubro/1994
- [UHR73] UHR, L.; *Pattern Recognition, Learning and Thought*. Englewood Cliffs, New Jersey/USA, Prentice-Hall, 1973
- [VIL92] VILELA, M.R.; *Nematóides: O Fantasma da Agricultura* in *Informe Agropecuário*, Belo Horizonte, Vol. 16 N° 172 p. 3, 1992
- [WAG74] WAGNER, R.A.; *The String-to-String Correction Problem*, Journal of the Assotiation for Computing Machinery, Vol. 21, N° 1, pp. 168-173, Janeiro/1974
- [YOR92] YORINORI, J.T.; CHANCHAR, M.J.A.; NASSER, L.C.B.; HENNING, A.A.; *Doenças da Soja e Seu Controle*, p. 333-397 in *Simpósio sobre Cultura de Soja nos Cerrados*, Uberaba, 1992.
- [YOU88] YOUNG, I.T.; ROOS, R.; *Acuity: Analysis for the Personal Computer*, p. 5-15 in *Pattern Recognition and Artificial Intelligence - Towards an Integration*, Amsterdan North-Holland, 1988
- [ZAD65] ZADEH, L.A.; *Fuzzy Sets*, Information and Control, N° 8, pp. 338-353, 1965
- [ZEM82] ZEM, A.C.; ZANON, J.I.; LORDELLO, L.G.E.; *Doses e Épocas de Aplicação do Nematicida "CARBOFURAN" no Controle de Meloidogyne javanica na Cultura da Batata*, Publicação N° 5, Piracicaba-SP, 1982
- [ZIM91] ZIMMERMANN, Hans J.; *Fuzzy Set Theory and Its Apliccations* - 2nd edition. Norwell, Massachussets/USA, Kluwer Academic Publisher, 1991

ANEXO I

Fitonematóides em Cultura de Soja

A soja (*Glycine max* (L.) Merrill) é uma das mais antigas plantas cultivadas no mundo. Atualmente apresenta destacada importância entre os produtos agrícolas, devido sua utilização tanto na nutrição humana quanto na animal como fonte de óleo e proteína, além de possuir múltiplos usos industriais ([ANT92]).

A seguir seguem considerações sobre aspectos econômicos do complexo soja apresentadas por Roessing e Guedes ([ROE92]):

A Soja no Brasil e no Mundo ([ROE92])

A soja é a mais importante oleaginosa cultivada no mundo. Da produção mundial total das oito principais oleaginosas (soja, algodão, amendoim, girassol, colza, linho, copra e palma) estimada em 225,44 milhões de toneladas para 1992/93, a soja deverá participar com 51%, ou seja, 115,2 milhões de toneladas. Seu alto teor de proteínas proporcionou múltiplas utilizações e a formação de um complexo industrial destinado ao seu processamento. Levando-se em conta apenas o valor bruto da produção mundial de soja, farelo e óleo, atinge-se, em média, considerando-se a estimativa para o ano de 1993 a cifra de 25 bilhões de dólares.

A participação da agricultura no PIB brasileiro tem se situado em torno de 10% na década de 80. Em 1990, para um PIB de US\$ 440,72 bilhões, a agricultura participou com US\$ 42,13 bilhões, representando 9,5%. A produção de soja, nesse ano, atingiu um valor bruto da ordem de US\$ 4,66 bilhões, representando, portanto, 1,05% do PIB brasileiro e 11,06% do PIB agrícola. O valor bruto da produção de soja dos Estados Unidos no mesmo ano, que foi de 54,6 milhões de toneladas, representou apenas 0,2% do PIB daquele país. Isso significa que, dadas as características do estágio de desenvolvimento brasileiro, a produção de soja é mais importante no Brasil que nos Estados Unidos, embora este seja o maior produtor mundial dessa oleaginosa.

A partir de 1973, a participação do valor bruto da produção de soja brasileira no PIB agrícola foi sempre maior que 10%, chegando a 18% em 1983 e 1984. A taxa média anual de crescimento do valor bruto da produção de soja, no período de 1970/1990 foi de 6,6% ao passo que, no mesmo período, o produto interno bruto agrícola cresceu apenas 2,96% ao ano e o PIB total, 4,16%. Sem a concorrência da soja, o crescimento tanto do PIB total como do PIB agrícola teria sido menor.

Outro aspecto que chama a atenção em relação à produção brasileira de soja é a participação do seu complexo agro-industrial nas receitas das exportações brasileiras. Os produtos agrícolas, embora tenham perdido terreno para os manufaturados, ainda participam em grande parcela da receita das exportações brasileiras. Em 1991 os produtos básicos representaram 28% do total das receitas das exportações; a soja participou com 1,8% e o farelo de soja, sozinho, com 4,4%. A partir de 1973 o valor das exportações do complexo soja (grão, farelo e óleo) tem participado com mais de 20% do total das exportações dos produtos básicos, constituindo-se na principal fonte de receita cambial do setor agropecuário, ao lado do café.

O produto mais valioso desse complexo é, sem dúvida, o farelo de soja. A receita de exportação de farelo de soja no período 1965-1991 atingiu US\$ 25,09 bilhões correntes, de um total de US\$ 41,82 bilhões do complexo soja, ou seja, 60% do total do complexo nesse período.

A produção mundial de soja, cujo volume participa do mercado internacional na formação de oferta e demanda pelo produto, está restrita principalmente a três países: EUA, Brasil e Argentina. Esses países participam com 80% da produção e 90% da comercialização mundial de soja.

A produção norte-americana tem representado mais da metade da produção mundial de soja. A partir de 1983-84, essa participação declinou, entre outros, devido à maior participação do Brasil, Argentina e Paraguai no comércio mundial de soja em grão. O comércio mundial de soja apresentou declínio em alguns anos anteriores, mas também tem apresentado grande capacidade de recuperação quando a economia mundial apresenta altas taxas de crescimento. O aquecimento da demanda de carnes (bovina, suína e aves) e produtos derivados está fortemente ligada ao aumento da renda dos consumidores. Em outras palavras, a elasticidade-renda da demanda de carnes e derivados tende a ser elástica. Isso provoca um aumento da demanda de farelos de alto teor protéico, que entram na composição das rações animais. Uma influência adicional na futura tendência do comércio mundial e produção mundial de soja é a política agrícola da Rússia e dos países que faziam parte da antiga URSS, especialmente quanto à prioridade dada às importações de farelos de alto teor protéico. O nível de proteína das rações animais nesses países está bem abaixo dos padrões ocidentais, e poderá haver um esforço para suprir esta deficiência através do aumento das importações de farelos com alto teor de proteína.

A participação brasileira no comércio mundial de farelo de soja tem variado de um ano para o outro, dependendo do rendimento da produção doméstica e da política governamental interna. A participação brasileira tem sido grande desde o início dos anos 70, quando praticamente os EUA e o Brasil supriam toda a demanda mundial dos produtos de soja. Segundo estimativas, o volume líquido da comercialização mundial de soja e farelo de soja para o biênio 1995-96 deverá ficar em torno de 55 milhões de toneladas.

QUADRO I-1

Produto (Em milhões Toneladas)	1986-87	1990-91	1995-96
Trigo	76,82	95,87	105,89
Grão para alimentação animal	69,96	82,18	96,69
Soja	25,20	26,63	29,55
Farelo de soja	17,60	20,91	25,88

Quadro I-1 Volume líquido (descontadas as importações) da comercialização mundial de trigo, grãos para alimentação animal, soja e farelo de soja em 1986-87 e previsão para 1990-91 e 1995-96. Fonte: WORLD FOOD INSTITUTE (1987) in [ROE92].

Diante da possibilidade do aumento significativo da produção brasileira de soja na próxima década, a política brasileira de incremento da exportação desse produto deverá ser mais agressiva.

No Brasil, a partir do final dos anos 60, a produção de soja teve um crescimento extraordinário. Entre 1970 e 1980, a produção de soja aumentou de 1,5 milhão de toneladas para 15 milhões. Dentre os vários fatores responsáveis pelo grande aumento da produção destacam-se:

- a) Significativo aumento real do preço internacional dos produtos primários no início da década de 70.
- b) Condições favoráveis do mercado externo à comercialização da soja brasileira, que acontece justamente na entressafra norte-americana.
- c) Adaptação das cultivares oriundas do sul dos EUA na região sul do Brasil.
- d) Possibilidade de cultivo do trigo na mesma área da soja, como cultura de inverno.
- e) Disponibilidade de uma estrutura cooperativista, montada no Sul do país.
- f) Aumento progressivo da capacidade de esmagamento da soja, resultante da necessidade de abastecer o mercado interno com óleos vegetais comestíveis, e da política brasileira de exportação, incentivando a exportação de produtos industrializados ou semi-industrializados, tal como o farelo de soja.
- g) Rápido crescimento da avicultura brasileira no final da década de 60 e início dos anos 70, quando foram adotadas modernas tecnologias de produção de aves de corte e para produção

de ovos. Isso reforçou a procura de farelos com alto teor protéico para composição das rações animais.

- h) Grande redução, no início dos anos 70, da produção mundial da farinha de peixe que, pelo seu alto valor protéico, é muito utilizada na composição de rações para animais.
- i) Incentivo da política agrícola brasileira, na década de 70, ao crescimento da produção.
- j) Apoio da pesquisa e da assistência técnica, possibilitando a criação de cultivares adaptadas a diferentes regiões climáticas do país e aumento da produtividade nas áreas já cultivadas tradicionalmente.
- k) Pagamento, pelo governo, aos produtores de café para que erradicassem suas plantações e as substituíssem por uma cultura de sua escolha ante a excessiva produção de café do início dos anos 60.

Até o início dos anos 80 a produção de soja no Brasil se concentrou na região Centro-Sul. A partir daí, a participação da região Centro-Oeste aumentou significativamente. De uma área de 1,29 milhão de hectares e uma produção de 2,2 milhões de toneladas em 1980, representando 14,51% da produção nacional, a região Centro-Oeste passou a ter 5,136 milhões de hectares com uma produção de 10,57 milhões de toneladas em 1989. nesse ano, a região representou 44,52% da produção nacional de soja.

Analisando-se o PIB e PIB agrícola dos estados brasileiros da fronteira agrícola do Centro-Oeste, percebe-se a importância que a cultura da soja passa a ter como fonte de receita.

QUADRO I-2

Estado		1970	1975	1980	1985
	Em US\$ bilhões				
MA	PIB	0,91	1,63	2,49	2,40
	PIB agrícola	0,39	0,69	0,79	0,35
	VBPS	-	-	-	-
BA	PIB	4,20	8,27	12,76	12,78
	PIB Agrícola	0,97	1,70	2,09	1,79
	VBPS	-	-	-	-
MG	PIB	9,13	18,87	27,75	24,72

	PIB Agrícola	1,64	3,23	4,94	4,29
	VBPS	-	0,02	0,07	0,16
GO	PIB	1,67	3,41	5,03	5,27
	PIB Agrícola	0,58	1,28	1,35	1,01
	VBPS	-	0,02	0,12	0,24
MT	PIB	1,20	0,95	1,79	2,17
	PIB Agrícola	0,44	0,24	0,44	0,30
	VBPS	-	-	0,03	0,20
MS	PIB	-	5,39	7,42	6,45
	PIB Agrícola	-	1,96	3,28	1,78
	VBPS	-	0,08	0,34	0,44

Quadro 1-2 PIB, PIB Agrícola e Valor Bruto da Produção de Soja (VBPS). Os valores foram transformados em dólares e depois deflacionados pelo "wholesale prices" para valores constantes de 1991. O VBPS foi calculado com base na produção de soja em cada ano, multiplicado pelo preço médio da soja "CIF Rotterdam" e descontado em média 20% de transporte e taxas diversas que não permanecem no Estado produtor. Fonte: IBGE (1991) in [ROE92]

Analisando-se o quadro 4 conclui-se que, sem sombra de dúvida, a cultura da soja é mais importante para os estados de Mato Grosso, Mato Grosso do Sul e Goiás. No estado de Goiás, de uma participação no PIB estadual de 2,39% em 1980, passou para 4,55% em 1985. Em Mato Grosso do Sul, de 4,58% em 1980 subiu para 6,82% do PIB estadual em 1985. Finalmente em Mato Grosso passou de 1,68% em 1980 para 9,22% em 1985.

Outro importante indicador do papel que a cultura da soja desempenha na região diz respeito ao ICMS. A arrecadação desse tributo em 1989, proveniente da comercialização da soja, esteve entre US\$ 50-100 milhões em cada um dos estados da região Centro-Oeste. Estimativas da FGV indicam que para cada 1,5 milhão de hectares haverá 50 mil empregos diretos e cerca de 100 mil indiretos com a ocupação agrícola dos cerrados. Dessa forma pode-se concluir que a soja contribuiu, nesta região, em 1989, com 357.100 empregos diretos e 714.200 empregos indiretos.

A soja é um produto agrícola de curto canal de comercialização e exportação. Um aspecto de destaque na sua comercialização é que não requer beneficiamento especial após a colheita. Alia-se isso a boa localização das firmas industriais, nas zonas de maior produção. Grande parte da indústria está

localizada junto às principais zonas produtoras, e o produto é adquirido diretamente do produtor. Nas mãos dos compradores (cooperativas e indústrias) a soja é processada e transformada nos seus produtos mais nobres, o farelo e o óleo.

A indústria brasileira produz o farelo de soja, que é ofertado aos mercados interno e externo e utilizado principalmente na elaboração de rações para alimentação animal. A demanda do farelo de soja é, portanto, derivada da demanda de carnes de aves, suínos, e bovinos. O óleo de soja abastece o mercado interno em alta quantidade, em torno de 90% da quantidade oferecida.

O espetacular desenvolvimento desta cultura, envolvendo todo o complexo agro-industrial, se dá sob severas críticas originárias de diferentes segmentos sociais. As posições de grupos ambientalistas, intelectuais de diferentes matizes, partidos políticos, confederações de trabalhadores rurais, movimentos sobre reforma agrária e setores governamentais, têm criticado, de maneira contundente, as possíveis consequências decorrentes do desempenho da economia da soja relacionadas a: degradação ambiental, concentração da terra e da renda, substituição de culturas alimentares, desemprego na mão-de-obra e outros temas.

Algumas verdades foram ditas, porém não foram revistas em função da evolução tecnológica verificada no sistema de produção da cultura. Significa dizer que em determinado momento a crítica procedia, hoje não mais. Muitos mitos foram criados por desinformação ou por posições, aprioristicamente, assumidas como verdadeiras.

Se for possível desvestir-se dos preconceitos e analisar de maneira imparcial o papel desempenhado pela lavoura de soja e a contribuição que o setor público da pesquisa agropecuária ofereceu para viabilização desta cultura, adaptando-a às diversas condições edafo-climáticas do país e sócio-econômicas do produtor, poder-se-á, facilmente, concluir que os benefícios sociais foram muito superiores aos custos decorrentes da alocação de fatores para a produção de soja.

Assim, fundamentado num cenário conservador de evolução da taxa de expansão da economia, é necessário prover medidas que assegurem um crescimento, para a produção de soja, em torno de 4% ao ano.

Doenças de soja ([YOR92])

Na safra de 1989/90 (produção de 20,1 milhões de toneladas) foi possível estimar uma perda de 5 milhões de toneladas ou o equivalente a US\$ 830 milhões. Isso explica, em parte, a baixa produtividade média da soja no país (1800 kg/ha). As perdas certamente serão maiores se os danos causados por outras doenças (principalmente o cancro da haste, a antracnose, os nematóides das galhas e a podridão de Sclerotina) e as reduções de qualidade de germinação forem acrescentadas. Além disso

as doenças reduzem o aproveitamento dos fertilizantes e de outros insumos, aumentando o custo de produção.

Entre as doenças de soja identificadas no Brasil despontam, doenças fúngicas, bacterianas, causadas por vírus, e por nematóides.

As principais doenças fúngicas da soja são:

- a) Crestamento foliar e mancha púrpura da semente (*Cercospora kikuchii*)
- b) Mancha foliar de *Alternaria* (*Alternaria* sp.)
- c) Mancha foliar de *Ascochyta* (*Ascochyta* sp.)
- d) Mancha parda (*Septoria glycines*)
- e) Mancha "olho-de-rã" (*Cercospora sojinai*)
- f) Mancha foliar de *Myrothecium* (*Myrothecium roridum*)
- g) Oídio (*Microsphaera diffusa*)
- h) Ferrugem (*Phakopsora pachyrhizi*)
- i) Mildio (*Pernospora manshurica*)
- j) Mancha foliar de *Phyllosticta* (*Phyllosticta* sp.)
- k) Mancha alvo e podridão da raiz (*Corynespora cassiicola*)
- l) Antracnose (*Colletotrichum dematium* var. *truncata*)
- m) Seca da haste e da vagem (*Phomopsis* spp.)
- n) Seca da vagem (*Fusarium* spp.)
- o) Mancha de levedura (*Nematospora corily*)
- p) Podridão branca da haste (*Phialophora gregata*)
- q) Cancro da haste (*Diaphorte phaseolorum*; *Phomopsis phaseoli*)
- r) Podridão negra da raiz (*Macrophomina phaseolina*)
- s) Podridão radicular de *Cylindrocladium* (*Cylindrocladium clavatum*)
- t) Tombamento e murcha de *Sclerotium* (*Sclerotium rolfsii*)
- u) Podridão radicular de *Rosellinia* (*Rosellinia* sp.)

As principais doenças bacterianas da soja são:

- a) Crestamento bacteriano (*Pseudomonas syringae*)
- b) Pústula bacteriana (*Xanthomonas campestris*)
- c) Fogo selvagem (*Pseudomonas syringae*)

As principais doenças causadas por vírus são

- a) Mosaico comum da soja (SMV)
- b) Queima do broto (TRSV)
- c) Mosaico amarelo do feijoeiro (BYMV)
- d) Mosaico cálico (AMV)

As principais doenças causadas por nematóides são

- a) Nematóides de galhas (*Meloidogyne incognita*; *Meloidogyne javanica*; *Meloidogyne arenaria*)
- b) Nematóide de cisto (*Heterodera glycines*)

A importância dos nematóides como parasitas da soja pode ser percebida na colocação de YORINORI ([YOR92]):

"A recente descoberta (safra 1991/92) do nematóide de cisto (*Heterodera glycines* Ichinohe), um dos mais temidos inimigos da soja na região dos cerrados, traz um novo desafio para a pesquisa e a cultura de soja no Brasil."

Nematóide de cisto da soja

Mendes ([MEN92]) tece comentários a respeito do nematóide de cisto de soja. O que segue abaixo é uma exposição de colocações da autora acerca do problema.

As perdas de produção em soja, devido aos nematóides, variam de insignificantes a 100%. No Brasil, estima-se que estas perdas sejam da ordem de 15%, excetuando-se aquelas ocasionadas por *Heterodera glycines*, o nematóide de cisto. O nematóide de cisto é considerado o problema mais importante para a cultura de soja nos EUA, desde sua descoberta em 1954. No Brasil, esta espécie foi identificada na safra 1991/1992 na Região Central, no Mato Grosso, Mato Grosso do Sul e em Minas Gerais e Goiás.

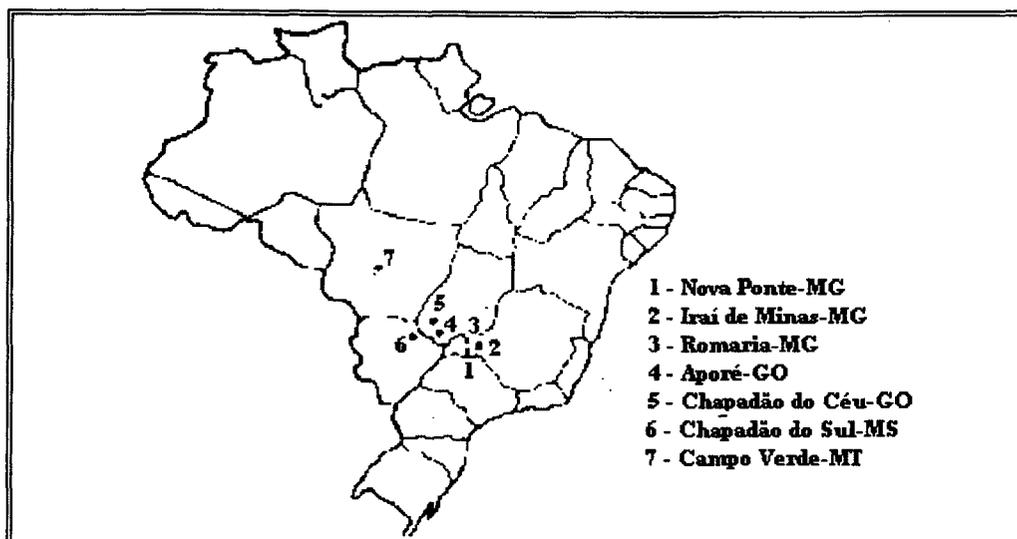


Fig. I-1.1 - Locais onde o nematóide de cisto de soja, *Heterodera glycines*, foi identificado no Brasil, na safra 1991/1992. Fonte: [MEN92]

É este nematóide que já ocorria no Japão desde o século XIX, causando a doença conhecida como "nanismo amarelo da soja". No Brasil, o nematóide foi diagnosticado na safra de 1991/92, em amostras de solo e raízes de soja provenientes de Nova Ponte-MG, Campo Verde-MT e Chapadão do Céu-GO e descrito inicialmente como *Heterodera* spp. Logo depois este nematóide foi identificado também em Aporé-GO, Iraí de Minas-MG, e Romaria-MG. A origem deste nematóide no Brasil é desconhecida.

O gênero *Heterodera* é caracterizado pela formação de cistos. O cisto é o corpo da fêmea adulta morta, cheio de ovos, o qual, no final do ciclo de vida torna-se um envoltório protetor, de cor marrom, altamente resistente. Devido a esta resistência, o cisto constitui a unidade de dispersão mais eficiente. Isto permite que o nematóide seja facilmente levado de uma área para outra, a curtas ou longas distâncias, por qualquer método que envolva movimento de solo. Também através das sementes de soja, ou outra espécie vegetal, provenientes de áreas infestadas podem conter torrões de cistos e serem portanto responsáveis pela introdução do patógeno em áreas onde ainda não ocorre.

Recentemente, o jornal "Correio Brasiliense", de Brasília, publicou uma nota a respeito do problema causado pelo nematóide de cisto nas plantações de soja. Tal nota é apresentada abaixo na íntegra:

QUADRO I-3

Mata Verde

Nematóide de cisto. Este é o nome da doença que mais preocupa o ministério da Agricultura atualmente. Mal de origem, característica e manifestação desconhecida, o nematóide representa uma ameaça mortal para a soja, que rende até US\$ 4 bilhões na pauta de exportação do Brasil - o segundo maior produtor mundial, com quase 25 milhões de toneladas em 1994.

Em 1992, seis municípios do Centro-Oeste apareceram infectados com o nematóide. Subiu para 13 em 1993, 28 em 1994 e contaminou agora 46 municípios de Minas, Mato Grosso, Rio Grande do Sul e São Paulo.

"O nematóide é a Aids da soja. Se atingir 100 municípios eu perco o controle da doença", adverte o ministro José Eduardo Andrade Vieira, que aguarda o exame da EMBRAPA, semana que vem, para iniciar a guerra para salvar a soja.

Quadro I-3 Nota sobre nematóide de cisto. Fonte: Jornal "Correio Brasiliense", Brasília-DF, outubro de 1995

Outros nematóides parasitas da soja

Fitonematóides causadores de galhas já encontrados em soja pertencente ao gênero *Meloidogyne* e espécimes típicos são: *Meloidogyne javanica*, *Meloidogyne incognita*, *Meloidogyne arenaria*, *Meloidogyne hapla*, *Meloidogyne bauruensis* e *Meloidogyne inornata*. Entretanto, as espécies *Meloidogyne javanica* e *Meloidogyne incognita* são as mais freqüentes na cultura de soja. ([ANT92]).

No ano agrícola de 1987/88 o laboratório de Nematologia da Empresa Brasileira de Pesquisa Agropecuária - Centro Nacional de Pesquisa de Soja (EMBRAPA-CNPSO) recebeu para análise nematológica 58 amostras de raízes de soja de várias regiões produtoras do Brasil. Quarenta e seis dessas amostras continham *M. javanica*, e nove continham *M. incognita*. Além deste gênero, outros gêneros também ocorrem em soja: *Pratylenchus* spp, *Helicotylenchus* spp., *Criconemella* spp., *Xiphinema* spp. e *Trichodorus* spp. ([ANT92]).

Segundo Antônio ([ANT92]), já em 1952 se alertava para o problema de nematóides das galhas em lavouras de soja, no estado de São Paulo, e consideravam-no na época, o único fator limitante da

ANEXO II

Um Exemplo do Cálculo da Distância Entre 2 Grafos AFT

Para apresentar a marcha (quase) completa dos cálculos envolvidos na distância D entre 2 AFT's $T1$ e $T2$ utilizam-se duas figuras bem simples, para que se possa entender com maior clareza as etapas inerentes aos cálculos. Além disso ressalta-se que este é um exemplo baseado nas características propostas no modelo de identificação de nematóides através da estrutura do estilete, ou seja, pode-se supor que os padrões apresentados neste exemplo, sejam dois tipos distintos de estilete.

Seja a tarefa de se comparar dois padrões, $P1$ e $P2$, apresentandos nas figuras AII.1, AII.2 e AII.3:

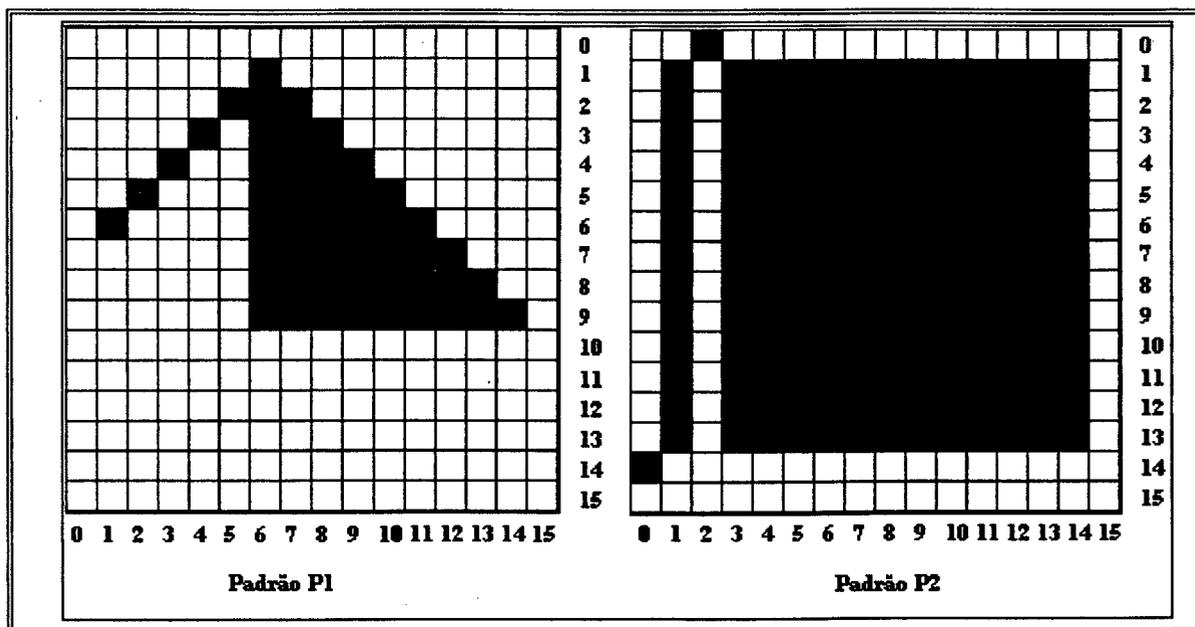


Fig. AII.1 - Padrões P1 e P2 a serem comparados pelo modelo de grafo AFT

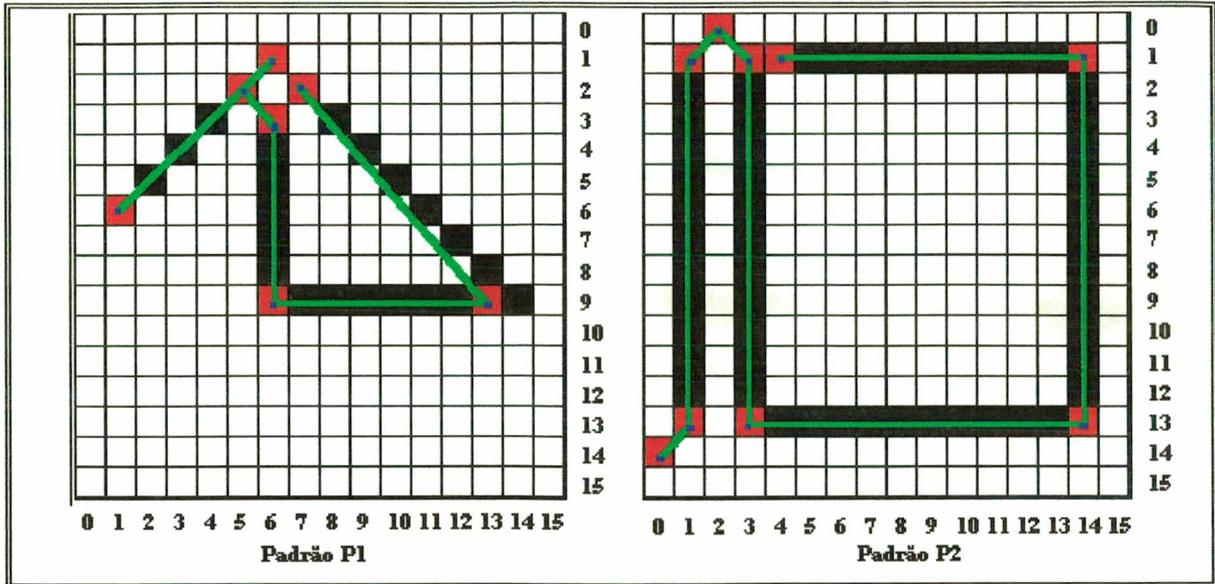


Fig. AII.2 - O polígono representativo do contorno. O padrão 1 apresenta 6 arestas, enquanto o padrão 2 possui 8.

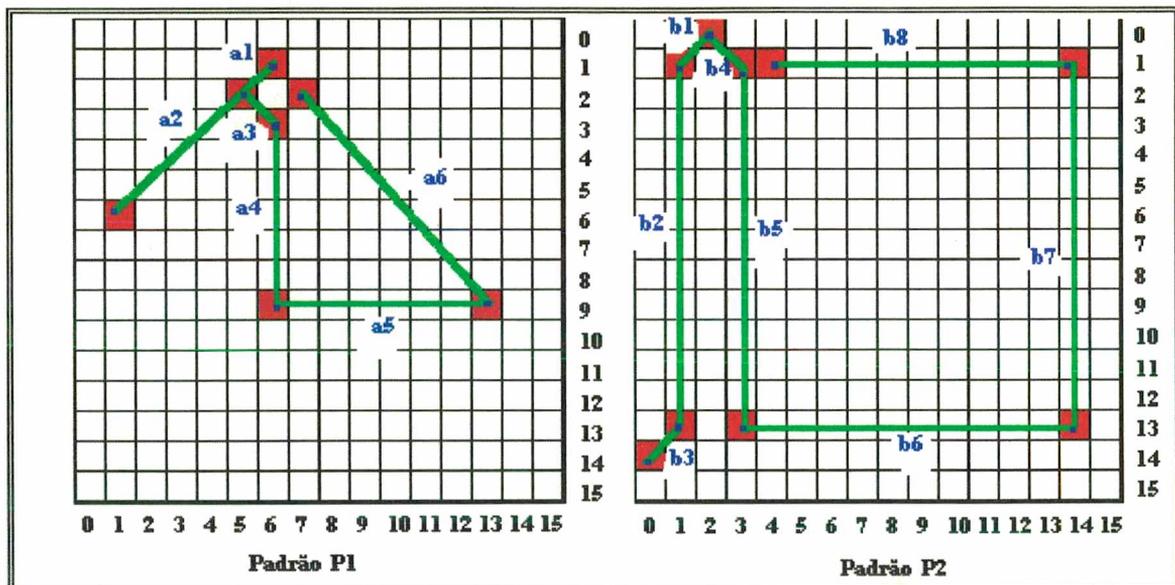


Fig. AII.3 - Um realce nas arestas dos padrões a serem comparados.

O grafo AFT $T1 = (V1, A1, B1, E1, X1, C1)$ para o Padrão 1

Seja $T1 = (V1, A1, B1, E1, X1, C1)$ onde

$$V1 = \{ a_1, a_2, a_3, a_4, a_5, a_6 \}$$

$$A1 = \{ p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15} \}$$

$$B1 = B \text{ (definido na seção 4.8)}$$

$$E1 = E \text{ (definido na seção 4.8)}$$

$$X1 = X \text{ (definido na seção 4.8)}$$

$$C1 = C \text{ (definido na seção 4.8)}$$

O grafo AFT $T2 = (V2, A2, B2, E2, X2, C2)$ para o Padrão 2

Seja $T2 = (V2, A2, B2, E2, X2, C2)$ onde

$$V2 = \{ b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8 \}$$

$$A2 = \{ q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, \\ q_{16}, q_{17}, q_{18}, q_{19}, q_{20}, q_{21}, q_{22}, q_{23}, q_{24}, q_{25}, q_{26}, q_{27}, q_{28} \}$$

$$B2 = B \text{ (definido na seção 4.8)}$$

$$E2 = E \text{ (definido na seção 4.8)}$$

$$X2 = X \text{ (definido na seção 4.8)}$$

$$C2 = C \text{ (definido na seção 4.8)}$$

As Unidades dos grafos AFT T1 e T2

Unidades de T1 (Padrão 1)	Unidades de T2 (Padrão 2)
$U_1 = (a_1, p_1, a_2)$	$U_1 = (b_1, q_1, b_2)$
$U_2 = (a_1, p_2, a_3)$	$U_2 = (b_1, q_2, b_3)$
$U_3 = (a_1, p_3, a_4)$	$U_3 = (b_1, q_3, b_4)$
$U_4 = (a_1, p_4, a_5)$	$U_4 = (b_1, q_4, b_5)$
$U_5 = (a_1, p_5, a_6)$	$U_5 = (b_1, q_5, b_6)$
$U_6 = (a_2, p_6, a_3)$	$U_6 = (b_1, q_6, b_7)$
$U_7 = (a_2, p_7, a_4)$	$U_7 = (b_1, q_7, b_8)$
$U_8 = (a_2, p_8, a_5)$	$U_8 = (b_2, q_8, b_3)$
$U_9 = (a_2, p_9, a_6)$	$U_9 = (b_2, q_9, b_4)$
$U_{10} = (a_3, p_{10}, a_4)$	$U_{10} = (b_2, q_{10}, b_5)$
$U_{11} = (a_3, p_{11}, a_5)$	$U_{11} = (b_2, q_{11}, b_6)$
$U_{12} = (a_3, p_{12}, a_6)$	$U_{12} = (b_2, q_{12}, b_7)$
$U_{13} = (a_4, p_{13}, a_5)$	$U_{13} = (b_2, q_{13}, b_8)$
$U_{14} = (a_4, p_{14}, a_6)$	$U_{14} = (b_3, q_{14}, b_4)$
$U_{15} = (a_5, p_{15}, a_6)$	$U_{15} = (b_3, q_{15}, b_5)$
	$U_{16} = (b_3, q_{16}, b_6)$
	$U_{17} = (b_3, q_{17}, b_7)$
	$U_{18} = (b_3, q_{18}, b_8)$
	$U_{19} = (b_4, q_{19}, b_5)$
	$U_{20} = (b_4, q_{20}, b_6)$
	$U_{21} = (b_4, q_{21}, b_7)$
	$U_{22} = (b_4, q_{22}, b_8)$
	$U_{23} = (b_5, q_{23}, b_6)$
	$U_{24} = (b_5, q_{24}, b_7)$
	$U_{25} = (b_5, q_{25}, b_8)$
	$U_{26} = (b_6, q_{26}, b_7)$
	$U_{27} = (b_6, q_{27}, b_8)$
	$U_{28} = (b_7, q_{28}, b_8)$

Operadores envolvidos para o cálculo da distância $D(T_1, T_2)$

A base para este trabalho é o modelo de grafo AFT . Shaout ([SHA92]) apresenta os conceitos e o exemplo de uma implementação de grafo AFT para reconhecer caracteres de imprensa escritos à mão. Para o modelo, 4 cálculos-chave estão envolvidos:

- O cálculo da distância D_n entre 2 nodos e D_a entre 2 arcos
- O cálculo da distância D_c entre 2 complexos
- O cálculo da distância D_u entre 2 unidades e
- O cálculo da distância D entre 2 AFT's .

Sejam

- $U(v,p,w) \in T_1$,
- $U(u,q,z) \in T_2$

então

		Natureza do operador
a)	$D_n(v,u) = d_n [X1(v), X2(u)]$	$d_n =$ Distância de Hamming
	$D_a(p,q) = d_a [C1(p), C2(q)]$	$d_a =$ Distância de Hamming
b)	$D_c [C(v,p,w), C(u,q,z)] = f_c [D_n(v,u), D_a(p,q), D_n(w,z)]$	$f_c = \max$
c)	$D_u [U(v,p,w), U(u,q,z)] = f_u [D_c [C(v,p,w), C(u,q,z)],$ $D_c [C(v,p,w), C(z,q',u)],$ $D_c [C(w,p',v), C(u,q,z)],$ $D_c [C(w,p',v), C(z,q',u)]]$	$f_u = \min$
d)	$D(T_1, T_2) = f_1 \prod_{p \in A1} f_2 \prod_{q \in A2} d_u [U(v,p,w), U(u,q,z)]$	$f_1 = \max$ e $f_2 = \min$

O que se obtém com a aplicação destes operadores é uma estrutura "rígida" de comparação. O operador \max para o cálculo da distância entre 2 complexos é o causador de tal rigidez. A proposta é: de posse de uma unidade do padrão que se quer determinar, então procurar esta unidade no padrão de *template*. Se não encontrar a unidade, o padrão será considerado *totalmente* diferente ($D = 1$),

justamente devido a rigidez do operador max para f_c , combinado com as operações $f_2 = \min$ e $f_1 = \max$ no cálculo de D .

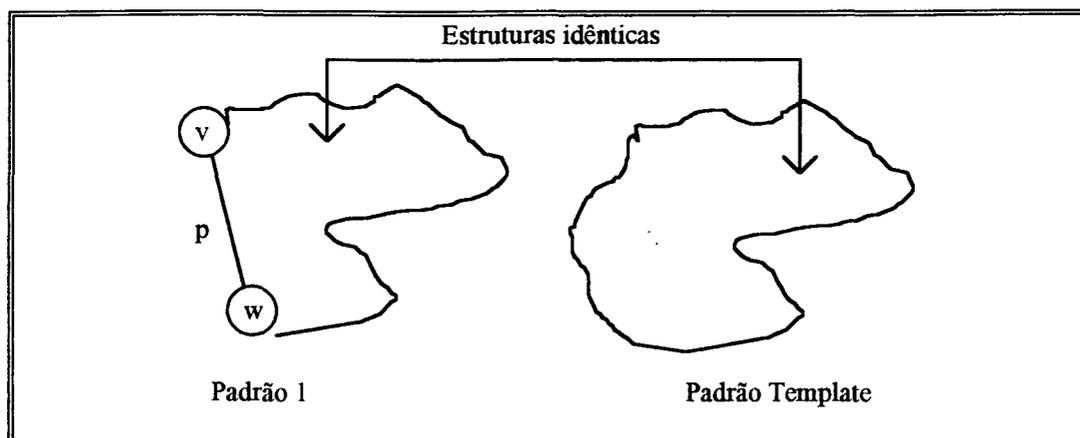


Fig. AII.4 - Unidade $U(v,p,w)$ do padrão 1 é a única não encontrada no padrão template. No resto, toda a estrutura é idêntica

Na figura AII.4, a unidade $U(v,p,w)$ confrontada com todas as unidades do padrão template apresentará a distância máxima que vale 1. E é esta a distância mínima de $U(v,p,w)$ (operação $f_2 = \min$). Ao se efetuar a operação $f_1 = \max$, a maior distância será 1, indicando que o padrão é completamente diferente do padrão *template*.

Para a situação de exemplo descrita por Shaout ([SHA92]), onde se deseja reconhecer caracteres de imprensa escritos à mão, a "rigidez" do operador max na distância entre dois complexos é útil, pois nos casos de caracteres escritos à mão, a ausência de uma unidade no cenário implica em alta probabilidade do padrão em teste não "concordar" com o padrão *template*, e a rejeição total é ideal.

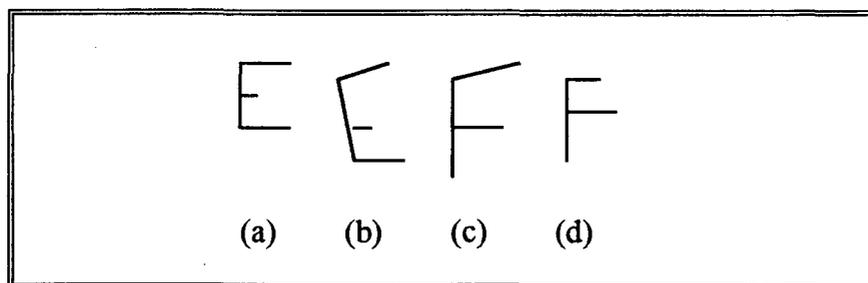


Fig. AII.5 - Padrões utilizados na aplicação proposta por Shaout ([SHA92])

Ao se construir os grafos AFT's para os cenários (a), (b), (c) e (d) da figura AII.5, poderá se perceber que graças à rigidez do operador max no cálculo da distância entre dois complexos, será

evitado tomar o padrão (a) como sendo uma instância do padrão (c), pois a unidade que representaria a terceira "perna" da letra "E" não será encontrada em (c), indicando distância 1.

Na figura AII.1, o padrão em formato triangular, apesar de não ser igual ao padrão de formato quadrado, pode ser considerado como tendo uma "ligeira" semelhança à este. Por isso, no *template matching* entre ambos, espera-se que a distância D assuma algum valor "alto", porém não o valor 1 (que representa uma situação de completa diferença). Só que, se forem submetidos à comparação, utilizando os operadores citados acima, a distância indicará exatamente 1.

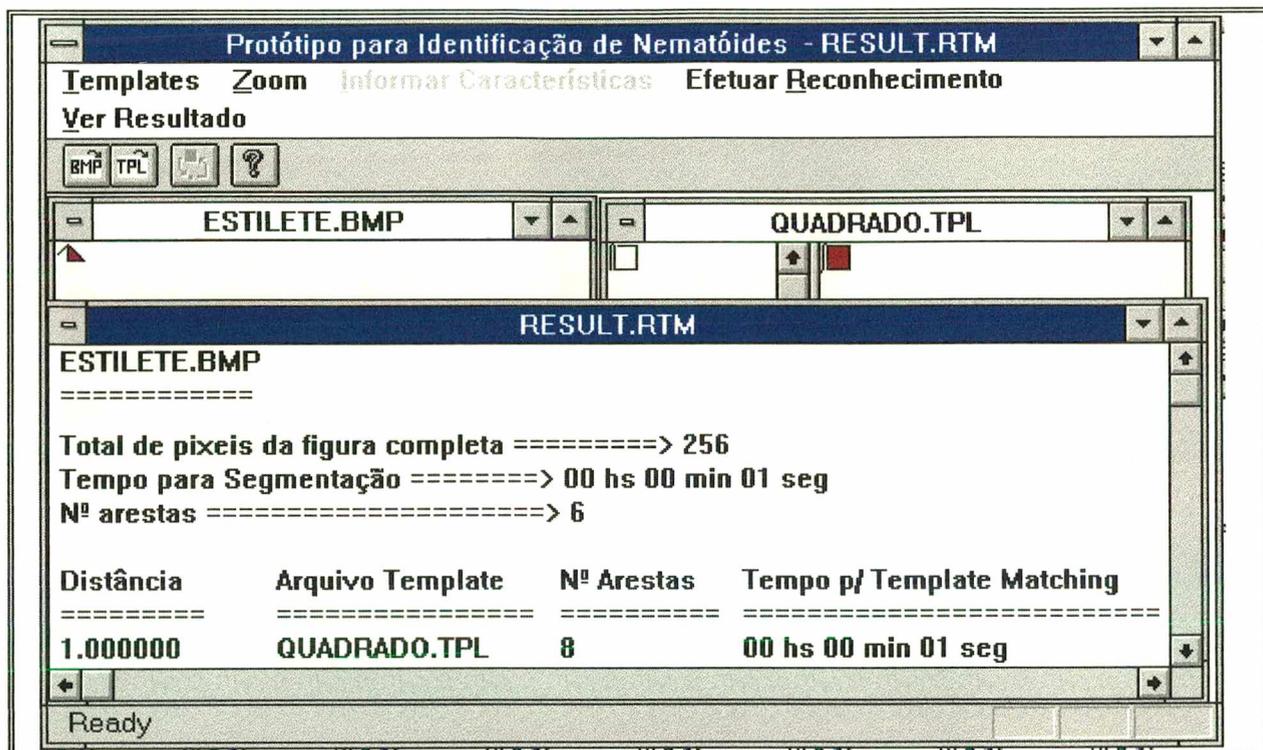


Fig. AII.6 - Template Matching efetuado entre o "triângulo" (janela ESTILETE.BMP) e o "quadrado" (janela QUADRADO.BMP). Foram utilizados no cálculo, os operadores propostos por Shaout ([SHA92]) inclusive max para f_c . (Programa executado em PC-486 DX-2 66 Mhz)

Conforme se observa na figura AII.6, a distância entre o "triângulo" e o "quadrado" da figura AII.1 vale 1, quando seria mais adequado algo "acima de 0.50" sem chegar a 1 pois, conforme dito, intuitivamente os padrões apresentam ligeira semelhança. O complexo $C_1 = (a_1, p_1, a_2)$ da unidade $U_1 = (a_1, p_1, a_2) \in T1$ origina $d_c = 1$ pois esta unidade não se encontra entre as unidades de $T2$ (ver figura AII.3).

Para o problema proposto neste trabalho, a rigidez do operador max em f_c não é desejada pois, diferente de caracteres escritos à mão, uma unidade que não exista no padrão *template*, não implica em rejeição total. É necessário definir um operador adequado para f_c a fim de se evitar sua rigidez. O

problema é "visualizar" duas unidades e definir uma distância para sua similaridade de complexos, tomando-se como referencial o fato de seus nodos representarem o contorno do estilete de nematóides.

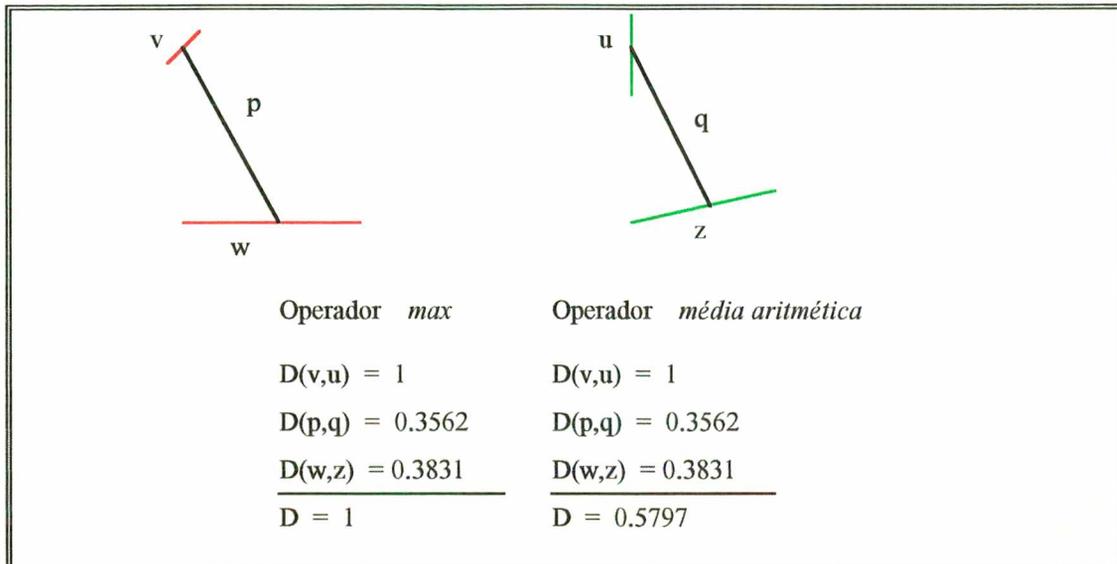


Fig. AII.7 - A distância entre 2 complexos, utilizando o operador *max* e utilizando-se a média aritmética da soma da distância dos nodos e do arco.

Uma proposta é a de utilizar, ao invés do operador *max*, a média aritmética da soma das distâncias entre os nodos e entre o arco do complexo. A idéia é distribuir um "peso" igual para cada um dos componentes da unidade, de modo que a distância esteja balanceada em função dos mesmos, e não somente em função daquele que apresentar maior distância. Para o caso do "triângulo" e do "quadrado", fazendo-se

$$D_c [C(v,p,w), C(u,q,z)] = f_c [D_n(v,u), D_a(p,q), D_n(w,z)] \text{ onde}$$

$$f_c = (D_n(v,u) + D_a(p,q) + D_n(w,z)) / 3,$$

obteve-se ao executar o programa, uma distância de 0.6595 do "triângulo" em relação ao "quadrado". E este valor reflete melhor a distância, levando-se em conta que o interesse não é caracteres escritos à mão, mas sim contorno de estilete de nematóides.

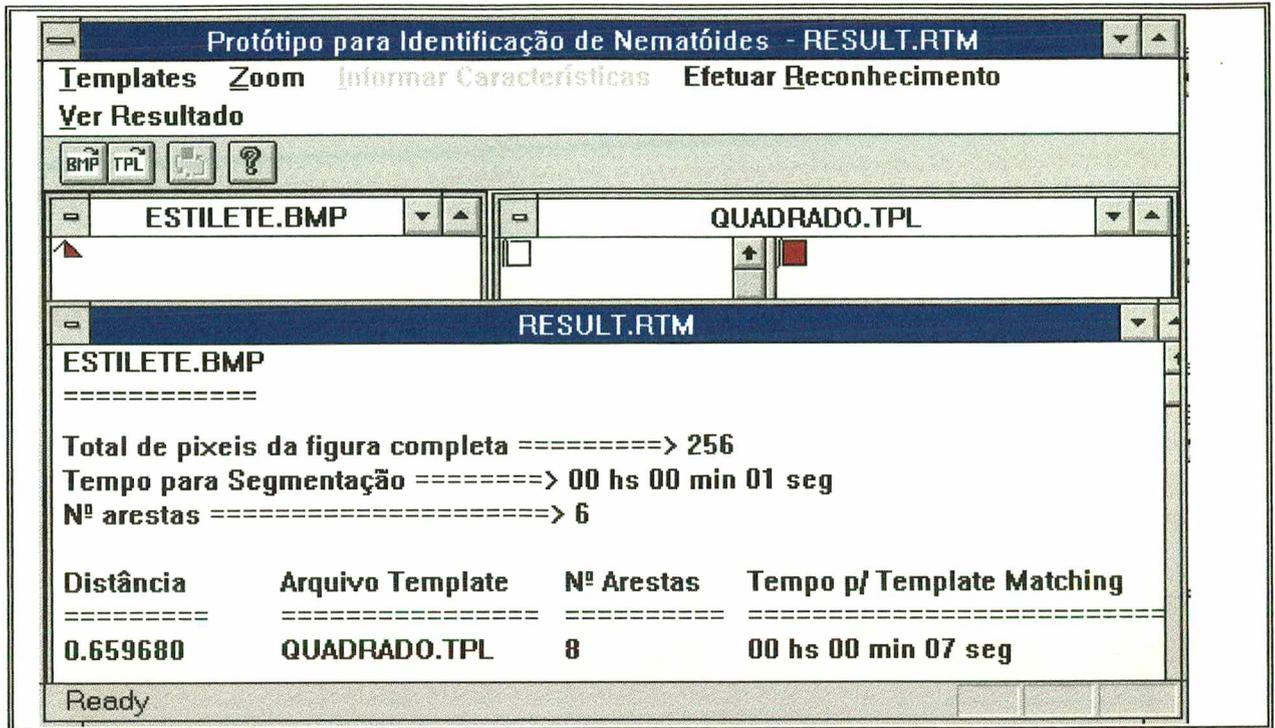


Fig. AII.8 - *Template matching* efetuado entre o "triângulo" (arquivo ESTILETE.BMP) e o "quadrado" (arquivo QUADRADO.TPL). Foram utilizados no cálculo, os operadores propostos por Shaout ([SHA92]) exceto o operador *max* para f_c . Neste caso foi efetuada a média aritmética entre as distâncias $D_n(v,u)$, $D_a(p,q)$ e $D_n(w,z)$. (Programa executado em PC-486 DX-2 66 Mhz)

Caracterização do unitário $U_5 \in T1$

Uma vez que se tenham definidos os operadores para os 4 cálculos-chave do modelo AFT, pode-se apresentar então a caracterização de uma determinada unidade de um padrão representado por grafo AFT. Escolheu-se arbitrariamente mostrar a caracterização da unidade $U_5 = (a_1, p_5, a_6) \in T1$.

Para $U_5 = (a_1, p_5, a_6) \in T1$ (ver figura AII.3) tem-se que :

a_1 = aresta definida pelos pontos (6,1) e (5,2). Ponto médio da aresta = (5.5,1.5)

a_6 = aresta definida pelos pontos (13,9) e (7,2). Ponto médio da aresta = (10,5.5)

p_5 = arco da relação entre a_1 e a_6 .

Observa-se nas figuras AII.1, AII.2 e AII.3, que a escala para o eixo das coordenadas está representado na forma matricial (tipo <linha,coluna>). A apresentação desta forma é devido o fato de o

raciocínio de todo trabalho iniciar com a representação da imagem, que no caso é um arquivo do tipo "bitmap". O formato de armazenamento no padrão "bitmap" é justamente no formato matricial, e daí as referências das figuras nesta forma. Contudo, nesta etapa de processamento é necessário visualizar *pontos* no sistema de coordenadas cartesianas (tipo <coluna,linha>), e não *elementos* de uma matriz.

Para o processamento porém, isto não é problema, requerendo apenas alguns cuidados durante a fase de manipulação destes valores. Como o arco p_5 é definido pela reta que passa pelo ponto médio das arestas a_1 e a_6 , deve-se ter o cuidado de mapear adequadamente as coordenadas (x,y) dos pontos.

Tomando como máximo para a linha o valor 15,

o ponto médio (5.5,1.5) passa a ser (5.5,(15-1.5)) = (5.5,13.5) e

o ponto médio (10,5.5) passa a ser (10,(15-5.5)) = (10,9.5).

A coordenada x (coluna) não apresenta problemas, tendo o mesmo tratamento no "formato" matricial e cartesiano. Agora, estando nesta forma, pode-se utilizar os conceitos de matemática elementar para caracterização de reta, que se baseia nos eixos de coordenadas cartesianas.

$$Y_r = mX_r + b$$

$$13.5 = 5.5m + b$$

$$9.5 = 10m + b$$

Resolvendo o sistema de equações obtém-se o coeficiente angular

$$m = \operatorname{tg} \phi = -0.8888.$$

$$\phi = \operatorname{arc} \operatorname{tg} m = -41.6335^\circ.$$

Como o sentido da reta importa e o sentido é do ponto (5.5,13.5) para (10,9.5),

$$\phi = 360 - 41.6335 = 318.366^\circ.$$

Tendo o ângulo ϕ da reta que une os dois pontos médios dos nodos a_1 e a_6 , pode-se determinar a relação difusa de *posição* para o arco p_5 :

Acima de	$\mu_{Ac} = 1 - \min \{ [90 - 318.366]/90, 1 \}$	= 0
Abaixo de	$\mu_{Ab} = 1 - \min \{ [270 - 318.366]/90, 1 \}$	= 0.4626
À Esquerda de	$\mu_E = 1 - \min \{ [180 - 318.366]/90, 1 \}$	= 0
À Direita de	$\mu_D = 1 - \min \{ \min [318.366 , 360 - 318.366]/90, 1 \}$	= 0.5374

A sentença (a_1, p_5, a_6) é lida como "a₆ está (acima de/ abaixo de/ à esquerda de/ à direita de) a₁", pois arbitrou-se, na implementação do programa que a relação refere-se à ação do nodo final sobre o nodo inicial.

A aresta a₁ (pontos (6,1) e (5,2)) é caracterizada por:

$$(6,1) = (6,15-1) = (6,14)$$

$$(5,2) = (5,15-2) = (5,13)$$

$$Y_r = mX_r + b$$

$$14 = 6m + b$$

$$13 = 5m + b$$

Resolvendo o sistema de equações tem-se o coeficiente angular

$$m = \text{tg } \theta = 1.$$

$$\theta = \text{arc tg } m = 45^\circ.$$

Como o sentido da reta importa e o sentido é do ponto (6,14) para (5,13),

$$\theta = 180 - 45 = 135 + 45 = 225^\circ.$$

A caracterização da *direção* de a₁ fica:

Horizontal	$\mu_H = 1 - \min \{ \min [225 , 180 - 225 , 360 - 225]/45, 1 \}$	= 0
Vertical	$\mu_V = 1 - \min \{ \min [90 - 225 , 270 - 225]/45, 1 \}$	= 0
Positivo	$\mu_P = 1 - \min \{ \min [45 - 225 , 225 - 225]/45, 1 \}$	= 1
Negativo	$\mu_N = 1 - \min \{ \min [135 - 225 , 315 - 225]/45, 1 \}$	= 0

Os comprimentos das arestas a_1, a_2, a_3, a_4, a_5 e a_6 , do grafo AFT T1 são

$a_1 (6,1) \rightarrow (5,2)$	$= [(6-5)^2 + (1-2)^2]^{1/2}$	$= 1.41421$
$a_2 (5,2) \rightarrow (1,6)$	$= [(5-1)^2 + (2-6)^2]^{1/2}$	$= 5.65685$
$a_3 (5,2) \rightarrow (6,3)$	$= [(5-6)^2 + (2-3)^2]^{1/2}$	$= 1.41421$
$a_4 (6,3) \rightarrow (6,9)$	$= [(6-6)^2 + (3-9)^2]^{1/2}$	$= 6.0$
$a_5 (6,9) \rightarrow (13,9)$	$= [(6-13)^2 + (9-9)^2]^{1/2}$	$= 7.0$
$a_6 (13,9) \rightarrow (7,2)$	$= [(13-7)^2 + (9-2)^2]^{1/2}$	$= 9.21954$

e a maior aresta, que é a_6 , mede $N = 9.21954$. Para a caracterização do conjunto difuso *tamanho*, o universo de discurso situa-se na faixa $[0, 9.21954]$. Para a aresta a_1 cujo tamanho $T = 1.41421$ tem-se

$$T/N = 1.41421 / 9.21954 = 0.15339 = 15.339 \% N$$

Muito Pequeno	$\mu_{OO} = (-10/N) T + 2.5,$ se $0.15*N \leq T \leq 0.25*N$	$= 0.96607$
Pequeno	$\mu_O = (10/N) T - 1.5,$ se $0.15*N \leq T \leq 0.25*N$	$= 0.03393$
Médio	$\mu_M = 0,$ se $T < 0.40*N$	
Grande	$\mu_G = 0,$ se $T < 0.65*N$	

Agora deve-se caracterizar a aresta a_1 através da interseção dos conjuntos difusos *direção* e *tamanho*. Para a operação de interseção pode-se escolher, entre outros, os operadores propostos por Hamacher, Yager, Dubois e Prade, Zimmerman e Zysno ([ZIM91]). Para maior simplicidade do modelo, no programa-protótipo foi utilizado o operador *min* para efetuar esta interseção. Daí obtém-se:

Direção		(Min)	Tamanho	
Horizontal	$= 0$		Muito Pequeno	$= 0.96607$
Vertical	$= 0$		Pequeno	$= 0.03393$
Positivo	$= 1$		Médio	$= 0$
Negativo	$= 0$		Grande	$= 0$

Horizontal-Muito Pequeno	= 0
Horizontal-Pequeno	= 0
Horizontal-Médio	= 0
Horizontal-Grande	= 0
Vertical-Muito Pequeno	= 0
Vertical-Pequeno	= 0
Vertical-Médio	= 0
Vertical-Grande	= 0
Positivo-Muito Pequeno	= 0.96607
Positivo-Pequeno	= 0.03393
Positivo-Médio	= 0
Positivo-Grande	= 0
Negativo-Muito Pequeno	= 0
Negativo-Pequeno	= 0
Negativo-Médio	= 0
Negativo-Grande	= 0

Agora deve-se aplicar procedimento semelhante para a aresta a_6 :

- Com os pontos que definem a_6 achar ângulo θ .
- Com o ângulo θ calcular a pertinência de a_6 no conjunto difuso *direção*
- Calcular o tamanho relativo de a_6
- Com o tamanho relativo, calcular a pertinência de a_6 no conjunto difuso *tamanho*
- Aplicar operador *min* para caracterizar a interseção dos conjuntos difusos *direção* e *tamanho*

Após esta operação obtém-se a caracterização do *complexo* $C_1(a_1, p_5, a_6)$ da unidade $U_5 = (a_1, p_5, a_6)$. No complexo, como se sabe, o sentido do arco p_5 importa. Para calcular o complexo $C_2(a_6, -p_5, a_1)$ da unidade $U_5 = (a_1, p_5, a_6)$, deve-se apenas efetuar uma "inversão" no grau de pertinência do arco p_5 (posição), gerando $-p_5$.

Acima de	$-\mu_{Ac} = \mu_{Ab}$	= 0.4626
Abaixo de	$-\mu_{Ab} = \mu_{Ac}$	= 0

À Esquerda de	$-\mu_E = \mu_D$	$= 0.5374$
À Direita de	$-\mu_D = \mu_E$	$= 0$

Ao finalizar esta operação tem-se os 2 complexos C_1 e C_2 de U_5 conseguindo-se portanto caracterizá-lo conforme o esperado.

Distância entre a unidade U_5 de $T1$ e a unidade U_6 de $T2$

Escolheu-se arbitrariamente duas unidades para se apresentar a marcha dos cálculos na distância entre unidades. As unidades a serem confrontadas serão $U_5 = (a_1, p_5, a_6) \in T1$ e $U_6 = (b_1, q_6, b_7) \in T2$. Para calcular a distância, 3 etapas de cálculo são necessárias:

1) Distância entre nodos e arcos

$$D_n(a_1, b_1), \quad D_n(a_1, b_7), \quad D_n(a_6, b_1), \quad D_n(a_6, b_7), \\ D_a(p_5, q_6), \quad D_a(p_5, -q_6), \quad D_a(-p_5, q_6), \quad D_a(-p_5, -q_6)$$

2) Distância entre complexos

$$D_c [C_1(a_1, p_5, a_6), C_1(b_1, q_6, b_7)] = f_c[d_n(a_1, b_1), d_a(p_5, q_6), d_n(a_6, b_7)] \\ D_c [C_1(a_1, p_5, a_6), C_2(b_7, -q_6, b_1)] = f_c[d_n(a_1, b_7), d_a(p_5, -q_6), d_n(a_6, b_1)] \\ D_c [C_2(a_6, -p_5, a_1), C_1(b_1, q_6, b_7)] = f_c[d_n(a_6, b_1), d_a(-p_5, q_6), d_n(a_1, b_7)] \\ D_c [C_2(a_6, -p_5, a_1), C_1(b_7, -q_6, b_1)] = f_c[d_n(a_6, b_7), d_a(-p_5, -q_6), d_n(a_1, b_1)]$$

3) Distância entre unidades

$$D_u [U_5(a_1, p_5, a_6), U_6(b_1, q_6, b_7)] = f_u[D_c [C_1(a_1, p_5, a_6), C_1(b_1, q_6, b_7)], \\ D_c [C_1(a_1, p_5, a_6), C_2(b_7, -q_6, b_1)], \\ D_c [C_2(a_6, -p_5, a_1), C_1(b_1, q_6, b_7)], \\ D_c [C_2(a_6, -p_5, a_1), C_1(b_7, -q_6, b_1)]]$$

As unidades $U_5 = (a_1, p_5, a_6) \in T1$ e $U_6 = (b_1, q_6, b_7) \in T2$, após serem devidamente caracterizadas, apresentam a seguinte configuração:

$U_5 = (a_1, p_5, a_6)$				
Direção-Tamanho	nodo a_1	nodo a_6	(a_6) Posição (a_1)	arco p_5
Horizontal-Muito Pequeno	= 0	= 0	Acima de	= 0
Horizontal-Pequeno	= 0	= 0	Abaixo de	= 0.4626
Horizontal-Médio	= 0	= 0	À Esquerda de	= 0
Horizontal-Grande	= 0	= 0	À Direita de	= 0.5374
Vertical-Muito Pequeno	= 0	= 0		
Vertical-Pequeno	= 0	= 0		
Vertical-Médio	= 0	= 0		
Vertical-Grande	= 0	= 0.09775		
Positivo-Muito Pequeno	= 0.96607	= 0		
Positivo-Pequeno	= 0.03393	= 0		
Positivo-Médio	= 0	= 0		
Positivo-Grande	= 0	= 0		
Negativo-Muito Pequeno	= 0	= 0		
Negativo-Pequeno	= 0	= 0		
Negativo-Médio	= 0	= 0		
Negativo-Grande	= 0	= 0.90225		

$U_6 = (b_1, q_6, b_7)$				
Direção-Tamanho	nodo b_1	nodo b_7	(b_7) Posição (b_1)	arco p_5
Horizontal-Muito Pequeno	= 0	= 0	Acima de	= 0
Horizontal-Pequeno	= 0	= 0	Abaixo de	= 0.3053
Horizontal-Médio	= 0	= 0	À Esquerda de	= 0
Horizontal-Grande	= 0	= 0	À Direita de	= 0.6947
Vertical-Muito Pequeno	= 0	= 0		
Vertical-Pequeno	= 0	= 0		
Vertical-Médio	= 0	= 0		
Vertical-Grande	= 0	= 1		
Positivo-Muito Pequeno	= 1	= 0		
Positivo-Pequeno	= 0	= 0		
Positivo-Médio	= 0	= 0		

Positivo-Grande	= 0	= 0
Negativo-Muito Pequeno	= 0	= 0
Negativo-Pequeno	= 0	= 0
Negativo-Médio	= 0	= 0
Negativo-Grande	= 0	= 0

De início é preciso calcular a distância entre nodos e arcos do grafo AFT, respectivamente D_n e D_a . Para ambas as distâncias, utiliza-se no programa-protótipo, a distância de Hamming. Então

$$D_n = D_a = [\sum |x_i - y_i|] / 2$$

e calculando-se por exemplo para $D_n(a_6, b_1)$ obtém-se

Direção-Tamanho	nodo a_6	nodo b_1	$ x_i - y_i $
Horizontal-Muito Pequeno	0	0	0
Horizontal-Pequeno	0	0	0
Horizontal-Médio	0	0	0
Horizontal-Grande	0	0	0
Vertical-Muito Pequeno	0	0	0
Vertical-Pequeno	0	0	0
Vertical-Médio	0	0	0
Vertical-Grande	0.09775	0	0.09775
Positivo-Muito Pequeno	0	1	1
Positivo-Pequeno	0	0	0
Positivo-Médio	0	0	0
Positivo-Grande	0	0	0
Negativo-Muito Pequeno	0	0	0
Negativo-Pequeno	0	0	0
Negativo-Médio	0	0	0
Negativo-Grande	0.90225	0	0.90225

$\sum x_i - y_i $	=	2
$[\sum x_i - y_i] / 2$	=	1

$D_n(a_6, b_1) = 1$. Efetuando-se os cálculos para os demais elementos chega-se a

$$\begin{aligned} D_n(a_1, b_1) &= 0.0339 \\ D_n(a_1, b_7) &= 1 \\ D_n(a_6, b_1) &= 1 \\ D_n(a_6, b_7) &= 0.9023 \\ D_a(p_5, q_6) &= 0.1573 \\ D_a(p_5, -q_6) &= 1 \\ D_a(-p_5, q_6) &= 1 \\ D_a(-p_5, -q_6) &= 0.1573 \end{aligned}$$

Agora é possível calcular a distância d_c entre dois complexos.

$$D_c [C_1(v, \alpha, w), C_1(u, \beta, z)] = f_c [D_n(v, u), D_a(\alpha, \beta), D_n(w, z)], \text{ onde}$$

$$f_c = (D_n(v, u) + D_a(\alpha, \beta) + D_n(w, z)) / 3,$$

$$\begin{aligned} D_c [C_1(a_1, p_5, a_6), C_1(b_1, q_6, b_7)] &= (0.0339 + 0.1573 + 0.9023) / 3 = 0.3645 \\ D_c [C_1(a_1, p_5, a_6), C_2(b_7, -q_6, b_1)] &= (1 + 1 + 1) / 3 = 1 \\ D_c [C_2(a_6, -p_5, a_1), C_1(b_1, q_6, b_7)] &= (1 + 1 + 1) / 3 = 1 \\ D_c [C_2(a_6, -p_5, a_1), C_1(b_7, -q_6, b_1)] &= (0.9023 + 0.1573 + 0.0339) / 3 = 0.3645 \end{aligned}$$

Agora, com o cálculo da distância entre 2 complexos é possível calcular a distância entre as duas unidades $U_5 = (a_1, p_5, a_6) \in T1$ e $U_6 = (b_1, q_6, b_7) \in T2$

$$\begin{aligned} D_u [U(v, \alpha, w), U(u, \beta, z)] &= f_u [D_c [C_1(v, \alpha, w), C_1(u, \beta, z)], \\ & D_c [C_1(v, \alpha, w), C_2(z, -\beta, u)], \\ & D_c [C_2(w, -\alpha, v), C_1(u, \beta, z)], \\ & D_c [C_2(w, -\alpha, v), C_1(z, -\beta, u)]] \end{aligned}$$

Utiliza-se para f_u , o operador *min*.

$D_u [U_5(a_1, p_5, a_6), U_6(b_1, q_6, b_7)] =$	$f_u [D_c [C_1(a_1, p_5, a_6), C_1(b_1, q_6, b_7)],$
	$D_c [C_1(a_1, p_5, a_6), C_2(b_7, -q_6, b_1)],$
	$D_c [C_2(a_6, -p_5, a_1), C_1(b_1, q_6, b_7)],$
	$D_c [C_2(a_6, -p_5, a_1), C_1(b_7, -q_6, b_1)]]$
$=$	$\min [0.3645, 1, 1, 0.3645] = 0.3645$

Distância entre 2 AFT's

A medida de distância proposta para dois AFT's é

$$D(T_1, T_2) = f_1 \underset{p \in A_1}{\min} f_2 \underset{q \in A_2}{\min} D_u [U(v,p,w), U(u,q,z)] \text{ ([SHA92])}$$

onde toma-se f_1 por operação *max* e f_2 por operação *min*. Definidas as operações f_1 e f_2 , toma-se a projeção de cada unidade de T1 e procura-se a menor distância ($f_2 = \min$) de tal unidade em relação às unidades de T2. A partir daí tem-se um vetor com a menor distância de cada unidade de T1 em relação a T2. Aplicando-se a operação *max* sobre este vetor encontra-se a distância desejada D. No caso, conforme mostrado na figura AII.8, $D = 0.65968$.

