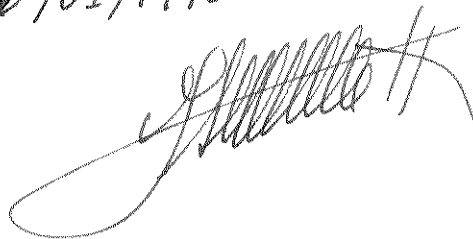


UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

RECONHECIMENTO DE CARACTERES ALFANUMÉRICOS,
ATRAVÉS DE TÉCNICAS DE LÓGICA NEBULOSA

THOMAS DE ARAUJO BUCK
ORIENTADOR: MÁRCIO LUIZ DE ANDRADE NETTO

Este exemplar corresponde à redação final da tese defendida por Thomas de Araujo Buck e aprovada pela Comissão Julgadora em 26/01/1990



TESE APRESENTADA À FACULDADE DE ENGENHARIA
ELÉTRICA DA UNIVERSIDADE ESTADUAL DE CAMPINAS
- UNICAMP - COMO PARTE DOS REQUISITOS EXIGIDOS
PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM
ENGENHARIA ELÉTRICA.

UNICAMP
BIBLIOTECA CENTRAL

Jürgen e Helena,
a vocês que são o início de tudo !

Para pensar :

O pós-computador é o novo primitivismo.

Ralf Hutter

AGRADECIMENTOS

Às minhas irmãs Carla e Barbara, que não ficaram chateadas por eu ter saído de Salvador,

Ao professor Márcio Luiz de Andrade Netto, pela acolhida, orientação e ensinamentos durante este longo período,

Aos tios José Antonio e Solange, que me hospedaram e me acolheram inicialmente em Campinas,

Aos amigos Elvio, Aluizio, José e Wolf, que me aguentaram na B. República, e aos amigos Castanho, Olga, Luciano, Guido, Manoel e Chico Boya, que bom ter conhecido vocês,

Aos meus familiares Carla, Mônica, Marcelo, Heliana, Márcia e Ricardo, que conviveram comigo em Campinas, e à Moara, que nasceu durante este trabalho,

A este amor de menina chamada Beatriz Giannini,

E a todos aqueles que, se não estão *explicitamente* referenciados aqui, contribuíram *implicitamente* para deixar a minha passagem por Campinas a mais alegre possível.

SUMÁRIO

O problema de Reconhecimento de Padrões é dividido em classificação e descrição dos padrões, nos quais são utilizadas técnicas de Lógica Nebulosa, de modo a permitir maior flexibilidade no processo de reconhecimento, dando capacidade de tolerância mais elevada para o sistema.

Dentre as várias técnicas matemáticas distintas usadas na solução do problema do reconhecimento de padrões, existe a chamada aproximação linguística, onde os padrões são representados como uma cadeia, árvore ou grafo de suas primitivas e relações, e o processo de decisão é geralmente um procedimento de análise gramatical.

Com ênfase na descrição estrutural dos padrões, a aproximação sintática tenta estabelecer uma analogia entre a análise da estrutura das imagens e a sintaxe de uma linguagem formal. Os padrões são vistos como sentenças em uma Teoria de Linguagens Formais para Visão por intermédio de uma gramática formal.

Uma das principais necessidades no projeto de um sistema estrutural de reconhecimento de padrões é o desenvolvimento experimental de uma gramática, capaz de representar a análise sintática necessária para se efetuar a classificação de forma automática.

Ainda que não exista um conjunto universal de primitivas para o reconhecimento linguístico de padrões, as expressões PDL de Alan Shaw são utilizadas para diversas aplicações, dentre as quais o reconhecimento de caracteres alfanuméricos.

ABSTRACT

The **Pattern Recognition** problem is divided into classification and description of the patterns, in which are used **Fuzzy Logic** techniques, giving flexibility in the recognition process and a tolerance capability to the system.

From the various distincts mathematical techniques used in the solution of the pattern recognition problem, there is the so called **linguistic approach**, where the patterns are represented as a string, tree or graph of its primitives and relations, and the decision process is in general a gramatical analysis procedure.

With emphasis in the structural description of patterns, the syntactic approach try to establish an analogy between the structural analysis of the images and the syntax of a formal language. The patterns are seen as sentences in a **Formal Theory of Vision Language** by a formal grammar.

One of the most important necessity in the project of a structural pattern recognition system is the experimental development of a grammar, capable of representing the needed syntactic analysis for doing the automatic classification.

Even if a universal primitive set for the linguistic recognition of patterns does not exist yet, the PDL expressions of Alan Shaw can be utilized for various applications, from among the recognition of alphanumeric characters.

ÍNDICE

VOLUME I

SUMÁRIO	5
ABSTRACT	6
ÍNDICE	7
CAPÍTULO I. Introdução	8
CAPÍTULO II. Noções de Lógica Nebulosa	18
CAPÍTULO III. Noções de Reconhecimento de Padrões	48
CAPÍTULO IV. Descrição do Reconhecimento Linguístico	66
CAPÍTULO V. Implementação e Resultados	97
CAPÍTULO VI. Conclusões e Sugestões	126
REFERÊNCIAS BIBLIOGRÁFICAS	140

VOLUME II

APÊNDICES	150
A. Padrões dos Caracteres Alfanuméricos Iniciais	151
B. Padrões dos Caracteres Alfanuméricos Finais	159
C. Padrões dos Caracteres Numéricos	167
D. Padrões dos Caracteres Alfabéticos Minúsculos	171
E. Famílias dos Padrões Alfanuméricos	179
F. Letras A para Testes	208
G. Letras A Contínuas para Tolerabilidade	216
H. Diversos Formatos para os Caracteres A e H	223
I. Detalhes das Máscaras de Casamento de Gabarito	230
J. Funções de Pertinência	236
K. Tabelas e Gráficos de Calibração	240
L. Trechos de Códigos em PROLOG	247
M. Glossário	253

CAPÍTULO I

INTRODUÇÃO

I.1. Histórico

I.2. Motivação

I.1. Histórico

O objetivo da área de reconhecimento de padrões é a extração automática de informações a partir de sinais sonoros, visuais, de frequência e outros. Pavlidis [43] ressalta que a palavra *padrão* é derivada da mesma raiz da palavra grega *patron* e, no seu uso original, significava algo selecionado como exemplar perfeito a ser imitado.

Dentre os vários objetivos que se pode atribuir a um sistema de reconhecimento e interpretação de imagens, está o de, dado um processo simples qualquer, automatizá-lo por tratamento computacional a fim de liberar a mão-de-obra especializada para atividades mais complexas, e que ainda dependem do ser humano. Mas descobriu-se que no campo do processamento automático de imagens, o esforço computacional era bastante elevado, exigindo computadores sofisticados, conseqüentemente, inviáveis economicamente. Portanto, durante um grande período da história esta área de pesquisa ficou de certo modo estagnada esperando pelo desenvolvimento de algoritmos mais simples e de computadores mais poderosos.

Entre o final da década de 1960 e o início da década de 1970 alguns pesquisadores, entre eles Richard Duda e Gerald Agin, trabalhavam no *Stanford Research Institute*. Dentre os vários projetos em andamento naquela época estava o do *processamento automático de imagens* por meio de computadores. Estes algoritmos foram tão importantes para o desenvolvimento da área de visão artificial por computadores que até hoje são referenciados como os algoritmos SRI (Usategui e Madrigal [26]).

Com isto, os pesquisadores acima trabalharam na sistematização de simplificações nos algoritmos e nas imagens a serem processadas para serem tratadas a tempo real. Como exemplo pode-se citar as três restrições mais fortes do SRI: 1) a imagem só pode ser tratada com dois níveis de cinza (preto e branco); 2) a imagem não deve possuir objetos superpostos, linhas escondidas ou qualquer outra imposição que não esteja explicitamente na imagem que se está tratando; 3) a imagem deve ser estável, ou seja, não é dinâmica e

nem muda de posição.

Apesar de simples estas restrições diminuem consideravelmente o volume de informações que os algoritmos devem processar para detectar os objetos de interesse na imagem e, conseqüentemente, diminuem o tempo de resposta do sistema de tal modo que ele possa ser aplicado em tempo real.

Assim, depois do desenvolvimento destas condições *restritoras* nas imagens, do desenvolvimento de equipamentos dedicados à área de tratamento de imagens e do desenvolvimento de hardware e algoritmos em paralelo, a utilização do computador no processamento e interpretação de imagens passou-se a ser cada vez mais viável tanto tecnológica quanto economicamente.

I.2. Motivação

Segundo Gonzalez e Thomason [9], as técnicas de reconhecimento de padrões (não apenas os visuais) estão entre as mais importantes ferramentas no campo da inteligência para máquinas, pois envolvem conceitos que são usados pelos humanos diariamente em seu cotidiano. Ainda segundo os mesmos autores, reconhecimento de padrões pode ser definido como a *categorização dos dados de entrada do sistema em classes identificáveis pela extração de atributos ou estruturas significativas dos dados, a partir de um conjunto de detalhes irrelevantes.*

Bezdek [37] define que *reconhecimento de padrões é a procura de estruturas nos dados.* A imagem nada mais é do que uma sequência de bytes que representam, para cada pixel, o nível de cinza devido à luminosidade da cena. Os algoritmos portanto fazem uma busca por estes bytes para detectar sequências ordenadas e bem definidas dos mesmos, que costuma-se chamar de *primitivas*. Em termos de informação, ainda segundo o autor, os *dados* a possuem, a *procura* a reconhece e a *estrutura* a representa.

Técnicas linguísticas na área de reconhecimento de padrões tem por finalidade a exploração das estruturas ou de relacionamentos existentes em um padrão bi- ou tri-dimensional presente em uma imagem. Se uma dada estrutura existe ou pode ser obtida a partir da imagem, então um padrão complexo pode ser reduzido em termos de sub-padrões, os quais por sua vez podem ser novamente reduzidos até que ocorram as primitivas que, como o próprio nome diz, não são divisíveis em novos sub-terminos. Sendo assim, é possível realizar a classificação por um estudo similar ao do reconhecimento de estruturas gramaticais numa linguagem formal. Esta decomposição dos padrões em sub-padrões no reconhecimento estrutural é o fundamento da metodologia dos sistemas de reconhecimento linguísticos.

A origem destes algoritmos linguísticos deveu-se ao fato de que os convencionais fazem uso apenas das informações residentes nos pixels, que são apenas uma fração das informações disponíveis na imagem conseqüentemente não

utilizando a informação contextual a nível de objeto. Leonardo Bins [39] caracteriza funcionalmente **sistemas de visão computacional** como processadores de informação que aceitam como entrada representações bidimensionais de cenas e produzem como saída uma descrição simbólica. A partir desta descrição é possível analisá-la para se extrair as informações desejadas.

Pela Teoria das Linguagens Formais, para cada gramática existe uma linguagem associada que é o conjunto de todas as cadeias de símbolos terminais que podem ser obtidas do símbolo inicial da gramática, por um número finito de aplicações de suas regras de produção.

O estudo das estruturas geradas de uma imagem é feito por um processo que recebe o nome de análise sintática, a partir de uma gramática formal, que aceita apenas as estruturas bem formadas segundo a gramática estabelecida e, particularmente no caso das gramáticas nebulosas, atribui um valor numérico que corresponde à semelhança entre a estrutura gerada da imagem e a estrutura do padrão que se quer reconhecer. Estas gramáticas nebulosas fazem uso da lógica nebulosa (também chamada difusa ou incerta) proposta por Zadeh [31] na década de 1960, com base em trabalhos originais de Post [89] e Lukaziewicz [90]. Consequentemente, o problema de reconhecer um objeto consiste, segundo Banerji [19], em aplicar um teste na imagem para determinar se ela pertence à gramática e, se o resultado for positivo, de avaliar a pertinência nas várias classes pré-definidas.

Pelo que se expôs acima, os sistemas de reconhecimento linguísticos evitam o uso de algoritmos exaustivos, como por exemplo os algoritmos de reconhecimento estatísticos, sendo deste modo mais fácil, rápido e inteligente a realização da dita tarefa. Uma vez formalizada a descrição estrutural de uma imagem, esta é usada para o seu reconhecimento.

No presente trabalho, conforme Pavlidis, existe um exemplar considerado perfeito adequadamente chamado de **padrão**, para cada classe de caracteres alfanuméricos que exista no sistema. O sistema utiliza uma metodologia de reconhecimento sintática de padrões que usa o conceito de expressões PDL desenvolvida por Alan C. Shaw nos laboratórios do acelerador linear da Universidade de Stanford na década de 1960.

O fato dessa metodologia ter sido descrita neste laboratório deve-se ao excessivo número de fotografias gerado naquela época em cada experiência realizada na câmara de bolhas no dito acelerador, fotografias essas que tinham que ser analisadas pelos físicos. Pensou-se portanto em uma técnica que pudesse ser utilizada nos computadores que automatizasse a parte inicial da análise destas fotografias. Assim, os físicos, ao invés de receberem a imagem pura e bruta, recebiam também um estudo já previamente feito pelo computador, que desse modo ajudava o estudo dos fenômenos ocorridos durante o experimento, pelo simples fato de que os físicos não perdiam tempo em busca de informações que podiam ser obtidas artificialmente, e portanto diminuía o tempo de resposta para cada experiência realizada. No apêndice B do livro de King Sun Fu [14] estão algumas fotografias dos experimentos, bem como todas as expressões PDL originais do trabalho de Shaw da época, usadas para reconhecer as partículas sub-atômicas. Um exemplo de reconhecimento de partículas pode ser visto na figura 1.1 a seguir.

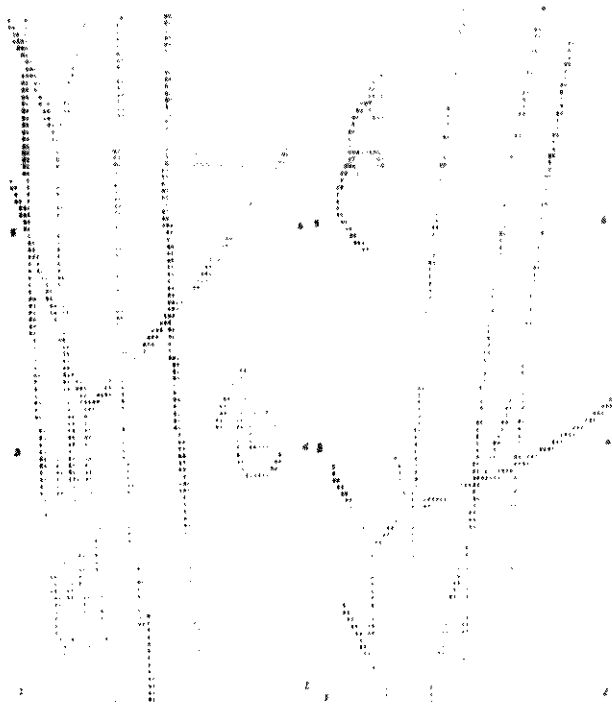


Figura 1.1 - Experiência na câmara de bolhas

Numa primeira aproximação, o sistema proposto tem o aspecto lógico de implementação mostrado na figura 1.2.

Se a expressão *visão computacional ou artificial* fôr empregada para indicar a aquisição e o uso das informações visuais para a tomada de decisões por uma máquina qualquer, para controlar certo processo por exemplo, ilustrado na figura 1.3, e se fôr também envolvido as duas seguintes fases, a aquisição da imagem e a análise de seus dados, pode-se dizer então que o presente trabalho envolve-se basicamente com a segunda parte. Em outras palavras, para a tomada de decisão pelo reconhecedor linguístico dos caracteres alfanuméricos existem rotinas que dão o necessário suporte para essa análise.

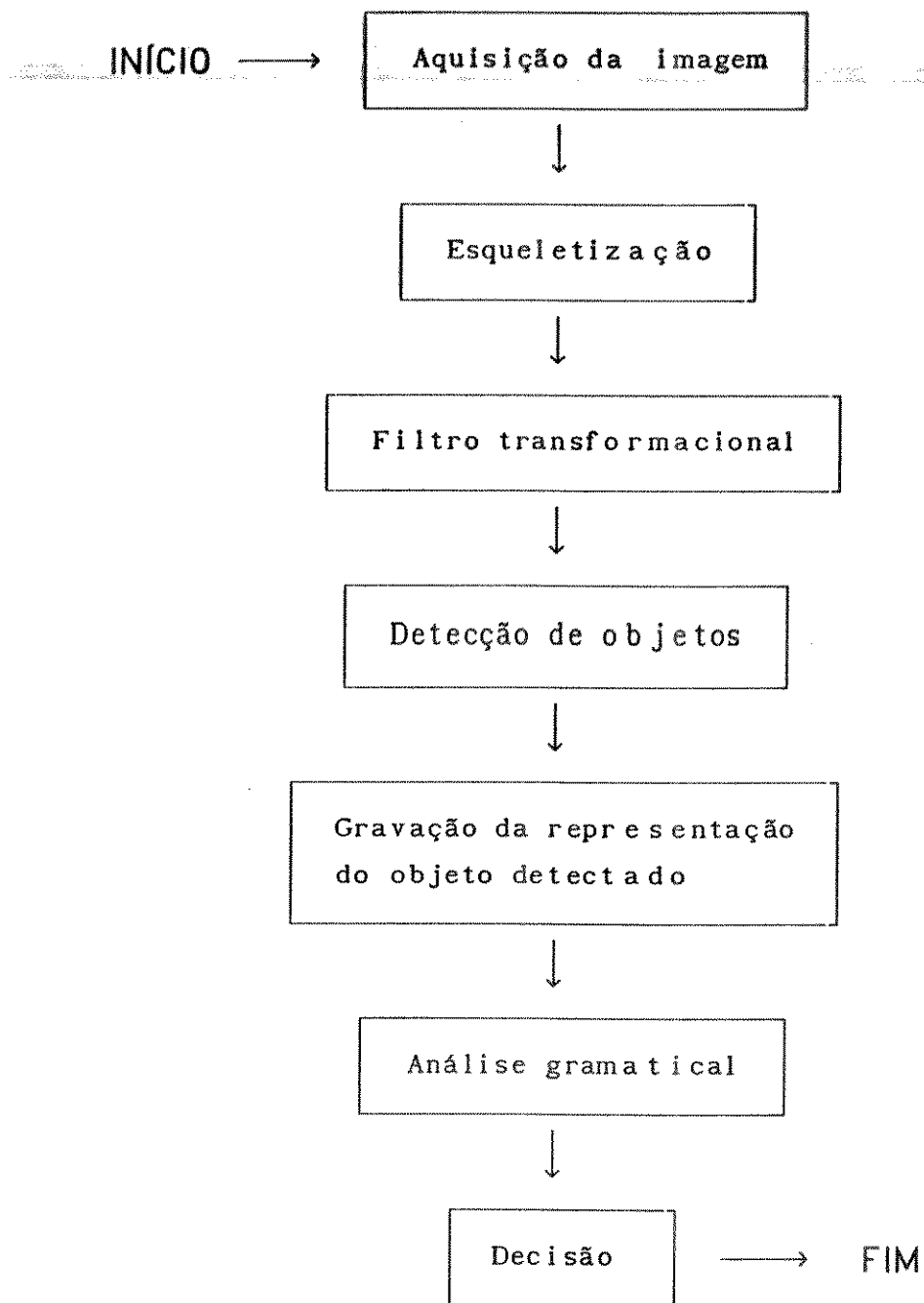


Figura 1.2 - Resumo do sistema

Segundo Rao e Jain [45], costuma-se dividir a tarefa de processamento de informações visuais por um sistema de visão em três níveis básicos. O nível mais baixo tenta extrair características primitivas das imagens, e a este nível costuma-se denominar de visão de baixo nível, ou processamento antecipado. Este tipo de processamento normalmente envolve um volume muito

grande de computação repetitiva a cada pixel da imagem. No nível intermediário as características como linhas, regiões, formas, orientação de superfície, etc, são extraídas. A este nível do processamento simbólico da visão é dado o nome de processamento intermediário, ou segmentação. O último e talvez o mais importante para o presente trabalho é o processamento chamado visão de alto nível ou processamento final, que lida com objetos e utiliza-se de conhecimentos do domínio específico para construir descrições detalhadas ou gerais da imagem, realizando uma interpretação consistente para uma determinada finalidade. Esta última fase portanto envolve a aplicação de operações de entendimento da imagem, implicando tipicamente numa grande complexidade, tanto no que diz respeito à quantidade de imagens diferentes que podem ser analisadas, quanto à quantidade de maneiras diferentes que as informações podem ser armazenadas.

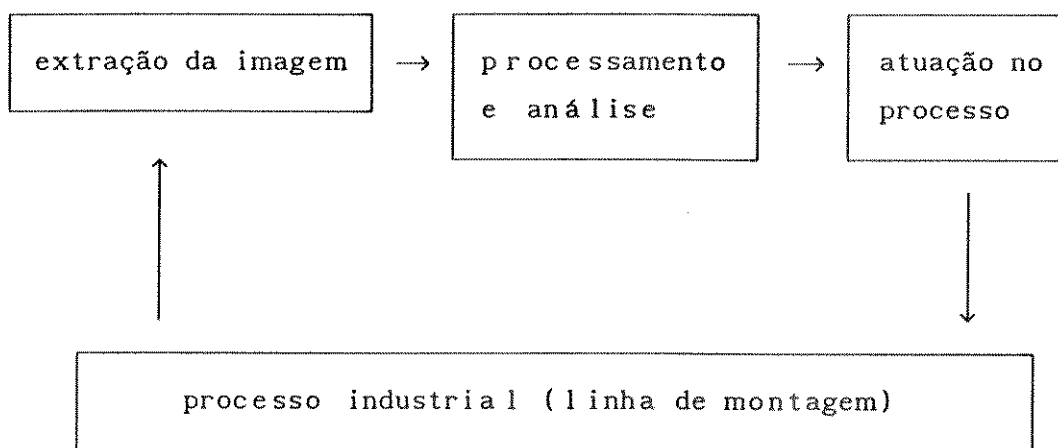


Figura 1.3 - Uso de visão artificial em ambientes industriais

O que se quer dizer com *entendimento* ? As propriedades previamente extraídas devem ser interpretadas com relação a decisões em termos de identidades, localizações, orientações, ou talvez sobre a tolerância dimensional dos objetos presente na imagem. Para isso torna-se necessária a aquisição de informações e/ou conhecimentos externos ao sistema de visão. Uma decisão a ser tomada sobre uma coleção de vértices e linhas da representação de um objeto necessita do acesso a algum modelo armazenado daquele objeto, para realizar então o estudo comparativo entre ambos denominado *casamento*.

Assim, a função da fase de mais alto nível da visão é a de tomar decisões a partir de atributos extraídos das fases mais baixas.

Concluindo, a motivação para a realização do presente trabalho tem sua origem nas seguintes considerações: 1) uso da metodologia de análise gramatical para se efetuar o reconhecimento; 2) distanciamento da aproximação decisão-teórica, já estudada por Bernardo Sotomayor [24] e Evandro Ruiz [61]; 3) aplicação de técnicas de inteligência artificial na metodologia e na implementação escolhidas.

O presente trabalho portanto visa a implementação em micro-computadores das rotinas linguísticas de reconhecimento (também chamadas de *gramaticais, estruturais, sintáticas ou descritivas*) baseadas nas expressões PDL geradas a partir da imagem dos caracteres alfanuméricos. O sistema recebe o nome de RENEALF, por REconhecedor NEbuloso de Caracteres ALFAnuméricos.

CAPÍTULO II

NOÇÕES DE LÓGICA NEBULOSA

- II.1. Introdução à Lógica Nebulosa
- II.2. Aplicações a Reconhecimento de Padrões
- II.3. Variáveis Linguísticas
- II.4. Exemplos de Funções Nebulosas

II.1. Introdução à Lógica Matemática

Segundo Zadeh [57], Lógica é a ciência dos princípios formais normativos do raciocínio. Neste sentido, Lógica Nebulosa é a parte da lógica matemática que concerne aos princípios formais do raciocínio incerto ou aproximado, com o raciocínio preciso como um caso especial e limitante. A lógica nebulosa pode ser encarada, ainda segundo o autor, como uma extensão da lógica multivalores. Muitos sistemas foram desenvolvidos baseados nesta nova metodologia de raciocínio, principalmente aqueles que modelam as imprecisões do pensamento, que tem um papel importante na habilidade humana de tomar decisões em ambiente com incertezas.

Segundo Shimura [11], eventos que ocorrem no mundo real podem ser descritos por meio de três classes ou categorias: determinísticos, probabilísticos e nebulosos. Sob vários pontos de vista o reconhecimento de padrões pode ser considerado essencialmente nebuloso porque não existe nenhuma fronteira precisa entre as diversas categorias de classificação. Um processo de decisão nebuloso é útil na classificação de padrões desconhecidos e exige funções de classificação mais simples.

Sintaxes formais em certos sistemas parecem suficientes e bastante estruturadas porque apenas as sentenças que estão totalmente sem erros são aceitas, mas existem certos sistemas em que uma análise léxica mais flexível ou tolerante a falhas é uma característica desejável, exemplificados pelos sistemas de reconhecimento de padrões sintáticos. Isto explica em parte porque os sistemas de visão que utilizam análise sintática se beneficiam com o uso de lógica nebulosa.

A lógica nebulosa é baseada no princípio de que o raciocínio humano raramente é preciso, e pelo seu uso os computadores também passam a representar e manipular dados e raciocínios imprecisos e até ambíguos, como se pode ver na figura 2.1.

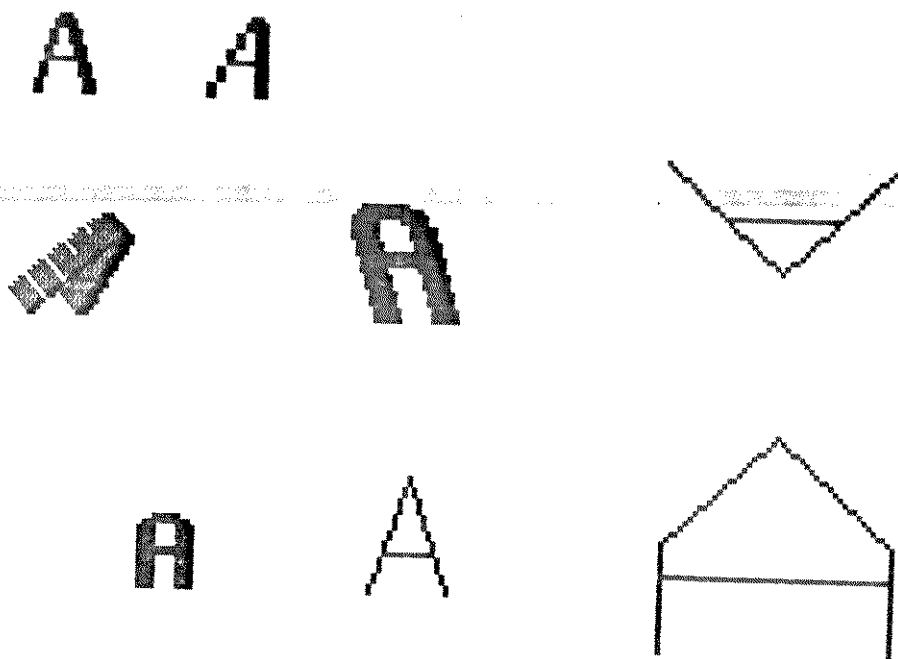


Figura 2.1 - Diversas letras As

Recentemente houve um sensível aumento na utilização da teoria de conjuntos e algoritmos nebulosos para aplicações em processamento de imagens, o que foi motivado por um desejo de se modelar a ambiguidade e ruídos contidos em imagens definidas digitalmente.

Para um lógico matemático (Giorno et alii [60]) um *programa* é um procedimento finitário para resolver um problema e *lógica* é sinônimo de sistema formal, ou seja, de uma linguagem formal acompanhada de uma abstração adequada para os princípios usados na decisão de quando uma afirmação (na linguagem do sistema) é consequência de outras. Programação em Lógica significa portanto utilizar um sistema formal para resolver problemas, enfatizando a sua estrutura lógica e não a construção de um procedimento para resolvê-lo.

O processo de criar um programa em lógica envolve essencialmente duas fases. Na primeira fase escolhe-se uma *linguagem simbólica* para exprimir os conceitos do problema ou situação em questão, eliminando ambiguidades e imprecisões inerentes ao uso de uma linguagem natural. Na segunda descrevem-se através de sentenças na linguagem escolhida, os fatos e propriedades pertinentes ao problema ou situação.

Segundo Negoita [12], **Raciocínio Plausível** é simplesmente a derivação de conclusões a partir de fatos que parecem estar corretos. Uma vantagem da aproximação por lógica nebulosa é a possibilidade de se representar variáveis numéricas e linguísticas de uma maneira uniforme, com um formalismo para manipulá-las e tratá-las. Ainda segundo o mesmo autor, **Aproximação Linguística** é uma função que mapeia os sub-conjuntos nebulosos para o conjunto de valores linguísticos.

As duas formas abaixo de representar a noção de pertinência a um conjunto são equivalentes:

- 1) $x \in A$, onde \in é o símbolo de pertence;
- 2) através da função característica $\mu_A(x)$, onde:

$$\mu_A(x) = \begin{cases} 0, & \text{se } x \notin A \\ 1, & \text{se } x \in A. \end{cases} \quad (1)$$

As duas representações acima são utilizadas na *Teoria Clássica de Conjuntos*, mas apenas a segunda forma, com ligeira modificação, é útil para a *Teoria de Conjuntos Nebulosos*.

Seja \mathcal{U} um conjunto denominado *conjunto de referência* e seja x um elemento de \mathcal{U} . Define-se um subconjunto nebuloso A de \mathcal{U} como um conjunto de pares ordenados da seguinte forma:

$$\{ (x \mid \mu_A(x)) \}, \forall x \in \mathcal{U} \quad (2)$$

onde $\mu_A(x)$ é o grau ou nível de pertinência de x em A . Portanto, se $\mu_A(x)$ tomar seus valores num conjunto \mathcal{M} , chamado *conjunto de pertinência*, pode-se dizer que x toma seus valores em \mathcal{M} através de uma função $\mu_A(x)$. Pode-se então escrever

$$x \underset{\mu_A}{\rightsquigarrow} \mathcal{M} \quad (3)$$

Esta função será chamada de *função de pertinência*.

Como se pode notar pela expressão (2) qualquer elemento do universo \mathcal{U} faz parte do subconjunto nebuloso A , só que alguns terão valor de pertinência nulo, e outros terão valores não nulos.

Neste trabalho, \mathcal{U} é o conjunto de todas as imagens que se quer analisar, e $A, B, C, \text{ etc.}$ são os subconjuntos das imagens da letra A, da letra B, e assim sucessivamente, até a letra Z.

Aplicando-se estes conceitos para o caso de imagens com caracteres alfanuméricos, se A for o conjunto nebuloso de imagens com a letra A, então qualquer imagem faz parte deste conjunto, porém a maioria delas terá valores de pertinência baixos.

Definem-se agora algumas operações básicas sobre subconjuntos nebulosos. Sejam A e B dois subconjuntos nebulosos.

$$A: \begin{array}{l} x \wedge \longrightarrow \mathcal{M} \\ \mu_A(x) \end{array} \quad (4)$$

$$B: \begin{array}{l} x \wedge \longrightarrow \mathcal{M} \\ \mu_B(x) \end{array}$$

com $\mathcal{M} = [0, 1]$.

Inclusão

Diz-se que o subconjunto nebuloso A está incluído no subconjunto nebuloso B se:

$$\forall x \in \mathcal{U}: \mu_A(x) \leq \mu_B(x) \quad (5)$$

o qual será denotado por $A \subset B$.

Igualdade

Diz-se que dois subconjuntos nebulosos A e B são iguais se e somente se :

$$\forall x \in \mathcal{U}: \mu_A(x) = \mu_B(x) \quad (6)$$

o que será denotado por $A = B$.

Complemento

Diz-se que dois subconjuntos nebulosos A e B são complementares se :

$$\forall x \in \mathcal{U}: \mu_A(x) = 1 - \mu_B(x) \quad (7)$$

denotando-se por $B = \bar{A}$

Intersecção

Define-se a intersecção de dois subconjuntos nebulosos A e B como segue :

$$\forall x \in \mathcal{U}: \mu_{A \cap B}(x) = \text{MIN} \{ \mu_A(x), \mu_B(x) \} \quad (8)$$

denotando-se por $A \cap B$.

União

Define-se a união de dois subconjuntos nebulosos A e B como segue :

$$\forall x \in \mathcal{U}: \mu_{A \cup B}(x) = \text{MAX} \{ \mu_A(x), \mu_B(x) \} \quad (9)$$

denotando-se por $A \cup B$.

A título de exemplo, suponhamos uma aplicação onde se deseja uma imagem com uma letra A e não R. A função de pertinência f desta situação será :

$$f_{A \text{ and } \bar{R}}(\lambda) = \text{MIN} [f_A, f_{\bar{R}}] = \text{MIN} [f_A, 1 - f_R]$$

Outro exemplo seria o de uma aplicação que necessite de uma imagem com um Q ou um O. Então temos :

$$f_{Q \text{ or } O}(\lambda) = \text{MAX} [f_Q, f_O]$$

Soma disjuntiva

A soma disjuntiva de dois subconjuntos nebulosos A e B é definida em termos de uniões e intersecções do seguinte modo :

$$A \oplus B = (A \cap \bar{B}) \cup (\bar{A} \cap B) \quad (10)$$

Diferença

A diferença de dois subconjuntos nebulosos A e B é dada pela seguinte equação :

$$A - B = A \cap \bar{B} \quad (11)$$

Produto Algébrico

Define-se o produto algébrico de dois subconjuntos nebulosos A e B como segue :

$$\forall x \in \mathcal{U}: \mu_{A \circ B}(x) = \mu_A(x) \cdot \mu_B(x) \quad (12)$$

e denota-se por $A \circ B$.

Soma Algébrica

Define-se a soma algébrica de dois subconjuntos nebulosos A e B como sendo:

$$\begin{aligned} \forall x \in \mathcal{U}: \\ \mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \circ \mu_B(x) \end{aligned} \quad (13)$$

e denota-se por $A + B$.

Distância Euclideana ou distância quadrática

Para \mathcal{U} contendo N elementos x_i a distância Euclideana será definida como:

$$e(A,B) = \sqrt{\sum_{i=1}^N (\mu_A(x_i) - \mu_B(x_i))^2} \quad (14)$$

portanto $0 \leq e(A,B) \leq \sqrt{N}$.

Distância Euclideana relativa

A distância Euclideana relativa é dada por:

$$\epsilon(A,B) = \frac{e(A,B)}{\sqrt{N}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mu_A(x_i) - \mu_B(x_i))^2} \quad (15)$$

portanto $0 \leq \epsilon(A,B) \leq 1$.

Norma Euclideana

A norma Euclideana é dada por:

$$e^2(A,B) = [e(A,B)]^2 \quad (16)$$

E a norma Euclideana relativa é dada pela seguinte expressão:

$$e^2(A,B) = \frac{1}{N} e^2(A,B) = \frac{1}{N} \sum_{i=1}^N (\mu_A(x_i) - \mu_B(x_i))^2 \quad (17)$$

Propriedades de Subconjuntos Nebulosos

Dados os subconjuntos nebulosos $A \subset \mathcal{U}$, $B \subset \mathcal{U}$ e $C \subset \mathcal{U}$, temos:

Propriedades Comutativas

$$A \cap B = B \cap A$$

$$A \cup B = B \cup A$$

Propriedades Associativas

$$(A \cap B) \cap C = A \cap (B \cap C)$$

$$(A \cup B) \cup C = A \cup (B \cup C)$$

Idempotência

$$A \cap A = A$$

$$A \cup A = A$$

Distributividade

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

$$A \cap \phi = \phi$$

$$A \cup \phi = A$$

$$A \cap \mathcal{U} = A$$

$$A \cup \mathcal{U} = \mathcal{U}$$

onde ϕ é um conjunto ordinário em que:

$$\forall x_i \in \mathcal{U} : \mu_{\phi}(x_i) = 0$$

e onde \mathcal{U} é um conjunto ordinário em que:

$$\forall x_i \in \mathcal{U} : \mu_{\mathcal{U}}(x_i) = 1$$

Involução

$$\overline{\overline{A}} = A$$

Teoremas de DeMorgan

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

Relações Nebulosas

Para se medir o quanto uma imagem é semelhante a outra, tem-se que compará-las, para extrair características ou primitivas que existam em ambas. Quanto mais semelhanças houver, maior a chance da imagem ser classificada na respectiva classe de padrões que se está comparando.

Para tornar aplicável todo este formalismo, utiliza-se a noção de relação.

Uma relação nebulosa \mathcal{R} pode ser definida a partir de dois conjuntos nebulosos \mathcal{C}_1 e \mathcal{C}_2 como em Kaufmann [15] :

$$x \in \mathcal{C}_1; y \in \mathcal{C}_2; \tag{18}$$

Usamos a notação $x \mathcal{R} y$ para designar a relação.

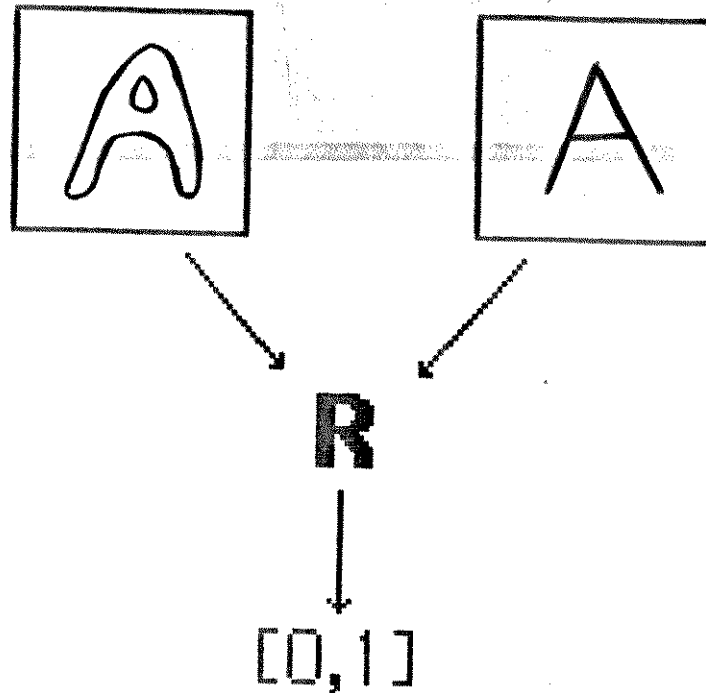


Figura 2.2 - Exemplo de uma relação nebulosa

Uma relação nebulosa portanto pode ser descrita como uma **comparação** entre dois conjuntos nebulosos, por exemplo, representando respectivamente imagens e classes de padrões.

O quanto uma imagem assemelha-se a outra é fornecida por meio do nível de pertinência.

Pode-se generalizar a noção de relação nebulosa \mathcal{R} para n conjuntos. Uma relação nebulosa \mathcal{R} em $U_1 \times U_2 \times \dots \times U_n$ é caracterizada por uma função de pertinência de n variáveis sobre a classe $U_1 \times U_2 \times \dots \times U_n$. Para o caso particular de 2 conjuntos nebulosos U e V , tem-se a seguinte notação:

$$\mathcal{R} = \left\{ (u_i, v_j) / \mu_{U,V}(u_i, v_j) \right\}; i = 1, \dots, n; j = 1, \dots, n;$$

onde u_i e v_j representam elementos de U e V respectivamente e (u_i, v_j) o par ordenado elemento de $U \times V$.

Composição Max-min

Sejam duas relações nebulosas $\mathcal{R}_1 \subset X \times Y$ e $\mathcal{R}_2 \subset Y \times Z$. Define-se então a relação de composição Max-Min de acordo com a seguinte expressão :

$$\mu_{\mathcal{R}_1 \circ \mathcal{R}_2}(x,z) = \text{MAX}_y [\text{MIN} (\mu_{\mathcal{R}_1}(x,y), \mu_{\mathcal{R}_2}(y,z))] \quad (19)$$

e denota-se por $\mathcal{R}_1 \circ \mathcal{R}_2$.

Primeira Projecção

Dados dois conjuntos nebulosos X e Y , o melhor casamento dentre todos os elementos y_1 de Y em relação à um único elemento x de X é chamado de *primeira projecção de \mathcal{R}* . Este melhor casamento é dado pelo maior valor de pertinência da relação nebulosa \mathcal{R} . Matematicamente, a função de pertinência a seguir define a primeira projecção de \mathcal{R} :

$$\mu_{\mathcal{R}}^{(1)}(x) = \text{MAX}_y \mu_{\mathcal{R}}(x,y) \quad (20)$$

Segunda projecção

Analogamente, a função de pertinência a seguir define a segunda projecção de \mathcal{R} :

$$\mu_{\mathcal{R}}^{(2)}(y) = \text{MAX}_x \mu_{\mathcal{R}}(x,y) \quad (21)$$

Projeção global

A primeira projeção da segunda projeção (ou vice-versa) é chamada *projeção global* de uma relação nebulosa \mathcal{R} , e é dada pela expressão a seguir:

$$h(\mathcal{R}) = \text{MAX}_x \text{MAX}_y \mu_{\mathcal{R}}(x,y) \quad (22)$$

e será denotada por $h(\mathcal{R})$. Voltar-se-á a este assunto no item IV.5.

Para o caso específico de processamento de imagens e de reconhecimento de padrões, se $h(\mathcal{R}) = 1$, a relação será dita *normal*, ou seja, a imagem corresponde fielmente a um certo padrão, e se $h(\mathcal{R}) < 1$ a relação será chamada *sub-normal*, ou seja, a imagem não coincide totalmente com nenhum dos padrões.

Suporte

Um subconjunto ordinário de pares ordenados (x,y) para o qual a função de pertinência $\mu_{\mathcal{R}}$ seja não nula será chamado *suporte* de \mathcal{R} :

$$S(\mathcal{R}) = \{ (x,y) \mid \mu_{\mathcal{R}}(x,y) > 0 \} \quad (23)$$

Envelope

Sejam duas relações \mathcal{R} e \mathcal{L} tais que :

$$\forall (x,y) \in U_1 \times U_2 : \mu_{\mathcal{R}}(x,y) \leq \mu_{\mathcal{L}}(x,y) \quad (24)$$

então \mathcal{L} é um *envelope* de \mathcal{R} , ou \mathcal{R} é uma *clausura* de \mathcal{L} .

União

A união de duas relações \mathcal{R} e \mathcal{L} é dada por :

$$\mu_{\mathcal{R} \cup \mathcal{L}}(x,y) = \text{MAX} [\mu_{\mathcal{R}}(x,y), \mu_{\mathcal{L}}(x,y)] \quad (25)$$

e será denotada por $\mathcal{R} \cup \mathcal{L}$.

Intersecção

A intersecção de duas relações \mathcal{R} e \mathcal{L} é dada por :

$$\mu_{\mathcal{R} \cap \mathcal{L}}(x,y) = \text{MIN} [\mu_{\mathcal{R}}(x,y), \mu_{\mathcal{L}}(x,y)] \quad (26)$$

e será denotada por $\mathcal{R} \cap \mathcal{L}$

Produto Algébrico

O produto algébrico de duas relações \mathcal{R} e \mathcal{L} é dado pela seguinte expressão:

$$\mu_{\mathcal{R} \circ \mathcal{L}}(x,y) = \mu_{\mathcal{R}}(x,y) \cdot \mu_{\mathcal{L}}(x,y) \quad (27)$$

e denota-se por $\mathcal{R} \circ \mathcal{L}$.

Soma Algébrica

A soma algébrica de duas relações \mathcal{R} e \mathcal{L} é dado por:

$$\mu_{\mathcal{R}+\mathcal{L}}(x,y) = \mu_{\mathcal{R}}(x,y) + \mu_{\mathcal{L}}(x,y) - \mu_{\mathcal{R}\circ\mathcal{L}}(x,y) \quad (28)$$

e denota-se por $\mathcal{R} + \mathcal{L}$.

Complemento

O complemento de uma relação \mathcal{R} é uma relação tal que :

$$\forall (x,y) \in \mathcal{U}_1 \times \mathcal{U}_2 : \mu_{\overline{\mathcal{R}}}(x,y) = 1 - \mu_{\mathcal{R}}(x,y) \quad (29)$$

e denota-se por $\overline{\mathcal{R}}$.

Soma disjuntiva

A soma disjuntiva de duas relações nebulosas \mathcal{R} e \mathcal{L} é dada pela expressão:

$$\mathcal{R} \oplus \mathcal{L} = (\mathcal{R} \cap \overline{\mathcal{L}}) \cup (\overline{\mathcal{R}} \cap \mathcal{L}) \quad (30)$$

e denota-se por $\mathcal{R} \oplus \mathcal{L}$.

Relação ordinária mais próxima de uma relação nebulosa

Seja \mathcal{R} uma relação nebulosa. A relação ordinária R mais próxima a \mathcal{R} é dada por :

$$\begin{aligned} \mu_R(x,y) &= 0 && \text{se } \mu_{\mathcal{R}}(x,y) < 0.5 \\ &= 1 && \text{se } \mu_{\mathcal{R}}(x,y) > 0.5 \\ &= 0 \text{ ou } 1 && \text{se } \mu_{\mathcal{R}}(x,y) = 0.5 \end{aligned} \quad (31)$$

Como exemplo, a relação mais próxima poderia ser interpretada como sendo a melhor imagem padrão em relação à imagem fornecida como entrada para o sistema. Assim, a imagem à direita da figura 2.3 seria dada como a mais próxima da imagem à esquerda que foi fornecida.

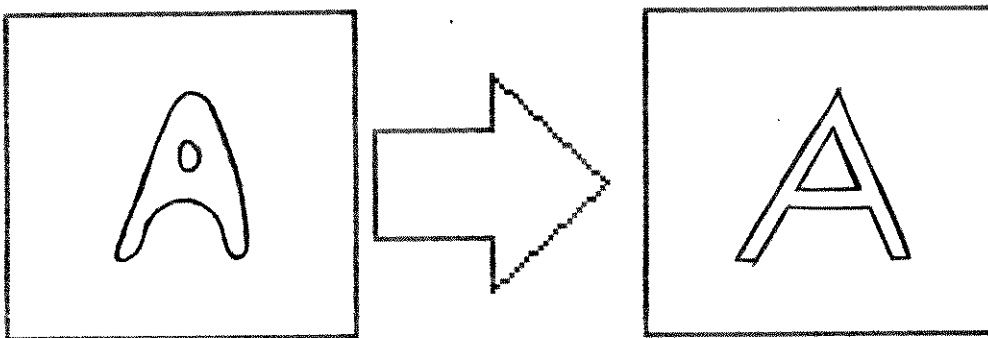


Figura 2.3 - Imagem mais próxima

A Teoria de Probabilidades trata da incerteza dada pela ocorrência ou não ocorrência de algum evento, enquanto que a incerteza na lógica nebulosa trata da pertinência ou da não pertinência de um objeto a um conjunto com fronteiras imprecisas.

Uma hipótese que se mostra neste trabalho é a de que quando nosso maior esforço é com o sentido da informação, ao invés de suas medidas, a própria estrutura da análise da informação é possibilística, e não probabilística, e portanto implica que a necessidade para essa análise é de uma teoria análoga à

teoria da probabilidade, que se chama *Teoria da Possibilidade*. Veja que neste caso o *sentido da informação* pode ser entendido como a extração e identificação do objeto presente na imagem, enquanto que suas *medidas* são o cálculo de alguns atributos presentes na imagem. Assim, torna-se possível a comparação dos algoritmos nebulosos com os decisão-teóricos.

Observe que o entendimento desta sutil diferença é de importância vital para usar-se o sistema da presente tese, pois ele apenas faz uso da *Teoria da Possibilidade*. Considere por exemplo quantas representações distintas da letra A maiúscula os seres humanos reconhecem corretamente. São inúmeras as variações como orientação, espessura e tamanho, dentre outras, que podem ocorrer sem perda de identidade [68].

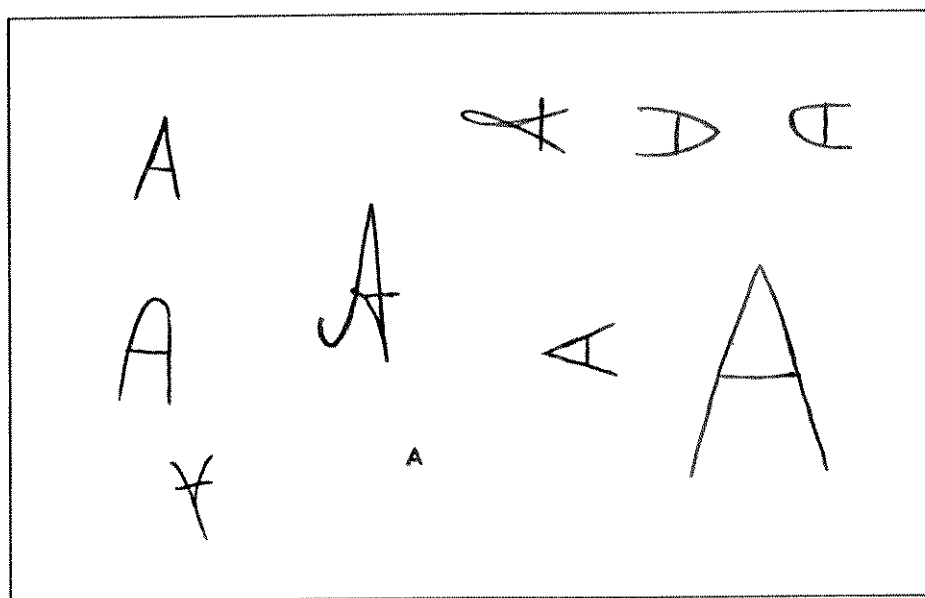


Figura 2.4 - Letras As manuscritas

Como um padrão é uma classe de equivalência consistindo de todas as configurações que gerem a mesma resposta pela máquina, a lógica nebulosa será então a responsável pela maneira como o sistema atuará sobre estas diversas configurações distintas possíveis.

II.2. Aplicações a Reconhecimento de Padrões

As aplicações da Teoria dos Subconjuntos Nebulosos à reconhecimento de padrões pode se dar sob 4 formas:

1. funções nebulosas;
2. variáveis linguísticas;
3. asserções condicionais nebulosas; e
4. algoritmos nebulosos.

A Lógica Nebulosa tem aplicabilidade em diversos sistemas de tratamento de informações. Dentre estas existem os sistemas de visão artificiais, que podem fazer grande uso da lógica nebulosa, como será explicado abaixo.

A precisão das Linguagens de Descrição de Padrões contrasta muito fortemente com a imprecisão dos padrões encontrados em problemas típicos de reconhecimento de padrões. A fim de reduzir esta diferença, é natural a introdução de incerteza na estrutura destas Linguagens de Descrição de Padrões, levando portanto ao conceito de Linguagens Nebulosas, com suas aplicações a reconhecimento de padrões.

Para exemplificar esta imprecisão na execução normal dos algoritmos de reconhecimento, a imagem de uma letra **A** pode vir distorcida, translacionada, rotacionada, de diversas maneiras não previsíveis. A lógica nebulosa pode *facilmente* classificar todas essas imagens com um mínimo de erro, seja por intermédio de funções nebulosas como também por meio de algoritmos nebulosos, uma vez que uma letra **A** distorcida pode ter o seu valor de pertinência dentro do intervalo fechado $[0,1]$. Seja por exemplo a figura 2.5.

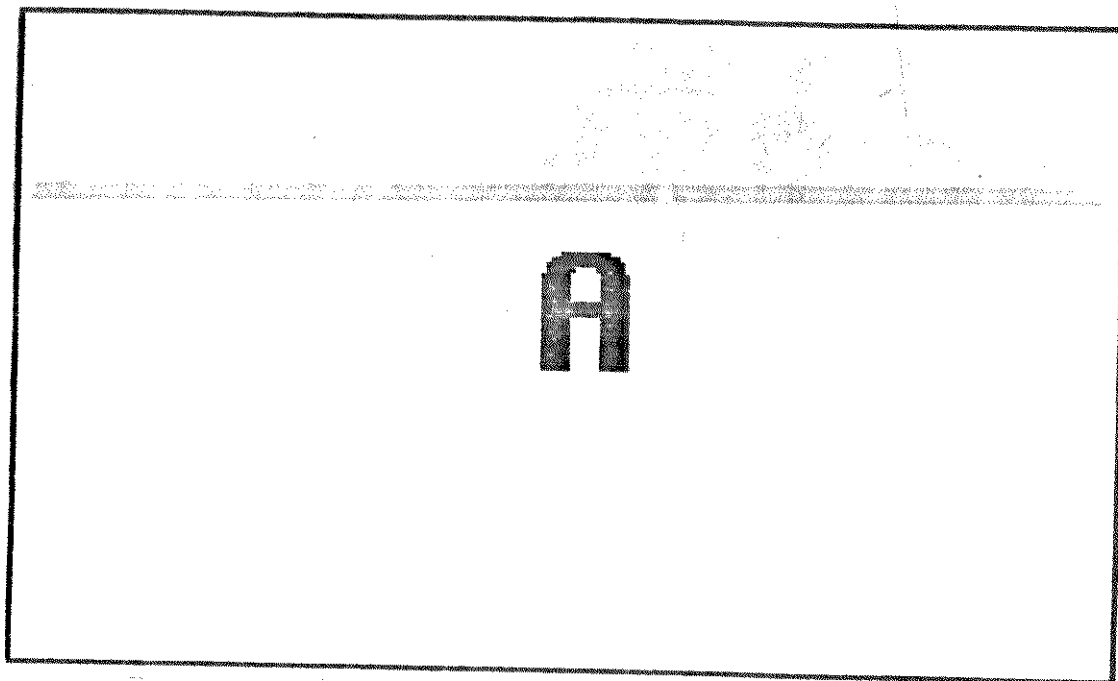


Figura 2.5 - Letra A perfeita

Usando-se uma gramática nebulosa, pode-se chegar à definição de uma "possível letra A" como sendo "um possível triângulo isósceles concatenado nas extremidades de sua base a duas retas aproximadamente verticais", e assim analogamente para os outros tantos caracteres alfanuméricos.

Porque é que se usou lógica nebulosa neste problema de reconhecimento de padrões ? As fronteiras entre os padrões não são exatamente bem definidas, como se pode notar entre as letras O, D e Q; existe o problema da ambiguidade; os padrões que se quer reconhecer são muitos (total de 26 maiúsculos, podendo chegar a 26 maiúsculos + 26 minúsculos + 10 algarismos), cada um deles podendo ter várias representações distintas (donde vem o problema da ambiguidade); e a matemática da *lógica nebulosa* oferece um ferramental apropriado para se tratar o problema da incerteza. Suponha que a imagem da figura 2.6 foi fornecida ao sistema de reconhecimento. Que caracter é este ?



Figura 2.6 - Exemplo de ambiguidade

Um dos principais problemas no projeto de um algoritmo *flexível* de reconhecimento é a capacidade de diferenciação das variações das imagens fornecidas, pois diferentes usuários desenham o mesmo caracter de maneiras distintas, assim como o mesmo usuário desenha o mesmo caracter repetidas vezes de maneira desigual. O algoritmo deve tolerar até um determinado nível de variação, sem perder sua habilidade de discriminar os caracteres que devem ser diferentes. É exatamente este ponto de vista que Silva e Roy [63] defendem, afirmando que os **descritores** (características ou peculiaridades dos objetos) devem ter quatro propriedades básicas:

1. **discriminação** - tomar valores significativos diferentes para objetos distintos;
2. **similaridade** - tomar valores os mais idênticos possíveis para objetos semelhantes;
3. **independência** - não deve haver correlação entre os descritores; e
4. **concisão** - suficiência com um pequeno número de descritores.

Muitas das ferramentas matemáticas indispensáveis para a manipulação dessas incertezas estão presentes na lógica nebulosa. Na maioria das aplicações práticas, existe uma determinada quantidade de incerteza no

processo sendo estudado.

Com a lógica nebulosa, pode-se buscar algumas semelhanças fortes ou fracas da imagem sendo analisada, com, por exemplo, a letra A, o algarismo 8, e assim sucessivamente. Isto se baseia no fenômeno conhecido como Princípio da Superposição da lógica nebulosa, que é o seguinte: "qualquer imagem fornecida ao sistema corresponde, com um certo valor de pertinência, a todas as classes de padrões".

Em outras palavras, dada uma imagem com um objeto que se quer reconhecer, esta é comparada com as 26 classes de padrões existentes, gerando então 26 valores de pertinência. O maior destes valores é que deverá fornecer a resposta, obviamente se este for maior que um *limiar mínimo*. Assim, a imagem 2.7 é definida como:

$$f(I) = \left\{ (X_A | 0.2), (X_B | 0.12), (X_C | 0.3), \dots, (X_I | 1.0), \dots \right\}$$

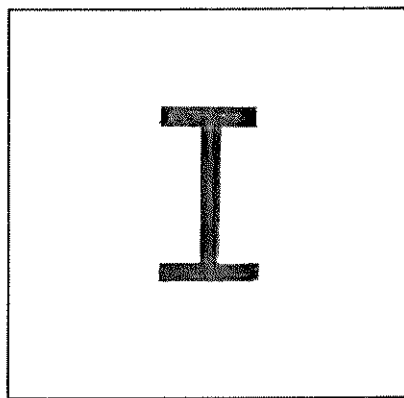


Figura 2.7 - Imagem da letra I

Pode-se facilmente concluir que a letra I acima corresponde simultaneamente a uma letra A com 0.2, a uma letra I com 1.0, e assim sucessivamente. Este Princípio da Superposição pode ser visualizado através do esquema 2.8 e do gráfico 2.9 a seguir.

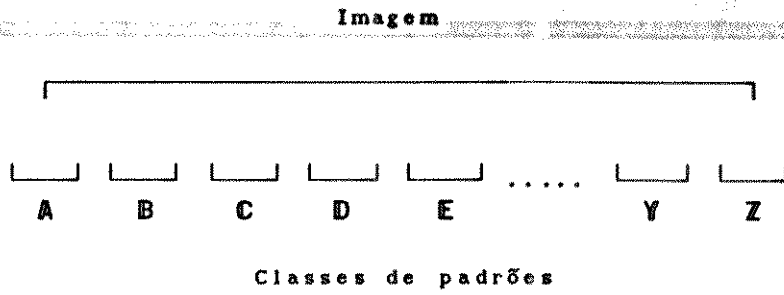


Figura 2.8 - Princípio da Superposição

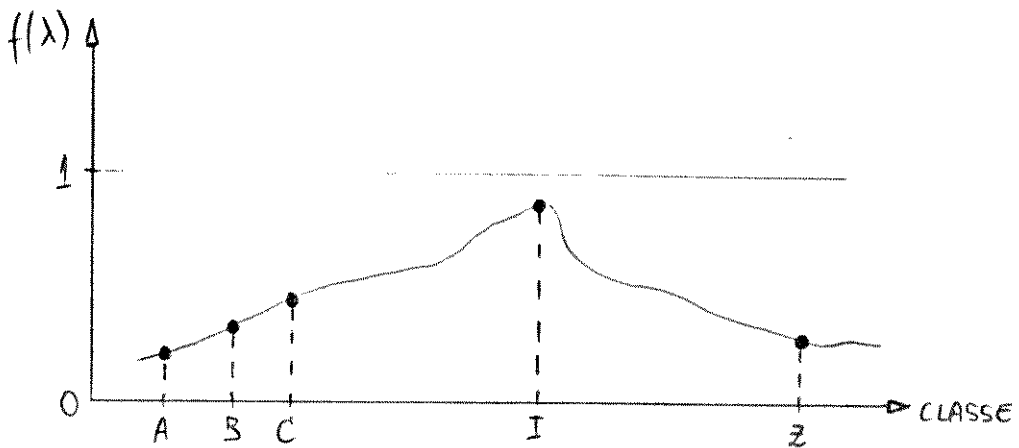


Figura 2.9 - Gráfico da função nebulosa

Assim, a imagem $\lambda \in \mathcal{U}$ (qualquer imagem λ do conjunto de referência \mathcal{U}) tem a seguinte notação:

$$\begin{array}{ccccccc} \lambda \in A, & \lambda \in B, & \lambda \in C, & \dots, & \lambda \in I, & \dots & \\ 0.2 & 0.12 & 0.3 & & 1.0 & & \end{array}$$

Para citar outro exemplo, suponha que existe um conjunto \mathcal{U} com 5 imagens.

$$\mathcal{U} = \{ I_1, I_2, I_3, I_4, I_5 \}$$

Agora suponha-se que existem apenas 3 classes de padrões representadas pelas classes das letras A, T e W. Então pode-se construir 3 subconjuntos nebulosos:

$$A = \{ (I_1 | 0.0), (I_2 | 0.1), (I_3 | 0.9), (I_4 | 0.2), (I_5 | 0.0) \};$$

$$T = \{ (I_1 | 0.1), (I_2 | 0.9), (I_3 | 0.0), (I_4 | 0.1), (I_5 | 0.1) \};$$

$$W = \{ (I_1 | 1.0), (I_2 | 0.1), (I_3 | 0.0), (I_4 | 0.1), (I_5 | 0.0) \};$$

Pode-se concluir que a imagem I_1 corresponde a uma letra W, a imagem I_2 é uma letra T, a imagem I_3 é uma letra A, e as imagens I_4 e I_5 não obtiveram o limiar mínimo necessário, ou seja, foram rejeitadas.

Seja uma relação nebulosa \mathcal{R} , um conjunto de imagens X de diversas classes de padrões e um conjunto Y de imagens a serem analisadas, dadas por:

$$\mathcal{R} = \{ (X_i, Y_j), \mu_{\mathcal{R}}(X_i, Y_j) \}$$

onde

$$\mu_{\mathcal{R}}(X_i, Y_j) = 1 - \frac{1}{\text{fator}} \left[\sum_{k=1}^P (PAD_i^k - IMG_i^k)^2 \right]^{1/2}$$

que calcula a distância entre uma imagem com P características discriminatórias com N classes de padrões. Veja que a equação acima consegue fornecer valores dentro de um contínuo [0,1].

Para exemplificar, suponha-se 4 classes de padrões e apenas 2 características discriminatórias. As classes de padrões formam nuvens de pontos no plano cartesiano de acordo com a figura 2.10 a seguir.

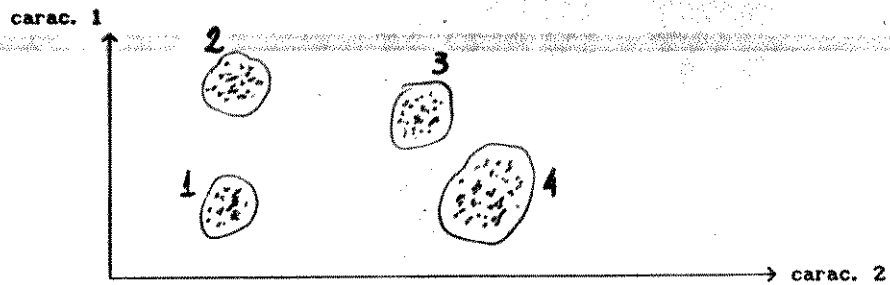


Figura 2.10 - Agrupamento de padrões

Achando o centro de gravidade para cada um destes agrupamentos, pode-se representar cada classe por um vetor que liga cada centro de gravidade com a origem, como na figura 2.11.

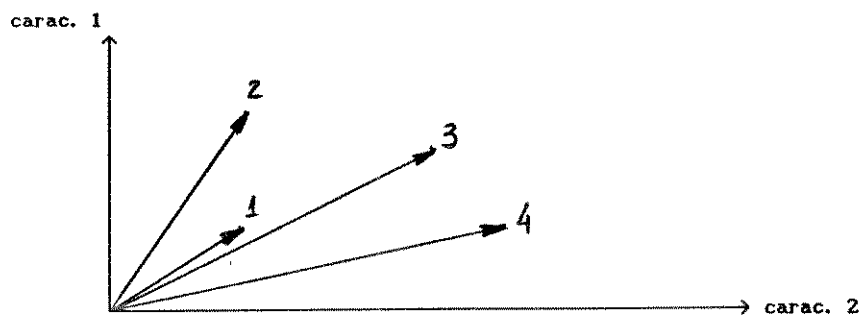


Figura 2.11 - Centros de gravidade

Quando se analisa uma certa imagem, esta corresponde a um determinado ponto neste mesmo plano cartesiano, e à menor distância entre este ponto e os centros de gravidade das N classes de padrões corresponde então a classe em que a imagem será classificada. Tem-se então a seguinte ilustração.

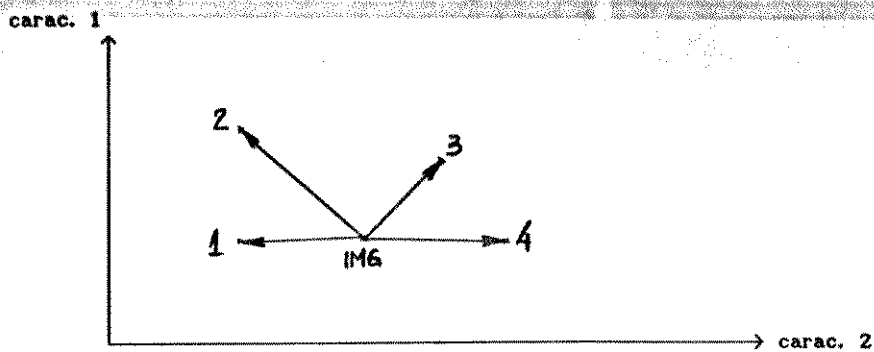


Figura 2.12 - Distâncias mínimas

Esta função, dada pela equação anterior, é a distância Euclideana entre dois pontos. Neste exemplo em particular, pode-se representá-la no plano cartesiano; entretanto, usando-se vários descritores, esta visualização passa a não ser mais possível.

Neste exemplo, cada classe de padrões e cada imagem tem 2 características calculadas, e portanto cada elemento é representado por meio do seguinte vetor:

$$V = [C_1, C_2]^T$$

e a função dada pela equação anterior faz então um mapeamento matematicamente definido pela expressão a seguir:

$$f(\lambda): \mathbb{R}^2 \longrightarrow \mathbb{R}$$

Para a formação destas "nuvens de pontos" mencionadas anteriormente, existem as técnicas nebulosas e não nebulosas (ou convencionais).

II.3. Variáveis Linguísticas

A lógica nebulosa faz uso de valores no contínuo $[0,1]$ para a função de pertinência, porém ao invés de explicitarmos que a função classificou a letra A com o valor $f(A) = 0.76$, seria útil a utilização de variáveis não numéricas ou não booleanas, que substituem o valor numérico por um valor menos preciso. A estas variáveis costuma-se chamar de *variáveis linguísticas*, pois os seus valores não são números, mas sim, como o seu próprio título diz, palavras ou sentenças em uma linguagem natural ou sintética, como por exemplo "*pouco*", "*muito*", "*nem tanto*", e assim por diante.

Zimmermann [71] define matematicamente uma variável linguística como estando caracterizada por uma quintupla $(\alpha, T(\alpha), U, G, M)$, na qual α é o nome da variável; $T(\alpha)$ denota o *conjunto-termo* de α , ou seja, o conjunto de nomes de valores linguísticos de α , com cada valor sendo uma variável nebulosa denotada geralmente por α com escopo sobre o universo de discurso U , que está associada a uma variável base μ ; G é uma regra sintática (que normalmente tem a forma de uma gramática) para geração do nome X de valores de α ; e M é uma regra semântica para associar U . Um X particular, que é um nome gerado por G , é chamado u . Deve ser notado que a variável base μ pode também ter valor por vetor.

Segundo Zadeh [7], a função principal das variáveis linguísticas é a de fornecer uma maneira sistemática para uma caracterização aproximada de fenômenos complexos ou mal-definidos. Em essência, distanciando-se do uso de variáveis quantificadas em favor do uso do tipo de descrição linguística empregada pelos humanos, adquire-se a capacidade de se manipular com relativa facilidade conhecimentos em sistemas que são bastante complexos para serem analisados através de técnicas tradicionais de matemática.

A mesma variável numérica com valor 0.76 seria substituída por uma variável linguística com valor "*não muito*", e assim, ao invés de se dizer que a imagem pertence à classe da letra A com 0.76 por cento de possibilidade, simplesmente diz-se que a imagem pertence "*não muito*" à classe da letra A.

Por que se diz que estas funções linguísticas de mapeamento são não determinísticas ? O motivo é muito simples, segundo Chang [56]. Dado um valor qualquer 0.76, dependendo da aplicação ou da função de mapeamento, esse valor numérico pode tomar qualquer valor linguístico. Por exemplo, se este valor fosse a taxa de erro de classificação de um sistema de visão, então 0.76 seria "muito alto", ou até "elevadíssimo", mas se fosse por exemplo a taxa de acerto do mesmo sistema, então seria linguisticamente "muito baixo", ou até "insignificante". Em outras palavras, não existe um algoritmo que, dado um valor numérico, forneça como resposta um valor linguístico. É por isso que a função é *heurística*, pois é construída de acordo com o usuário e com a aplicação.

Os graus intermediários podem ser mapeados linguisticamente como no exemplo da figura 2.13 a seguir.

$f(\lambda) \leq 0.1$	\rightarrow	"não parece"
$0.1 < f(\lambda) \leq 0.3$	\rightarrow	"parece muito pouco"
$0.3 < f(\lambda) \leq 0.5$	\rightarrow	"parece pouco"
$0.5 < f(\lambda) \leq 0.7$	\rightarrow	"parece muito"
$0.7 < f(\lambda) \leq 0.9$	\rightarrow	"realmente parece"
$f(\lambda) \geq 0.9$	\rightarrow	"é totalmente"

Figura 2.13 - Um possível mapeamento linguístico

As expressões PDL, que são utilizadas como *medidoras* do nível de pertinência do objeto com relação a cada classe de padrões, são traduzidas linguisticamente então como "*regra aproximada de uma possível letra A*", "*regra aproximada de uma possível letra B*", até a última, a "*regra aproximada de uma possível letra Z*".

É claro que este é um conceito extremamente interessante na área de reconhecimento de padrões. Tanto os métodos *decisão-teóricos*, considerados como clássicos, como os métodos *estruturais*, serão sensivelmente melhorados com o uso das variáveis linguísticas.

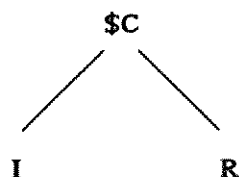
II.4. Exemplos de Funções Nebulosas

Qualquer função nebulosa que mapeia uma aplicação tem um contra-domínio dentro de um contínuo $[0,1]$. Simbolicamente, tem-se que:

$$f: D(\lambda) \longrightarrow [0,1]$$

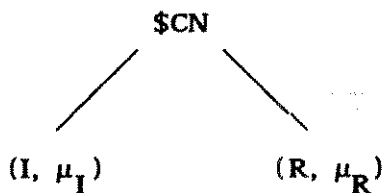
onde no presente trabalho o domínio $D(\lambda)$ representa o conjunto das imagens que se quer analisar. Se λ for um elemento, então $f(\lambda) = 0$ significa *nenhuma pertinência*, e $f(\lambda) = 1$ significa *pertinência total*.

Lee [30,51] exemplifica algumas funções nebulosas para o reconhecimento de *casas*, que a título ilustrativo, é definida simplificada como um "*triângulo isósceles concatenado verticalmente com um retângulo*". A casa pode então ser representada por meio da seguinte árvore:



onde I representa o triângulo isósceles, R representa o retângulo e a relação sintática $\$C$ entre I e R é a concatenação vertical.

Pelo uso do conceito de uma linguagem nebulosa, uma *casa aproximada* pode ser definida como sendo um "*triângulo aproximadamente isósceles concatenado verticalmente com um retângulo aproximado*". Faz-se necessário então uma mudança na representação da "casa nebulosa", agora através da seguinte árvore:



onde μ_I e μ_R são os graus de pertinência de "triângulo isósceles aproximado" e "retângulo aproximado", respectivamente.

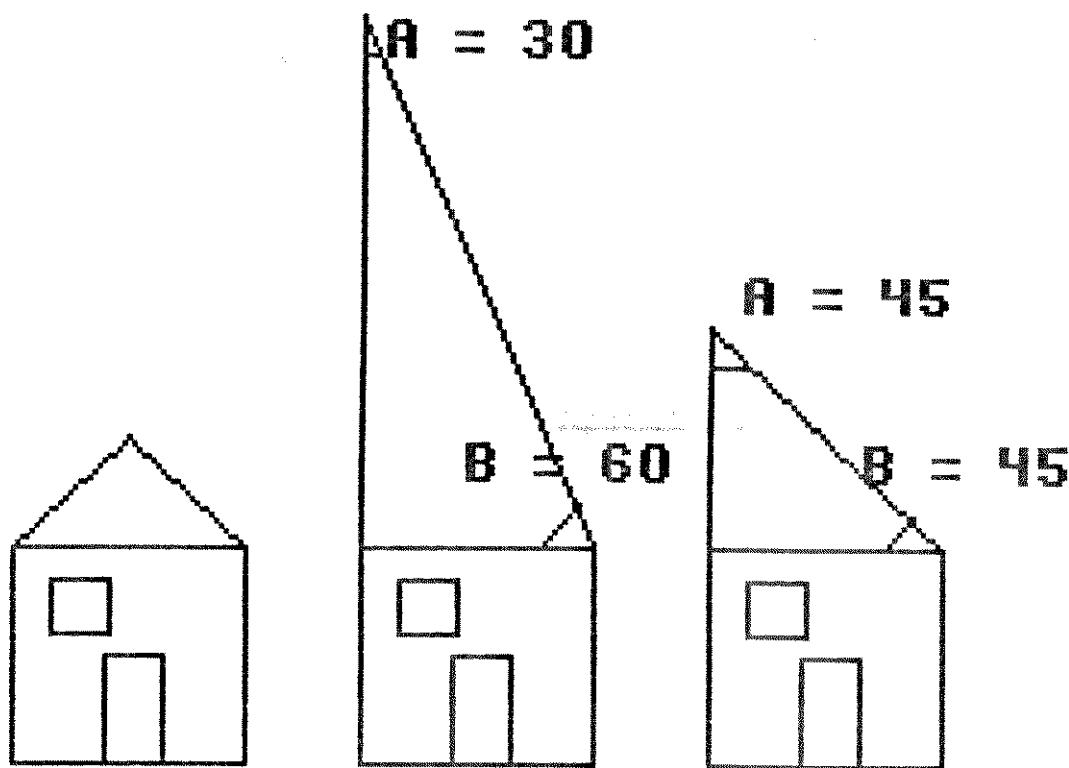
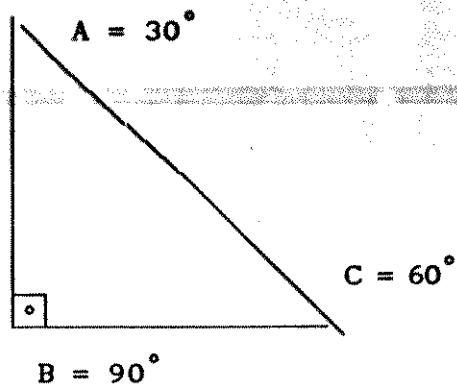


Figura 2.14 - Casas

Para o cálculo da função $f_I(\lambda) = \mu_I$, suponha um triângulo ∇ABC com os ângulos B e C como sendo os ângulos da base. A medida quantitativa da similaridade deste triângulo isósceles pode ser definida como:

$$\mu_I(\nabla ABC) = 1 - \frac{|B - C|}{90^\circ}$$

Por exemplo, suponha o triângulo a seguir.



A função acima citada resultará em:

$$\begin{aligned} \mu_I(\nabla ABC) &= 1 - \frac{|90 - 60|}{90} \\ &= 1 - \frac{30}{90} \\ &= 2/3 \end{aligned}$$

Linguisticamente, então, pode-se dizer que "a figura anterior do triângulo representa com 2/3 de certeza a possibilidade deste ser um *isósceles*". Se por acaso o retângulo der um resultado integral, ou seja, $\mu_R(\lambda) = 1$, então podemos inferir que a imagem representada por este triângulo e pelo retângulo representa com certeza de 2/3 a possibilidade da figura ser uma casa.

CAPÍTULO III

NOÇÕES DE RECONHECIMENTO DE PADRÕES

III.1. Classificação dos Reconhecedores

III.2. Casamento

III.2.1. Medida de Correlação Básica

III.3. Métodos Decisão-Teóricos

III.4. Métodos Estruturais

III.4.1. Aproximação Poligonal

III.5. Métodos Híbridos

III.1. Classificação dos Reconhedores

Entende-se o registro de imagens como o processo de encontrar um relacionamento forte entre pontos ou regiões de uma imagem com uma representação, de preferência não ambígua, de um certo objeto que se quer localizar nesta imagem. Os classificadores são entendidos como máquinas ou autômatos que associam determinados objetos em uma imagem à uma classe de padrões com um mínimo de intervenção humana.

Alguns autores dividem um sistema genérico de reconhecimento de padrões em unidades lógicas que são : 1) sensores; 2) pré-processador; 3) extrator de características; 4) classificador; e 5) um processo de treinamento e aprendizado, estes últimos ainda sub-divididos em *supervisionados* e *não supervisionados*, conforme a figura 3.1.

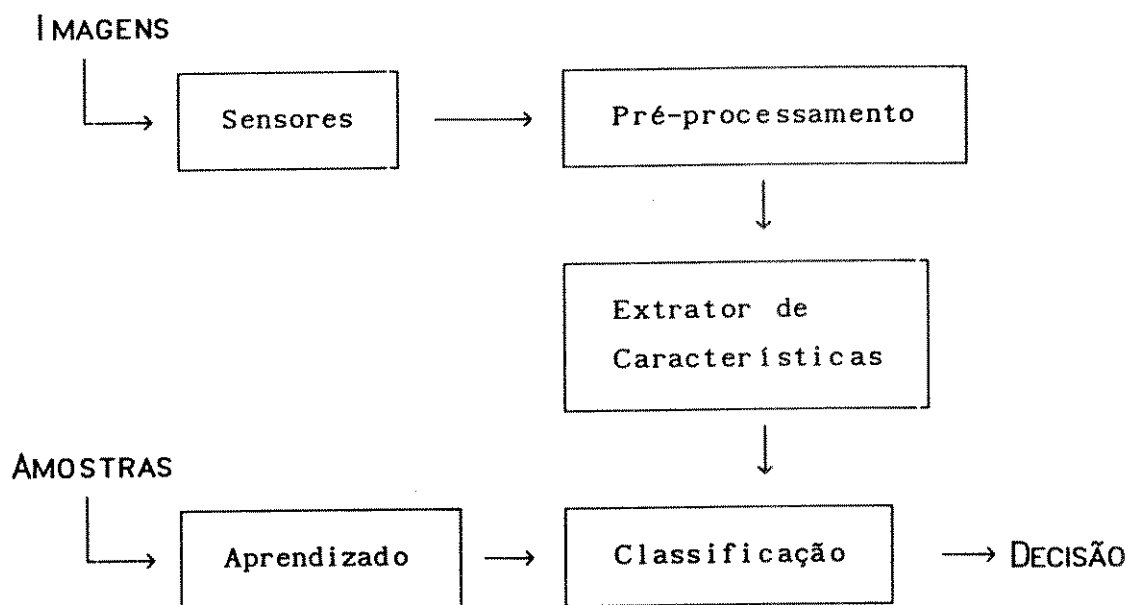


Figura 3.1 - Um sistema Geral de Reconhecimento

Desde o início do processamento de imagens sempre foi de interesse

classificar os sistemas de reconhecimento de padrões, não só com o fim de facilitar o estudo desses sistemas, como também pelo fato do agrupamento permitir uma generalização para sistemas pertencentes à mesma categoria.

Uma taxonomia possível seria:

- 1. Casamento com Gabarito
- 2. Método Decisão-Teórico
- 3. Método Estrutural
- 4. Métodos Híbridos

O presente trabalho situa-se no terceiro grupo, pois baseia-se em uma metodologia estrutural para classificar as formas existentes na imagem.

Entretanto, Ruiz [61] classifica os métodos de reconhecimento em apenas dois tipos fundamentais, os baseados no sinal e os de registro simbólico, onde a primeira abordagem (efetuada por processamento numérico) engloba o casamento com gabarito e a segunda (efetuada sob processamento simbólico) engloba os métodos decisão-teórico e estrutural.

Bunke [69] constrói uma tabela em que classifica os métodos de reconhecimento em quatro categorias, como mostra a figura 3.2.

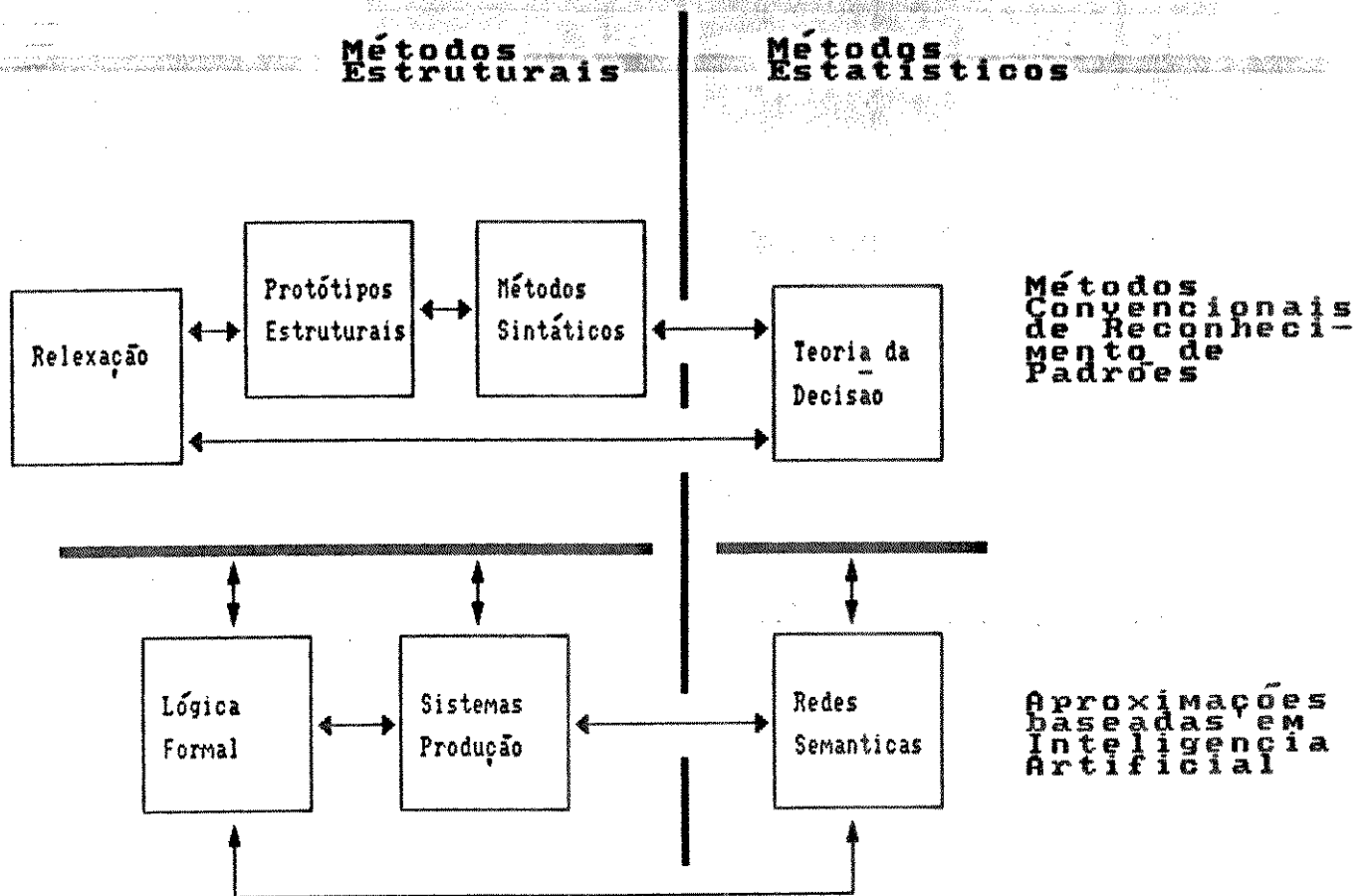


Figura 3.2 - Tabela de métodos híbridos

Outra forma de classificação dos sistemas de reconhecimento divide estes em também quatro outras grandes classes :

- 1. heurísticos
- 2. determinísticos
- 3. estatísticos
- 4. incertos ou nebulosos

É desejável também que o reconhecedor rejeite um padrão que tenha pouca semelhança com todas as classes de padrões que o sistema tenha capacidade de reconhecer.

III.2. Casamento

Este método é o mais simples e o mais fácil de se interpretar, consistindo basicamente na aplicação em conjuntos de pixels da imagem uma máscara que representa o padrão que se quer reconhecer. Se esta máscara não se casar com nenhum conjunto de pixels, conclui-se que o padrão não se encontra na imagem; caso contrário, o padrão é extraído e reconhecido.

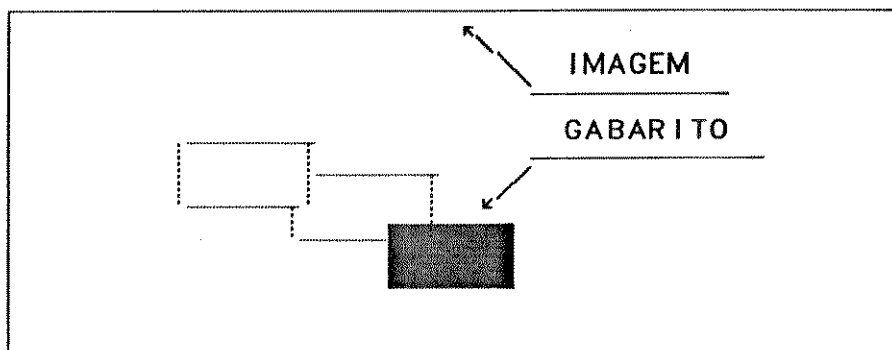


Figura 3.3 - Método *template matching*

Se o padrão que se quer reconhecer for muito simples, como por exemplo um círculo ou um quadrado, este método seria apropriado, mas se o padrão que se quer reconhecer for complexo, como por exemplo um *caracter manuscrito*, este método não será capaz de reconhecer o padrão na grande maioria das imagens. Outro problema que este método apresenta é quanto ao tratamento do escalamento e rotação dos padrões na imagem.

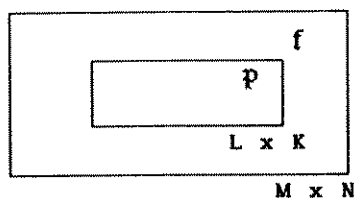
III.2.1. Medida de Correlação Básica

A técnica clássica de medida de semelhança entre duas figuras baseia-se no cálculo da medida de correlação entre as funções que determinam as imagens. Por exemplo, pode-se associar as funções f e p a valores binários em matrizes correspondentes à imagem, f com dimensão $M \times N$ a matriz da imagem e p com dimensão $L \times K$ a matriz do padrão.

Esta medida em sua forma normalizada, é expressa como segue:

$$R(u,o) = \frac{\sum_{i=1}^K \sum_{j=2}^L p(i,j) \times f(i+u, j+o)}{\sqrt{\sum_{i=1}^K \sum_{j=2}^L [f(i+u, j+o)]^2}}$$

onde, o par (u,o) é o intervalo de deslocamento do padrão p dentro da imagem f , como na figura abaixo, e a relação $p(i,j) < f(i,j)$ somente é válida se $f(i,j) = c * p(i,j)$, $\forall i,j$ e $c =$ constante.



De modo geral, a função de correlação básica normalizada $R(u,o)$ deve ser medida para $(N - K + 1) \times (M - L + 1)$ pontos, com o número máximo de comparações entre a janela e o padrão será $K \times L \times (N - K + 1) \times (M - L + 1)$.

III.3. Métodos Decisão-Teóricos

Nesta metodologia procura-se localizar um objeto numa imagem extraíndo-se da mesma uma certa quantidade de atributos considerados significativos para o processo de reconhecimento e que melhor representem a forma e a localização do objeto. Estes são armazenados em um **vetor de atributos**, que será utilizado pelos critérios de classificação.

Como exemplo dos atributos que se podem extrair de uma imagem citam-se : área, perímetro, momentos invariantes, momentos de primeira e segunda ordens, centro de gravidade, circularidade, energia de fronteira, número de cantos, dentre outros.

Cada classe de padrões do sistema possui um **vetor padrão de atributos** que será comparado ao vetor extraído da imagem, e a classe de padrões que apresentar o melhor resultado nesta operação será dada como sendo a classe à qual pertence o objeto da imagem.

Dentre as várias formas de se comparar dois vetores de atributos, existem os chamados **classificadores Bayesianos** que utilizando medidas da média e do desvio padrão, avaliam por meio de funções de verossimilhança a classe de padrões que mais se aproxima do vetor extraído da imagem. Tipicamente os classificadores são projetados com o critério de se minimizar a probabilidade condicional de erro.

A grosso modo pode-se dizer que existindo **N** classes de padrões no sistema de reconhecimento, necessita-se de **X** atributos discriminantes que se relacionam através da seguinte equação:

$$2^X \geq N$$

onde **X** é o menor número inteiro que satisfaça a expressão acima. Se,

entretanto, os atributos escolhidos não forem suficientemente discriminadores, então será necessário um número maior de atributos para se efetuar a classificação.

Uma crítica feita a esta aproximação do reconhecimento (Nandhakumar e Aggarwal [8]), refere-se ao fato de que como as técnicas estatísticas de reconhecimento de padrões são independentes do domínio da aplicação, estas ignoram propriedades estruturais específicas de sinais individuais. Outro fator crítico importante é de que o método decisão-teórico de reconhecimento de padrões permite apenas a descrição da classe do padrão, não descrevendo o próprio padrão de modo a permitir sua geração, como os métodos gramaticais discutidos a seguir.

III.4. Métodos Estruturais

Em sistemas tradicionais de reconhecimento de padrões a análise de uma cena é entendida como a designação de uma classe da qual esta faria parte. Já em sistemas mais modernos, com aplicações mais especializadas, há maior interesse pela descrição do objeto, e não apenas pela classificação da cena.

Suponha-se, por exemplo, o caso das ferramentas para CAD (Bhanu [65]). Ao entrarmos com a imagem padrão de *uma porca e de um parafuso* para o sistema, qual seria a vantagem de se armazenar apenas a imagem, como no caso de "template matching", ou ainda de se classificar "*é uma porca ou um parafuso*", como no método decisão-teórico ? Neste caso melhor seria armazenar algumas estruturas no Banco de Dados do CAD que fossem representativas de *uma porca e de um parafuso*. Considere-se a cena da figura 3.4 (Duda e Hart [13]).

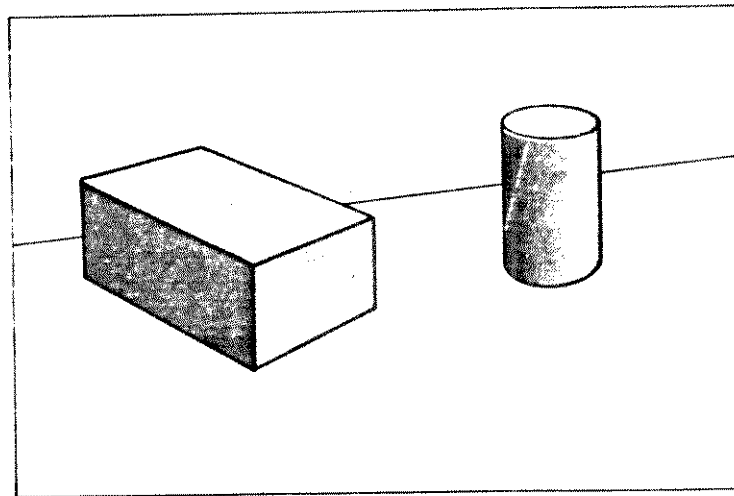


Figura 3.4 - Foto a ser analisada

De uma maneira muito grosseira pode-se dizer que a figura caracteriza-se por conter *uma caixa e um cilindro*. Se há interesse numa descrição estrutural da cena, pode-se representá-la com mais detalhes como: *uma caixa à esquerda do cilindro*. Se fosse necessário um nível maior de detalhamento, poder-se-ia escrever que a caixa está dividida em 3 quadriláteros, e assim sucessivamente até o nível de detalhe que se queira, como na figura 3.5.

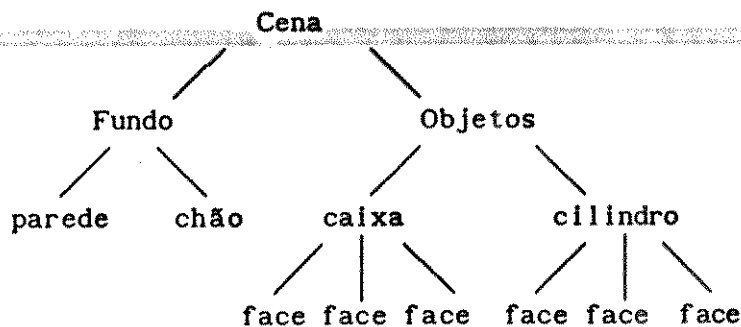


Figura 3.5 - Descrição hierárquica da foto

Segundo Thomason [70], Bunke [69] e Wong [66], existem várias formas de se descrever estruturalmente os padrões que se quer reconhecer, dentre as quais podem-se citar as seguintes:

1. grafos aleatórios;
2. morfismos;
 - 2.1. funções injetoras (*monomorfismo*)
 - 2.2. funções sobrejetoras (*epimorfismo*)
 - 2.3. funções bijetoras (*isomorfismo*)
3. grafos relacionais com atributos;
4. ilustrações adicionais;
5. cadeias de símbolos;
6. árvores rotuladas.

Dentre os métodos de reconhecimento estruturais citam-se :

1. geradores de *strings*;
2. gramáticas de árvores;
3. gramáticas lineares;
4. gramáticas de grafos.

Algumas propostas para a representação de padrões por grafos são: *linguagem de descrição de padrões* (PDL), proposta por Shaw [48, 49], o *grafo de adjacência de regiões* (RAG), proposta por Pavlidis [43], e *grafos relacionais com atributos* (ARG) proposto por Tsai e Fu [32]. Os métodos sintáticos, além

disto, são uma importante sub-classe da aproximação estrutural.

Usualmente em reconhecimento sintático os padrões são demasiadamente complexos, sendo normalmente descritos por decomposição em sub-padrões mais simples, difíceis de serem tratados pela abordagem decisão-teórica ou *template matching*. O porque disto reside no fato de que a possibilidade de variação na forma e tamanho dos padrões mais complexos é muito grande, dificultando portanto a construção de um gabarito que reconheça todas estas possíveis variações. Para o caso da análise decisão-teórica, a quantidade de parâmetros que deve-se utilizar no *vetor de atributos* torna-se uma solução muito cara do ponto de vista computacional.

Os classificadores estruturais reconhecem os objetos através da definição recursiva de seus elementos mais complexos em função de seus elementos mais simples ou *sub-termos*, até que ocorram as primitivas de reconhecimento. Para o caso estrutural, os extratores de primitivas tentam extrair estruturas mais globais, como retas, curvas, vértices, etc, ao contrário dos métodos decisão-teóricos, que tentam calcular área, perímetro, etc.

O primeiro passo para o reconhecimento por meio de métodos estruturais é a escolha do modelo sintático para a descrição do padrão. Define-se o conjunto de primitivas que serão utilizadas, sendo que esta escolha é muito influenciada pelos tipos de padrões que se quer reconhecer, pela tecnologia disponível para a implementação do sistema e pelos recursos computacionais que se dispõe para o sistema. Não existe até a presente data nenhuma solução geral para o problema da seleção de primitivas para o caso da aproximação estrutural.

Supondo-se a existência de uma solução satisfatória para o problema da seleção de primitivas, o próximo passo para o sistema de reconhecimento estrutural é a construção de processos ou mecanismos que farão a geração da linguagem que descreverá os padrões em estudo, por exemplo uma gramática para caracteres alfanuméricos.

A estrutura definida sintaticamente através de uma gramática pode ser interpretada como uma árvore gerada pelas regras de produção da gramática, onde os símbolos terminais são as *folhas* e os não-terminais são os *arranjos*

permitidos de sub-grafos ou sub-estruturas.

O passo seguinte será o projeto de um módulo que reconheça os padrões gerados pela gramática. Para uma entrada representada pela sentença β , o problema do reconhecimento pode ser matematicamente representado pela seguinte questão :

$$\beta \in L(G_i), \text{ para } i = 1, 2, \dots, N ?$$

onde L é a linguagem gerada pela gramática G para os N padrões existentes.

Dada a sentença β representando um objeto desconhecido, o processo de reconhecimento linguístico fará uma análise sintática da sentença afim de descobrir : (1) se β está gramaticalmente bem formada; e (2) a qual das N classes de padrões mais se aproxima a sentença β .

Para se averiguar se a sentença β é bem formada, é necessário testar a pertinência de β na linguagem $L(G_i)$, o que pode ser feita de duas maneiras distintas. Utilizando a lógica tradicional, a pertinência é dada por 0 representando a não pertinência, ou seja, a sentença β está gramaticalmente mal formada, e 1 representando a pertinência, ou seja, β é bem formada. Já com a utilização da lógica nebulosa, a questão da pertinência passa a assumir os valores dentro do contínuo $[0,1]$, onde 0 significa a não pertinência e 1 significa a pertinência total, com a possibilidade de diversos graus intermediários de pertinência.

O critério para um casamento com sucesso no método estrutural é a existência da medida de acerto entre a sentença representando o objeto desconhecido, e a sentença representando a classe de padrões que se está comparando. Esta medida de acerto deve ser suficientemente maior para apenas uma classe de padrões existente no sistema, e que será então dada como a classe à qual pertence a imagem fornecida.

Um sistema de reconhecimento linguístico de padrões consiste de três partes principais representadas graficamente pela figura 3.6.

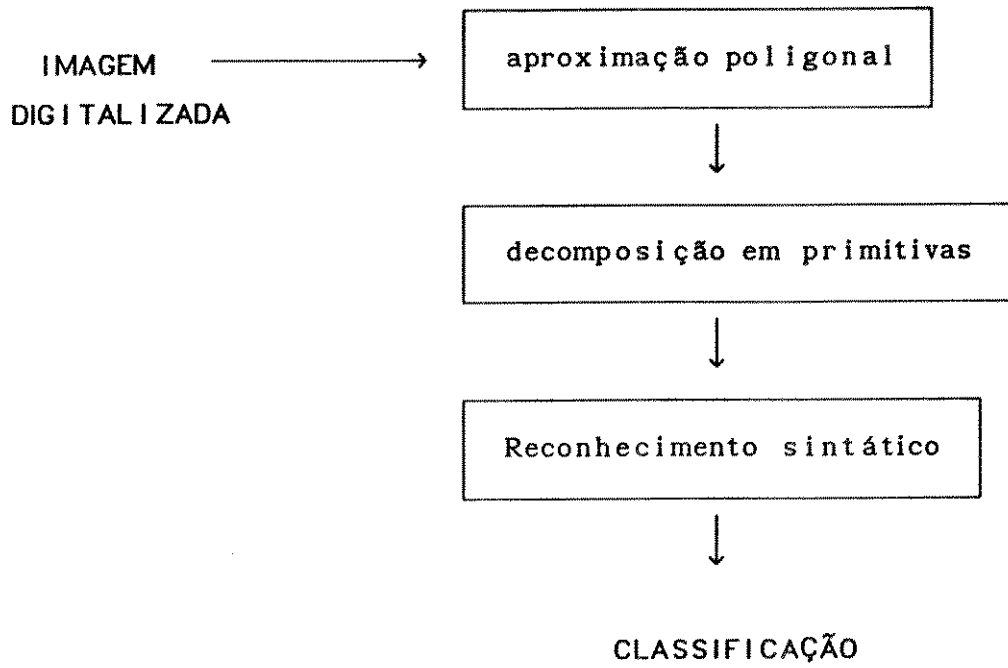


Figura 3.6 - O reconhecimento linguístico

III.4.1. Aproximação Poligonal

Segundo Ruiz [61], nota-se na aproximação sintática que os elementos de comparação tem estruturas hierárquicas mais elevadas, portanto a complexidade da fase de pré-processamento cresce, e é muito comum utilizar-se de aproximação poligonal para reduzir esta complexidade e facilitar a tarefa do classificador.

Segundo Pavlidis [41, 42], a principal característica desta aproximação é o uso de pré-processamento numérico significativo dos dados (*pixels*) sem o uso de informação semântica, seguido por uma análise baseada em semântica dos dados agora fortemente reduzidos. Em outras palavras, a aproximação poligonal transforma uma sequência desordenada de pixels em uma estrutura ordenada, como uma reta, uma curva, etc, de tal modo que a detecção das primitivas do método estrutural se torne significativamente mais fácil e rápida, gerando características simples mas poderosas para a classificação. A principal dificuldade da aproximação sintática no caso de reconhecimento de padrões reside na extração de informações desta sequência desordenada de pixels sobre retas, ângulos, concavidades, etc, das imagens digitalizadas.

Tomemos por exemplo a figura 3.7 do algarismo 2.



Figura 3.7 - Algoritmo 2 real

Pode-se representá-lo através das aproximações das diversas curvas que ele possui através das retas contínuas também mostradas na figura. Deste modo, o algoritmo 2 pode ser representado pela figura 3.8.

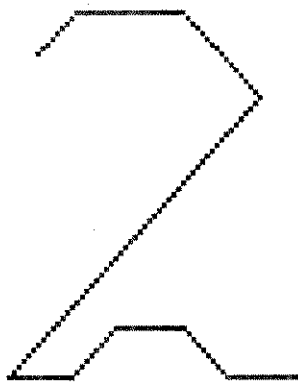
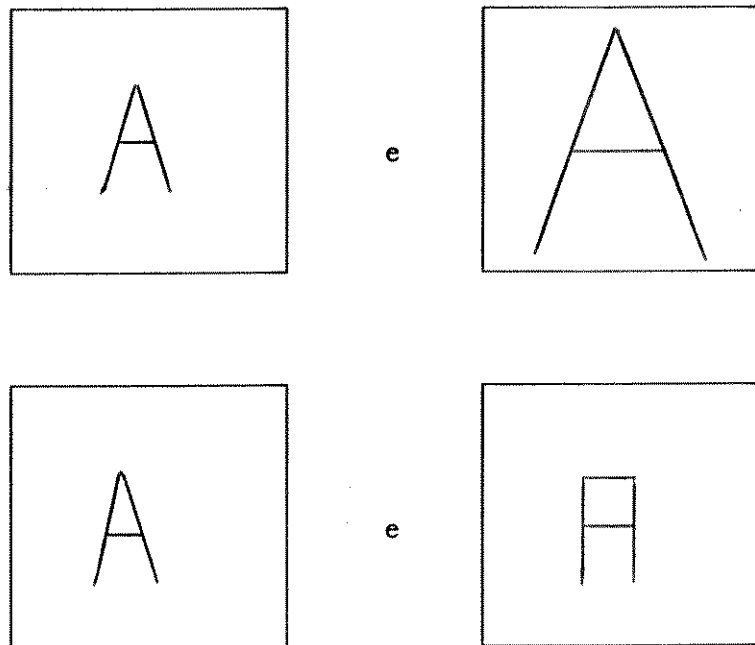


Figura 3.8 - Algoritmo 2 aproximado

Esta aproximação tem a importante missão de *melhorar a qualidade da imagem em termos estruturais*, tentando concatenar várias poligonais pequenas em algumas grandes, bem como de tirar as poligonais pequenas restantes, que normalmente são inúteis para a fase de análise.

Existe uma diferença fundamental entre as imagens dos grupos I e II a seguir.



No primeiro par de imagens, a diferença entre elas reside no fato das primitivas usadas serem de tamanhos diferentes, mas são do mesmo tipo e possuem a mesma quantidade de primitivas usadas. Isto quer dizer que a **estrutura do padrão é a mesma**. No segundo par de imagens, a diferença consiste nos tipos usados e em que quantidades, apesar de serem do mesmo tamanho. Isso implica numa **diferença estrutural** entre ambas.

Um dos fortes motivos para o uso destas aproximações poligonais é a sua relativa facilidade para o uso com as expressões PDL a serem descritas no próximo capítulo, bem como uma certa facilidade para o uso com uma função de pertinência nebulosa quando a operação de concatenação não fôr perfeita.

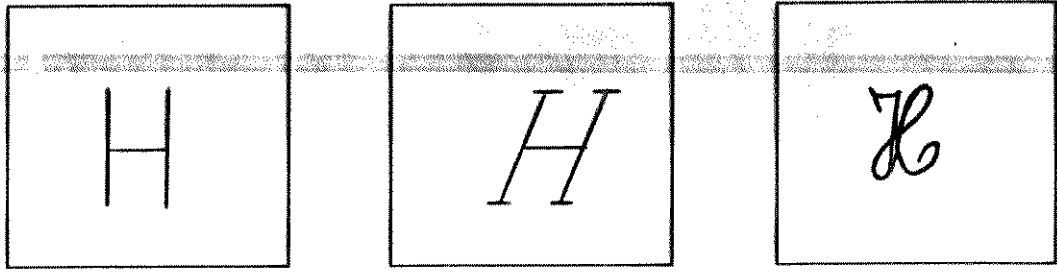


Figura 3.9 - Diferenças estruturais

III.5. Métodos Híbridos

Estes métodos híbridos se aplicam a sistemas onde os métodos descritos anteriormente atuando em separado não conseguem obter a mesma eficiência que conseguem os sistemas que têm estes métodos combinados.

Existem sistemas por exemplo que utilizam os métodos *template matching* e/ou decisão-teóricos para efetuar a tarefa de classificação, pois necessitam de uma decisão, e que utilizam os métodos estruturais para a descrição dos padrões analisados, pois é necessário o seu armazenamento.

Concluindo, se as duas primeiras aproximações para o problema de reconhecimento de padrões são métodos quantitativos, o último (o estrutural) é um método qualitativo, e é por isto que ele é um método tão interessante para se resolver este problema de reconhecimento e classificação de padrões

CAPÍTULO IV

DESCRIÇÃO DO RECONHECIMENTO LINGUÍSTICO

IV.1. Teoria das Linguagens Formais

IV.2. Expressões PDL

IV.2.1. Propriedades e Teoremas Básicos

IV.2.2. Exemplos

IV.3. Fase de Pré-processamento

IV.4. Fase de Detecção

IV.5. Fase de Análise e Decisão

IV.1. Teoria das Linguagens Formais

As linguagens podem ser classificadas em *naturais* e *artificiais*. No final da década de 1950 Noam Chomsky propôs uma teoria para se estudar linguagens que ficou conhecida pelo nome de **Teoria das Linguagens Formais**.

As linguagens naturais (desprovidas de formalismos, e carregadas de semântica e pragmática) são extremamente complexas de serem tratadas computacionalmente, por isto não são usadas em reconhecimento de padrões. Esta é aliás uma sub-área da Inteligência Artificial, e ainda pode ser sub-dividida em *naturais* (as linguagens completas) e *jargão* (sub-domínio especializado de uma linguagem).

Uma gramática formal é um conjunto de regras de produção com a forma geral :

$$\langle \text{lado esquerdo} \rangle ::= \langle \text{lado direito} \rangle$$

expressa em BNF (Backus-Naur Form) e onde se explicita o desdobramento de $\langle \text{lado esquerdo} \rangle$ em outros termos constantes em $\langle \text{lado direito} \rangle$.

Segundo Chomsky as gramáticas são classificadas de acordo com a forma de suas regras de produção, convenientemente chamada de *hierarquia de Chomsky*. Devido à variedade de conjuntos que obedecem a estas regras de produção, Chomsky tentou dividi-los em dois grandes sub-grupos, os de *gramática irrestrita* (os quais Fu [14] considera muito difíceis de serem tratados computacionalmente) e os de *gramática restrita*.

As gramáticas irrestritas ou do tipo 0 (zero) não têm restrições na forma de suas regras. Assim, pode-se ter literalmente *qualquer* coisa dos dois lados das regras de produção. Novamente Fu prova que o problema de se verificar se uma certa cadeia de caracteres foi gerada por uma gramática irrestrita é em geral indecidível.

Como não é interessante do ponto de vista computacional o uso das gramáticas irrestritas, costuma-se utilizar as gramáticas restritas, onde algumas restrições na forma de suas regras de produção as tornam passíveis de serem tratadas computacionalmente.

As gramáticas sensíveis (ou sensitivas) ao contexto, também chamadas de gramáticas do tipo 1, são gramáticas onde as regras de produção só podem ser disparadas dependendo do contexto onde o lado esquerdo da regra de produção estiver inserido. Daí a origem do nome.

As gramáticas livres de contexto, também chamadas gramáticas do tipo 2, são gramáticas onde, independente do contexto onde o lado esquerdo das regras de produção estiver inserido, se houver casamento entre este lado da regra e um sub-termo da sentença que está sendo analisada, então a regra é disparada.

E por último vêm as gramáticas regulares ou de estados finitos, também chamada de gramáticas do tipo 3, que são gramáticas onde as regras de produção só podem ser das seguintes formas:

$$V_n ::= V_t \text{ ou } V_n, \text{ ou então}$$

$$V_n ::= V_t,$$

onde V_n e V_t são os conjuntos de símbolos não terminais e terminais da gramática, respectivamente.

Esta hierarquia pode ser vista na figura 4.1.

O reconhecimento de padrões pela aproximação sintática torna esta tarefa semelhante à análise sintática de linguagens, porque as primitivas de reconhecimento podem ser interpretadas como letras em uma linguagem, a concatenação de primitivas como palavras e, finalmente, uma imagem como um texto linguístico completo.

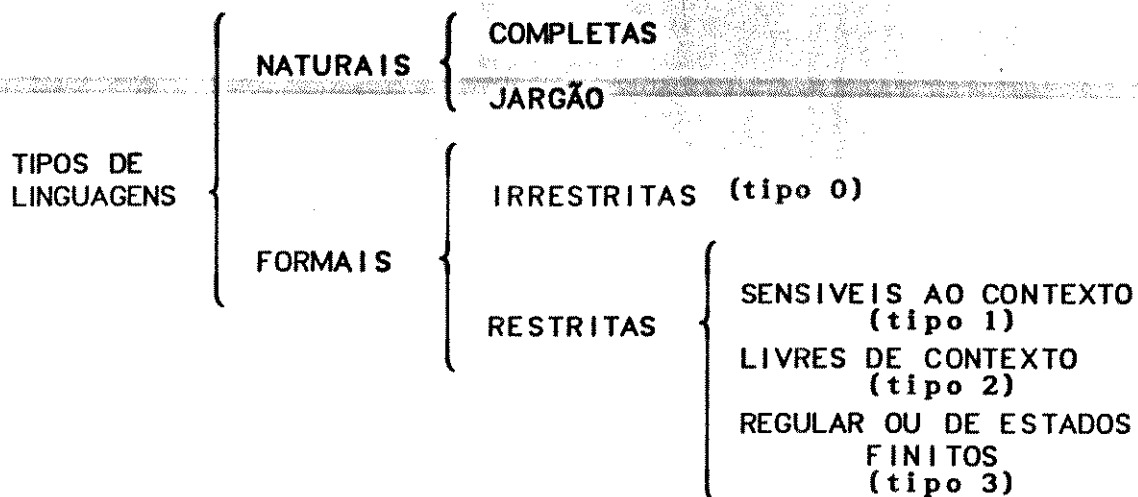


Figura 4.1 - Hierarquia de gramáticas de Noam Chomsky

Um sistema básico estudado em *Teoria das Linguagens Formais* é um sistema que permita a construção de conjuntos finitos de regras de produção que gerem exatamente o conjunto de sentenças em uma linguagem especificada. A estes conjuntos de regras de produção dá-se o nome de gramática que pode ser de vários tipos, conforme descrito a seguir.

Uma gramática ordinária G é uma quádrupla $G = (V_t, V_n, S, P)$, onde V_t é um conjunto finito de símbolos terminais; V_n é um conjunto também finito de símbolos não-terminais, conseqüentemente $V = V_t \cup V_n$ é o vocabulário da gramática; $S \in V_n$ é o símbolo de partida; e P é um conjunto finito de regras de produção da forma :

$$\alpha \longrightarrow \beta,$$

onde α e β são membros do conjunto V^* de todas as sentenças (incluindo a sentença nula λ) sobre o vocabulário V .

Como bem definido em Furtado e Casanova [22], as gramáticas livres de contexto são gramáticas restritas às regras de produção com um único símbolo não-terminal na parte esquerda e uma parte direita não-vazia, e o processo de aplicar uma certa gramática livre de contexto em uma sentença representativa de um objeto pode ser representada por uma árvore de derivação, como

consequência da propriedade supra-citada de suas regras de produção.

Uma gramática G qualquer pode ser usada tanto no modo gerativo, quanto no modo analítico. Se as regras de produção são usadas da esquerda para a direita, então a gramática está no modo gerativo, e se são usadas da direita para a esquerda, a gramática está no modo analítico. Uma cadeia de símbolos terminais construída usando-se uma dada gramática no modo gerativo é chamada de sentença. O conjunto de todas as sentenças que podem ser construídas usando a gramática é chamado de linguagem da gramática. Pelo processo descrito acima, se imagens podem ser analisadas linguisticamente, elas também podem ser geradas pela reversão do processo de análise.

Uma extensão ao conceito de gramáticas ordinárias é a idéia de gramáticas nebulosas, baseadas no conceito de sub-conjuntos nebulosos. Nebulosidade é uma propriedade adicionada à linguagem pela modificação da gramática, como segue.

Uma gramática nebulosa produz uma linguagem que é um conjunto nebuloso de sentenças, com o valor de pertinência de cada sentença à linguagem dentro do intervalo $[0,1]$, onde 0 indica nenhuma pertinência e 1 indica total pertinência. Estas linguagens mostraram-se, segundo DePalma e Yau [10], promissoras no tratamento de problemas de reconhecimento de padrões, onde o conceito embutido é possibilístico ou nebuloso.

Uma gramática nebulosa FG é uma 6-tupla $FG = (V_t, V_n, P, S, J, f)$, onde V_t , V_n , P e S são respectivamente o alfabeto terminal, o alfabeto não-terminal, o conjunto das regras de produção e o símbolo de partida, como nas gramáticas ordinárias; $J = \{r_i \mid i=1, 2, \dots, n\}$ é o conjunto de rótulos distintos para as produções em P , e f é a função de pertinência nebulosa definida por $f: J \rightarrow [0,1]$. Estes rótulos distintos dados pelo conjunto J utilizam as variáveis linguísticas para o cálculo da função de pertinência nebulosa f .

Pode-se notar que as gramáticas ordinárias são um caso especial das gramáticas nebulosas, exatamente quando $f(r_i) = 1$ para todo r_i em J . Gramáticas nebulosas podem ser classificadas também de acordo com as suas regras de produção, pela mesma hierarquia anterior.

Exemplificando a Teoria Formal das Linguagens, suponha-se que se quer reconhecer os capacitores e os resistores em um circuito elétrico, e testar para ver se o circuito está funcionalmente correto ou não. As primitivas seriam as figuras de um resistor e de um capacitor, e a gramática G_1 necessária faria o teste de consistência do circuito, onde a disposição do circuito seria estudada. Esta gramática seria dada pelos seguintes componentes:

$$\begin{aligned} V_n &= \{S\} \\ V_t &= \{a, b\} \\ S &= \{S\} \\ P &= \{ S \rightarrow aS, \\ &\quad S \rightarrow b \}. \end{aligned}$$

As primitivas a e b estão representadas pela figura 4.2.

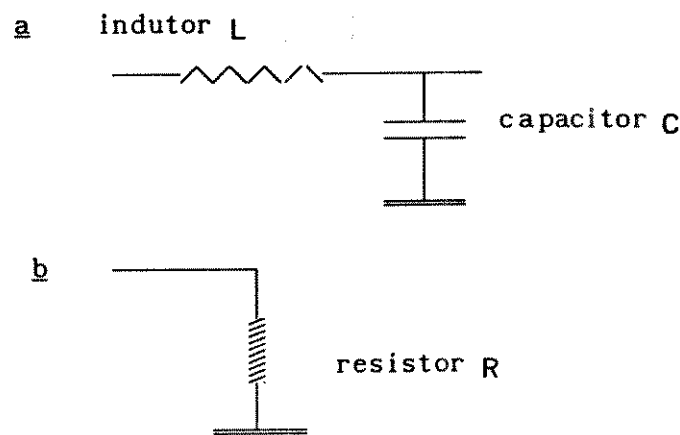


Figura 4.2 - Primitivas da gramática G_1

Já o padrão mostrado na figura 4.3 corresponde à sentença "aab", que foi gerada a partir da seguinte sequência de aplicações das regras de produção:

$$S \rightarrow aS \rightarrow a(aS) \rightarrow a(ab) \rightarrow aab$$

Como foram aplicadas de maneira única, a gramática não é ambígua. A gramática reconheceu no circuito elétrico a sequência de dois blocos de capacitores e um

bloco de resistor, nesta ordem. Se esta sequencia não estiver correta, o circuito está rejeitado.

O problema do reconhecimento será : dada a sentença aab, verificar se esta faz parte da gramática. Uma das maneiras de realizar a verificação é aplicar as regras de produção no sentido inverso ao das setas.

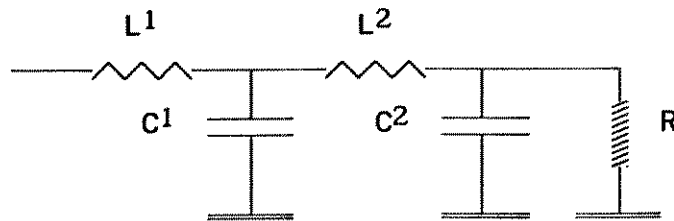


Figura 4.3 - Padrão a ser reconhecido

Uma vez solucionado o problema da definição de uma gramática, volta-se a atenção para o problema de usar uma gramática para analisar uma sentença formada por primitivas representando uma certa imagem. Ao processo de responder se uma dada sentença foi gerada ou não de uma dada gramática dá-se o nome de *análise sintática*. Se existe mais de uma sequência de regras que chegue a esta mesma sentença, então a estrutura da sentença é dita ser *sintaticamente ambígua*.

Da escolha certa de uma gramática, com sua família de primitivas anexadas, depende todo o sucesso, eficiência e velocidade de um sistema de reconhecimento de padrões.

IV.2. Expressões PDL

Segundo Negoita [16], **representação** é definida como o conjunto de convenções para se descrever objetos, padrões, idéias, etc. Shaw [48] propôs uma linguagem artificial para o reconhecimento de padrões em experiências realizadas com câmara de bolhas, chamada de **Linguagem de Descrição de Padrões**, e que foi também utilizada para fazer o reconhecimento de caracteres alfanuméricos, conforme descrito em Fu [14].

Uma *primitiva de figura* é qualquer padrão n -dimensional ($n \geq 1$) com dois pontos distintos chamados *cabeça* e *cauda*. Uma concatenação de duas primitivas quaisquer pode-se dar apenas em sua cabeça e/ou sua cauda, de modo que estas operações possam ser representadas por um grafo.

PDL é uma linguagem de cadeias lineares representada por uma gramática linear livre de contexto, dada a seguir.

$$\begin{aligned}
V_n &= \{ S, SL \} \\
V_t &= \{ h1, h2, h3, d1, d2, d3 \} \cup \\
&\quad \{ v1, v2, v3, g1, g2, g3 \} \cup \\
&\quad \{ +, -, x, *, \neg, /, (,) \} \cup \\
&\quad \{ \lambda \} \\
S &= \{ S \} \\
P &= \{ S \rightarrow \text{prim}, \\
&\quad S \rightarrow (S \text{ op } S),^1 \\
&\quad S \rightarrow \neg S, \\
&\quad S \rightarrow SL, \\
&\quad S \rightarrow (/SL), \\
&\quad SL \rightarrow S, \\
&\quad SL \rightarrow (SL \text{ op } SL),^2 \\
&\quad SL \rightarrow (\neg SL), \\
&\quad SL \rightarrow (/SL) \}
\end{aligned}$$

onde λ é a primitiva de comprimento nulo; / é usada para permitir que cabeça e cauda de uma primitiva sejam definidas arbitrariamente; \neg é o símbolo de inversão de cabeça e cauda; os operadores *op* são +, -, x e *; e as primitivas *prim* são λ , h1, h2, h3, v1, v2, v3, g1, g2, g3, d1, d2, d3, todas elas segmentos orientados de retas em diversas direções representadas na figura 4.4.

Uma sentença *S* escrita em PDL fornece uma descrição estrutural T_s de uma figura por nomeação de todas as suas primitivas e suas conectividades na cabeça e na cauda.

Uma figura pode ser concatenada *apenas* em seus pontos cabeça e/ou cauda, assim a aplicação dos operadores de concatenação torna-se implementável, e uma primitiva pode ser representada como um *nodo dirigido por rótulo em um grafo*, apontando do nodo de sua cauda para o nodo de sua cabeça. Esta abstração será

¹ Esta regra de produção é nebulosa, pois nem sempre a concatenação é perfeita, ou seja, $0 \leq \text{concatenação} \leq 1$.

² *idem*

bastante útil, como se verá a seguir.

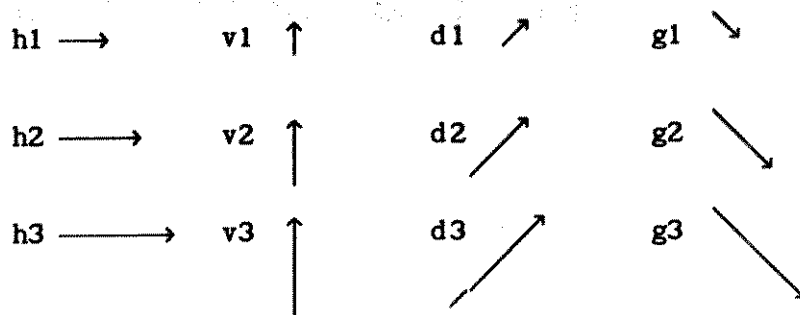


Figura 4.4 - Primitivas PDL

O sentido dos operadores de concatenação +, -, x e * é apresentado abaixo pela definição de $P(S1 \text{ op } S2)$, onde $S1$ e $S2$ pertencem à família PDL, $a1$ pertence a $P(S1)$ e $a2$ pertence a $P(S2)$. A expressão *cat* significa concatenado em :

$$P(S1 + S2) = \{ a1, a2 \mid \text{cabeça}(a1) \text{ cat cauda}(a2) \};$$

$$P(S1 - S2) = \{ a1, a2 \mid \text{cabeça}(a1) \text{ cat cabeça}(a2) \};$$

$$P(S1 \times S2) = \{ a1, a2 \mid \text{cauda}(a1) \text{ cat cauda}(a2) \};$$

$$P(S1 * S2) = \{ a1, a2 \mid \text{cabeça}(a1) \text{ cat cabeça}(a2) \text{ e} \\ \text{cauda}(a1) \text{ cat cauda}(a2) \}.$$

A representação destas operações está ilustrada na figura 4.5.

Em um reconhecimento bem sucedido, a função de reconhecimento não somente classifica corretamente o objeto presente na imagem, como também gera uma descrição estrutural das primitivas; de modo inverso, a descrição do padrão serve de entrada para a função gerativa.

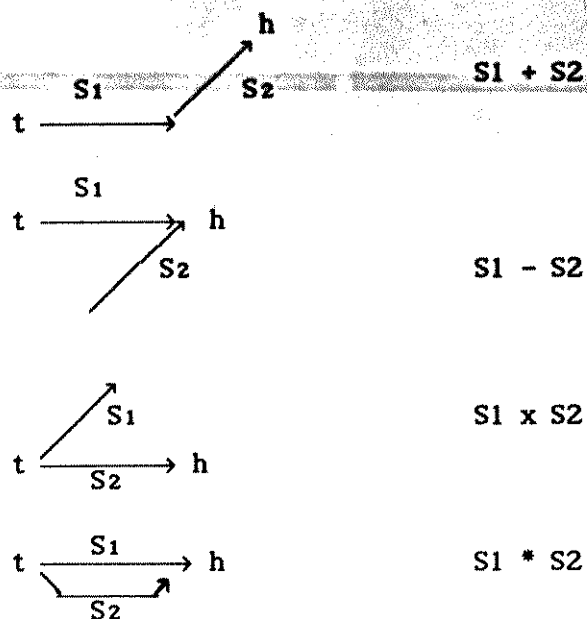


Figura 4.5 - Concatenações PDL

IV.2.1. Propriedades e Teoremas Básicos

ASSOCIATIVIDADE:

Cada um dos operadores de concatenação binários é associativo:

- (a) $(S1 + S2) + S3 = S1 + (S2 + S3)$
- (b) $(S1 - S2) - S3 = S1 - (S2 - S3)$
- (c) $(S1 x S2) x S3 = S1 x (S2 x S3)$
- (d) $(S1 * S2) * S3 = S1 * (S2 * S3)$

COMUTATIVIDADE:

O único operador comutativo é o * :

- (e) $(S1 * S2) = (S2 * S1)$

OPERADOR \neg :

Este operador atua como um complementador em uma álgebra Booleana:

$$(f) \neg (S1 + S2) = (\neg S2) + (\neg S1)$$

$$(g) \neg (S1 * S2) = (\neg S2) * (\neg S1)$$

O operador \neg apresenta as propriedades de DeMorgan com respeito a $\underline{=}$ e \underline{x} :

$$(h) \neg (S1 x S2) = (\neg S2) - (\neg S1)$$

$$(i) \neg (S1 - S2) = (\neg S2) x (\neg S1)$$

Este operador unário também tem a propriedade da involução:

$$(j) \neg (\neg S1) = S1$$

O operador $/$ admite as duas propriedades a seguir:

$$(k) / (/ S1) = (/ S1)$$

$$(l) / (S1 op S2) = (/ S1) op (/ S2)$$

A primitiva nula λ tem as propriedades abaixo relacionadas :

$$(m) S1 + \lambda = \lambda + S1 = S1$$

$$(n) S1 - \lambda = S1$$

$$(o) S1 x \lambda = S1$$

$$(p) \neg \lambda = \lambda$$

$$(q) \lambda op \lambda = \lambda$$

No caso em que se usa a representação da imagem em expressões PDL, deve-se ter alguns teoremas que assegurem esta representação como um método de inferência, ou seja, como metodologia para o reconhecimento de padrões. A prova destes Teoremas pode ser encontrada em [48].

Teorema 1

"Qualquer expressão PDL válida descreve uma única conectividade de primitivas".

Teorema 2

"Qualquer conjunto conectado de primitivas pode ser descrito por uma expressão PDL válida".

Teorema 3

"Dada uma expressão PDL válida S_1 descrevendo um conjunto de primitivas conectadas cujo grafo correspondente tem N nodos, é possível derivar $2^N - 1$ (e não mais) outras expressões PDL válidas S_1, S_2, \dots, S_{N^2} , tais que

- $$\left. \begin{array}{l} (1) S_i \equiv S_j \\ (2) S_i \neq S_j \end{array} \right\} \text{ ambos se } i, j = 1, 2, \dots, N^2; i \neq j$$
- (3) cada $S_i, i=1, 2, \dots, N^2$ é equivalente a uma expressão tendo

uma das seguintes formas:

- (i) $(/ S_{11}) + (SL_1 + (/ S_{12}))$
- (ii) $(/ S_{11}) + SL_1$
- (iii) $SL_1 + (/ S_{12}),$

onde SL_1 é obtida de S_1 dando os mesmos rótulos àquelas primitivas em S_1 que aparecem em S_{11} e/ou S_{12} ".

Teorema 4

"O conjunto de operadores $\{+, -, \times, *, \neg\}$ não é suficiente para a descrição de qualquer conjunto de conexões de primitivas".

Teorema 5

"Qualquer expressão PDL válida é equivalente a uma outra que utiliza apenas o conjunto de operadores $\{+, \neg, / \}$. Mais ainda, estes operadores são independentes entre si".

IV.2.2. Exemplos

Seja por exemplo a expressão PDL da letra A maiúscula, dada em Fu [14].

$$(d2 + (((d2 + g2) * h2) + g2))$$

A representação estrutural da expressão acima é dada pela figura 4.6.

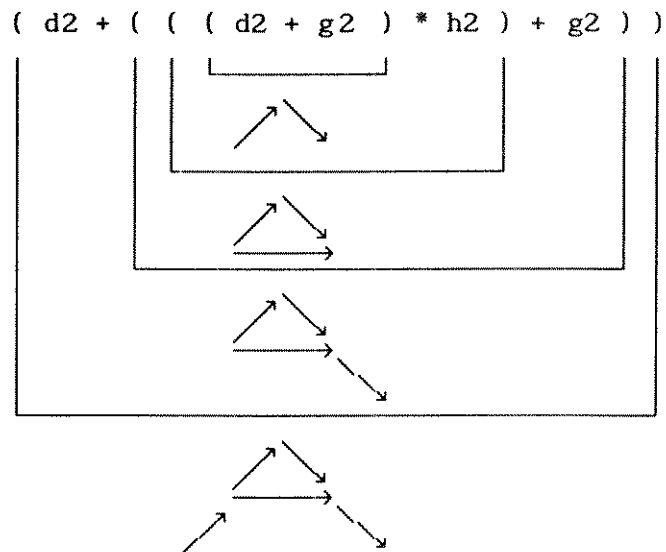


Figura 4.6 - Expressão PDL descrevendo o caracter A

A descrição estrutural da expressão PDL acima é dada pela árvore de

derivação na figura 4.7 a seguir. As figuras ilustram o fato de que a análise das expressões PDL geram uma descrição estrutural do padrão, além de classificá-lo.

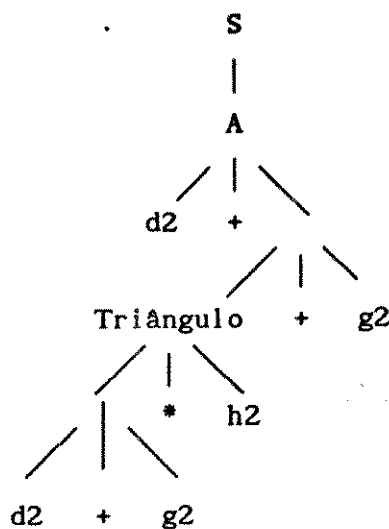


Figura 4.7 - Descrição estrutural do caracter A

O motivo da escolha das expressões PDL de Shaw e da lógica nebulosa não apenas como forma de reconhecimento, mas também como metodologia de representação do objeto presente na figura, deveu-se à alguns fatos. O primeiro deles é que a representação pictorial de um caracter alfanumérico bem definido em uma imagem é facilmente reconhecido através dessas expressões; o mesmo não ocorre se a figura estiver com *ruidos*, e por isso foram usadas as técnicas de lógica nebulosa. O segundo motivo é que pode-se analisar detalhes estruturais da figura representada, como por exemplo a parte superior da figura da letra B quando comparado com a parte superior da letra R, e chegar a alguma conclusão. Narasimhan [47] cita inclusive alguns exemplos de comparações que se pode fazer apenas através de um reconhecimento estrutural, tais como: 1) até que ponto uma letra A se parece com uma letra B ?; 2) é esta realmente uma letra A ?; 3) qual a diferença entre esta letra A e aquela letra A ?; e assim sucessivamente. E o último motivo dessa escolha foi o de facilitar o reconhecimento de letras deformadas com um determinado grau de certeza.

O problema da representação é uma das mais importantes questões para a operação de reconhecimento, porque o modelo selecionado deve ter as características necessárias para se efetuar a classificação. A decomposição do objeto presente na imagem em um conjunto de primitivas e seus interrelacionamentos, e a geração de uma árvore como resultado de sua análise sintática, são as características principais de um sistema estrutural de reconhecimento de padrões. Em outras palavras, o ato de reconhecer é feito através de uma análise da árvore gerada, e a decisão é feita sobre a melhor análise. Os sistemas de reconhecimento são compostos normalmente por 3 fases: (a) pré-processamento; (b) detecção; e (c) análise e decisão. Um esquema simplificado encontra-se na figura 4.8, que será analisada nos próximos itens deste capítulo.

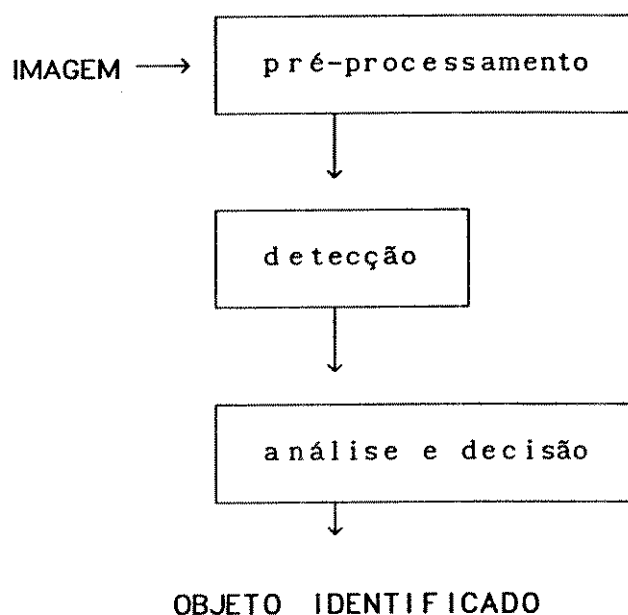


Figura 4.8 - Um sistema de reconhecimento

IV.3. Fase de Pré-processamento

A captação de imagens consiste na extração de uma cena do mundo real com o propósito de reconhecer e/ou localizar algum objeto de interesse, através de uma câmera. Da digitalização dos pixels forma-se a matriz pictórica representativa da cena, e o armazenamento desta matriz faz-se usualmente no chamado *frame grabber*.

Quando se faz a captação de imagens para o posterior processamento por computadores, estas normalmente chegam com defeitos, ruídos e outros problemas, que devem ser minimizados para melhor e mais rápido processamento pela máquina. A esta fase inicial de preparação da imagem para se efetuar operações de reconhecimento é dado o nome de fase de pré-processamento.

Dentre as várias técnicas usadas na fase de pré-processamento estão as operações de *limiar*, detecção de contorno, crescimento de regiões e esqueletização, dentre outras.

Uma das técnicas mais utilizadas nesta fase é a aplicação do operador de limiar que matematicamente pode ser descrita como o seguinte :

$$f_t(x,y) = \begin{cases} b_0, & \text{se } f(x,y) < t \\ b_1, & \text{se } f(x,y) \geq t \end{cases}$$

Figura 4.9 - Algoritmo de Limiar

Algoritmos para analisar imagens binárias são normalmente mais rápidos e mais simples que algoritmos para processamento de imagens com vários níveis de cinza, pois precisam de menos esforço computacional e de menos armazenamento de dados [2].

Os métodos de limiar são aqueles que determinam um valor particular t^* de

t para a função acima baseados em algum critério, que pode ser global ou local. As técnicas de limiar globais são aquelas que aplicam à toda a imagem um valor único para o limiar, enquanto que técnicas de limiar locais são aquelas que particionam a imagem dada em sub-imagens e determinam para cada uma delas um valor de limiar.

Na fase de pré-processamento é fundamental o conceito de *afinamento*, também conhecido por *enfraquecimento* ou *esqueletização* [18]. Afinamento é o nome mais comum para o processo de reduzir a largura de um objeto tipo linha com várias camadas de pixels para apenas uma linha fina e única. Outros nomes dados a estes algoritmos são *incêndio de pastos* e *linha de fogo*, pois "o lugar geométrico dos pontos de extinção do fogo forma o esqueleto da figura". Vide figura 4.10.

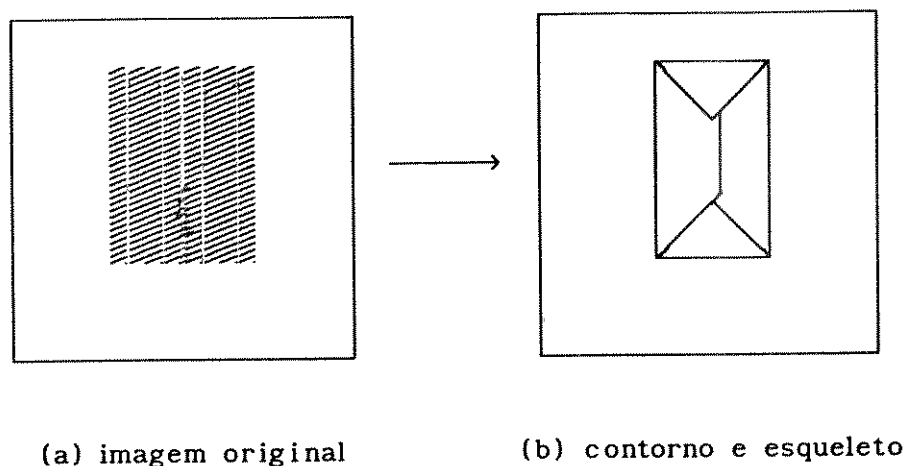


Figura 4.10 - Aplicação de esqueletização

Como os algoritmos propostos são antigos e como a tecnologia das câmeras e monitores melhorou de forma significativa, a resolução e a qualidade das imagens aumentou muito, de tal modo que, se a alguns anos uma linha do mundo real era representada por uma camada de 4 pixels, hoje a mesma imagem traria uma camada de, por exemplo, 10 pixels, o que tornaria o processamento de imagens ainda mais lento caso não houvesse estes algoritmos de pré-processamento. Por isto é que esta fase vem ganhando importância em sistemas de visão mais novos, diminuindo o volume de pixels necessários à

extração das informações de uma certa imagem para a tomada de alguma decisão.

A operação de esqueletização consiste no apagamento dos pontos da fronteira da figura de tal modo que: 1) não remova pontos extremos (por exemplo, os pixels finais de uma reta); 2) não quebre a conectividade dos pontos pertencentes ao padrão; e 3) não cause excessiva erosão à figura original [29].

O que se quer dizer com excessiva erosão? O algoritmo de esqueletização deve ser tal que informações suficientes devam ser mantidas na imagem afinada para se extrair todas as informações estruturais necessárias ao reconhecimento. Em outras palavras, são tirados apenas aqueles pixels que não eram imprescindíveis ao reconhecimento da figura, chamados pixels redundantes, deixando-a o mais próximo possível da estrutura original. Mais ainda, o algoritmo deve ser tal que permaneçam no esqueleto do objeto informações suficientes para reconstruí-lo o mais próximo possível do original. A esta propriedade dos algoritmos de esqueletização dá-se o nome de **reconstrutibilidade**.

Um exemplo de esqueletização das imagens com caracteres alfanuméricos é o da figura 4.11.

Detecção de bordas é uma técnica de segmentação de imagens baseada na detecção da descontinuidade. A presença de uma certa borda na imagem pode ser descrita quantitativamente por uma mudança abrupta no nível de cinza dos pixels. Estes pixels, quando da execução dos algoritmos de detecção de bordas, não são apagados da imagem, e ao término destes algoritmos, tornam-se as bordas do objeto.

As técnicas de detecção de bordas conseguem extrair da imagem apenas o contorno dos objetos presentes nesta imagem, passando então este contorno para a fase de detecção dos sistemas de reconhecimento. É interessante notar que após a detecção das bordas em uma certa imagem, normalmente ainda submete-se a imagem ao tratamento da *esqueletização*, pelo fato de que muitas vezes as bordas ainda possuem mais de um pixel de espessura, o que deve ser reduzido.

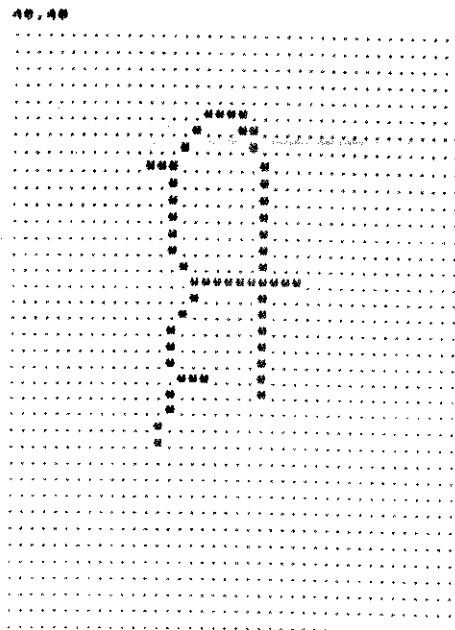
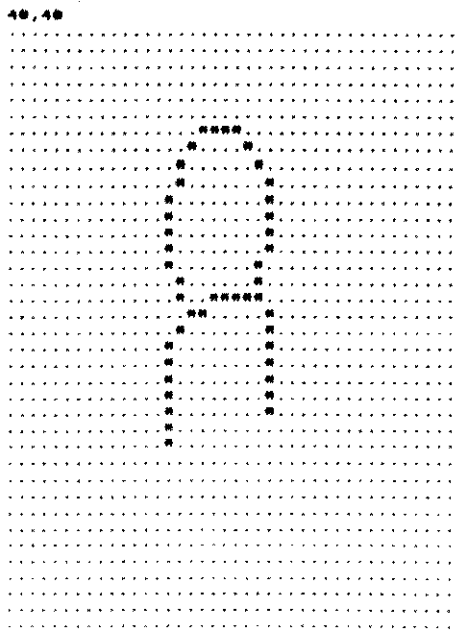
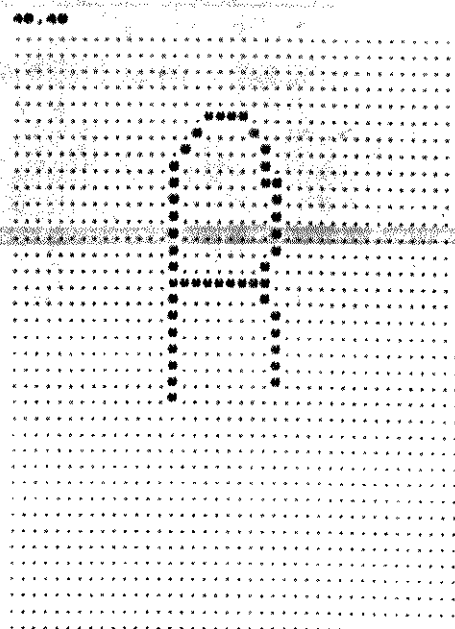
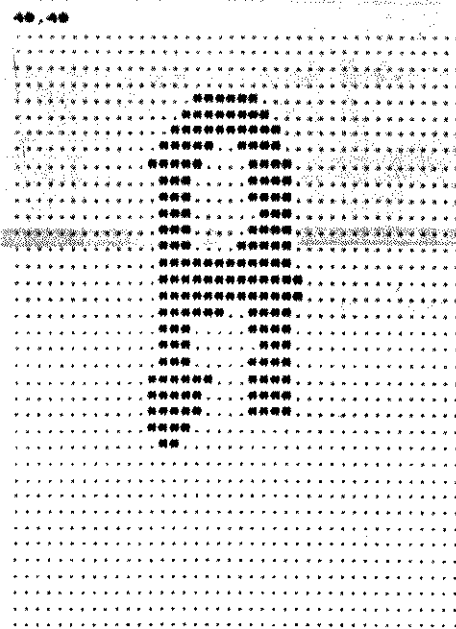


Figura 4.11 - Uso dos algoritmos de esqueletização

As técnicas de detecção de bordas, que também podem ser usadas como técnicas de segmentação, são divididas basicamente em dois grupos: (1) técnicas para imagens binárias; e (2) técnicas para imagens multi-tonais.

Para o caso de imagens multi-tonais, é difícil especificar *a priori* quais bordas delimitam objetos relevantes na imagem. Dependendo da meta desejada, diferentes variações locais de intensidade podem provocar o apagamento de bordas desejáveis. Além disto, bons detetores devem se precaver das falhas de detecção ocasionadas por confusão com ruídos.

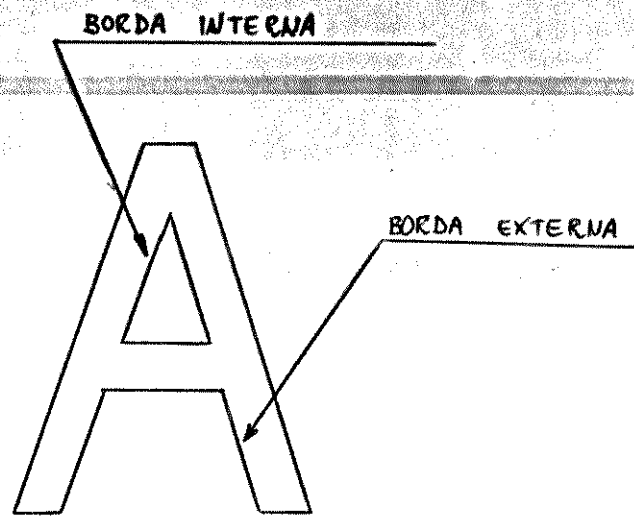


Figura 4.13 - Bordas de um objeto

IV.4. Fase de Detecção

Faz-se necessária a localização na imagem de certos atributos discriminatórios, no caso da metodologia decisão-teórica, e de primitivas linguísticas, no caso da metodologia estrutural, que são também chamadas de *primitivas de reconhecimento*, e a fase do sistema que é responsável por esta localização chama-se *fase de detecção de primitivas*.

Os detectores de primitivas são algoritmos que visam extrair informações da imagem, sem obrigatoriamente alterar a cena captada. Tais informações servirão de entrada para outros algoritmos, tais como reconhecimento de padrões, computação cognitiva de cenas, descrição computacional de primitivas nas cenas, extração de características de objetos componentes de cenas, dentre muitos outros. Para o método decisão-teórico, a presença da matriz representativa da imagem é importante, mas para o reconhecimento estrutural a representação do objeto é muito mais importante que a própria imagem.

Nesta fase de detecção em geral faz-se uso de técnicas tais como geração do código da cadeia, aproximação poligonal, dentre outras.

Uma imagem binária pode ser representada como um conjunto de pontos claros e escuros. O método de representação que consiste na determinação da sequência de pixels que formam o objeto é chamado de *código da cadeia* e pode ser implementado por hardware dedicado ou por software. Para aplicações mais simples, normalmente usa-se a segunda opção. O código da cadeia, proposto por Freeman [67], é uma estrutura de dados representativa dos limites do objeto por meio de oito diferentes vetores, como mostra a figura 4.14.

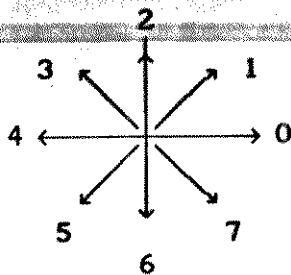


Figura 4.14 - Vetores direção do código da cadeia de Freeman

Segue abaixo um exemplo da geração deste código.

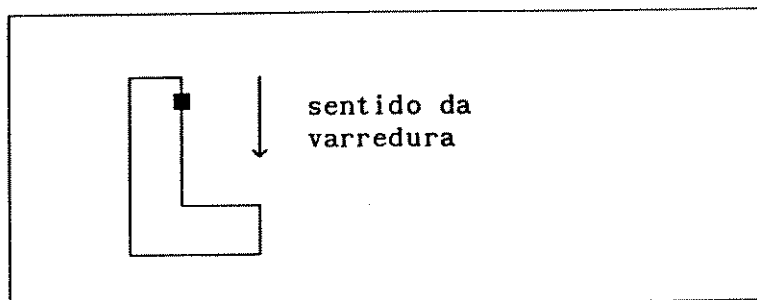


Figura 4.15 - Geração do código da cadeia

A representação da imagem por esta forma faz o rastreamento do objeto na imagem, e a cada grupo de pixels o sistema gera um código que representa a direção que a varredura tomou em relação ao pixel anterior. Costuma-se analisar este código por meio das chamadas gramáticas de cadeia, porque à medida que o objeto é rastreado, uma cadeia de símbolos é gerada, fazendo a representação. A análise desta cadeia pode ser feita pelas gramáticas G e FG discutidas anteriormente.

O uso da *detecção de bordas*, ao contrário do uso da *esqueletização*, contribue para que o padrão fique com suas primitivas fechadas, ou seja, saindo de um ponto qualquer das primitivas, consegue-se voltar a ele sem

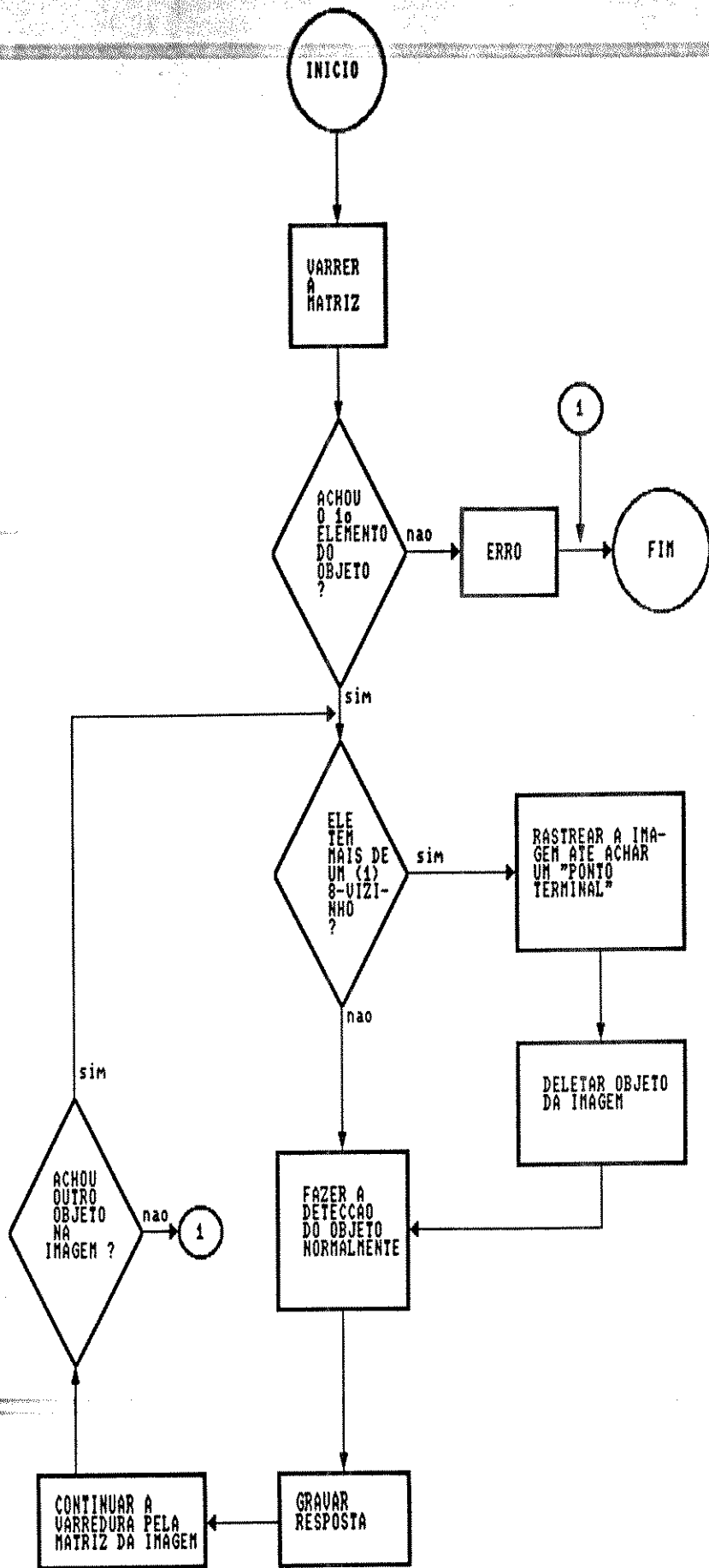
abandonar os pixels que compõem o objeto. O processo de esqueletização normalmente deixa o mesmo padrão com as primitivas abertas, mas em compensação, o uso das expressões PDL de Shaw torna-se simples quando se usa o processo da esqueletização. O uso das gramáticas G e FG se faz com a detecção de bordas.

Para a utilização das expressões PDL, é necessária a detecção de todas as primitivas (de um total de 12 tipos distintos) que existem na imagem. Esta extração, no entanto, torna-se complexa quando se faz a análise a nível pictórico. As inúmeras sequências de pixels que podem existir em uma imagem tornam o problema ambíguo e impreciso, e alguns algoritmos utilizados na suavização deste problema encontram-se no item *aproximação poligonal* já discutido anteriormente. O algoritmo que implementa a extração das primitivas PDL da imagem se encontra no fluxograma a seguir.

Entre a fase de pré-processamento e a fase de detecção de primitivas, existe uma fase intermediária denominada *segmentação da imagem*. Sua finalidade principal é a de dividir a imagem, não apenas fisicamente, em termos de regiões de pixels, mas também logicamente, em termos de se obter diversos objetos de interesse em um mesmo segmento. Esta segmentação em regiões têm a finalidade de se estudar alguma propriedade local importante, como por exemplo no caso de análise de cenas.

Segundo Fu [14], estas técnicas de segmentação da imagem podem ser classificadas em três sub-categorias: 1) limiar de atributos característicos; 2) detecção de bordas; e 3) extração de regiões.

Conforme Ruiz [61], a "*geração do código da cadeia* é por si só um método de segmentação da imagem, com uma representação compacta e conveniente para cada figura de forma isolada, a partir das quais são feitas medidas ou extrações de atributos". A geração do *código da cadeia* realiza, de maneira fraca mas eficiente, uma segmentação da imagem, pois cada cadeia gerada corresponde, de modo geral, a um objeto na imagem. Diz-se neste caso que é fraco porque o algoritmo não tem capacidade de distinguir se a imagem possui dois objetos, ou se apenas um objeto que se encontra "quebrado".



IV.5. Fase de Análise e Decisão

Diversas técnicas são usadas na chamada de fase de análise e decisão, quando se faz o uso das primitivas extraídas na fase anterior para se promover o reconhecimento. Costuma-se dizer que o cérebro de um sistema de reconhecimento encontra-se nesta fase, e que os 5 sentidos humanos (audição, olfato, visão, tato e gustação) realizam, conjuntamente, as duas fases anteriores, numa analogia à maneira humana de se reconhecer.

Assim que todas as primitivas dentro da imagem são identificadas, o processo de reconhecimento é executado por meio de uma análise sintática da sentença descrevendo o padrão dado, para determinar se a expressão está sintaticamente correta ou não com respeito à gramática especificada, no caso, pelas expressões PDL. A análise sintática também produz uma descrição estrutural da sentença que está representando o padrão dado, que é a que interessa, por exemplo, no caso de CAD já anteriormente mencionado.

Segundo van der Helm et alii [54], a aproximação pela informação estrutural trata a organização perceptual dos padrões refletida pela descrição deste padrão. Entretanto, um padrão pode ser descrito por vários códigos distintos. De modo a ser mais específico sobre a descrição de padrões, necessita-se de dois níveis de códigos: o nível sintático e o nível semântico. Uma sequência de símbolos é chamada *código primitivo*, cujos símbolos são chamados *elementos primitivos*. O nível sintático trata apenas da aplicação direta da gramática, ou seja, da representação do objeto, enquanto o nível semântico trata da análise e interpretação destes símbolos. O nível sintático é tratado nas fases de pré-processamento e detecção, enquanto que o nível semântico é tratado na presente fase de um sistema de reconhecimento de padrões.

Sistemas que trabalham do estado inicial para o estado final ou objetivo são chamados de sistemas com encadeamento à frente, também chamado *top-down* ou *goal-directed*, e sistemas que trabalham do estado final em direção ao estado inicial são chamados de sistemas com encadeamento para trás, também chamado

bottom-up ou *data-driven*. Alguns autores como Toussaint [75] admitem também a estratégia híbrida,

Para que se possa usar funções de distância em métodos linguísticos para resultar em valores de saída dentro do contínuo [0,1], simples de serem manipuladas pelos operadores da lógica nebulosa, tem-se então que implantar os seguintes procedimentos:

1. cada regra de produção deve ter um valor, que significa o quanto é importante a regra para o reconhecimento de um certo padrão ou primitiva;

2. faz-se a aplicação de todas as regras para descobrir-se todos os padrões possíveis de serem reconhecidos; e

3. a regra que fornecer o maior valor (maior que um certo limiar) será então considerado como a que representa o carácter reconhecido.

Cabe apenas ao usuário a decisão final na escolha de uma *função de distância* que sirva melhor ao ambiente do sistema que se está implementando, dado que não existe um algoritmo que diga qual é a melhor função. Nesta escolha terão efeitos a experiência do programador em sistemas já implementados, ou os testes com um certo número de funções, donde se extrai a melhor delas.

Dado que existe uma função que consegue medir a distância entre a figura presente na imagem e as classes dos padrões, pode-se calcular a melhor delas da seguinte maneira. Na fase de pré-processamento, rotaciona-se a imagem de 90° em 90° , e são então geradas quatro representações distintas para cada imagem.

Contando a parte de 4 rotações de imagens com as 26 letras maiúsculas, então o resultado final pode ser representado por uma relação nebulosa com a

forma da figura 4.16 a seguir.

	0°	90°	180°	270°	
A	μ_{a1}	μ_{a2}	μ_{a3}	μ_{a4}	μ_a
B	μ_{b1}	μ_{b2}	μ_{b3}	μ_{b4}	μ_b
C	μ_{c1}	μ_{c2}	μ_{c3}	μ_{c4}	μ_c
.....					..
.....					..
Y	μ_{y1}	μ_{y2}	μ_{y3}	μ_{y4}	μ_y
Z	μ_{z1}	μ_{z2}	μ_{z3}	μ_{z4}	μ_z
.....				
	μ_1	μ_2	μ_3	μ_4	μ

Figura 4.16 - Projeção global de uma relação

Veja que μ pode ser interpretado como sendo o resultado final de um sistema de reconhecimento de padrões.

Cada representação dos quatro estados de rotação possíveis é comparada à expressão PDL da classe, e é retirado então o melhor destes resultados, através da seguinte equação nebulosa, representando a primeira projeção da relação nebulosa citada :

$$f_A(\lambda) = f_0(\lambda) \vee f_{90}(\lambda) \vee f_{180}(\lambda) \vee f_{270}(\lambda) \quad (1)$$

onde o operador \vee é o operador MAX. Ao final existirão 62 valores $f(\lambda)$ representando o casamento da imagem com cada uma das classes de padrões. A partir daí, a decisão de qual classe pertence o objeto de interesse é dado por outra equação nebulosa, representando a segunda projeção da relação nebulosa anterior :

$$f(\lambda) = f_A(\lambda) \vee f_B(\lambda) \vee \dots \vee f_Y(\lambda) \vee f_Z(\lambda) \quad (2)$$

Existe apenas mais um fator restritivo, que é o chamado limiar final mínimo, L_{\min} , ou coeficiente de confiança. Faz-se a verificação de $f(\lambda)$ final, para se evitar que o maior valor, conforme a projeção global nebulosa anterior, fique abaixo do valor L_{\min} considerado como *nível de certeza mínimo* que se deve ter ao se fazer qualquer decisão, filtrando assim qualquer resposta abaixo deste valor. Deste modo eliminam-se os casos em que não existem evidências suficientes para garantir o reconhecimento. Matematicamente, o fato acima exposto pode ser escrito da seguinte forma :

$$f(\lambda) \geq L_{\min}$$

Pode-se escolher qualquer valor numérico para este limiar, apesar da existência de um compromisso entre um valor grande, que garante uma elevada certeza na decisão contra um alto índice de rejeição, e um valor pequeno, que praticamente não rejeitaria imagem alguma, mas comprometeria a execução de qualquer outro sistema que dependesse da resposta do reconhecedor. Evidentemente que este valor deve ser estudado para todo sistema implementado no paradigma sintático, pois não se pode dizer *a priori* qual seria seu valor ótimo.

Resumidamente, o que se apresentou no presente capítulo pode ser visualizado pela figura 4.17 a seguir.

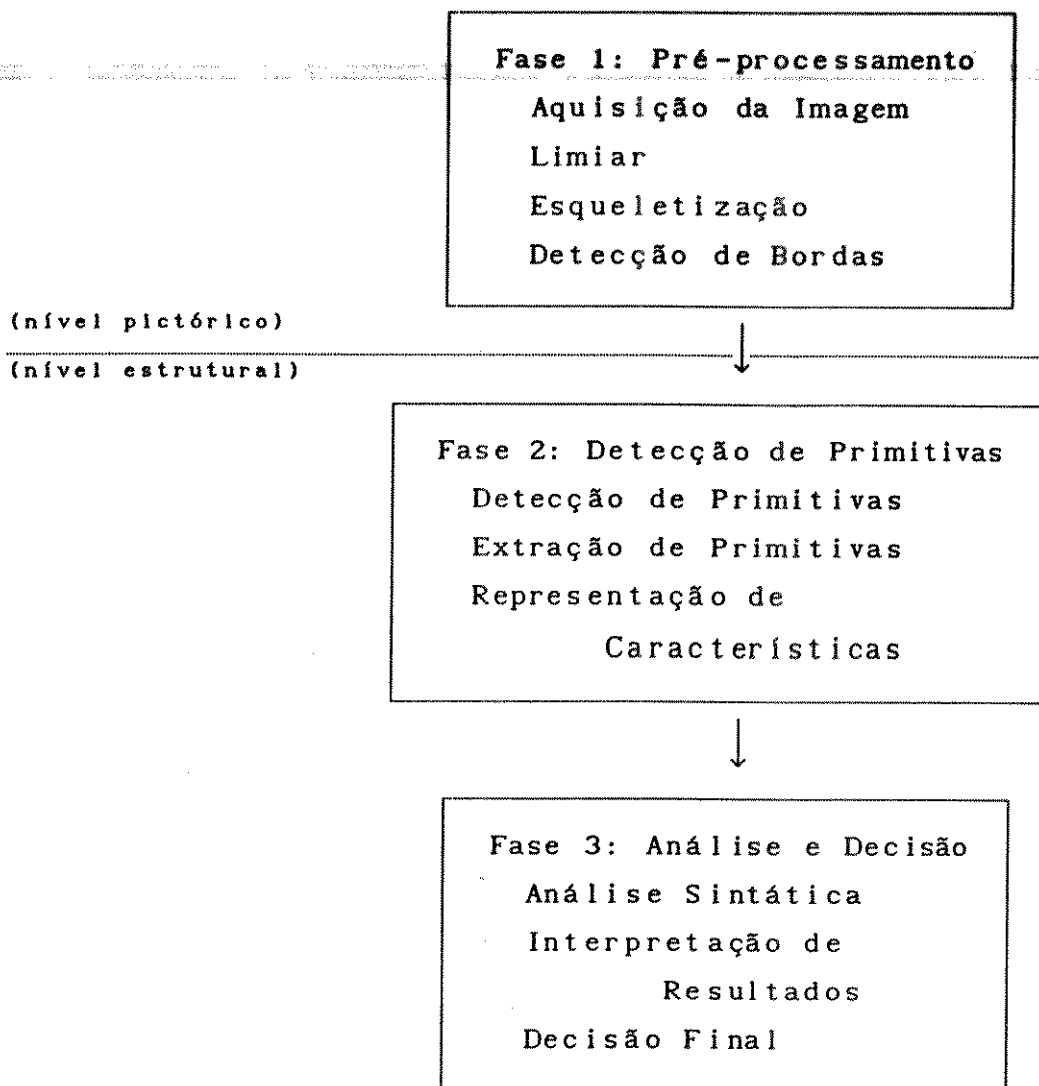


Figura 4.17 - Resumo final

CAPÍTULO V

IMPLEMENTAÇÃO E RESULTADOS

V.1. Características do Sistema

V.2. Experiências com o Sistema

V.3. Considerações Gerais

V.1. Características do Sistema

O sistema apresentado neste trabalho tem como objetivo reconhecer caracteres alfanuméricos, podendo também ser utilizado para o reconhecimento de formas bi-dimensionais. Este consiste na implementação dos algoritmos que foram definidos nos capítulos anteriores.

Devido à diferença do nível de abstração do conhecimento entre a imagem (conhecimento e/ou informações implícitas nos pixels) e sua representação (conhecimento e/ou informação explícitas nas primitivas), fica claro pelo exposto na descrição do reconhecimento estrutural que o volume de informações que uma imagem traz é sensivelmente diminuído com as técnicas de esqueletização e detecção de primitivas, pois ocorre uma *especialização das informações* contidas na imagem. Não só houve um aumento na *qualidade* das informações, como também uma diminuição na *quantidade* de informações da imagem.

Algumas restrições foram feitas no início deste trabalho na tentativa de amenizar os problemas envolvidos na complexidade do reconhecimento de padrões. Para que haja um perfeito entendimento do texto, tais restrições são mencionadas abaixo:

1. a imagem só pode ter 2 *níveis de cinza* (em aplicações industriais esta restrição é quase sempre usada);
2. cada imagem só pode *conter um objeto*;
3. a imagem deve ser *estática*.

No presente trabalho foi estudado não apenas o uso de metodologias estruturais de reconhecimento, mas também a influência exercida pelos diversos algoritmos no problema do reconhecimento, tais como esqueletização, detecção de bordas, rotação, dentre outros. O motivo desta análise é o fato de que

tendo conhecimento desta influência, consegue-se minimizá-la a fim de melhorar a eficiência do sistema, tornando-o tolerante a determinados problemas.

Segundo Greanias [52], um sistema de reconhecimento deve empregar um método direto e simples que, a partir da geração de determinadas grandezas (no caso de reconhecimento decisão-teórico) ou primitivas (no caso de reconhecedores estruturais), consiga diferenciar imagens, tendo assim portanto a capacidade de classificar. Para isto, o referido autor utiliza cinco critérios de tolerância relacionados a seguir:

1. *registro de caracteres* - se for necessário, os caracteres devem ser reconhecidos em uma sub-área da imagem;

2. *tamanho de caracter* - o sistema deve reconhecer caracteres que tenham um tamanho dentro de uma relação máxima de 4 para 1;

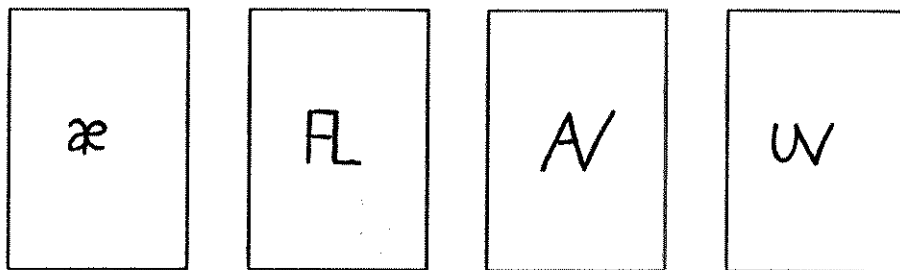
3. *formas* - a tolerância das formas de um caracter alfanumérico é definida em termos das expressões PDL que se quer utilizar na representação estrutural do dito caracter;

4. *qualidade da linha* - é claro que imagens "grosseiras" são mais difíceis de serem reconhecidas do que imagens "esbeltas", e a linha deve então ter uma tolerância quanto à qualidade;

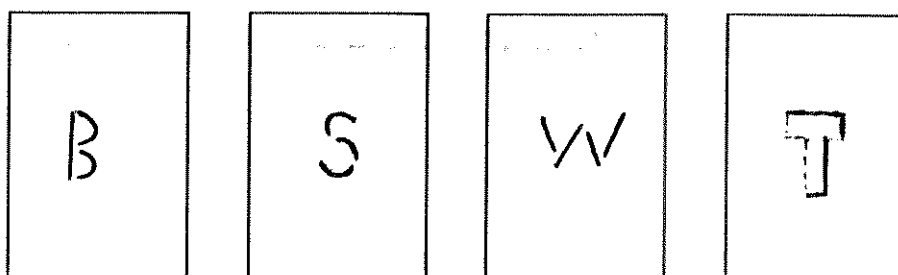
5. *rotação do caracter* - o caracter deve ter uma rotação máxima tolerável de $\pm 20^\circ$.

Existem alguns problemas que são próprios da área de reconhecimento de caracteres alfanuméricos, que são os causadores de erros e/ou rejeições frequentes:

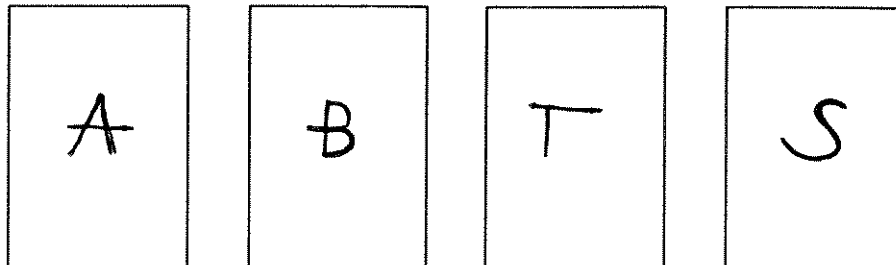
1. *caracteres ligados ou que se tocam;*



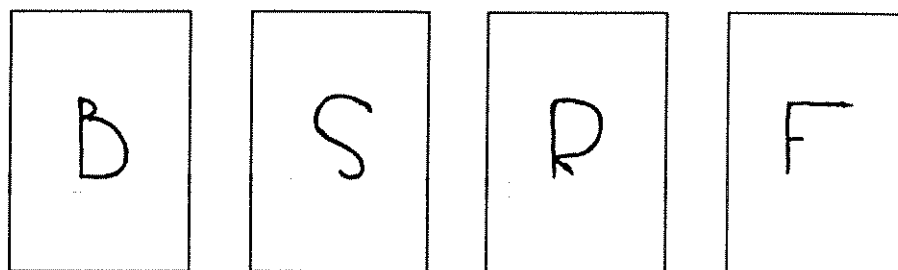
2. *caracteres quebrados;*



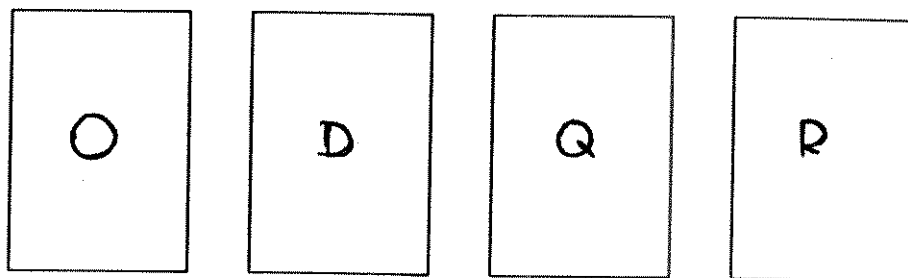
3. *caracteres com segmentos estranhos;*



4. *caracteres com proporções exageradas;*



5. caracteres similares a outros pela forma.



A imagem consiste de uma matriz de 40x40 pixels, simulada em ambiente controlado de ruído e com apenas 2 níveis de cinza, em microcomputadores PC. Um exemplo de imagem usada pelo sistema é dada na figura 5.1 abaixo.

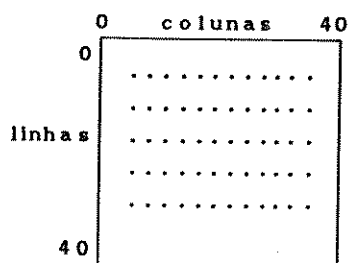


Figura 5.1 - Representação de uma imagem

Numa primeira aproximação para o reconhecimento de caracteres alfanuméricos, foi utilizada a esqueletização da imagem com a posterior extração das primitivas PDL necessárias ao reconhecimento. O processo de esqueletização, eliminando pixels redundantes, faz com que o número de primitivas extraídas torne-se mínimo, facilitando os algoritmos de reconhecimento e diminuindo o tempo de resposta do sistema.

O próximo passo para o reconhecimento de caracteres alfanuméricos foi dado com a utilização de um algoritmo grosseiro de rotação das imagens, seja no sentido horário de 90 em 90°, seja no sentido anti-horário também de 90 em 90°. Uma imagem sendo rotacionada e analisada tornaria o sistema, com uma

certa tolerância, independente da rotação dos caracteres alfanuméricos dentro da imagem, enquanto que a lógica nebulosa seria responsável pelos $\pm 20^\circ$ de tolerância mencionados por Greanias anteriormente. Assim, uma figura que ilustre os graus de rotação cobertos pelo sistema seria o representado pela figura 5.2.

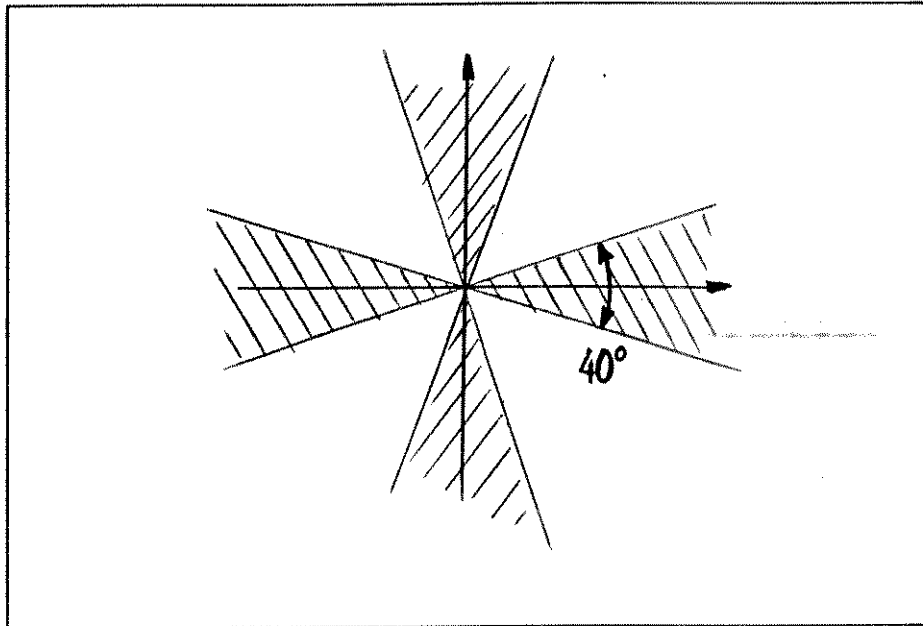


Figura 5.2 - Graus de rotação do sistema

Outra tentativa para o reconhecimento sintático de padrões foi com o uso de uma máscara nebulosa tipo casamento de gabarito, dada pela figura 5.3. Maiores detalhes da máscara podem ser obtidos no apêndice I.

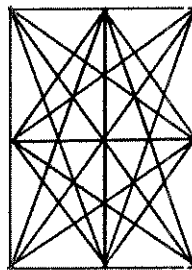


Figura 5.3 - Máscara do casamento de gabarito

A máscara consiste de 38 primitivas previamente selecionadas que são usadas para efetuar o reconhecimento dos caracteres alfanuméricos, através de um mapeamento entre o caracter que se quer reconhecer e a máscara. A figura 5.4 a seguir ilustra por exemplo o mapeamento da letra S.

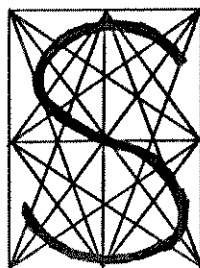


Figura 5.4 - Uso da máscara sobre a letra S

O motivo da escolha desta máscara deve-se ao fato de que as primitivas que se encontram na máscara são *extremamente* úteis no reconhecimento de determinadas letras que possuem apenas retas, tais como as letras A, L, I, etc, e *relativamente* úteis para o caso de caracteres alfanuméricos *curvos*, como S, B, O, etc. É evidente que o uso de outras máscaras seria possível também para o reconhecimento, como na figura 5.5. Entretanto as primitivas existentes na máscara escolhida possuem relação com as expressões PDL dos caracteres alfanuméricos.

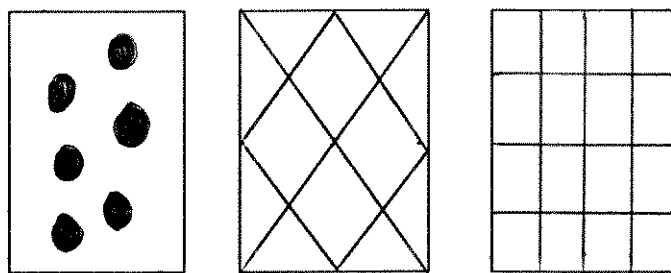


Figura 5.5 - Possibilidade do uso de outras máscaras

Posteriormente, a máscara foi alterada de 38 para 50 primitivas, para se

ter um mapeamento melhor entre as primitivas da máscara e as primitivas das expressões PDL.

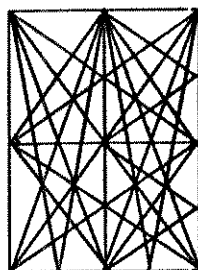


Figura 5.6 - Máscara com 50 primitivas

Com o início da utilização destas máscaras no reconhecimento dos caracteres alfanuméricos, fez-se necessária a implementação do paradigma da aprendizagem, já que é extremamente difícil prever o mapeamento das diversas letras com relação à máscara nebulosa. Existe um módulo de aprendizado supervisionado dentro do sistema, que tem por função obter este mapeamento para as diversas letras do alfabeto.

O processo estrutural de reconhecimento nada mais é do que uma análise sintática, e o algoritmo por ele responsável é normalmente um processo *não-determinístico*. Este é um dos motivos que levou à escolha do PROLOG como sendo a linguagem de programação responsável pela fase de análise do sistema, basicamente por causa dos mecanismos de *depth-first* e *backtracking* nela existentes, essenciais na instanciação da computação não determinística.

Dada uma imagem para o sistema de reconhecimento, este detecta todas as primitivas, que são então analisadas sintaticamente de acordo com as expressões PDL, formando assim uma *árvore de reconhecimento*, que é então comparada com a árvore de cada uma das classes de padrões existentes, conforme a figura 5.7.

A fase de análise do sistema calcula o grau de semelhança entre as duas árvores, atribuindo custos a qualquer *ramo* ou *folha* incompleta, faltante ou em excesso que possa aparecer no processo de reconhecimento.

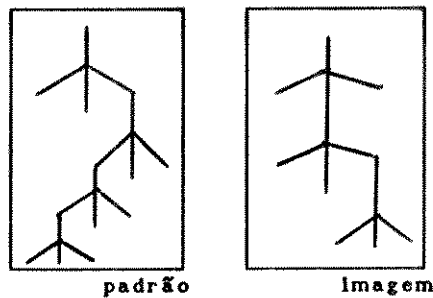


Figura 5.7 - Árvores da letra A

Pode-se dizer que a melhor classe de padrões, encontrada nestas comparações pelas funções de pertinência do sistema, é dada como sendo a classe de padrões na qual a imagem analisada é classificada. A única restrição que é feita neste caso é que a melhor destas comparações deve resultar em um valor numérico maior que o coeficiente de certeza, para que se possa ter uma *certeza mínima* sobre a decisão tomada.

Normalmente discute-se qual a melhor função de pertinência para ser usada por um sistema de reconhecimento. A função escolhida deve ter : (1) alta capacidade discriminatória; (2) capacidade de não separar caracteres semelhantes ; e (3) apresentar o maior valor sempre acima do *coeficiente de certeza*.

Todas as funções utilizadas pelo sistema foram implementadas em PROLOG, pois o cálculo é não-determinístico e se baseia nos fatos simbólicos listados a seguir :

1. número de primitivas existentes na imagem;
2. número de primitivas ausentes da imagem;
3. número de primitivas utilizadas pelas expressões PDL;
4. distâncias envolvidas nas concatenações necessárias ao reconhecimento;
5. tamanho das primitivas.

As funções estão listadas no apêndice J. Elas devem sempre fornecer um resultado que esteja dentro do intervalo [0,1] da lógica nebulosa, significando a distância entre o caracter lido na imagem e as 26 classes de padrões existentes. Estas funções têm sempre um formato do tipo :

$$F_{\lambda}(x) = 1 - \text{SOMATÓRIO DE ERROS}(x)$$

onde se percebe que se o somatório de erros for quase nulo, a função gera então valores perto da unidade (1), enquanto que se houver muitos erros, a função aproxima-se do zero (0).

A maneira pela qual a matriz 40x40 é varrida pelo sistema gera primitivas com posicionamento relativo distinto, problema que foi constatado, fazendo com que a implementação contivesse diversas maneiras diferentes das imagens serem varridas em busca das primitivas de reconhecimento. A figura 5.8 ilustra algumas destas maneiras distintas.

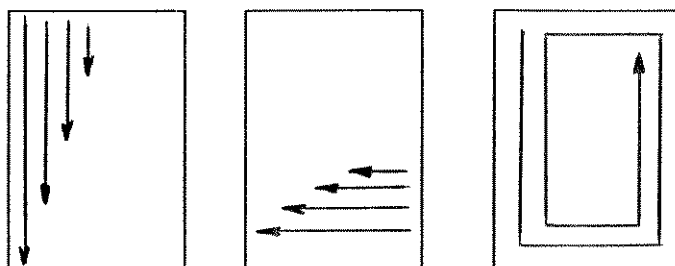


Figura 5.8 - Diferentes rastreamentos na imagem

Um dos motivos que levou à escolha das expressões PDL de Shaw como forma de representação dos padrões foi a sua independência da questão do tamanho das primitivas encontradas na imagem, ou seja, a questão do *escalamento*. Assim, dada uma imagem de uma letra O "pequena" e uma letra O "grande", ambas serão reconhecidas corretamente.

A questão da *translação* também beneficia as expressões PDL, ou seja,

tanto a imagem de uma letra H no canto superior esquerdo quanto no canto inferior direito, por exemplo, serão corretamente classificadas.

O mesmo já não se pode dizer a respeito da *rotação*, para o caso das expressões PDL. Uma imagem de uma letra V invertida de 180° tem uma grande chance de ser reconhecida como letra A, porque as primitivas g1 e d1 da letra V tornam-se primitivas d1 e g1, respectivamente, na letra A invertida. Para se tornar independente do problema da rotação, deve existir um código especial dentro do sistema.

Um resumo para o caso da expressões PDL está ilustrado na tabela 5.9.

	Rotação	Translação	Escalamento
Independência	NÃO	SIM	SIM

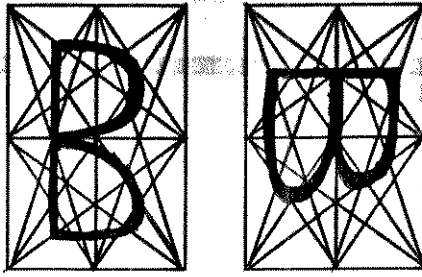
Tabela 5.9 - Atributos das expressões PDL

No caso das máscaras para o casamento de gabarito nebuloso usadas nas experiências, elas são extremamente dependentes quanto aos fatores acima estudados, refletindo-se na tabela 5.10.

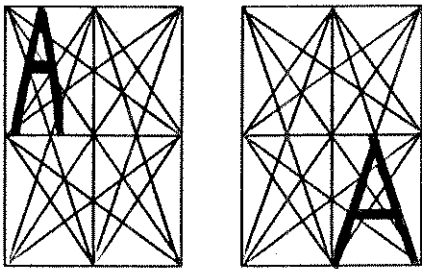
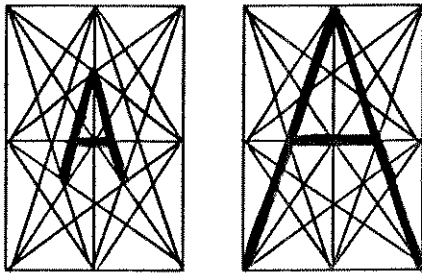
	Rotação	Translação	Escalamento
Independência	NÃO	NÃO	NÃO

Tabela 5.10 - Atributos das máscaras nebulosas

No caso da rotação, por exemplo, se fôr ensinado ao sistema o objeto da esquerda da imagem abaixo, então dificilmente ele vai reconhecer corretamente a imagem da letra B.



O mesmo pode-se dizer a respeito de escalamento e translação, respectivamente representados pelos pares de imagens a seguir.



V.2. Experiências com o Sistema

Foi mostrado o funcionamento do sistema proposto nos itens precedentes, passando-se agora aos resultados obtidos nas experiências, resultantes do uso com imagens de caracteres alfanuméricos, algumas das quais estão no apêndice F deste trabalho.

As experiências são relatadas em ordem cronológica, de modo a ressaltar os problemas ocorridos e as soluções implementadas.

É evidente que a pesquisa experimental para obter a melhor configuração de reconhecimento para o sistema demanda um certo tempo. Existem muitas variáveis que influenciam o processo de classificação e decisão do sistema, como a quantidade de algoritmos de esqueletização, formas de rastreamento, limiar para o sistema, porcentagem de desvio padrão para cima e para baixo, muitas funções de pertinência, etc. O número de combinações possíveis destes fatores é bastante grande, o que torna o problema da calibração do sistema de reconhecimento bastante trabalhoso.

O sistema rastreia a imagem para obter as 12 primitivas dos tipos **h**, **v**, **g** e **d** das expressões PDL, primitivas estas que são geradas pela esqueletização do objeto presente na imagem, extraídas por um processo de varredura pela imagem ainda 40x40.

Posteriormente, aumentou-se de 4 para 8 a quantidade de informações extraídas das imagens. De cada imagem analisada são extraídas 8 informações simbólicas, que são:

1. linha inicial;
2. linha final;
3. coluna inicial;
4. coluna final;
5. número de primitivas **h**;

- 6. número de primitivas v;
- 7. número de primitivas g;
- 8. número de primitivas d.

Chatterji [72], além dos itens 5 a 8 acima, menciona o uso de 4 outras características, a quantidade de curvas do tipo **A**, **U**, **C** e **D**, que são também usadas no processo de reconhecimento, com as quais afirma ter conseguido um índice de reconhecimento de até 96 %.

Qualquer sistema de reconhecimento será tanto melhor quanto mais informações puderem ser extraídas das imagens, pagando-se o preço do sistema ficar mais complexo e sua resposta ficar mais lenta. Eventualmente existe um número pequeno de características *ótimas* que podem ser selecionadas para que o reconhecimento de padrões tenha um alto índice de acertos, gerando o compromisso entre número de evidências e taxa de reconhecimento.

Um exemplo da extração destas oito informações encontra-se a seguir. A extração das informações 5 a 8 anteriores é feita pela contagem de cada um dos quatro tipos de primitivas detectadas na imagem.

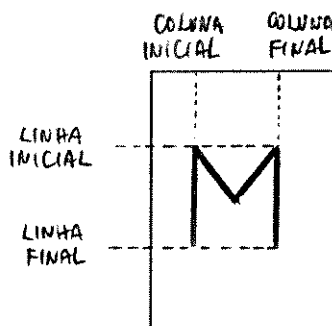


Figura 5.11 - Extração das 8 informações

O sistema implementado faz uso da seguinte equação da lógica nebulosa, onde a variável i representa todos os caracteres alfanuméricos possíveis, e j representa os 4 tipos de primitivas PDL que podem existir.

$$F(\lambda) = 1 - \left[\frac{1}{\text{fator}} \sum_{i=1}^{26} \sum_{j=1}^4 (\text{imagem}_i^j - \text{padrão}_i^j)^2 \right]^{1/2} \quad (I)$$

onde *fator* é um *fator de normalização* para que o resultado do somatório forneça valores entre 0 e 1.

Esta equação nos diz que o menor valor para o somatório

$$S = (H_{im} - H_{pd})^2 + (V_{im} - V_{pd})^2 + (G_{im} - G_{pd})^2 + (D_{im} - D_{pd})^2$$

é a melhor classe de *padrões*, ou seja, a que mais se aproxima da *imagem*. As experiências realizadas com o algoritmo acima estão resumidas na tabela 5.12.

	Total	Porcentual
Número de imagens	55	100,00
Acertos	14	25,45
Erros	39	70,91
Rejeições	2	4,45

Tabela 5.12 - Resultados da experiência 6

Como ainda não havia um conjunto de algoritmos suficientemente eficientes, de modo a obter um índice geral de reconhecimento satisfatório, buscou-se a solução através da máscara representativa do "mapeamento" entre as imagens e as expressões PDL.

Esta máscara consiste de 38 primitivas poligonais, que através de um processo que determina qual delas possuía um índice de presença de pixels superior a um determinado limiar mínimo, separa então as 8 informações simbólicas mencionadas anteriormente para a fase de análise e decisão do sistema. Portanto, muito embora o processo de busca das informações simbólicas

tenha sido alterado, continua-se com a equação (I) anterior como algoritmo de reconhecimento.

É fácil notar que a própria máscara faz o serviço dos algoritmos de esqueletização, deixando portanto de haver perdas estruturais das informações. A máscara aplicada à imagem (i) gera a estrutura simbólica da imagem (ii).

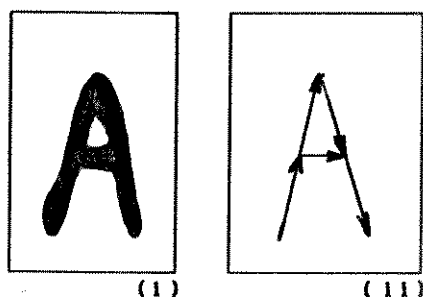


Figura 5.13 - Aplicação da máscara

Os resultados conseguidos com as experiências com este novo método são descritos pela tabela 5.14. Na experiência número 12, portanto, conseguiu-se uma surpreendente melhora no desempenho do sistema.

	Total	Porcentual
Número de imagens	26	100.00
Acertos	20	76.92
Erros	6	23.08
Rejeições	0	0.00

Tabela 5.14 - Resultados da experiência 12 (b)

Percebe-se que a equação (I), apenas com 4 características das imagens, não tem capacidade discriminatória suficiente, ou seja, são necessários mais critérios de discriminação. Um outro fato interessante de se notar é que o

critério de discriminação é o mesmo até aqui, muito embora a maneira de como estas características são extraídas da imagem é que varia nas experiências.

Fez-se necessário um levantamento estatístico, para uso com a equação (1), do compromisso entre a *porcentagem mínima de pixels* da imagem que as primitivas deveriam conter e o *número de primitivas PDL* que o sistema enviava para a fase de análise. Tome-se o exemplo da letra A. Com a utilização da referida máscara com 38 poligonais, se a porcentagem de pixels for 0%, então obtém-se 38 poligonais, se for 5% obtém-se 33 poligonais, e assim sucessivamente, até que se for 95% obtém-se 1 poligonal, e nenhuma poligonal no caso de 100%. O levantamento deste compromisso foi feito para se saber qual porcentagem deve ser usada para se conseguir o melhor índice de reconhecimento. Segue abaixo o gráfico para a letra A maiúscula do alfabeto.

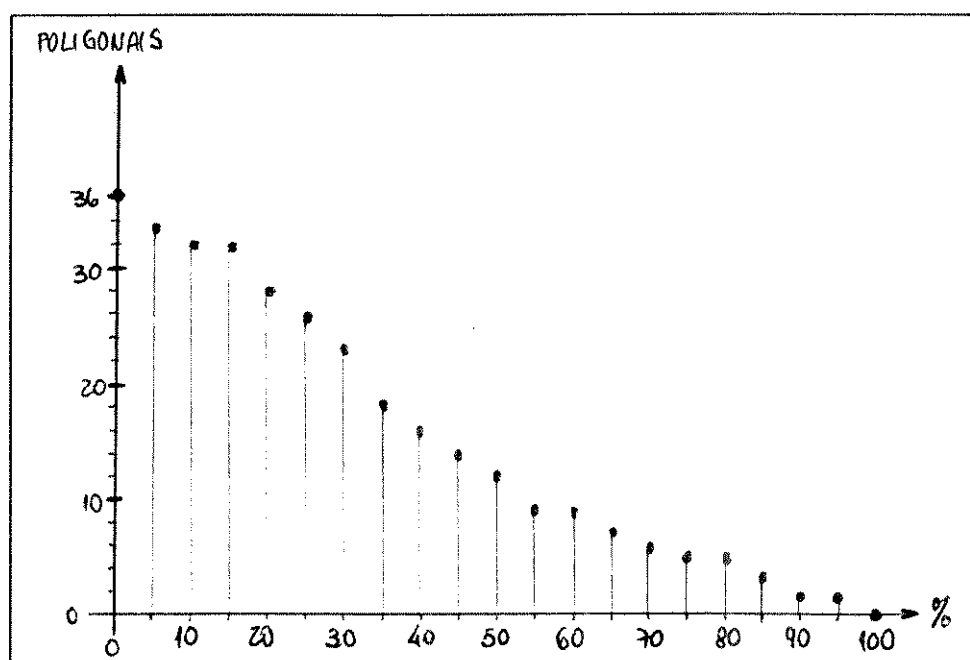


Figura 5.15 - Gráfico do compromisso porcentagem \times poligonais

Uma vez tendo um algoritmo suposto razoável, em termos de capacidade de discriminação de classes, partiu-se para o aspecto de *aprendizagem* com o sistema, ensinando-lhe diversas famílias de padrões, obtendo como resposta a quantidade de imagens corretamente classificadas.

	Família NOV		Família CHE		Família FIN	
	Total	Porcen.	Total	Porcen.	Total	Porcen.
Nº de Imags	26	100.00	26	100.00	26	100.00
Acertos	9	34.62	8	30.76	25	96.15
Erros	9	34.62	12	46.16	1	3.85
Rejeições	8	30.76	6	23.08	0	0.00

Tabela 5.16 - Resultados da experiência 14 (a)
(aprendizagem com a família FIN)

	Família NOV		Família CHE		Família FIN	
	Total	Porcen.	Total	Porcen.	Total	Porcen.
Nº de Imags	26	100.00	26	100.00	26	100.00
Acertos	26	100.00	24	92.30	13	50.00
Erros	0	0.00	1	3.85	11	42.30
Rejeições	0	0.00	1	3.85	2	7.70

Tabela 5.17 - Resultados da experiência 14 (b)
(aprendizagem com a família NOV)

	Família NOV		Família CHE		Família FIN	
	Total	Porcen.	Total	Porcen.	Total	Porcen.
Nº de Imags	26	100.00	26	100.00	26	100.00
Acertos	26	100.00	24	92.30	13	50.00
Erros	0	0.00	2	7.70	11	42.30
Rejeições	0	0.00	0	0.00	2	7.70

Tabela 5.18 - Resultados da experiência 14 (c)
(aprendizagem com a família CHE)

Desta experiência pôde-se tirar algumas conclusões. Onde ocorreu o maior número de erros foi quando se ensinou a família FIN, pois é a família que mais difere entre as três ensinadas, como se pode notar no apêndice E. A diferença reside no formato geral dos caracteres ensinados. Portanto é muito mais fácil ensinar a família NOV ou a família CHE para reconhecer a família FIN do que o contrário. Observou-se que no caso de serem ensinadas as imagens das famílias NOV ou CHE, alguns pixels a mais ou a menos no processo de reconhecimento vão fazer menos diferença do que se fossem ensinadas as imagens da família FIN. Este fenômeno deve-se ao fato de que a família NOV ou CHE tem mais *pixels ensinados* do que na família FIN, portanto o reconhecedor "sente" menos a ausência ou presença de pixels acesos.

Durante estas experiências utilizou-se a máscara para o casamento com gabarito da maneira clássica (ou não nebulosa), pois o resultado final era primitiva ausente ou primitiva presente. Isto acarreta algumas consequências, por exemplo, utilizando-se o *limiar mínimo* para presença de pixels de 70%, uma primitiva poligonal que possua 69% não será levada em consideração, o que torna o sistema muito pouco flexível.

Partiu-se então para uma nova fase, em que o parâmetro para a equação (I) será a porcentagem de presença de pixels em cada uma das 38 poligonais, e não mais a quantidade de cada um dos 4 tipos de primitivas existentes nas

expressões PDL. Portanto, a quantidade de informações discriminatórias, representado pela variável J , passa de 4 para 38 itens. A equação então torna-se a seguinte:

$$F(\lambda) = 1 - \left[\frac{1}{\text{fator}} \sum_{i=1}^{26} \sum_{j=1}^{38} (\text{imagem}_i^j - \text{padrão}_i^j)^2 \right]^{1/2} \quad (II)$$

Por exemplo, suponha-se uma poligonal que no padrão da classe ensinada possui 60 pixels e na imagem aparece com 30 pixels. Isto não significa que a poligonal não existe, mas sim que ela existe "um pouco". Se fossem encontrados 50 pixels nesta mesma poligonal da imagem, então a primitiva estaria "quase com certeza", e assim sucessivamente. Portanto, o algoritmo passa a ter *capacidade nebulosa* de trabalhar com poligonais que não estão *totalmente presentes* na imagem.

A experiência seguinte foi onde as características nebulosas do sistema foram postas à prova, saindo-se muito bem, como se pode observar pela tabela 5.19.

	Total	Porcentual
Número de imagens	26	100.00
Acertos	24	92.30
Erros	2	7.70
Rejeições	0	0.00

Tabela 5.19 - Resultados da experiência 15

Na figura 5.21, a partir da imagem real (a) fornecida ao sistema, consegue-se as informações simbólicas descritas em (b), que podem ser novamente mapeadas para uma *pseudo-imagem* de 40x40 como na figura (c). Dentre todas estas primitivas, o sistema seleciona apenas aquelas que ele precisa

para efetuar o reconhecimento (d), o que pode ser descrito estruturalmente de acordo com as figuras (e) e (f). O resultado final para o usuário aparece como a figura (g), e os resultados da experiência estão na tabela 5.20.

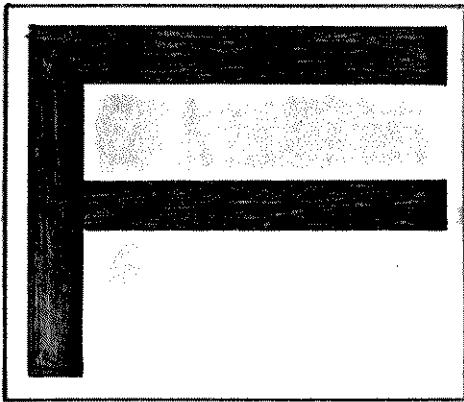
	Total	Porcentual
Número de imagens	26	100.00
Acertos	26	100.00
Erros	0	0.00
Rejeições	0	0.00

Tabela 5.20 - Resultados da experiência 16

Posteriormente concluiu-se em meio a numerosas experiências, que a máscara para o casamento com gabarito nebuloso ainda não era apropriada para a perfeita descrição em termos das expressões PDL de todos os caracteres alfanuméricos que o presente sistema deveria reconhecer. Fez-se então uma segunda máscara, semelhante à primeira, que extraía 50 primitivas. Assim, o parâmetro j da equação (III) permite uma maior capacidade de discriminação.

$$F(\lambda) = 1 - \left[\frac{1}{\text{fator}} \sum_{i=1}^{26} \sum_{j=1}^{50} (\text{imagem}_i^j - \text{padrão}_i^j)^2 \right]^{1/2} \quad (\text{III})$$

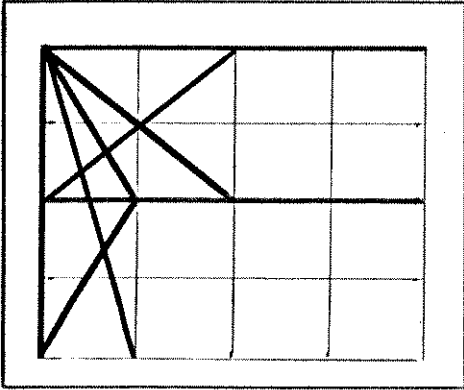
Da primeira experiência com a nova máscara os seguintes resultados foram obtidos, conforme a tabela 5.22.



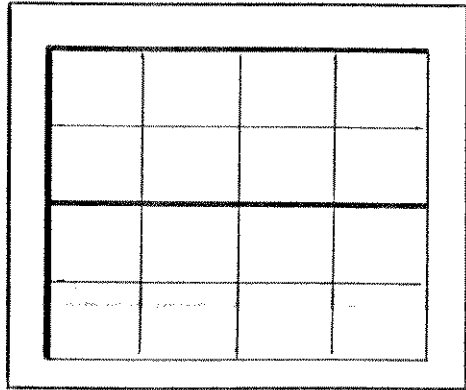
(a)

reTa(1,1,1,20).
 reTa(1,20,1,40).
 reTa(20,1,20,20).
 reTa(20,20,20,40).
 reTa(20,1,1,1).
 reTa(40,1,20,1).
 reTa(20,1,1,20).
 reTa(1,1,20,20).
 reTa(1,1,20,10).
 reTa(40,1,20,40).
 reTa(20,10,20,30).
 reTa(1,1,20,5).
 reTa(20,5,40,10).

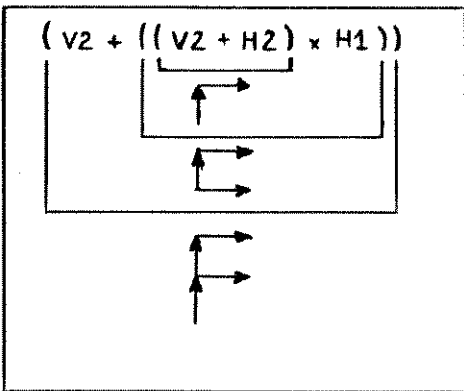
(b)



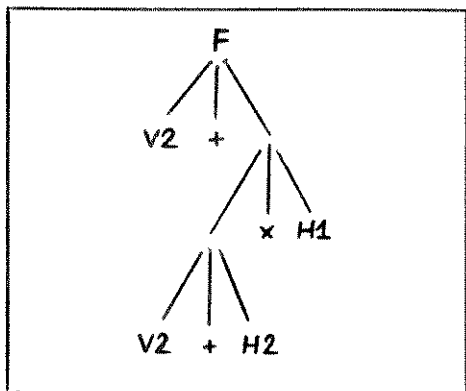
(c)



(d)



(e)



(f)

O CARACTER É A
 LETRA F MAIÚSCULA,
 TOTALMENTE.
 $F(\lambda) = 0,957404$

(g)

Figura 5.21 - Sistema RENEALF em funcionamento

	Total	Porcentual
Número de imagens	26	100.00
Acertos	26	100.00
Erros	0	0.00
Rejeições	0	0.00

Tabela 5.22 - Resultados da experiência 17

Até esta fase de experimentação testou-se apenas a capacidade de discriminar caracteres alfanuméricos distintos, ou seja, caracter A de W, T de K, etc. A seguir verificou-se a capacidade de discriminar entre letras A de diferentes tipos.

Foi feita uma experiência com a letra A em diversos *formatos* distintos, que foram : 1. standard; 2. pequeno; 3. itálico; 4. tremado; 5. matemático; 6. sublinhado; 7. expandido; 8. manuscrito; 9. gótico; e 10. quadrado. Os resultados estão descritos na tabela 5.23 e as imagens podem ser observadas no apêndice H.

Aqui cabe ressaltar o seguinte aspecto. A máscara para o casamento com gabarito nebuloso, seja a de 38 quanto a de 50 primitivas poligonais, não é apropriada para o reconhecimento de letras góticas ou manuscritas, pois ambas as máscaras reconhecem preferencialmente *retas*, enquanto que estes dois formatos são *construídos essencialmente por curvas*. Outro fato importante é que todas as letras A's tem que ser *estruturalmente semelhantes* à letra A do padrão ensinado. Em outras palavras, estes formatos são *estruturalmente distintos* dos padrões normais do sistema. Assim, pelo que se pode notar na figura 5.24, nem todas as imagens serão classificadas corretamente.

FONTES	VALORES DE PERTINÊNCIA
standard	0.98965347
pequeno	<i>não alcançou limiar mínimo</i>
itálico	0.97138754
tremado	0.88791892
matemático	0.88791892
sublinhado	0.99056592
expandido	0.99056592
manuscrito	0.93404977
gótico	<i>não alcançou limiar mínimo</i>
quadrado	0.97138754

Tabela 5.23 - resultados da experiência 19

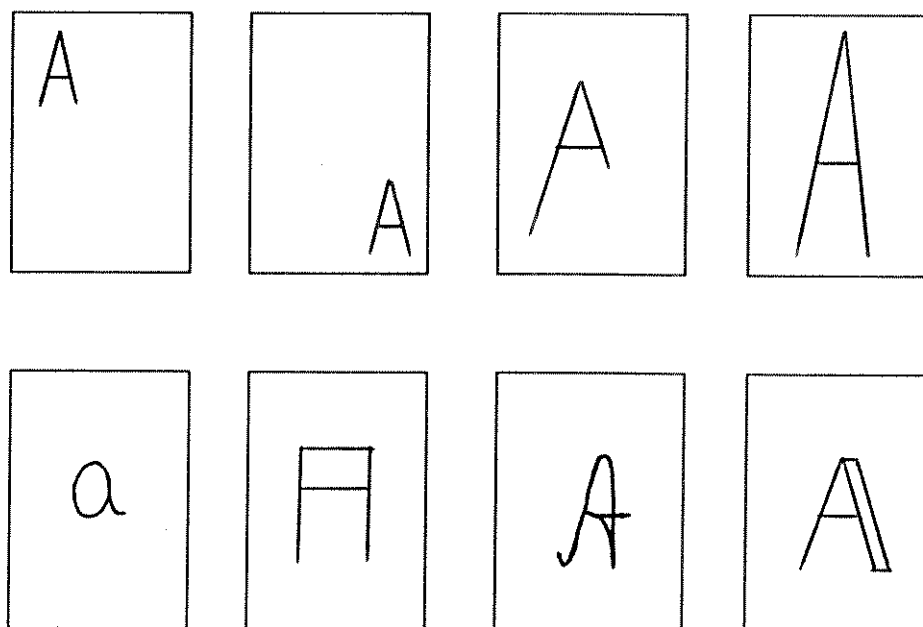


Figura 5.24 - Grupos de Imagens

Por fim, tratou-se o problema da calibração do sistema objetivando a elevação do índice de reconhecimento, através da variação dos parâmetros de entrada do sistema. A calibração tem a intenção de tornar o sistema o mais especializado possível para uma determinada aplicação, tornando-o capaz de realizar a tarefa de reconhecer caracteres alfanuméricos bastante específicos, ensinando-se exatamente a família de imagens que se quer reconhecer, para tornar o sistema poderoso e confiável. É esta família de imagens que se usa no processo de calibração, conforme a figura 5.25.

Foram feitas seis experiências baseadas em seis valores para o índice mínimo de presença de pixels, que foram 66, 71, 76, 81, 86 e 91 %, cada qual testado com 3 valores para o *limiar mínimo* para a função de pertinência, que foram de 0.7500, 0.8250 e 0.9000. Todas as seis experiências foram feitas com a letra A, conforme pode-se observar na sequência de tabelas do apêndice K.

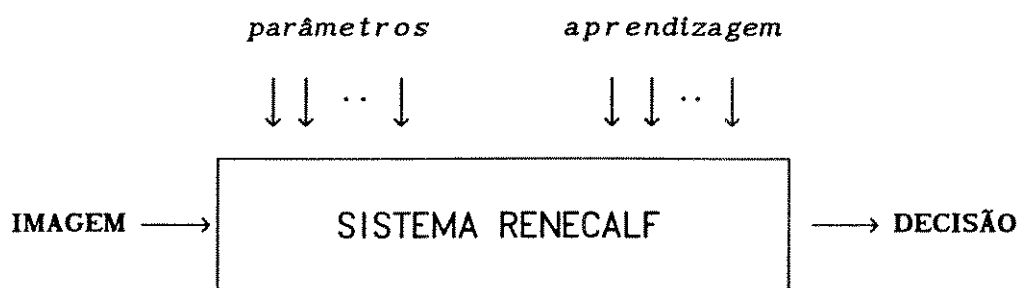


Figura 5.25 - Configuração final

Foram feitas 22 imagens, sendo que desta vez as imagens estão correlacionadas entre si, pois uma difere da outra pela redução da espessura das primitivas, como se pode observar no apêndice G. A intenção ao tirar as camadas de pixels dos caracteres era a de testar a tolerância máxima que o sistema conseguia suportar. Em outras palavras, até que valor pode-se alterar a imagem de uma letra A e o sistema continuar convergindo para a classe de padrões da letra A.

Como resultado do processo de calibração do sistema, passam a ocorrer mais *rejeições* de imagens do que *erros*, o que pode ser considerado positivo. É preferível que uma imagem seja rejeitada do que ser classificada erroneamente,

pois o prejuízo causado pela primeira decisão é menor do que o prejuízo causado por uma decisão errada.

Foi gerado um gráfico final que se encontra nas figuras 5.26 e 5.27 a seguir. Por este gráfico, para o limiar mínimo de 0,9000, vendo-se que a quantidade de acertos decresce com o crescimento do índice de presença de pixels, sugere-se portanto a utilização do sistema com 66 % de presença de pixels nas primitivas poligonais da máscara nebulosa. Note-se que no índice de 76 %, ainda não ocorreu nenhum erro de classificação, entretanto ocorreram 10 rejeições.

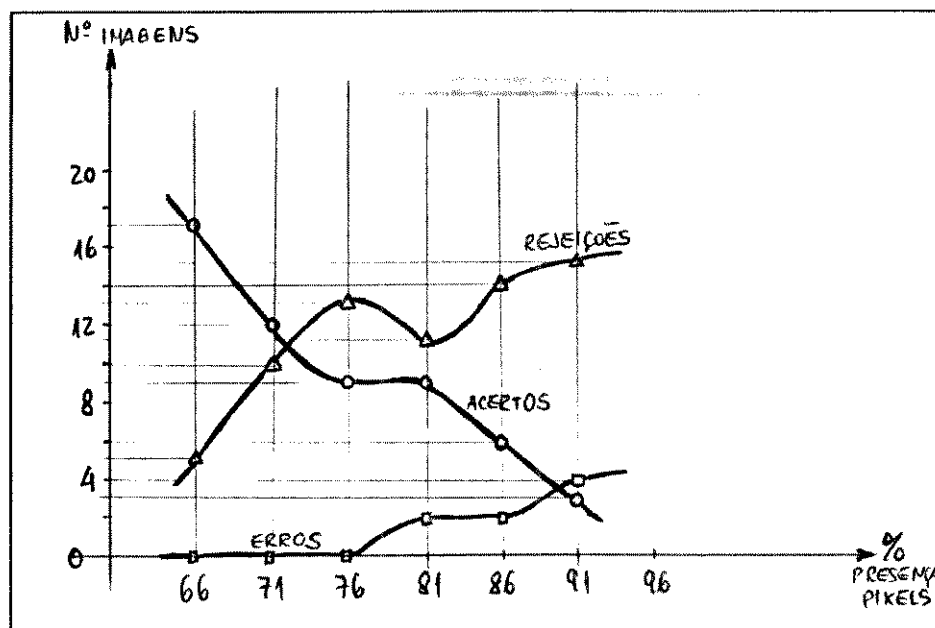


Gráfico 5.26 - Calibração do sistema com limiar 0,9000

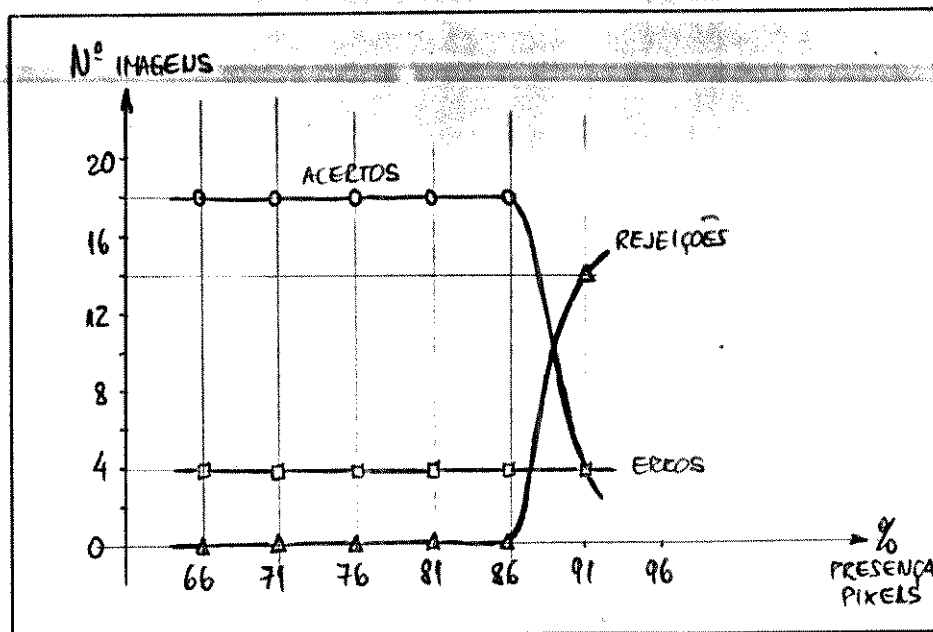


Gráfico 5.27 - Calibração do sistema com limiar de 0,7500

Entretanto, quando o limiar mínimo é de 0,7500, o comportamento do sistema é completamente diferente, como se pode notar pelo segundo gráfico. Neste caso específico a quantidade de acertos, erros e rejeições mantem-se constante dentro da faixa de 66 a 86 % de presença de pixels, quando então ocorre uma grande queda na quantidade de acertos.

Ao se fazer o gráfico de calibração para todos os *limiares mínimos* que se queira, notar-se-á que os resultados dos comportamentos terão diferenças entre si, levando à conclusão que a escolha final fica a cargo do usuário do sistema de reconhecimento de padrões.

V.3. Considerações Gerais

A intenção do presente trabalho foi a de se estudar as dificuldades que aparecem para solucionar o problema de reconhecimento de padrões, para o caso específico de caracteres alfanuméricos. Após um estudo teórico partiu-se para a implementação, pois é na prática que se percebe o quanto é difícil automatizar este processo. Um reconhecedor de caracteres alfanuméricos deve ter um alto índice de desempenho, medido por meio de uma alta quantidade de caracteres corretamente reconhecidos e por uma baixa quantidade de caracteres rejeitados e/ou erroneamente reconhecidos.

Inicialmente foram dadas como entrada para o sistema imagens de caracteres alfanuméricos que não apresentavam nenhuma deformação. Em outras palavras, as imagens estavam "bem comportadas", e neste caso o sistema obteve resultados bastante satisfatórios, levando em consideração os modelos de caracteres selecionados para o sistema.

Quando estes caracteres são ligeiramente alterados, ou seja, têm alguns de seus pixels acesos ou apagados aleatoriamente, o sistema começa a cair ligeiramente de desempenho, apresentando algumas rejeições e ainda quase nenhum erro de classificação. Quanto mais se altera, também mais se deteriora a resposta do sistema, pois começam a surgir os primeiros erros de classificação, e este resultado prático encontra-se ilustrado no gráfico 5.28.

Quando se atinge um determinado nível de mudanças nas imagens o índice de reconhecimento fica muito abaixo de qualquer expectativa mínima, ou seja, o sistema torna-se inviável como uma *máquina de reconhecimento*. Imagens aleatórias de caracteres manuscritos, por exemplo, dificilmente serão corretamente classificadas.

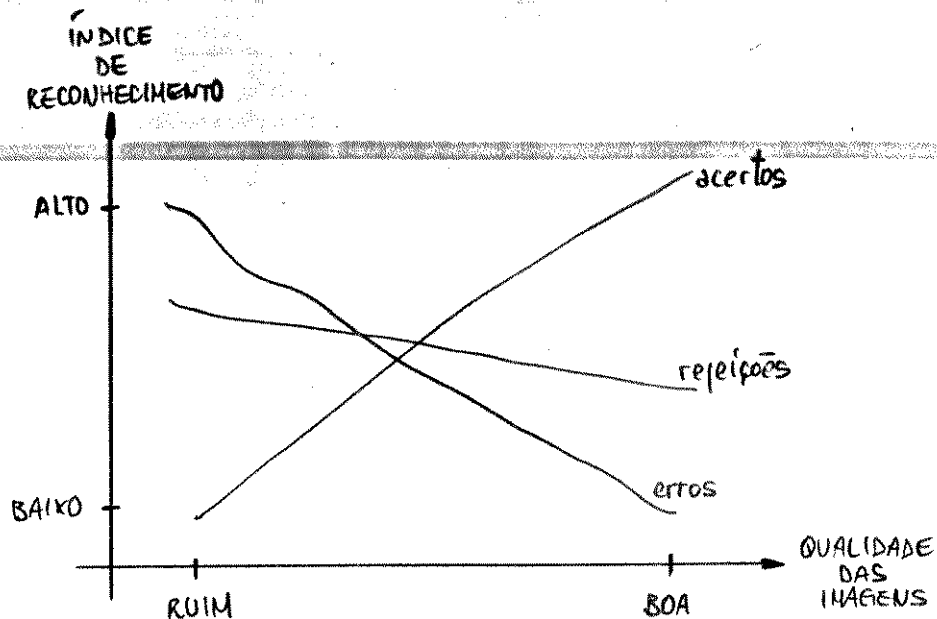


Gráfico 5.28 - Acertos e erros com a qualidade da imagem

Observa-se também que a *resolução das expressões PDL*, medidas em termos do número de primitivas PDL que a compõem, e o *tamanho da matriz da imagem* não podem ser muito diferentes. Tanto não devem ser comparadas imagens pequenas com expressões PDL com muitas primitivas, como também não devem ser comparadas imagens grandes com expressões PDL com poucas primitivas. No primeiro caso, imagens pequenas com muitas primitivas, ou seja, uma *gramática poderosa* para um sistema de visão de baixa resolução, facilitaria o aparecimento de *clusters* com caracteres complexos, tais como **B** ou **S**. Pode-se notar que o grande número de primitivas que ficariam em excesso debilitariam muito o valor de pertinência para os caracteres mais simples, tais como **I** ou **L**. Para o segundo caso, de imagens grandes com poucas primitivas, ou seja, uma *gramática leve* para um sistema de visão sofisticado (como no caso do HOMUK, por exemplo), aconteceria exatamente o oposto. A escassez de primitivas debilitariam desta vez as funções de pertinência dos caracteres complexos, formando assim *clusters* maiores para os caracteres simples.

CAPÍTULO VI

CONCLUSÕES E SUGESTÕES

VI.1. Análise dos Resultados

VI.3. Sugestões para Desenvolvimentos Posteriores

VI.4. Considerações Finais

VI.1. Análise dos Resultados

Segundo Lucena [84], "a visão processada por computadores capacita a máquina a ver, identificar ou compreender o que vê, localizar o que se está procurando, e assim sucessivamente". O objetivo da visão portanto é o de capacitar o computador a receber e processar informações visuais.

A seleção das características a serem usadas na implementação de um problema particular de reconhecimento de padrões é complexo e depende de muitos fatores (Troxel [82]), como taxa de erro requerida, tipos de erros que devem ser toleráveis, grau de confiança necessário para se obter uma resposta, o custo da implementação, velocidade em que os padrões devem ser classificados, necessidades de armazenamento, e assim por diante.

É evidente que quanto mais certeza existe acerca da capacidade discriminatória de uma característica específica, tanto melhor para o sistema de reconhecimento, pois o índice final de acertos será certamente bastante promissor. Isto implica em que o problema de reconhecimento de padrões baseia-se fortemente no Problema da Seleção de Primitivas. Quanto melhor a primitiva, em termos *qualitativos*, mais imagens serão corretamente classificadas, enquanto que se não houver nenhuma primitiva com poder discriminatório suficiente, tanto mais primitivas serão necessárias, em termos *quantitativos*, para se obter um índice de acerto equivalente.

Segundo Rosenfeld [83], uma característica do objeto terá poder decisório se aparece em apenas um dos *objetos alvo*. Outra característica, mesmo que seja facilmente detectável em uma imagem, terá valor de discriminação quase nulo se ela aparecer em vários *objetos alvo*. O peso da importância de certa característica depende portanto do número de objetos a serem reconhecidos.

Viu-se neste trabalho que apenas as 4 características de Chatterji, que são as quantidades de cada uma dos 4 tipos de primitivas PDL existentes, não têm o poder necessário de classificar a maioria das imagens nas classes

corretas. Mudou-se a maneira como as primitivas eram extraídas das imagens, através da mudança dos algoritmos de esqueletização, dos algoritmos de rastreamento, do índice de presença de pixels nas primitivas das máscaras, e assim sucessivamente, tentando analisar a influência de cada um destes fatores no processo global de reconhecimento de padrões. Estes resultados foram vistos no capítulo anterior.

Partiu-se então para a implementação das 2 máscaras para casamento com gabarito nebuloso, uma com 38 primitivas e a outra com 50 primitivas, passando então a haver 38 e 50 características com poder discriminatório, respectivamente, ao invés dos quatro fatores acima mencionados. O índice de reconhecimento com as características *nebulosas* de ambas, dentro de certos limites de tolerância para as imagens dos caracteres alfanuméricos, foi de praticamente 100 %. Em outras palavras, o sistema pode ser utilizado na prática.

Pelo que se concluiu das funções de pertinência, no total de nove funções propostas, viu-se que quanto mais fatores se leva em consideração, tanto melhor para a capacidade de acerto do sistema. Foram cinco os fatores usados como *medidores de proximidade* (custos) entre as árvores PDL da classe do padrão e da imagem:

1. concatenação mal feita;
2. excesso de primitivas;
3. ausência de primitiva;
4. concatenação impossível; e
5. tamanho da primitiva.

Nada impede de serem tomados outros fatores estruturais em consideração, ou deixar de tomar todos os 5 fatores acima numa mesma função, ou até mesmo fazer-se uma ponderação entre os diversos fatores. Os critérios são os mais variados possíveis, mas é claro que certos critérios são dependentes da aplicação que se deseja implementar, enquanto que outros são independentes da aplicação.

Como se pode notar os métodos sintáticos, particularmente as expressões

PDL, são não apenas discriminatórios, mas também descritivos, conforme as pretensões da metodologia de reconhecimento e exemplificado através da árvore de reconhecimento mostrada nas experiências. Os padrões são portanto armazenados como uma sequência de símbolos que conseguem descrevê-los não ambigualmente, podendo com isto serem comparados total ou parcialmente a outros padrões, respondendo assim às perguntas formuladas por Narasimhan [47].

Em qualquer sub-área da inteligência artificial, um dos maiores problemas que se enfrenta é o Problema da Representação do Conhecimento, haja vista que interessa na maior parte das aplicações apenas a descrição da imagem, e não a sua representação pictórica. A representação do objeto pertencente à imagem, no caso do presente trabalho, dá-se através destas expressões PDL, onde cada primitiva corresponde a um componente básico da imagem e vice-versa. Esta correspondência biunívoca é que faz com que não ocorram ambiguidades na descrição do dito objeto.

Como foi observado no capítulo V deste trabalho, a partir de uma imagem são gerados os predicados para cada objeto nas imagens, e a partir da melhor escolha destes predicados, pode-se armazenar a informação da letra correspondente à imagem, onde estes predicados representam os pontos de controle dos caracteres nas imagens.

Percebeu-se também a influência de alterações nas imagens quanto à rotação, escalamento e translação de seus objetos constituintes, tirando conclusões quanto à dependência ou independência dos métodos de reconhecimento em relação a estas 3 entidades. As máscaras são extremamente dependentes em relação aos fatores acima citados sendo, contudo, extremamente importantes quanto ao poder de discriminação. Quanto às expressões PDL, viu-se em experiências que uma vez bem determinadas as primitivas da imagem, então o sistema de inferência consegue reconhecer o carácter alfanumérico presente na imagem mesmo sob efeito de escalamento e/ou de translação, mas infelizmente não consegue reconhecê-lo sob efeito da rotação.

Outro fator que teve peso significativo foi a observação do desempenho do sistema quanto à diferenciação de diversos tipos de *formatos* para as letras do alfabeto. Esta é a propriedade de *similaridade*, ou seja, o sistema não pode

separar imagens diferentes do mesmo caracter alfanumérico, tais como A itálico, A gótico, e assim sucessivamente. O sistema deve ter a capacidade de diferenciar os diversos formatos do mesmo caracter, sem entretanto classificá-los erroneamente. Isto foi feito sob o paradigma da aprendizagem supervisionada. As imagens dos caracteres alfanuméricos a serem reconhecidas não podem ser *qualsquer*, mas sim devem ser *próximas* das imagens representantes das diversas classes ensinadas na *fase de implantação* do sistema.

Evidentemente que em qualquer dos casos acima, imagens representativas de padrões distintos muito *próximas* exigem maior capacidade discriminatória no processo de classificação, pois a convergência para a classe de padrões correta vai se tornar mais difícil. Ao se escolher os padrões que servirão de modelo para cada classe, deve-se ter o cuidado de tomar exemplares os mais distantes possíveis um dos outros. Assim dá-se maior possibilidade para que o algoritmo que realiza a tarefa de classificação cumpra sua tarefa corretamente.

O uso da lógica nebulosa se faz necessário por vários fatores, dentre eles a fronteira não bem definida entre os caracteres, a proximidade de um caracter a outro, a possibilidade de vários tipos de formatos para o mesmo caracter, a possibilidade da análise de diversos critérios para as funções de pertinência das imagens, e assim sucessivamente. O conceito de conjuntos nebulosos pode ser inteiramente aplicado à classificação de figuras geométricas simples e/ou complexos, tais como os caracteres alfanuméricos.

Quando se calculam os 26 valores de pertinência (pelas equações do capítulo V) em relação a cada classe de padrões, havendo apenas um valor de $F(\lambda)$ muito maior do que todos os outros valores, torna-se fácil a decisão final sobre qual a classe de padrões a que pertence o objeto (figura 6.1).

Entretanto, se existem comparações cujos resultados são semelhantes, como no caso 6.2 a seguir, então deve existir um critério de desempate para fazer a atribuição de classe para o objeto. A escolha deste critério depende da aplicação que se faz com o sistema de reconhecimento, e no caso do presente trabalho, esta escolha foi feita pelo operador, *manualmente*.

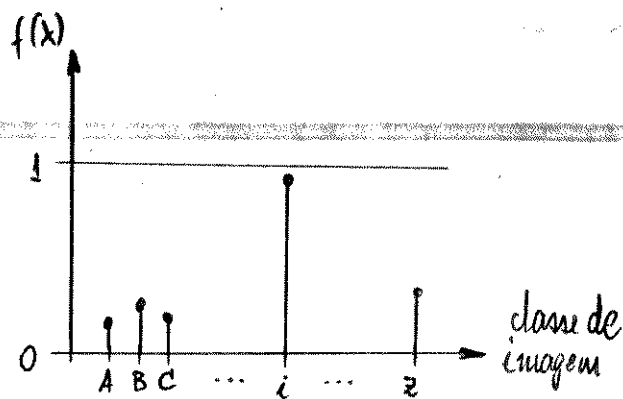


Figura 6.1 - Gráfico de decisão

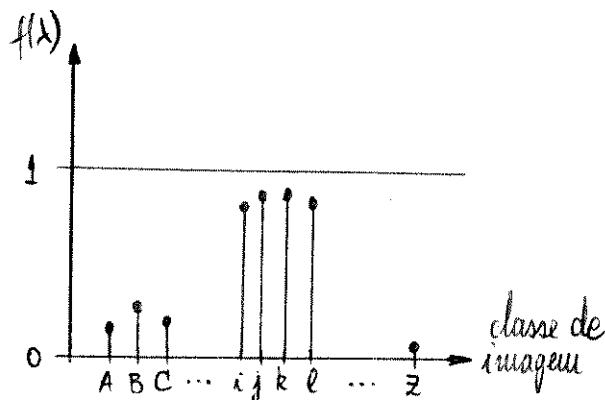


Figura 6.2 - Segundo gráfico de decisão

Enfatiza-se novamente que este trabalho teve a intenção não somente de implementar um reconhecedor linguístico de padrões, tentando construir uma eficiente máquina de leitura, mas também de testar os muitos fatores que influenciam a construção desta máquina, através de uma análise do processo de reconhecimento. Estas influências podem ser observadas no pré-processamento, na detecção das primitivas, na rotação da imagem, na qualidade dos objetos na imagem, no uso ou não de gramáticas transformacionais, nos valores numéricos para os *limiares*, e assim por diante.

Por fim foi possível observar o processo de **calibração** do sistema, onde os mais diversos fatores (parâmetros do sistema, variáveis globais, etc) que exercem influência no reconhecimento são alterados na tentativa de se obter a **melhor combinação**. Em outras palavras, selecionam-se os valores numéricos para os fatores do processo que resultem em taxa de acerto elevada para o sistema de reconhecimento.

VI.2. Sugestões para Desenvolvimentos Posteriores

Atualmente segue-se dedicando esforços consideráveis à investigação na área de Visão Artificial. Atenção especial está-se dedicando aos sistemas em três dimensões (3D), à representação e ao modelamento de sólidos e suas relações com os sistemas PAC, a incorporação e importância da cor como informação adicional da cena, e está-se iniciando a análise de cenas em movimento.

Prevê-se para futuro próximo :

1. sistemas que usufruam de sensores óticos melhores e mais baratos;
2. projetos de arquiteturas de processadores dedicados para o processamento digital de imagens;
3. utilização de linguagens de alto nível para o processamento de imagens;
4. incorporação de cor, tridimensionalidade, textura, dinamismo, etc;
5. redução dos custos totais destes sistemas.

Destaca-se, como sugestão, o reconhecimento de textos completos através de sua imagem. Se existirem algoritmos de segmentação implementados, pode-se reconhecer vários caracteres alfanuméricos em uma mesma imagem, sendo possível então o reconhecimento de palavras, frases e de textos completos.

Suponha-se que tenham sido implementados algoritmos de segmentação para uma imagem qualquer, por exemplo, a imagem de um texto. Poder-se-ia implementar uma rotina que navegasse as 2 máscaras (de 38 e de 50 primitivas) pela imagem, a fim de detectar as letras constituintes de um documento, idéia esta que pode ser visualizada pela figura 6.3. Para este caso específico, porém, seriam extremamente importantes a implementação de técnicas de visão de baixo nível, como filtragem de frequências, realçamento, suavização, etc, pelo fato de que sendo as imagens realmente extraídas do mundo real, seriam adquiridas por meio de uma câmera e estariam portanto em ambientes não

controlados, com possibilidade de ruídos.

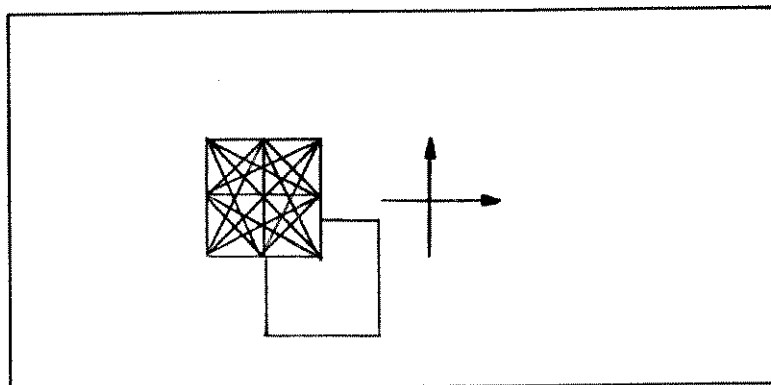


Figura 6.3 - Um sistema de navegação de máscaras

Sistemas de processamento de imagens que possuem uma *resolução* muito grande não poderiam utilizar diretamente os algoritmos deste trabalho, pois ocorreriam muitos erros devido à diferença na *qualidade* das imagens, conforme a figura 6.4 a seguir. Note-se que imagens com 512x512 pixels tem uma *qualidade de definição* muito superior às imagens usadas neste trabalho, portanto a extração das primitivas PDL teria de ser alterada para uso com esta nova resolução. É claro que quanto maior a resolução das imagens, maior também o número de primitivas PDL que vão ser extraídas, portanto, novos critérios discriminatórios teriam de ser implementados.



Second carbon copy



Example of offset lithography printing

Figura 6.4 - Qualidade das imagens

Como exemplo de reconhecimento de textos completos, Shaw [49] fornece uma gramática, também em PDL, para se fazer o reconhecimento em uma dada imagem. Uma página de texto pode ser analisada sintaticamente por meio de sentenças, linhas, palavras e caracteres. Primitivas nulas estabelecem a conectividade de palavras em linhas (iws), caracteres em palavras (ics) e caracteres em linhas (ils). *Esquerda*, *Direita* e *Rodapé* são indicações de posicionamento na página. Segue um exemplo de gramática PDL para reconhecimento de textos.

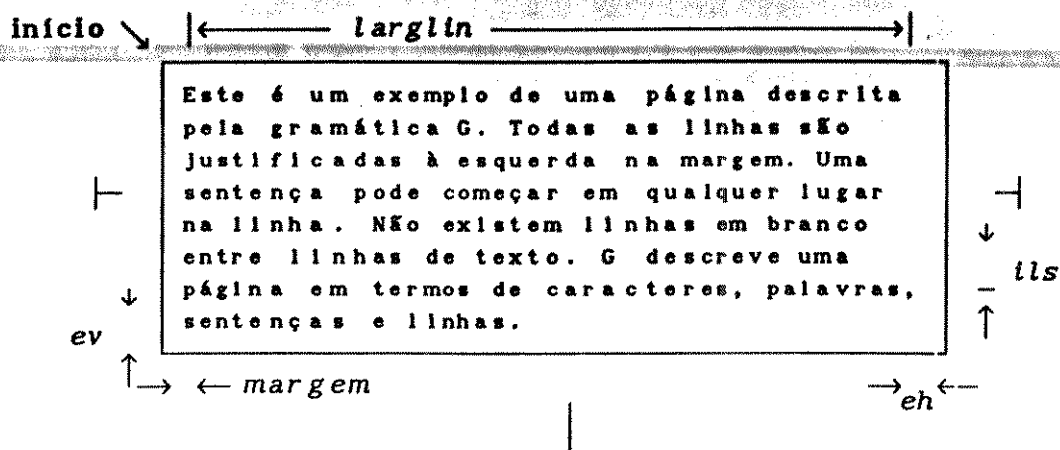


Figura 6.5 - Exemplo de reconhecimento de textos completos

Os símbolos mostrados na figura 6.5, que são as primitivas, representam o seguinte:

- esquerda |—
- direita —|
- rodapé |
- margem —
→
- larglin ~~~~~
←
- ils | ↓
- ponto -
- início ↘
- iws —
→
- eh —
→
- ics —
→
- ev | ↓

A gramática usada para inferir o texto anterior pode ser representada pelas regras BNF a seguir.

PÁGINA ::= (início + (início + EOP))
S ::= SENT
 ::= S + SENT
SENT ::= (INÍCIOSENT + (L + (ics + ponto)))
INÍCIOSENT ::= tws
 ::= (eol + ((larglin + esquerda) + (ils + margem)))
 ::= λ
L ::= LINHA
 ::= (L x ((ils + margem) + LINHA))
LINHA ::= PALAVRA
 ::= (LINHA + (tws + PALAVRA))
PALAVRA ::= CARAC
 ::= (PALAVRA + (ics + CARAC))
CARAC ::= A : B : C : D ... Y : Z
EOP ::= (ev + rodapé)
EOL ::= (eh + direita)

Figura 6.6 - Gramática para reconhecimento de textos

Para se reconhecer apenas uma letra não é necessário o uso de gramáticas sensíveis ao contexto, mas quando se lê palavras, frases ou textos inteiros, faz-se então necessário o uso destas. A necessidade de contexto pode ser facilmente visualizada pelo exemplo da figura 6.7. O leitor interpreta diferentemente o caracter do centro se este for lido horizontal ou verticalmente.

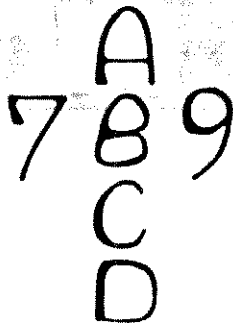


Figura 6.7 - Exemplo de interpretação por contexto

Assim como no exemplo acima diversas outras aplicações podem ser implementadas mediante a adoção de gramáticas adequadas, como por exemplo reconhecimento de caracteres chineses, análise de impressões digitais, interpretação automática de eletroencefalogramas (EEG) [23, 50].

Outra sugestão é a criação de um ambiente de muito alto nível que possa aprender *novas representações* para os padrões já previamente ensinados, ou que aprenda representações para *novos padrões*. Portanto, estes procedimentos de reconhecimento teriam entrada para um **dicionário** com os seus respectivos códigos. O código de qualquer nova instância de um padrão poderia ser colocado neste dicionário, e então a imagem deste novo objeto poderia ser imediatamente reconhecida.

Estas linguagens de muito alto nível para processamento de imagens podem ser representadas pela figura 6.8.

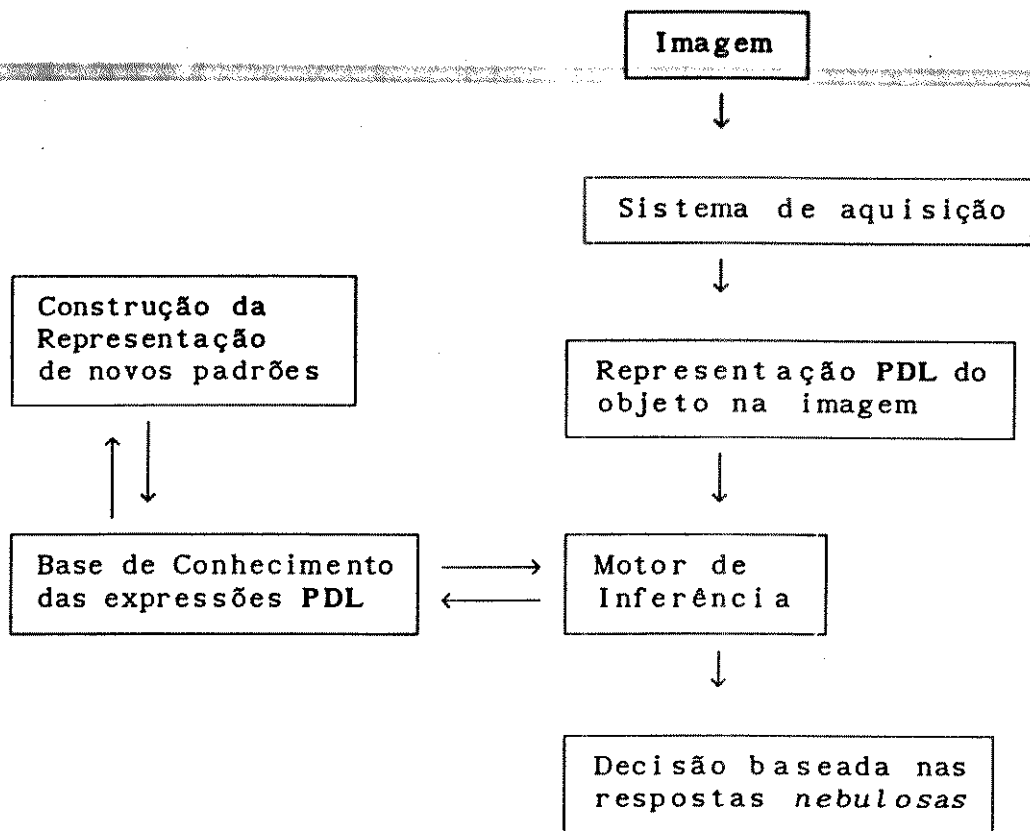


Figura 6.8 - Representação de linguagens de muito alto nível

VI.3. Considerações Finais

Na área de **Visão Artificial para Computadores** a tomada de decisão tem como objetivo enquadrar *corretamente* uma imagem em uma e apenas uma classe de padrões, sendo esta uma das *classes ativas* ou a *classe de rejeição*. Para isto o reconhecedor tem que possuir todas as características, atribuições e/ou capacidades discriminatórias para análise e classificação de uma imagem.

Diversos aspectos da **Inteligência Artificial** são levados em consideração atualmente em construções de poderosos sistemas de computação. Para o caso específico de reconhecimento de padrões, é óbvio que quanto mais conhecimentos o sistema possuir, tanto menos imagens serão erroneamente classificadas, dado que muitos destes algoritmos usam *técnicas cognitivas* associadas à inteligência artificial para inferir a forma e a identidade dos objetos de forma precisa.

Uma dificuldade que pode determinar a não utilização dos métodos estruturais é que as gramáticas usadas para caracterizar a estrutura dos padrões e para gerenciar a análise sintática podem ser ambíguas, fazendo com que um mesmo padrão possa ser analisado sintaticamente de diversas maneiras. Além disto, podem gerar algumas sentenças que não são descrições estruturais de padrões do mundo real (não padrões). Uma das sugestões que se faz quanto a este problema é o uso de gramáticas sensíveis ao contexto.

As aplicações do reconhecimento estrutural de padrões são muitas, conforme já citado anteriormente, além de inspeção de peças manufaturadas, sensoriamento remoto, previsão de metereologia, dentre outras.

O sistema desenvolvido neste trabalho é mais uma contribuição para a área de **Visão Artificial** e certamente poderá ser ampliado e melhorado em suas características de reconhecimento e também em suas aplicações, de modo a torná-lo mais abrangente.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Malers, J. e Sherif, Y. S.; "Applications of Fuzzy Set Theory", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-15, No. 1, Jan/Feb 1985, páginas 175 até 189
- [2] Kent, E. e Schneier, M.; "Eyes for Automatons"; *IEEE Spectrum*, March 1986; páginas 37 até 45
- [3] Hall, L. e Kandel, A.; "Studies in Possibilistic Recognition"; *Fuzzy Sets and Systems*, Vol. 17, No. 2, página 166, 1985
- [4] Brooks, R.; "Symbolic Reasoning Among 3-D Models and 2-D Images"; *Artificial Intelligence*, Vol. 17, páginas 285 até 348, 1981 (Special Issue on Computer Vision)
- [5] Ismail, M. A. e Selim, S.; "Fuzzy C-means: Optimality of Solutions and Effective Termination of the Algorithm"; *Pattern Recognition*, Vol. 19, No. 6, páginas 481 até 486, 1986
- [6] Hathaway, R. e Bezdek, J.; "Local Convergence of the Fuzzy C-means Algorithms"; *Pattern Recognition*, Vol. 19, No.6, páginas 477 até 480, 1986
- [7] Zadeh, L.; "Outline of a new Approach to the Analysis of Complex Systems and Decision Processes"; *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-3, No.1, Jan 1973, páginas 28 até 44
- [8] Nandhakumar, N. e Aggarwal, J.; "The Artificial Intelligence Approach to Pattern Recognition: A Perspective and an Overview"; *Pattern Recognition*, Vol. 18, No. 6, páginas 383 até 389, 1985
- [9] Gonzalez, R. e Thomason, M.; "Syntactic Pattern Recognition"; *Addison Wesley*, 1978
- [10] DePalma, G. F. e Yau, S. S.; "Fractionally Fuzzy Grammar with Application to Pattern Recognition"; *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, Lotfi Zadeh et alii, Academic Press, 1975, páginas 329 até 351

- [11] Shimura, M.; "An Approach to Pattern Recognition and Associative Memories using Fuzzy Logic"; *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, Lotfi Zadeh et alii, Academic Press, 1975, páginas 449 até 476
- [12] Negoita, C. V.; "Expert Systems and Fuzzy Systems"; *The Benjamin Cummins Publ. Co.*, 1985
- [13] Duda, R. O. e Hart, P. E.; "Pattern Classification and Scene Analysis"; *John Wiley and Sons*, 1973
- [14] Fu, K. S.; "Syntactic Methods in Pattern Recognition"; *Academic Press*, 1974
- [15] Kaufmann, A.; "Introduction to the Theory of Fuzzy Subsets"; *Academic Press*, 1975
- [16] Negoita, C. V. e Ralescu, D. A.; "Applications of Fuzzy Sets to Systems Analysis"; *John Wiley and Sons*, 1975
- [17] Marcus, C.; "Prolog Programming"; *Addison Wesley*, 1986
- [18] Smith, R. W.; "Computer Processing of Line Images: a Survey"; *Pattern Recognition*, Vol. 20, No. 1, páginas 7 até 15, 1987
- [19] Banerji, R. B.; "A Language for Pattern Recognition"; *Pattern Recognition*, Vol. 1, No. 1, páginas 63 até 74, 1968
- [20] Schildt, H.; "Artificial Intelligence using C"; *McGraw Hill*, 1987
- [21] Buck, T.; "Uma Monografia sobre o Uso de Lógica Nebulosa Aplicada a Reconhecimento de Padrões"; *Relatório Interno*, 16 de dezembro de 1987, UNICAMP, Campinas, SP
- [22] Furtado, A. L. e Casanova, M. A.; "String Pattern Matching in Prolog"; *IBM Technical Report*, CCR045

- [23] Fu, K. S.; "Syntactic Pattern Recognition, Applications"; *Springer Verlag*, 1977
- [24] Sotomayor, B. O. A. B.; "Reconhecimento de Padrões para Aplicações em CAD/CAM"; *Tese de Mestrado*, FEE - UNICAMP, Dezembro de 1984
- [25] Cohen, P. R. e Feigenbaum, E. A.; "The Handbook of Artificial Intelligence, Vol. III"; *Addison Wesley*, 1982
- [26] Usategui, J. M. A. e Madrigal, R. I.; "Visión Artificial por Computador"; *Ed. Paraninfo*, Madrid, 1986
- [27] Shiozaki, A.; "Edge Extraction using Entropy Operator"; *Computer Vision, Graphics and Image Processing*, Vol. 36, páginas 1 até 9, 1986
- [28] Davis, L. S.; "A Survey of Edge Detection Techniques"; *Computer Vision, Graphics and Image Processing*, Vol. 4, páginas 248 até 270, 1975
- [29] Naccache, N. J. e Shinghal, R.; "SPTA - A Proposed Algorithm for Thinning Binary Patterns"; *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-14, No. 3, May/June 1984, páginas 409 até 418
- [30] Lee, E. T.; "Applications of Fuzzy Set Theory to Image Sciences"; *Journal of Cybernetics*, Vol. 10, No. 1-3, páginas 127 até 136, 1980
- [31] Zadeh, L. A.; "Fuzzy Sets"; *Information and Control*, Vol. 8, 1965, páginas 338 até 353
- [32] Sanfeliu, A. e Fu, K. S.; "A Distance Measure Between Attributed Relational Graphs for Pattern Recognition"; *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, No. 3, May/June 1983, páginas 353 até 362
- [33] Sanfeliu, A.; "A Distance Measure Based on Tree Graph Grammars: a Way of Recognizing Hidden and Deformed 3D Complex Objects"; *Instituto de Cibernética - Diagonal 647*, Barcelona 28, Espanha

- [34] Zhang, T. Y. e Suen, C. Y.; "A Fast Parallel Algorithm for Thinning Digital Patterns"; *Communications of the ACM*, Vol. 27, No. 3, March 1984, páginas 236 até 239
- [35] Martínez-Pérez, M. P. et alii; "A Thinning Algorithm Based on Contours"; *Computer Vision, Graphics and Image Processing*, Vol. 39, páginas 186 até 201, 1987
- [36] Chin, R. T. et alii; "A One Pass Thinning Algorithm and its Parallel Implementation"; *Computer Vision, Graphics and Image Processing*, Vol. 40, páginas 30 até 40, 1987
- [37] Bezdek, J.; "Pattern Recognition with Fuzzy Objective Function Algorithms"; *Plenum Press*, 1981
- [38] Rosenfeld, A. e Davis, L.; "A Note on Thinning"; *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-6, No. 3, March 1976, páginas 226 até 228
- [39] Bins, L. S.; "Uso de Casamento Estrutural para Registro de Imagens de Satélite"; *Tese de Mestrado*, 26 de Julho de 1988, INPE, São José dos Campos, SP
- [40] Huntsberger, T. L. et alii; "Representation of Uncertainty in Computer Vision Using Fuzzy Sets"; *IEEE Transaction on Computers*, Vol. C-35, No. 2, Feb. 1986, páginas 145 até 156
- [41] Feng, H. F. e Pavlidis, T.; "Decomposition of Polygons into Simpler Components: Feature Generation for Syntactic Pattern Recognition"; *IEEE Transactions on Computers*, Vol. C-24, No. 6, June 1975, páginas 636 até 650
- [42] Pavlidis, T. e Ali. F.; "Computer Recognition of Handwritten Numerals by Polygonal Approximation"; *IEEE Transactions on System, Men and Cybernetics*, Vol. SMC-5, No. 6, Nov. 1975, páginas 610 até 614

- [43] Pavlidis, T.; "Structural Pattern Recognition"; *Springer Verlag*, 1977
- [44] Govindan, V. K. e Shivaprasad, A. P.; "A Pattern Adaptive Thinning Algorithm"; *Pattern Recognition*, Vol. 20, No. 6, páginas 623 até 637, 1987
- [45] Rao, A. R. e Jain, R.; "Knowledge Representation and Control in Computer Vision Systems"; *IEEE Expert*, Spring 1988, páginas 64 até 79
- [46] Lee, E. T.; "Fuzzy Tree Automata and Syntactic Pattern Recognition"; *IEEE Transactions on Pattern Anal. and Machine Intell.*, Vol. PAMI-4, No. 4, July 1982, páginas 445 até 449
- [47] Narasimhan, R.; "On the Description, Generation and Recognition of Classes of Pictures"; *Automatic Interpretation and Classification of Images*, A. Grasselli, Academic Press, 1969, páginas 1 até 42
- [48] Shaw, A. C.; "A Formal Picture Description Scheme as a Basis for Picture Processing Systems"; *Information and Control*, Vol. 14, 1969, páginas 9 até 52
- [49] Shaw, A. C.; "Parsing of Graph-representable Pictures"; *Journal of the ACM*, Vol. 17, No. 3, July 1970, páginas 453 até 481
- [50] Fu, K. S.; "Syntactic (Linguistic) Pattern Recognition"; *Digital Pattern Recognition*; K. S. Fu; Springer Verlag, 1980; páginas 95 até 134
- [51] Lee, E. T.; "Proximity Measures for the Classification of Geometric Figures"; *Journal of Cybernetics*, Vol. 2, No. 4, Oct-Dec 1972, páginas 43 até 59
- [52] Greanias, E. C. et alii; "The Recognition of Handwritten Numerals by Contour Analysis"; *IBM Journal of Research and Development*, Vol. 7, No. 1, páginas 14 até 22, 1963
- [53] Kandel, A.; "Fuzzy Mathematical Techniques with Applications"; *Addison Wesley*, 1986

- [54] van der Helm, P. A. et alii; "Avoiding Explosive Search in Automatic Selection of simplest Pattern Codes"; *Pattern Recognition*, Vol. 19, No. 2, páginas 181 até 191, 1986
- [55] Fu, K. S. et alii; "A Survey on Image Segmentation"; *Pattern Recognition*, Vol. 13, No. 1, 1981, páginas 3 até 16
- [56] Chang, S. S. L.; "On a Fuzzy Algorithm and its Implementation"; *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-8, No. 1, January 1978, páginas 31 até 32
- [57] Zadeh, L. A.; "Fuzzy Sets as a Basis for a Theory of Possibility"; *Fuzzy Sets and Systems*, Vol. 1, 1978, páginas 3 até 28
- [58] Sahoo, P. K. et alii; "A Survey of Thresholding Techniques"; *Computer Vision, Graphics and Image Processing*, Vol. 41, páginas 233 até 260, 1988
- [59] Mc Lean, G. F. et alii; "Hierarchical Edge Detection"; *Computer Vision, Graphics and Image Processing*, Vol. 44, páginas 350 até 366, 1988
- [60] Giorno, F. A. de C. et alii; "Programação em Lógica"; *V Escola de Computação*, SBC, Belo Horizonte, 1986
- [61] Ruiz, E. E. S; "Comparação de Técnicas e Métodos para Visão Computacional em Ambientes Industriais"; *Tese de Mestrado*, FEE - UNICAMP, 05 de Maio de 1989
- [62] Bratko, I.; "PROLOG Programming for Artificial Intelligence"; *Addison Wesley*, 1986
- [63] Silva, M. e Roy. A.; "Inteligencia Artificial y Robotica Industrial"; *Universidad de Zaragoza*, 1983
- [64] Bhanu, B.; "CAD-Based Robot Vision"; *IEEE Computer*, Vol. 20, No. 8, August 1987, páginas 12 até 16

- [65] Bhanu, B. e Ho, C. C.; "CAD-Based 3D Objetc Representation for Robot Vision"; *IEEE Computer*, Vol. 20, No. 8, August 1987, páginas 19 até 35
- [66] Wong, A. K. C.; "Structural Pattern Recognition: a Random Graph Approach"; *Pattern Recognition Theory and Applications*, Pierre Devijver e Josef Kittler; Springer Verlag, 1987, páginas 323 até 345
- [67] Freeman, H.; "On the Encoding of Arbitrary Geometric Configurations"; *Pattern Recognition: Introduction and Foundations*, Jack Sklansky; Dowden, Hutchinson and Ross, 1973, páginas 257 até 265
- [68] Dinneen, G. P.; "Programming Pattern Recognition"; *Pattern Recognition: Introduction and Foundations*, Jack Sklansky; Dowden, Hutchinson and Ross, 1973, páginas 250 até 256
- [69] Bunke, H.; "Hybrid Methods in Pattern Recognition"; *Pattern Recognition Theory and Applications*, Pierre Devijver e Josef Kittler; Springer Verlag, 1987, páginas 367 até 382
- [70] Thomason, M.; "Structural Methods in Pattern Recognition"; *Pattern Recognition Theory and Applications*, Pierre Devijver e Josef Kittler; Springer Verlag, 1987, páginas 307 até 321
- [71] Zimmermann, H. J.; "Fuzzy Sets in Pattern Recognition"; *Pattern Recognition Theory and Applications*, Pierre Devijver e Josef Kittler; Springer Verlag, 1987, páginas 383 até 391
- [72] Chatterji, B. N.; "Character Recognition using Fuzzy Similarity Relations"; *Approximate Reasoning in Decision Analysis*, Madam M. Gupta e E. Sanchez; Amsterdam, New York, Oxford, 1982, páginas 131 até 138
- [73] Blesser, B. A. et alii; "Empirical Tests for Feature Selection based on a Psychological Theory of Character Recognition"; *Pattern Recognition*, Vol. 8, No. 2, April 1976, páginas 77 até 85
- [74] Gonzalez, R. C. e Wintz, P; "Digital Image Processing"; *Addison Wesley*, 1987

- [75] Toussaint, G. T.; "The Use of Context in Pattern Recognition"; *Pattern Recognition*, Vol. 10, No. 3, 1978 (Special Issue), páginas 189 até 204
- [76] Duda, R. O.; "Elements of Pattern Recognition"; *Adaptive, Learning and Pattern Recognition Systems*; K. S. Fu e J. M. Mendel; Academic Press, 1970; páginas 3 até 33
- [77] Viglione, S. S.; "Applications of Pattern Recognition Technology"; *Adaptive, Learning and Pattern Recognition Systems*; K. S. Fu e J. M. Mendel; Academic Press, 1970; páginas 115 até 162
- [78] Rosenfeld, A. et alii; "Recent Developments in Digital Pattern Recognition"; *Digital Pattern Recognition*, K. S. Fu; Springer Verlag, 1980; páginas 205 até 232
- [79] Clowes, M. B.; "Transformational Grammars and the Organization of Pictures"; *Automatic Interpretation and Classification of Images*, A. Grasselli, Academic Press, 1969, páginas 43 até 77
- [80] "Character Recognition"; *The British Computer Society*, 1967
- [81] Allan, R. "Machine Vision"; *Electronic Design*, November 14, 1985, páginas 106 até 116
- [82] Troxel, D. E.; "Feature Selection for Low Error Rate OCR"; *Pattern Recognition*, Vol. 8, No. 2, April 1976, páginas 73 até 76
- [83] Rosenfeld, A. et alii; "DIALOG: Expert Vision Systems"; *Computer Vision, Graphics and Image Processing*; Vol. 34, 1986, páginas 99 até 117
- [84] Lucena, C. P.; "Inteligência Artificial e Engenharia de Software"; *JPE Editores*, 1987
- [85] Zadeh, L. A.; "Making Computers Think like People"; *IEEE Spectrum*, August 1984, páginas 26 até 32

- [86] Duff, M. J. e Leviardi, S.; "Languages and Architectures for Image Processing"; *Academic Press*, 1981
- [87] Letwin, G.; "Explorando o OS/2"; *Microsoft Press*, 1988
- [88] Kirsch, R.; "Computer Interpretation of English Text and Picture Patterns"; *IEEE Transactions on Electronic Computers*; Vol. EC-13, 1964, páginas 363 até 376
- [89] Post, E. L.; "Introduction to a General Theory of Elementary Propositions"; *American J. Math.* 43, 1921, páginas 163 até 185
- [90] Lukaciewicz, J.; "Logike Trojwartoscieweg"; *Ruch. Filozoficzne* 169, 1920
- [91] Zadeh, L. A.; "Fuzzy Logic"; *IEEE Computer*, April 1988, páginas 83 até 93
- [92] Cai, Z.; "Restoration of Binary Images Using Contour Direction Chain Codes Description"; *Computer Vision, Graphics and Image Processing*, Vol. 41, páginas 101 até 106, 1988
- [93] Basu, S. et alii; "Image Segmentatin by Syntactic Methods"; *Pattern Recognition*, Vol. 20, No. 1, páginas 33 até 44, 1987
- [94] Kim, S.-D. et alii; "A New Chain-Coding Algorithm for Binary Images Using Run-Length Codes"; *Computer Vision, Graphics and Image Processing*, Vol. 41, páginas 114 até 128, 1988
- [95] Fournier, A. et alii; "On the Power of the Frame Buffer"; *ACM Transactions on Graphics*, Vol. 7, No. 2, páginas 103 até 128, April 1988
- [96] Veenstra, J. et alii; "Line Drawings of Octree-Represented Objects"; *ACM Transactions on Graphics*, Vol. 7, No. 1, páginas 61 até 75, January 1988
- [97] Scheifler, R. W. et alii; "The X Window System"; *ACM Transactions on Graphics*, Vol. 5, No. 2, páginas 79 até 109, April 1986

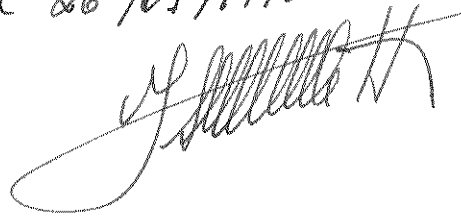
RECONHECIMENTO DE CARACTERES ALFANUMÉRICOS
ATRAVÉS DE TÉCNICAS DE LÓGICA NEBULOSA

VOLUME II

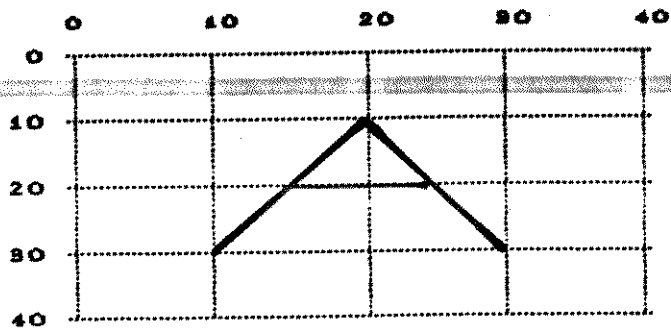
THOMAS DE ARAUJO BUCK

ORIENTADOR: MÁRCIO LUZ DE ANDRADE NETTO

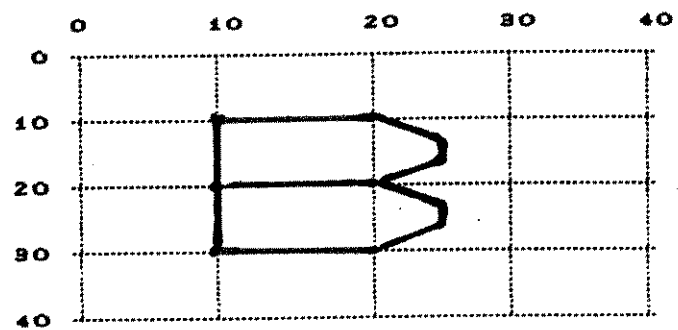
Este exemplar corresponde à redação final da Tese defendida por Thomas de Araújo Buck e aprovada pela Banca Julgadora em 26/05/1990



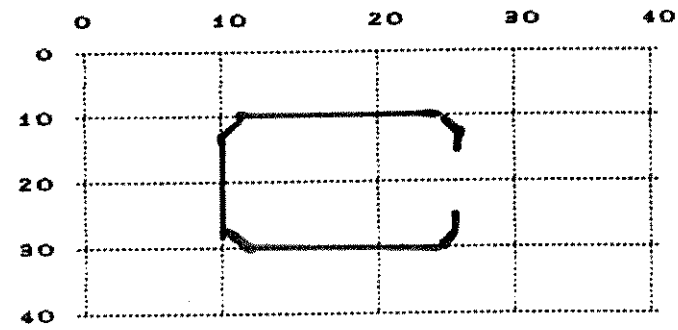
TESE APRESENTADA À FACULDADE DE ENGENHARIA
ELÉTRICA DA UNIVERSIDADE ESTADUAL DE CAMPINAS
- UNICAMP - COMO PARTE DOS REQUISITOS EXIGIDOS
PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM
ENGENHARIA ELÉTRICA.



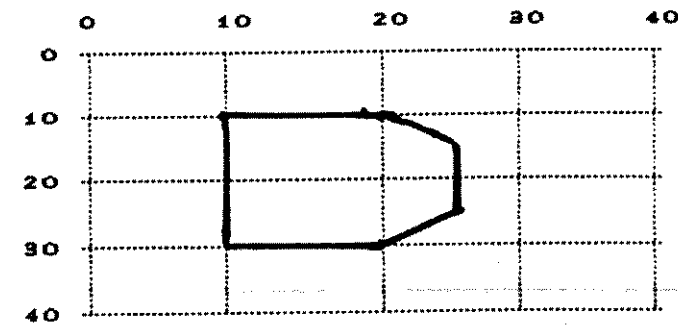
$$((d2 + ((d2 + g2) * h2)) + g2)$$



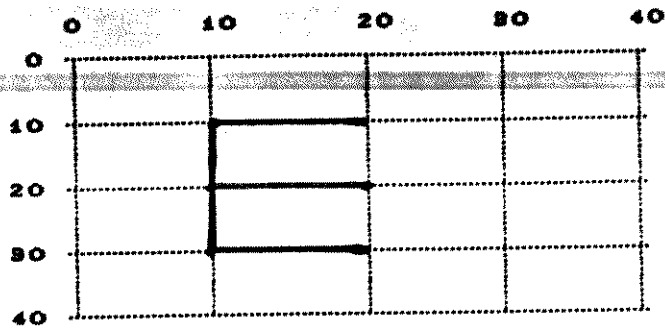
$$((v2 + ((v2 + h2 + g1 + (-(d1 + v1)))) * h2) + g1 + (-(d1 + v1))) * h2)$$



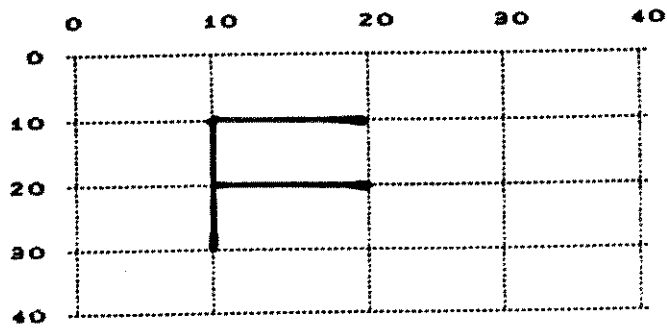
$$((-g1 + v2 + d1 + h1 + g1 + -v1) * (h1 + (d1 + v1)))$$



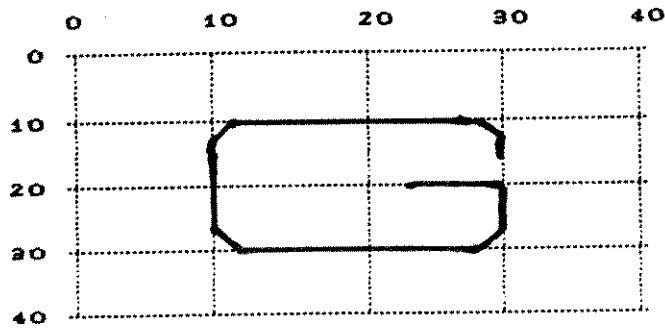
$$(h2 * (v3 + h2 + g1 + (-(d1 + v2))))$$



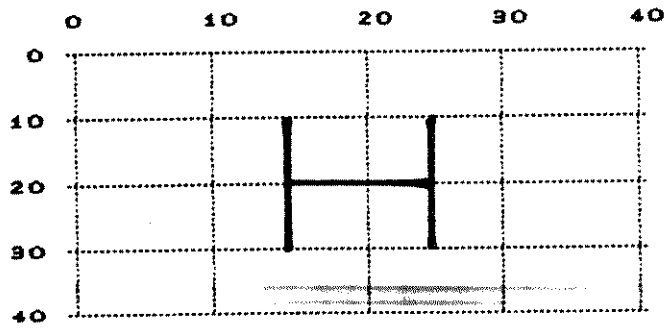
$$((v_2 + ((v_2 + h_2) \times h_2)) \times h_2)$$



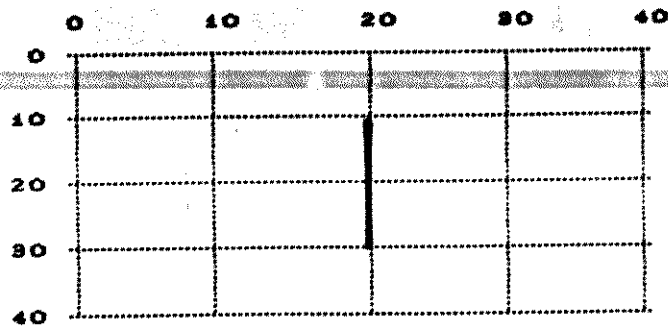
$$(v_2 + ((v_2 + h_2) \times h_2))$$



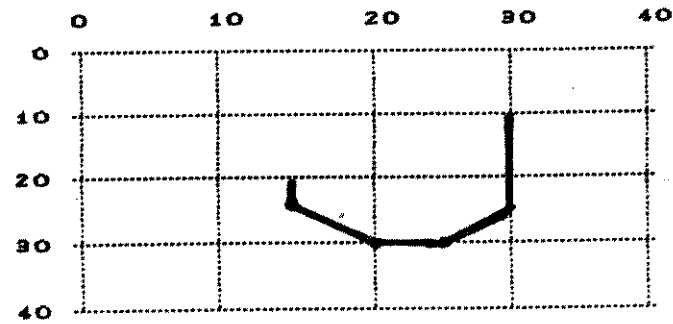
$$((-g_1 + v_2 + d_1 + h_1 + g_1 + -v_1) \times (h_1 + (d_1 + v_1 - h_1)))$$



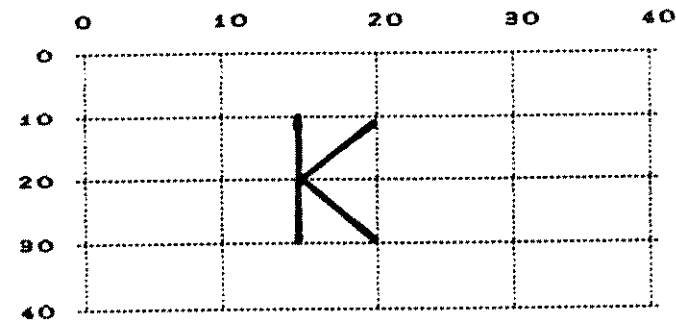
$$(v_2 + (v_2 \times (h_2 + (v_2 \times -v_2))))$$



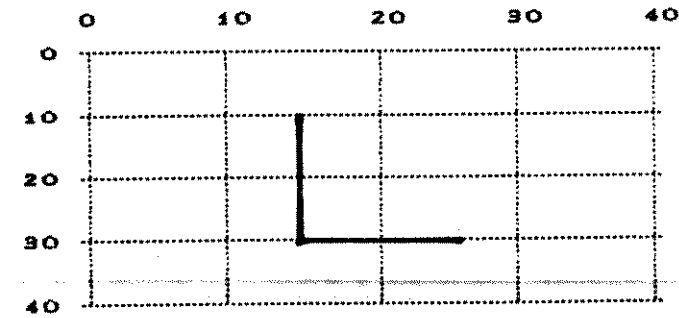
$$(v3 \times \lambda)$$



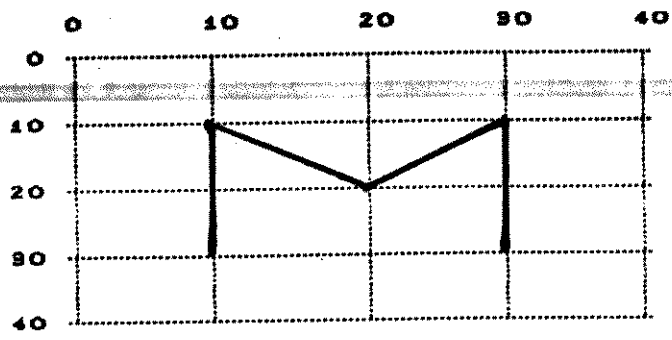
$$(((-g1 + v1) \times h1) + (d1 + v3))$$



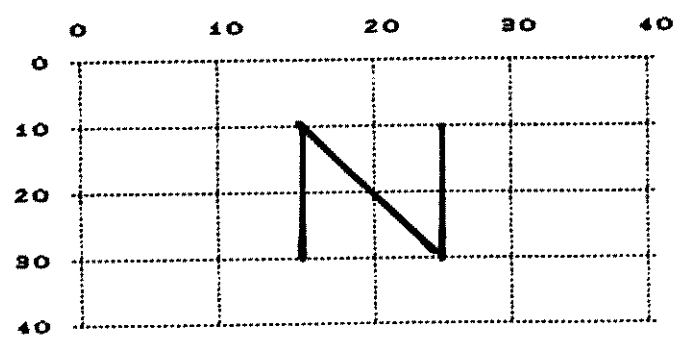
$$(v2 + (v2 \times d2 \times g2))$$



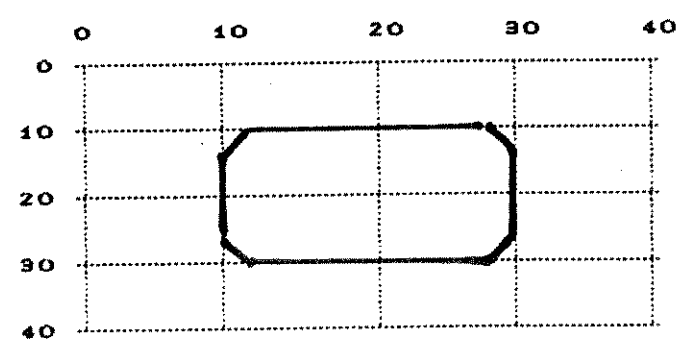
$$(v3 \times h2)$$



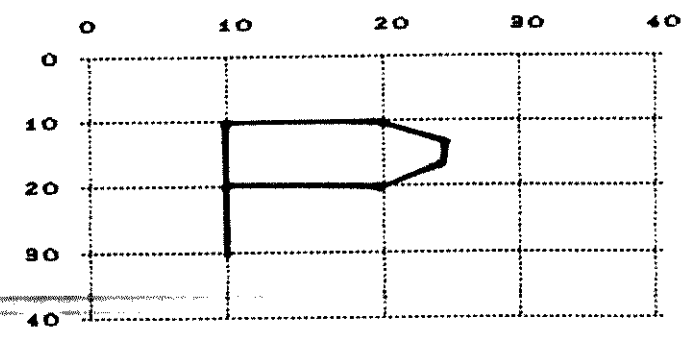
$$(v3+g3+d3+(-v3))$$



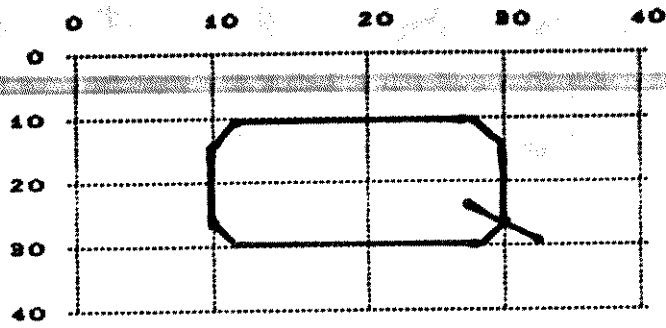
$$(v3+(g3+v3))$$



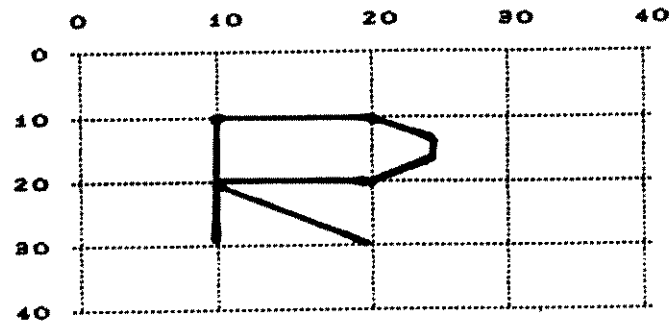
$$(h1 * (-q1 + v2 + d1 + h1 + g1 + (-d1 + v2)))$$



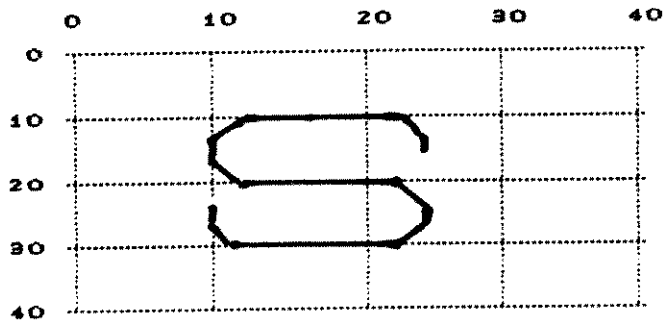
$$(v2 + ((v2 + h2 + g1 + (-d1 + v1))) * h2)$$



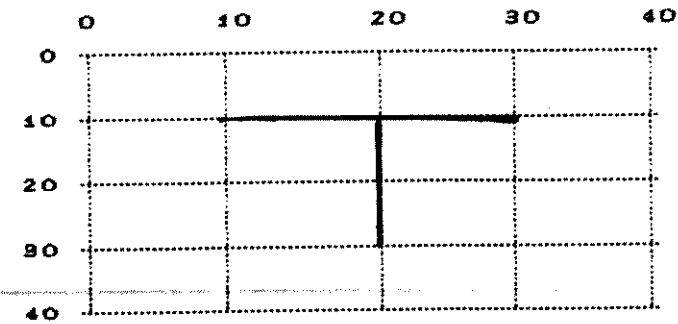
$$(h1 * (-g1 + v2 + d1 + h1 + g1 + (-(-d1 + (-g1 * xg1) + v2))))$$



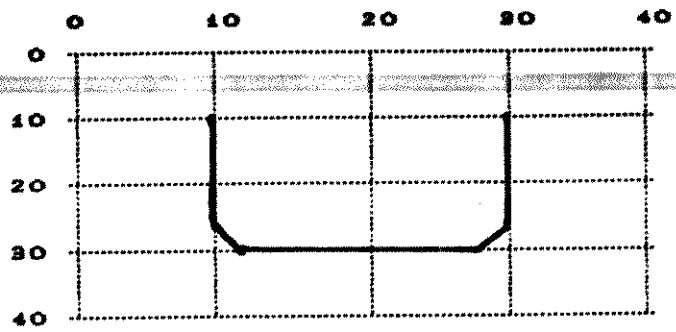
$$(v2 + (h2 * (v2 + h2 + g1 + (-(-d1 + v1)))) * xg2))$$



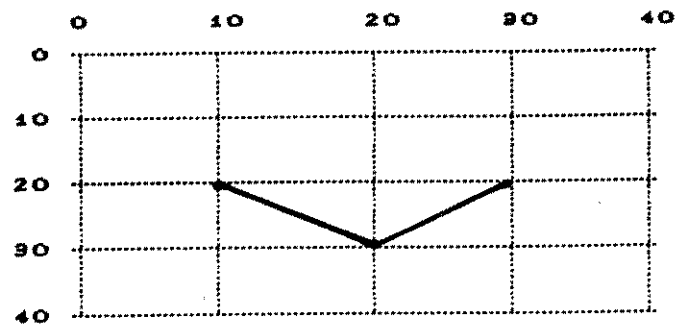
$$(((-g1 + v1) * xh1) + (d1 + v1 + (-(-g1 + h1 + g1))) + v1 + d1 + h1 + g1 + (-v1)))$$



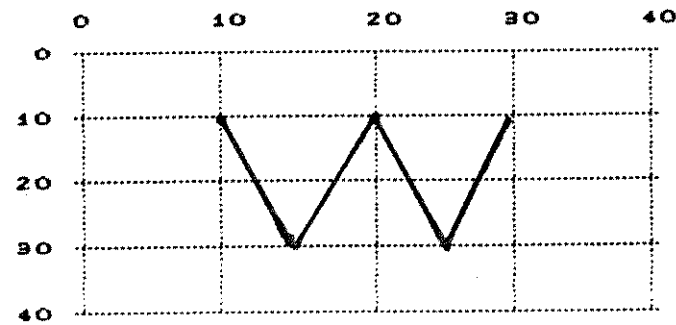
$$(v3 + (h1 * x - h1))$$



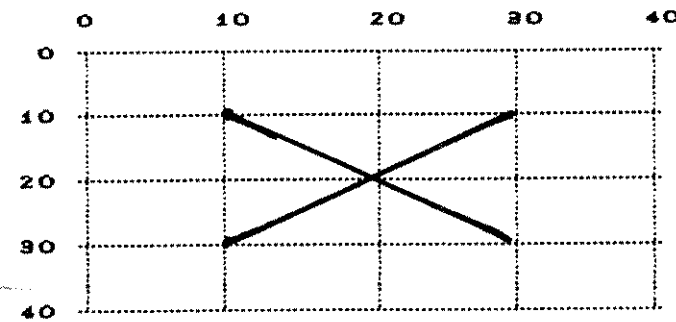
$$((-g_1 + v_3) \times (h_1 + (d_1 + v_3)))$$



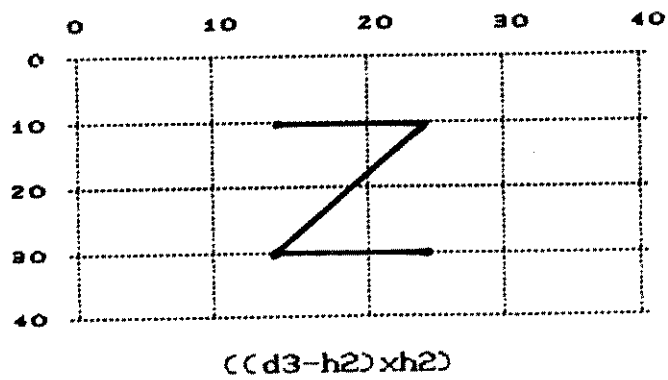
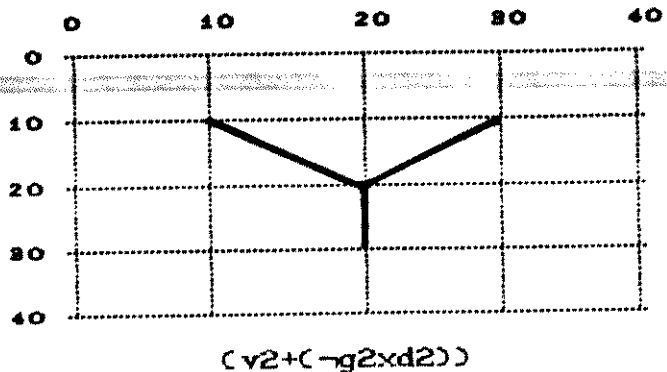
$$(-g_3 \times d_3)$$



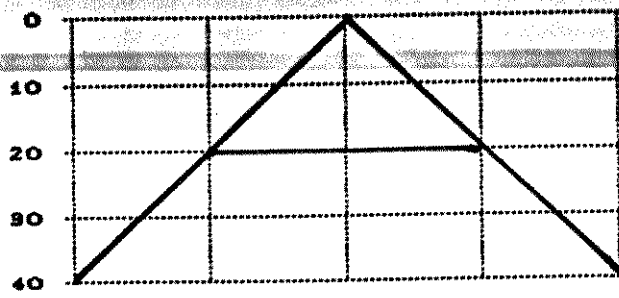
$$((-g_3 \times d_3) + g_3) + d_3$$



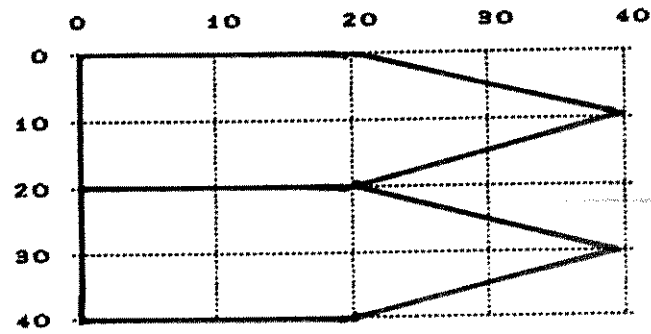
$$(d_2 + ((-g_2) \times d_2) \times g_2)$$



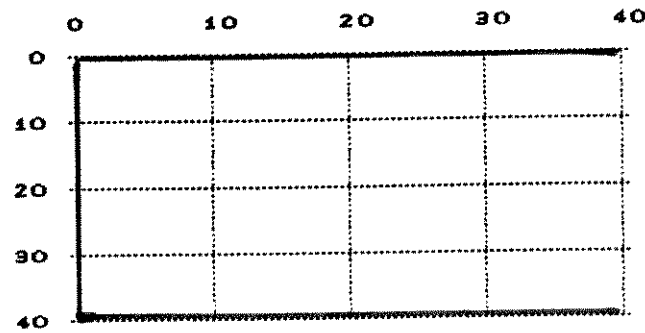
PADRÕES DOS CARACTERES ALFANUMÉRICOS FINAIS



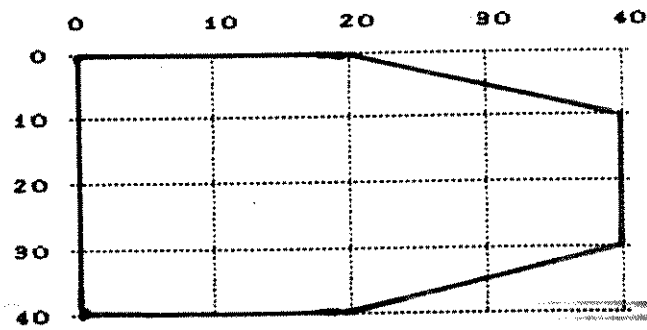
$$((d^2 + ((d^2 + g^2) * h^2)) + g^2)$$



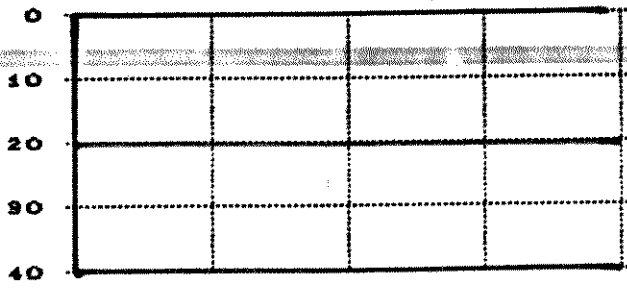
$$((v^2 + (((v^2 + h^2) + g^2) + -d^2) * h^2)) * (h^2 + d^2) + -g^2))$$



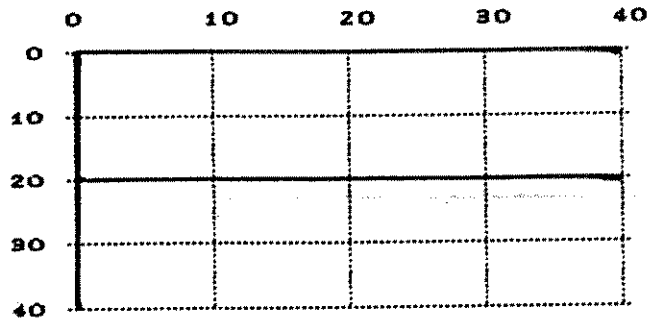
$$(((v^2 + v^2) + h^2) + h^2) * (h^2 + h^2))$$



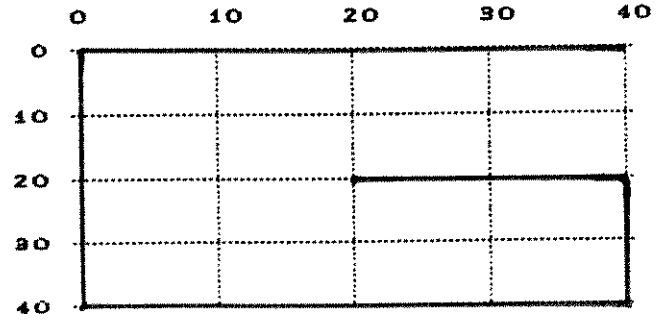
$$(((v^2 + v^2) + h^2) + g^2) * ((h^2 + d^2) + v^2))$$



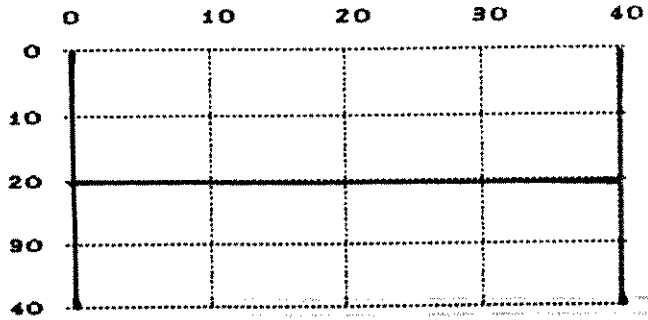
$$((v_2 + ((v_2 + h_2) + h_2) \times (h_2 + h_2))) \times (h_2 + h_2))$$



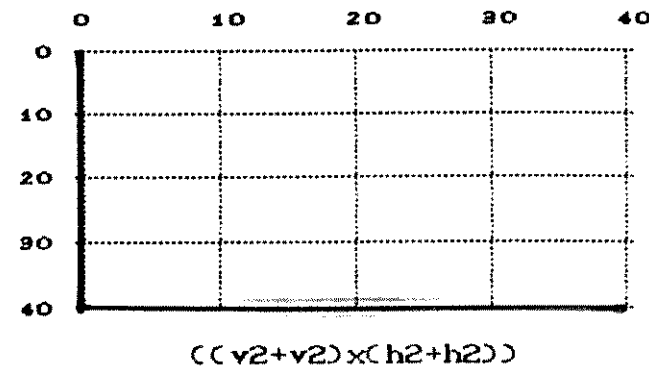
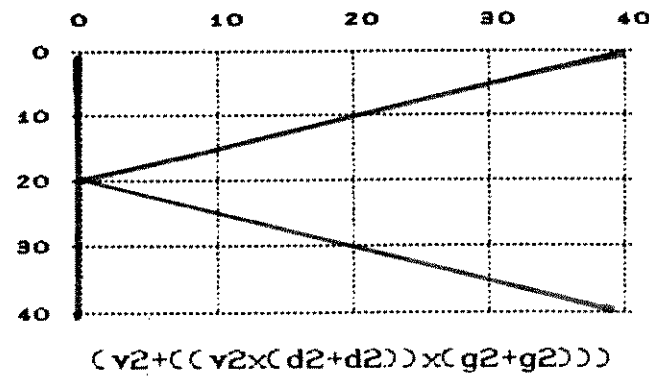
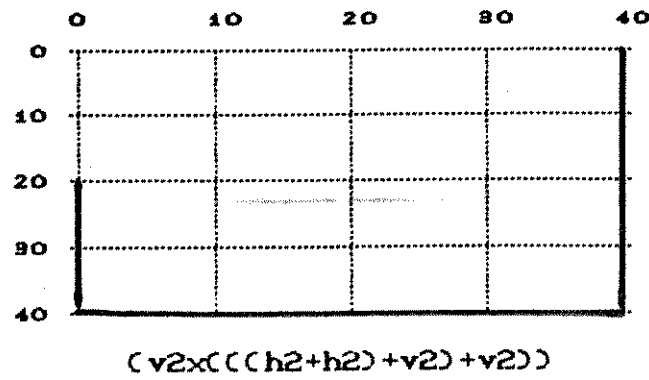
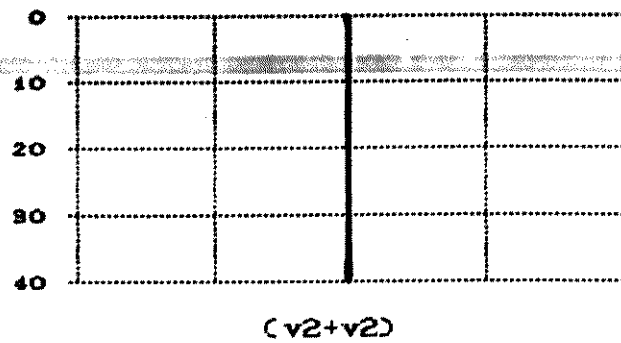
$$(v_2 + ((v_2 + h_2) + h_2) \times (h_2 + h_2))$$

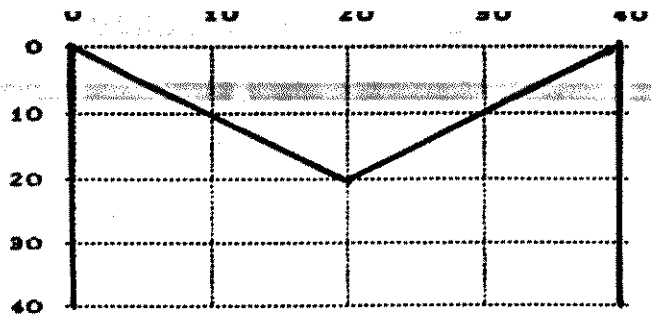


$$(((v_2 + v_2) + h_2) + h_2) \times ((h_2 + h_2) + v_2) + (-h_2)$$

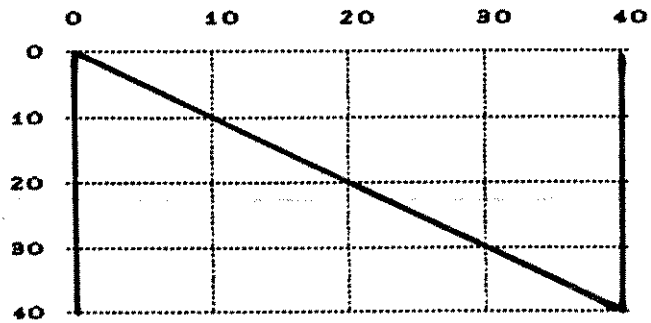


$$(v_2 + (v_2 \times (h_2 + h_2) + (v_2 \times -v_2)))$$

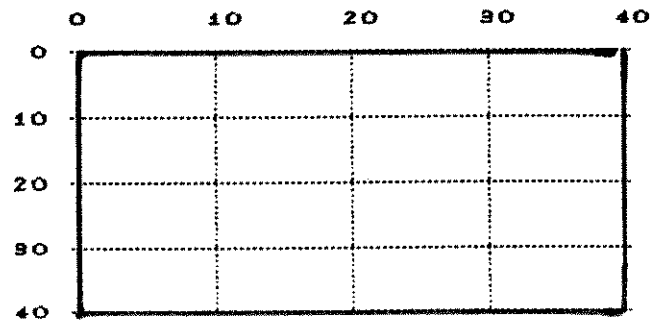




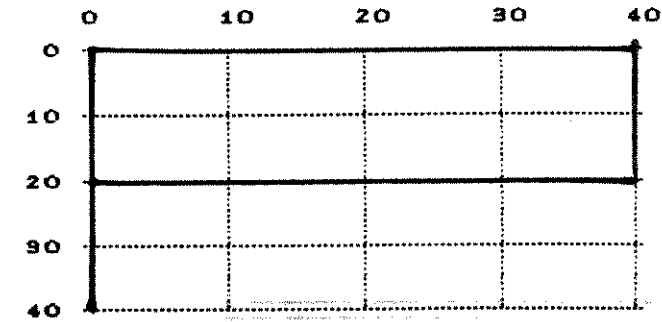
$$(((v^2 + v^2) + g^2) + d^2) + v^2 + v^2$$



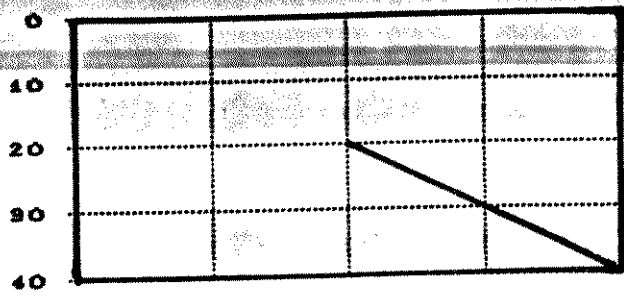
$$(((v^2 + v^2) + g^2) + g^2) + v^2 + v^2$$



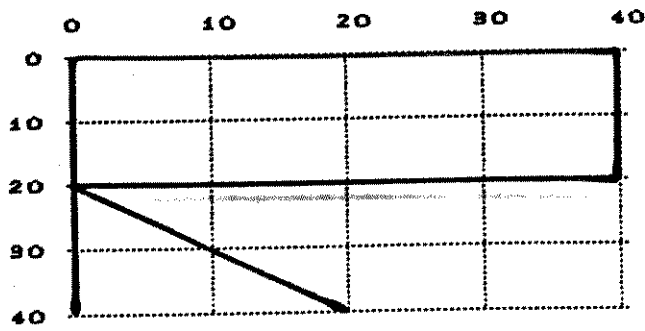
$$(((v^2 + v^2) + h^2) + h^2) * ((h^2 + h^2) + v^2) + v^2$$



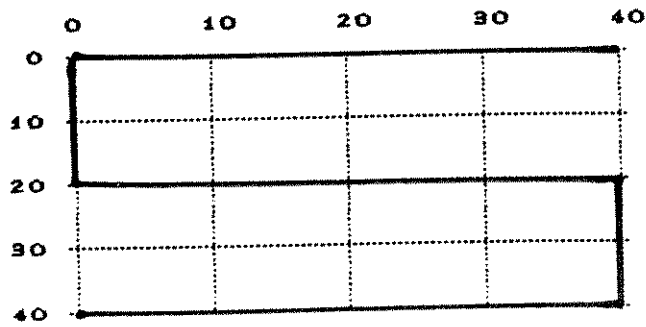
$$(v^2 + (((v^2 + h^2) + h^2) + h^2) * ((h^2 + h^2) + v^2))$$



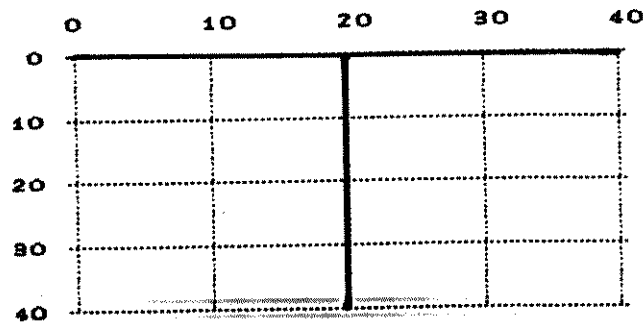
$$(((v^2+v^2)+h^2)+h^2) \times ((h^2+h^2)+(-g^2 \times (v^2+v^2)))$$



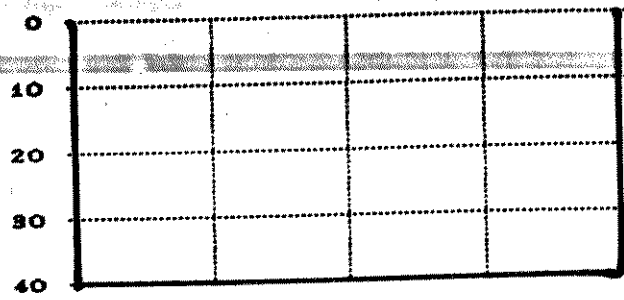
$$(v^2 + (((v^2+h^2)+h^2) \times ((h^2+h^2)+v^2)) \times g^2)$$



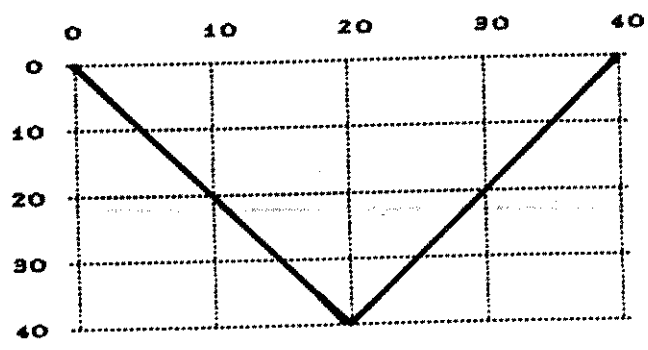
$$(((h^2+h^2)+v^2) - (((v^2+h^2)+h^2) \times (h^2+h^2)))$$



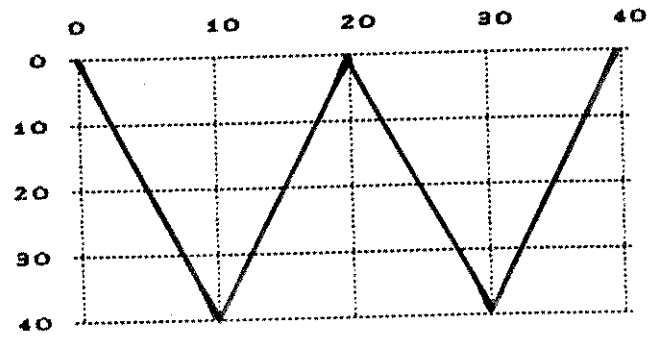
$$(v^2 + (v^2 + (h^2 + -h^2)))$$



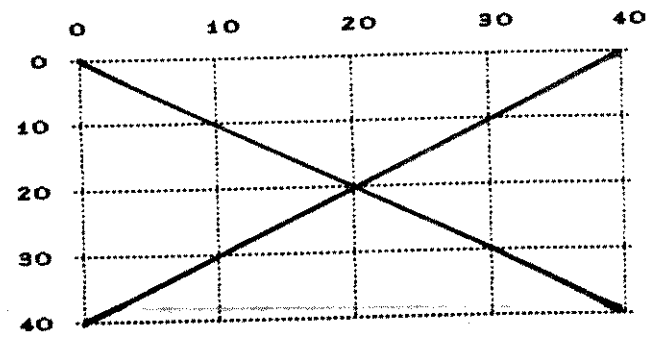
$$((v_2+v_2) \times ((h_2+h_2)+v_2)+v_2)$$



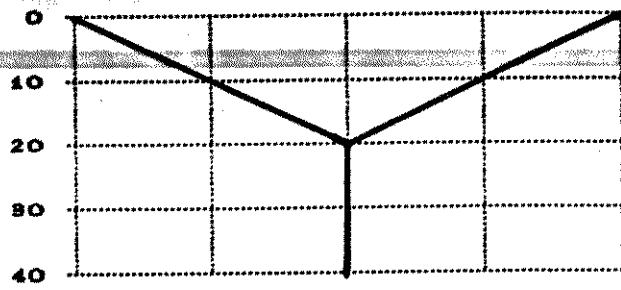
$$((g_2+g_2)+d_2)+d_2$$



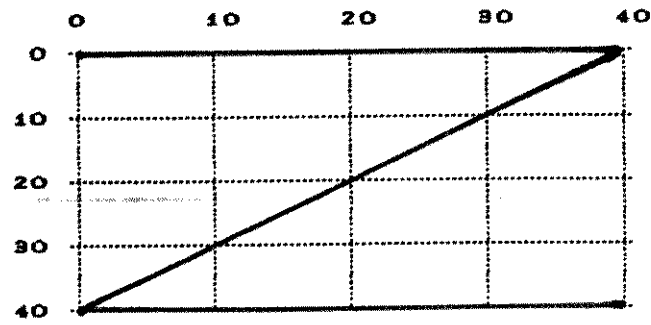
$$((((((g_2+g_2)+d_2)+d_2)+g_2)+g_2)+d_2)+d_2$$



$$(d_2+((-g_2 \times d_2) \times g_2))$$

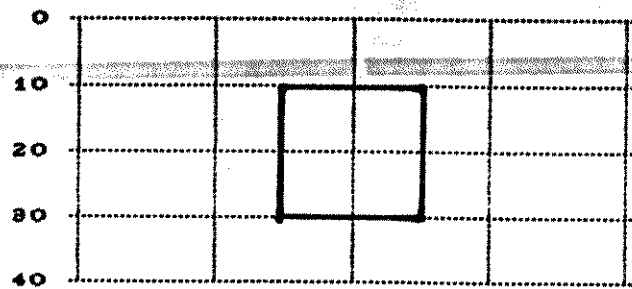


$$(v^2 + (-g^2 x d^2))$$

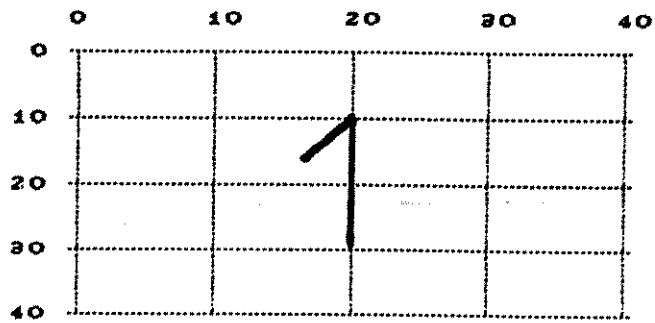


$$((d^2 + d^2) - (h^2 + h^2)) \times (h^2 + h^2)$$

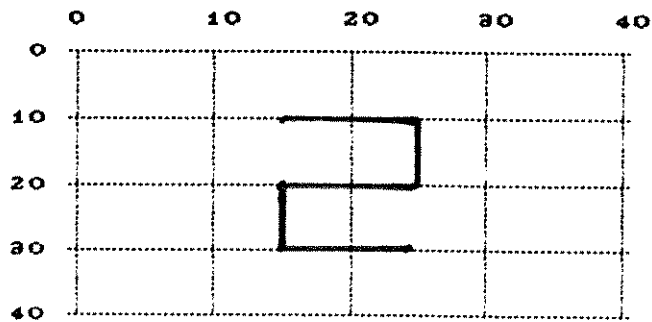
APENDICE C
PADRÕES DOS CARACTERES NUMÉRICOS



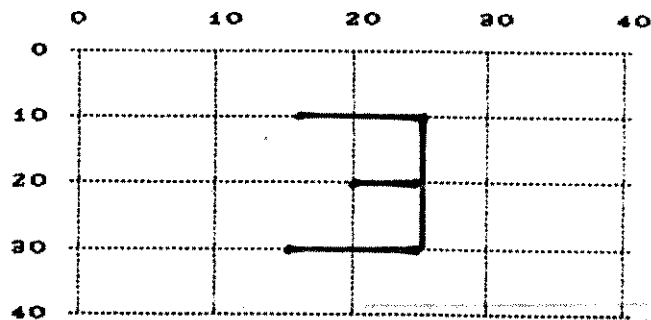
$$((h2+v3+v3) \times (v3+v3+h2))$$



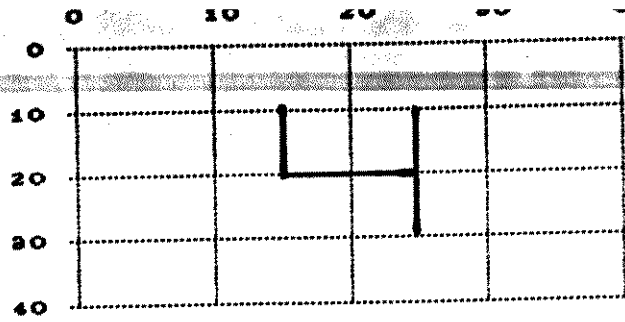
$$((v3+v3)+h1)$$



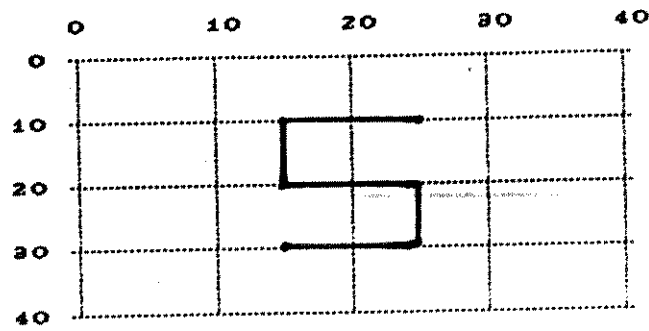
$$(h2 \times (v3+h2+v3+h2))$$



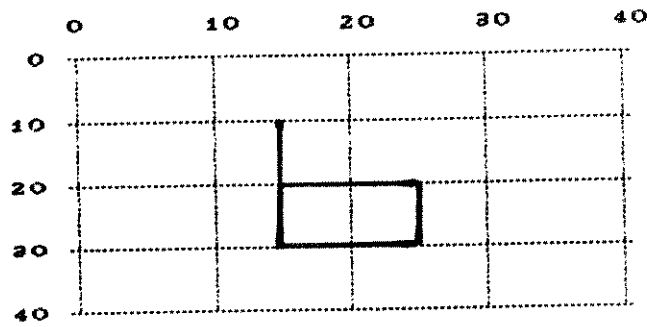
$$(h2+v3+((-h1) \times (v3+h2)))$$



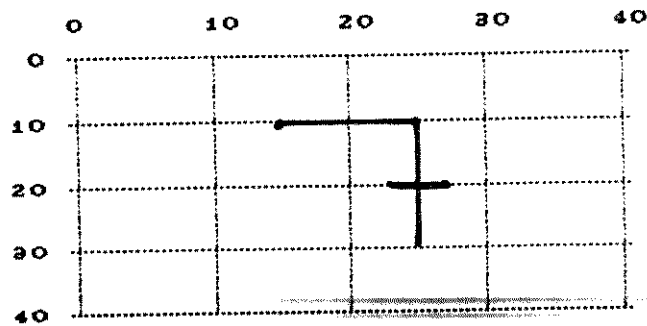
$$((v3 \times (h2 + (v3 + -v3))))$$



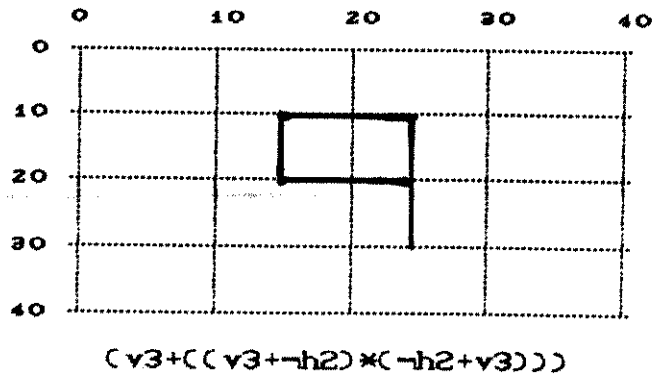
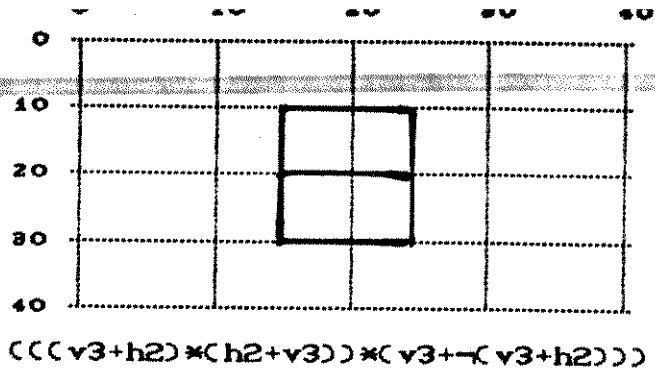
$$(h2 + v3 + (-h2) + v3 + h2)$$



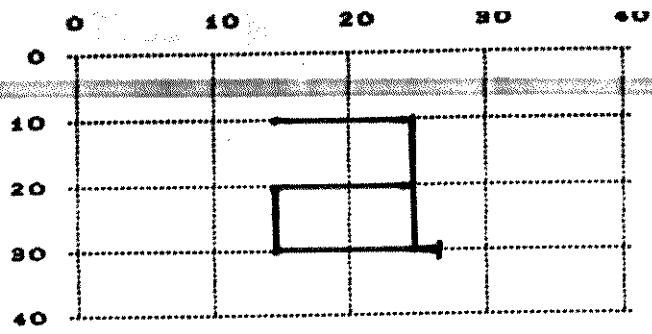
$$(((v3 \times (h2 + v3 + (-h2)))) + v3)$$



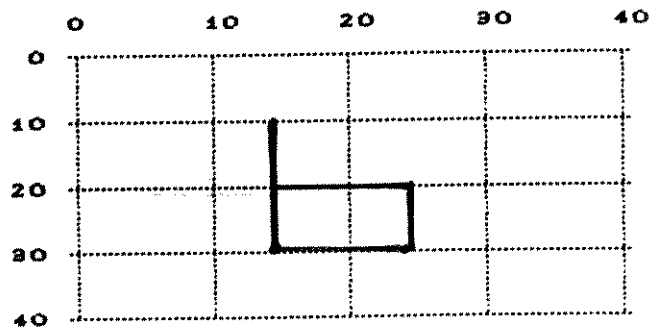
$$((v3 + (-h1 \times h1)) + v3 + -h2)$$



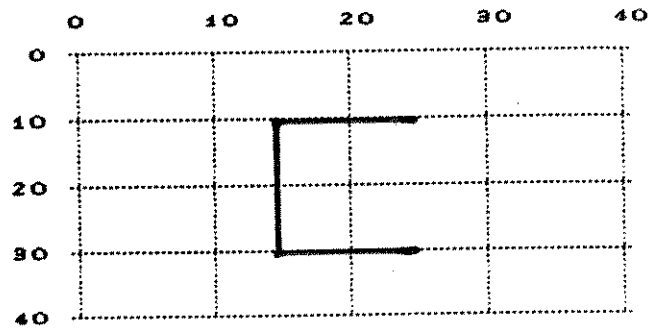
PADRÕES DOS CARACTERES ALFABÉTICOS MINÚSCULOS



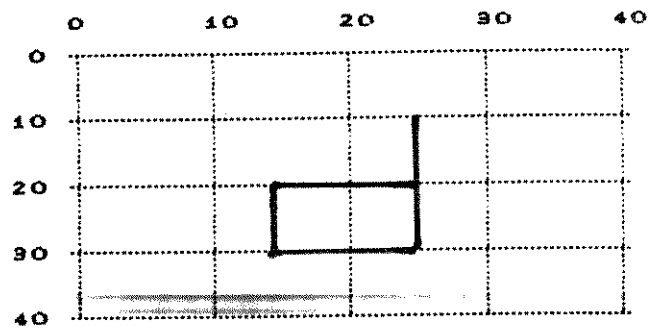
$$(((v3+h2) \times (h2+(h1 \times v3))) + v3 + -h2)$$



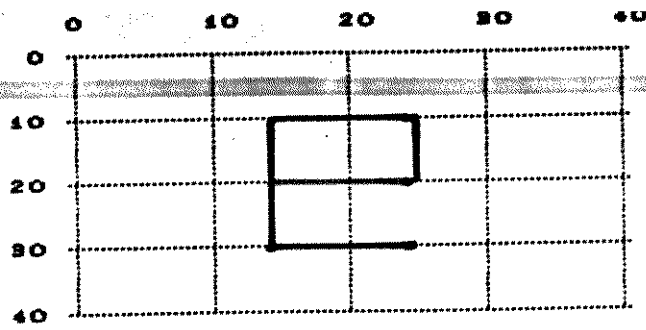
$$((v3 \times (h2+v3+-h2)) + v3)$$



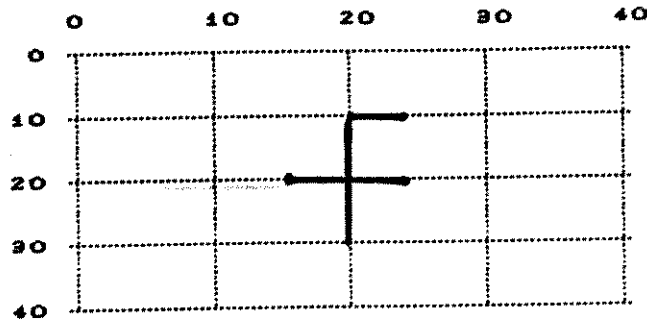
$$(h2 \times (v3+h2))$$



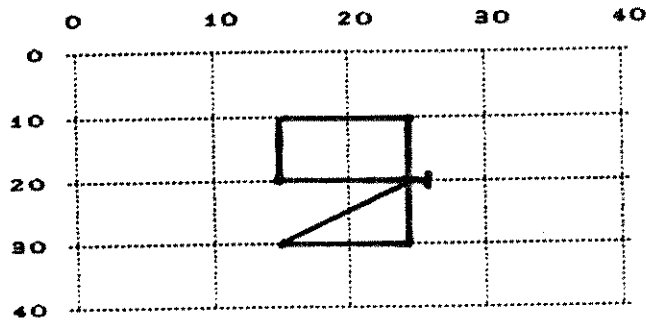
$$(((v3+h2) \times (h2+v3)) + v3)$$



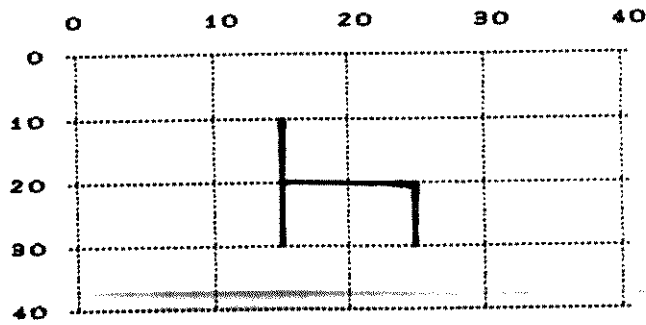
$$((v3+((v1+h2)*(h2+v1)))*h2)$$



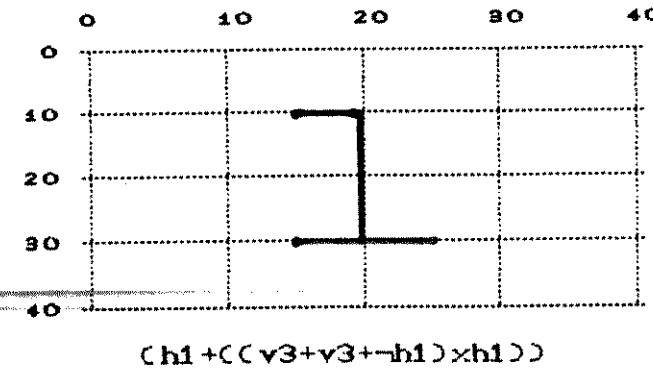
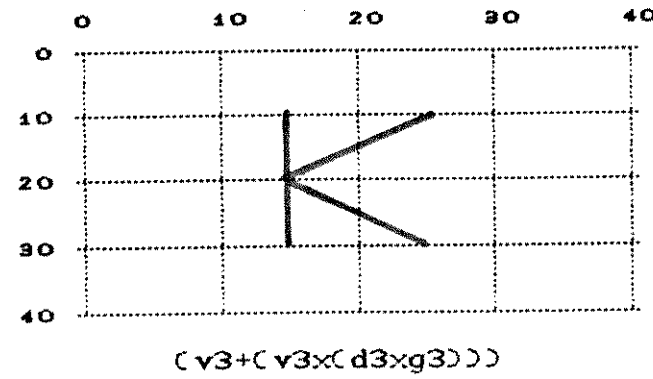
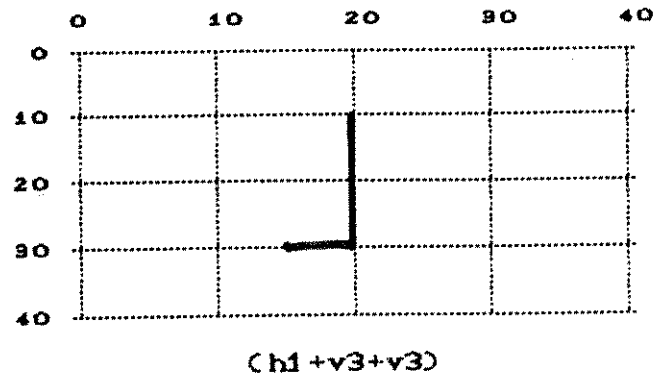
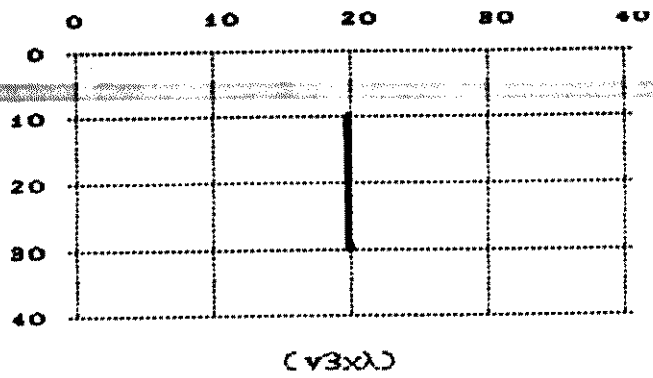
$$((v3+(h1x-h1))+v3+h1)$$

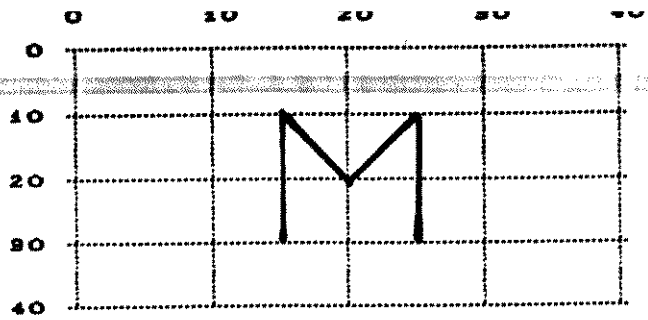


$$(((v3+h2+(-v3))*h2)*(-v3+-h2+d3))+h1)$$

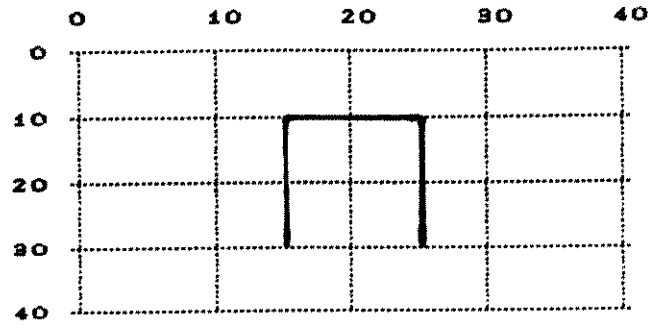


$$(v3+(v3x(h2+-v3)))$$

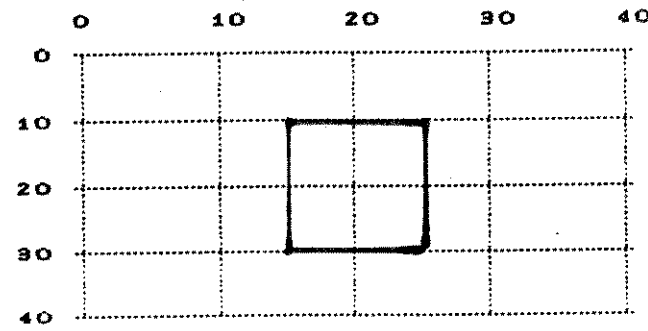




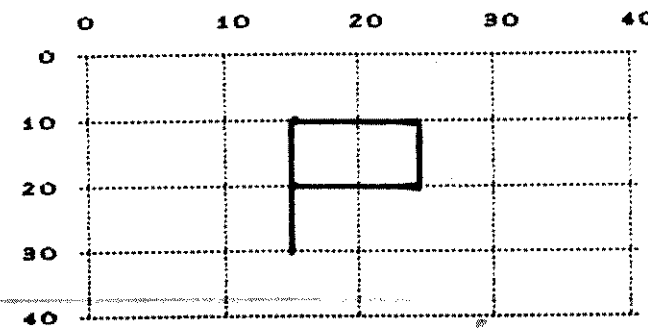
$$(v3+g3+d3+(-v3))$$



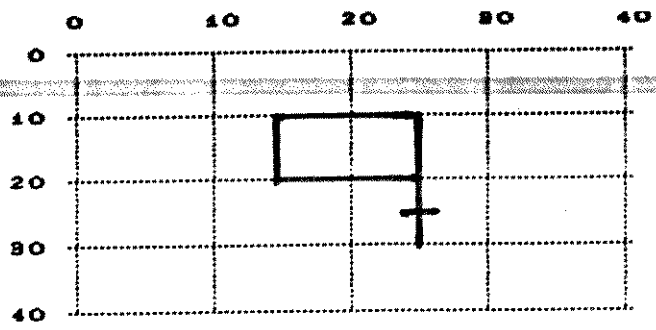
$$(v3+h2+(-v3))$$



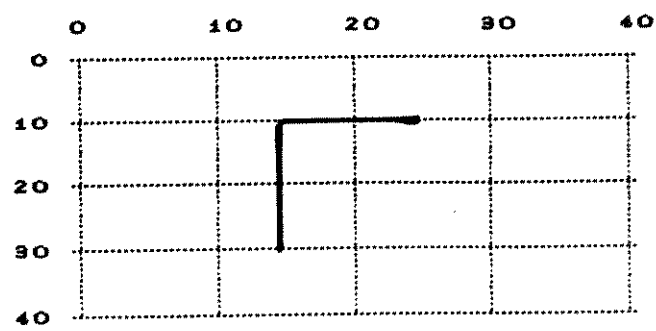
$$((v3+h2)*(h2+v3))$$



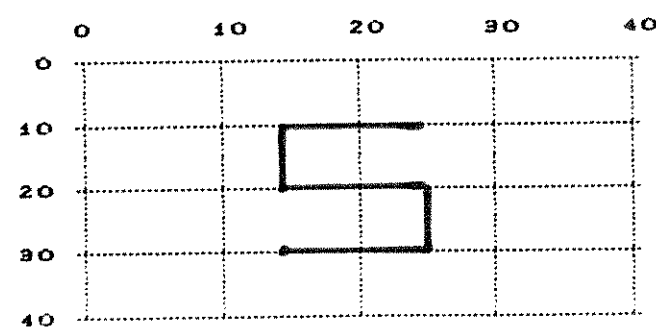
$$(v3+((v3+h2)*(h2+v3)))$$



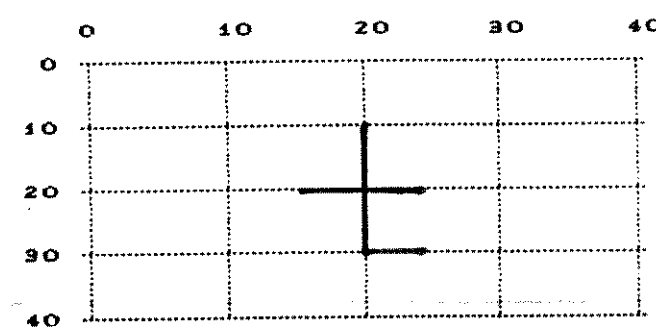
$$(((v3+h2+(-v3))*h2)-(v1+h1*(v1*-h1)))$$



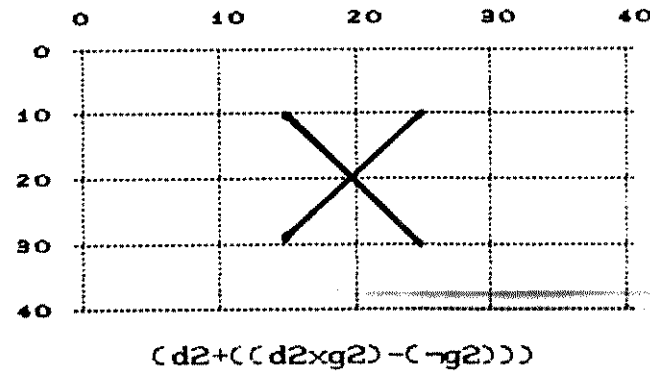
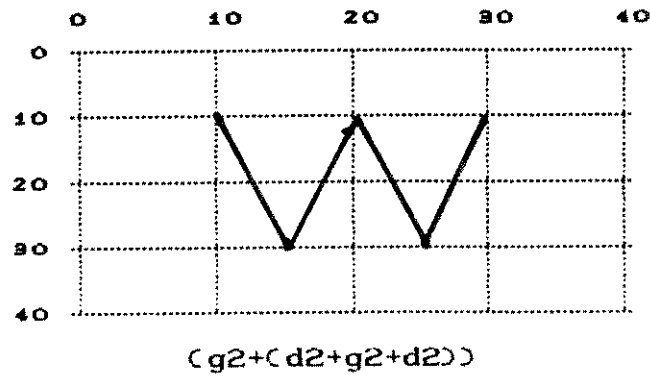
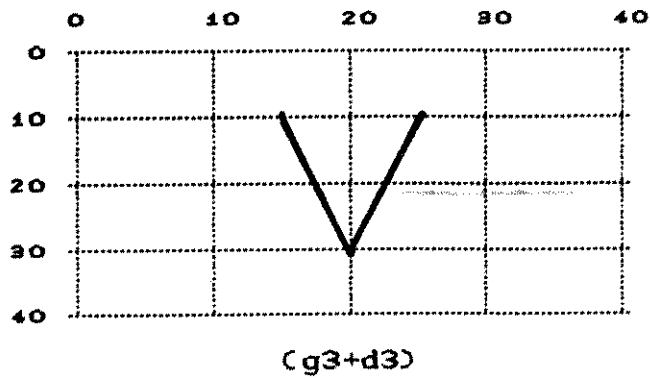
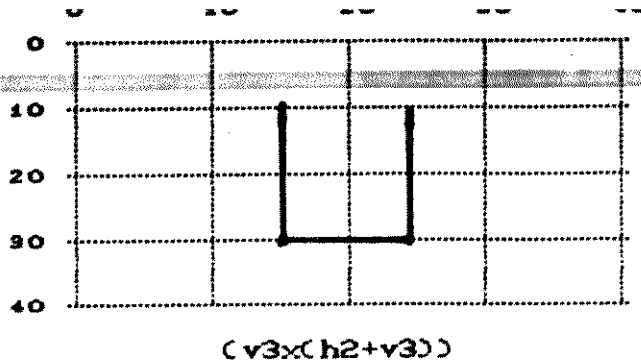
$$(v3+h2)$$

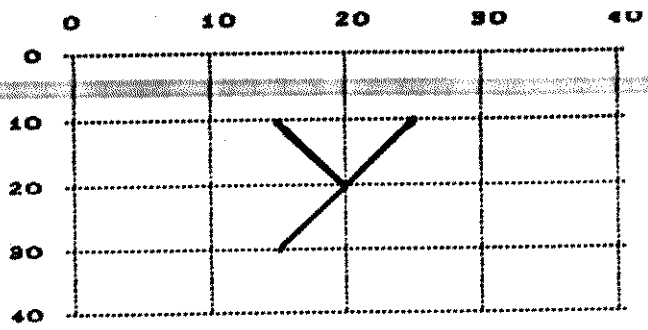


$$(h2+v3+(-h2)+v3+h2)$$

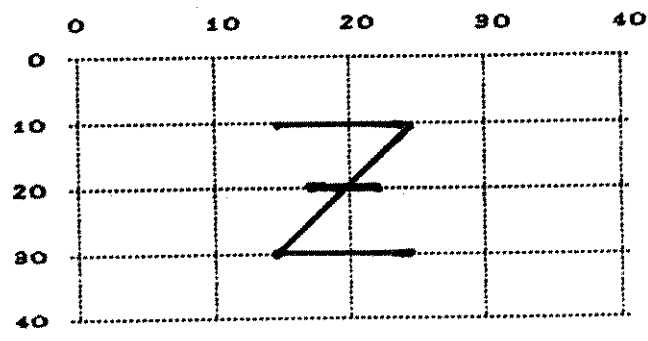


$$((v3+(h1*(v1*-h1))))*h1)$$





$$(d2+(d2x-g2))$$



$$(((d2-h2) \times h1) \times h1) - (d2 \times h2))$$

40, 40

Handwritten text in a grid pattern, consisting of multiple columns of characters.

40, 40

Handwritten text in a grid pattern, consisting of multiple columns of characters.

40, 40

Handwritten text in a grid pattern, consisting of multiple columns of characters.

40, 40

Handwritten text in a grid pattern, consisting of multiple columns of characters.

40, 40

Vertical text columns on the left page, containing repetitive characters and symbols.

40, 40

Vertical text columns on the right page, containing repetitive characters and symbols.

40, 40

Vertical text columns on the left page, containing repetitive characters and symbols.

40, 40

Vertical text columns on the right page, containing repetitive characters and symbols.

40, 40

Vertical text column on the left page, consisting of multiple lines of small, dense characters.

40, 40

Vertical text column on the right page, consisting of multiple lines of small, dense characters.

40, 40

Vertical text column on the left page, consisting of multiple lines of small, dense characters.

40, 40

Vertical text column on the right page, consisting of multiple lines of small, dense characters.

48, 48

Vertical text column on the left side of the page, containing multiple lines of text.

48, 48

Vertical text column on the right side of the page, containing multiple lines of text.

48, 48

Vertical text column on the left side of the page, containing multiple lines of text.

48, 48

Vertical text column on the right side of the page, containing multiple lines of text.

49, 49

Handwritten text in a grid format, consisting of multiple columns of characters.

49, 49

Handwritten text in a grid format, consisting of multiple columns of characters.

49, 49

Handwritten text in a grid format, consisting of multiple columns of characters.

49, 49

Handwritten text in a grid format, consisting of multiple columns of characters.

40, 40

Handwritten text on a dot grid background, appearing as a list of entries with varying indentations.

40, 40

Handwritten text on a dot grid background, appearing as a list of entries with varying indentations.

40, 40

Handwritten text on a dot grid background, appearing as a list of entries with varying indentations.

40, 40

Handwritten text on a dot grid background, appearing as a list of entries with varying indentations.

LETRAS A PARA TESTES

Handwritten text on a dotted grid background, consisting of multiple lines of cursive script.

Handwritten text on a dotted grid background, consisting of multiple lines of cursive script.

Handwritten text on a dotted grid background, consisting of multiple lines of cursive script.

Handwritten text on a dotted grid background, consisting of multiple lines of cursive script.

DIVERSOS FORMATOS PARA OS CARACTERES A E H

DETALHES DAS MÁSCARAS DE CASAMENTO DE GABARITO

Handwritten text on a grid background, appearing as a list or series of entries. The text is dense and difficult to read due to the small font and grid lines.

Handwritten text on a grid background, appearing as a list or series of entries. The text is dense and difficult to read due to the small font and grid lines.

Handwritten text on a grid background, appearing as a list or series of entries. The text is dense and difficult to read due to the small font and grid lines.

FUNÇÕES DE PERTINÊNCIA

Em todas as equações abaixo, o cálculo de CAP, CEP e CCI retornam para equação, respectivamente, a quantidade de predicados *ausência_de_primitivas/4*, a quantidade de primitivas *reta_do_artefato/4* e a quantidade de predicado *concatenação_impossível/3*.

Onde não houver mencionamento específico, o cálculo da concatenação *m feita* retorna o seguinte:

$$CCMF_{\#} = \frac{1}{\text{número de concatenações}} \sum_{i=1}^n \text{distâncias}$$

A função número 1 a seguir leva em consideração apenas o erro devido concatenações. Foi a primeira função usada no trabalho, e não leva consideração excesso ou ausência das primitivas *reta_do_artefato/4*.

$$F_1(\%) = 1 - (CCMF_1)^2$$

Já na segunda função mudou-se o esquema do cálculo da concatenação *m feita* passando a ser uma média entre a melhor e a pior concatenação $(\frac{\text{max} + \text{min}}{2})$, e passou-se a "prejudicar" as classes de padrões com poucas concatenações, como as letras I e L, através da introdução de um erro por excesso de primitivas. O valor *número de primitivas* desempenha o papel fator de normalização na equação.

$$F_2(\%) = 1 - CCMF_2 - \frac{CEP_2}{\text{número de primitivas}_2}$$

A parte do cálculo do *excesso de primitivas* e a parte que diz respeito cálculo da *ausência de primitivas* referem-se ao caso em que, respectivamente na imagem existem mais primitivas *reta_do_artefato/4* do que é necessário para a instanciação das expressões PDL, e na imagem existem menos primitivas do que o necessário. Em ambos os casos, o valor referente a esses erros de excesso e/ou escassez deve ser abatido do valor máximo da lógica nebulosa, caso esse que ocorre quando todas as concatenações forem perfeitas e não sobram ou faltam qualquer das primitivas.

$$F_3(X) = \left| 1 - \text{CCMF}_3 - \frac{\text{CEP}_3}{\text{número de primitivas}_3} \right|$$

O caso do excesso de primitivas pode ocorrer, por exemplo, quando se está comparando a imagem de uma letra W (que possui por volta de 8 primitivas) com a classe da letra I (que possui por volta de 2 primitivas). Deve ser gerado artificialmente um erro simbólico para evitar que uma imagem de uma letra muito complexa seja classificada como a imagem de uma letra muito simples.

O caso da ausência de primitivas ocorre, por exemplo, quando tem-se uma imagem de uma letra L e quer-se classificá-la como sendo da classe da letra S que possui bem mais primitivas em seu padrão de classe do que a anterior. Neste caso deve ser gerado analogamente um erro simbólico para evitar que uma imagem com uma letra muito simples seja reconhecida como uma letra muito complexa. Veja que é o oposto do caso anterior.

$$F_4(X) = \left| 1 - \text{CCMF}_4 - \frac{\text{CEP}_4}{\text{número de primitivas}_4} \right|$$

Deve ser notado que as três funções acima (2, 3 e 4) não levam em consideração os cálculos da ausência de primitivas, tamanho das primitivas e concatenação impossível.

A equação abaixo leva pela primeira vez em consideração o cálculo da concatenação impossível, juntamente com seu fator de normalização.

$$F_5(X) = \left| 1 - \text{CCMF}_5 - \frac{\text{CEP}_5 + \text{CAP}_5}{\text{número de primitivas}_5} - \frac{\text{CCI}_5}{\text{Nr. concatenações}_5} \right|$$

onde $\text{CCMF}_5 = \text{CCMF}_2$.

$$F_6(X) = \sqrt{\left| 1 - \left(\text{CCMF}_6 + \frac{\text{CEP}_6}{\text{número de primitivas}_6} \right)^2 \right|}$$

Na equação abaixo, o cálculo da ausência de primitivas é mais importante do que o cálculo do excesso de primitivas, como se pode notar pelos pesos de ponderação, e também não leva em consideração o cálculo do tamanho das primitivas nem o cálculo devido às concatenações impossíveis.

$$F_7(X) = \sqrt{\left| 1 - \left(\text{CCMF}_7 + \frac{0.2 * \text{CEP}_7 + 0.8 * \text{CAP}_7}{\text{número de primitivas}_7} \right)^2 \right|}$$

A função número 8 leva em consideração o cálculo do tamanho das primitivas, mas com fator de ponderação baixo.

$$F_8(X) = \sqrt{\left| 1 - \left(\text{CCMF}_8 + \frac{0.2 * \text{CEP}_8 + 0.7 * \text{CAP}_8 + 0.1 * \text{TAM}_8}{\text{número de primitivas}_8} \right)^2 \right|}$$

A função número 9 é a que usa o cálculo da concatenação impossível, com todos os outros cálculos, de acordo com a equação a seguir.

$$F_9(X) = \sqrt{\left| 1 - (\text{ERROS}_9)^2 \right|}, \text{ onde:}$$

$$\text{ERROS}_9 = \text{CCMF}_9 + \frac{0.2 * \text{CEP}_9 + 0.3 * \text{CAP}_9}{\text{no de primitivas}_9} + \frac{0.5 * \text{CCI}_9}{\text{no de concatenações}_9}$$

Em todos os casos, portanto, para evitar que a imagem seja classificada erroneamente, gera-se um erro simbólico que será levado em consideração quando da chamada do predicado `cálculo_da_pertinência/4`, que implementa as 9 funções listadas anteriormente.

O número que aparece em parênteses na legenda de cada tabela e gráfico seguir indica a porcentagem de presença de pixels em cada primitiva extraída das imagens.

	Totais	Percentuais
Acertos	18	81.82
Erros	4	18.18
Rejeições	0	0.00
Totais	22	100.00

Tabela 1 - Resultados da experiência 23 (68%)

	Totais	Percentuais
Acertos	18	81.82
Erros	4	18.18
Rejeições	0	0.00
Totais	22	100.00

Tabela 2 - Resultados da experiência 23 (71%)

	Totais	Percentuais
Acertos	18	81.82
Erros	4	18.18
Rejeições	0	0.00
Totais	22	100.00

Tabela 3 - Resultados da experiência 23 (76%)

	Totais	Percentuais
Acertos	18	81.82
Erros	4	18.18
Rejeições	0	0.00
Totais	22	100.00

Tabela 4 - Resultados da experiência 23 (81%)

	Totais	Percentuais
Acertos	18	81.82
Erros	4	18.18
Rejeições	0	0.00
Totais	22	100.00

Tabela 5 - Resultados da experiência 23 (86%)

	Totais	Percentuais
Acertos	18	81.82
Erros	4	18.18
Rejeições	0	0.00
Totais	22	100.00

Tabela 6 - Resultados da experiência 23 (91%)

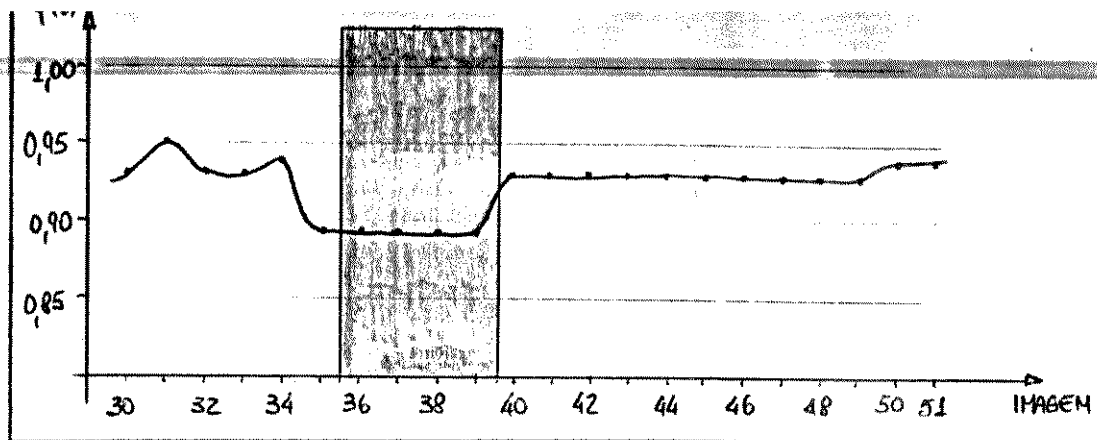


Gráfico 7 - Valores de pertinência para a experiência 23 (86%)

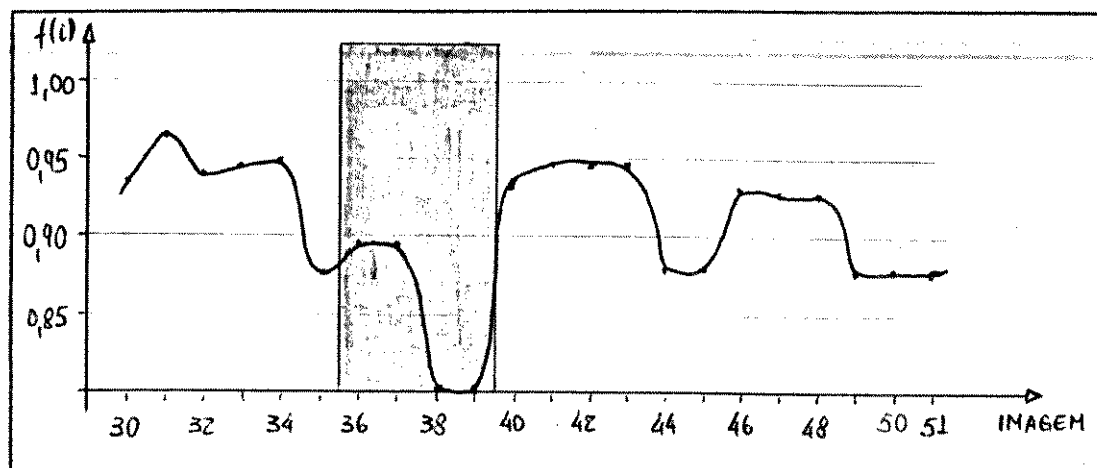


Gráfico 8 - Valores de pertinência para a experiência 23 (71%)

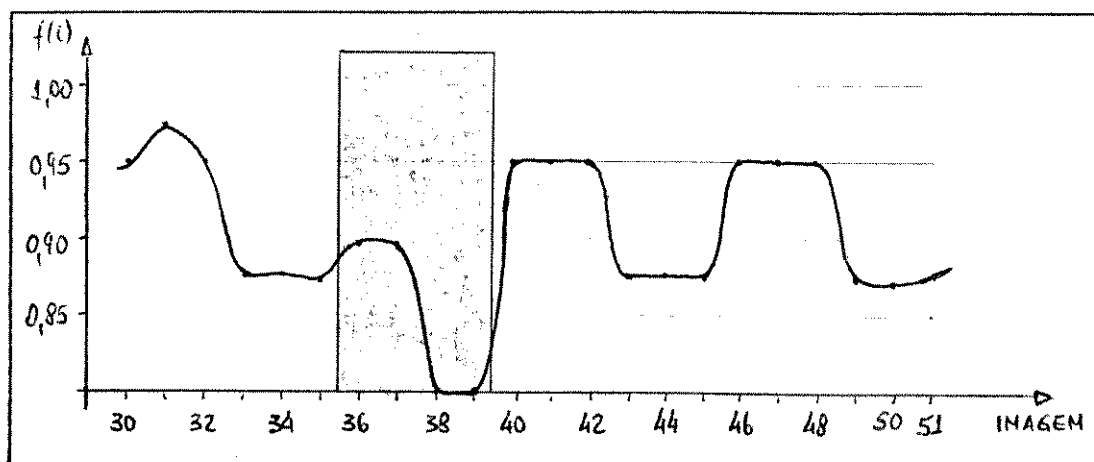


Gráfico 9 - Valores de pertinência para a experiência 23 (76%)

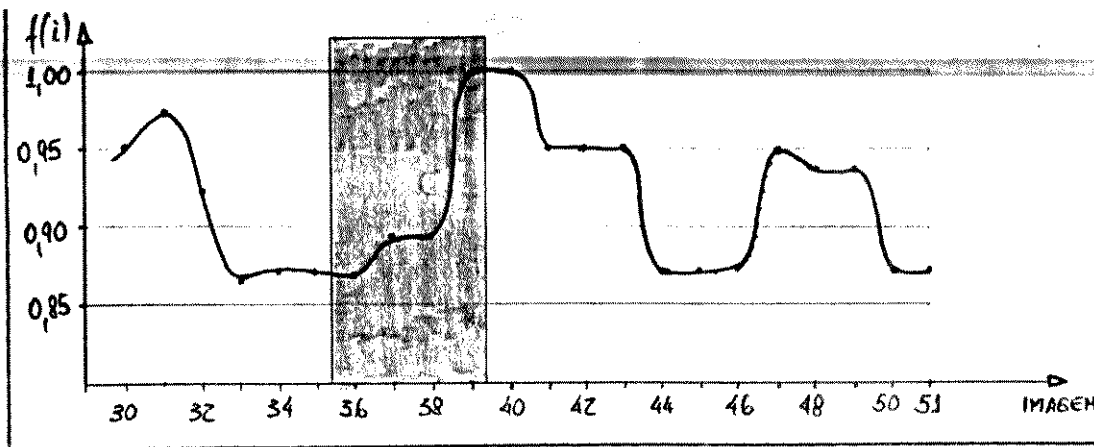


Gráfico 10 - Valores de pertinência para a experiência 23 (81%)

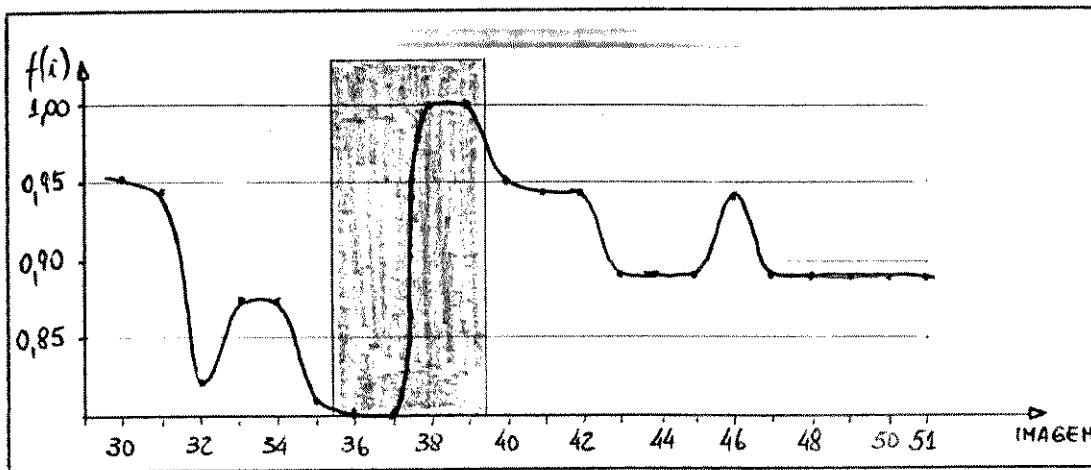


Gráfico 11 - Valores de pertinência para a experiência 23 (86%)

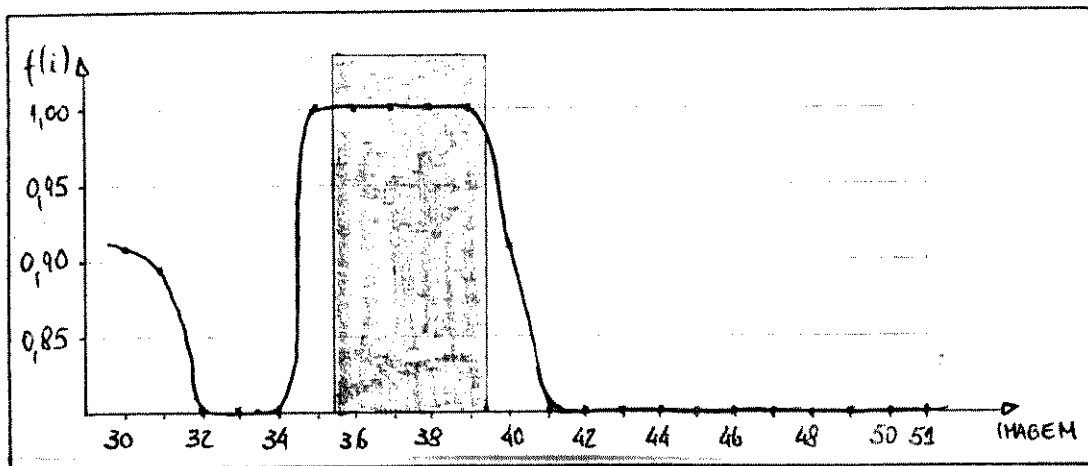


Gráfico 12 - Valores de pertinência para a experiência 23 (91%)

LIMIAR MÍNIMO	RECONHECIMENTO		
	Acertos	Erros	Rejeições
0.7500	18	4	0
0.8250	18	4	0
0.9000	17	0	5

Tabela 13 - Resultados (b) para a experiência 23 (66%)

LIMIAR MÍNIMO	RECONHECIMENTO		
	Acertos	Erros	Rejeições
0.7500	18	4	0
0.8250	18	2	2
0.9000	12	0	10

Tabela 14 - Resultados (b) para a experiência 23 (71%)

LIMIAR MÍNIMO	RECONHECIMENTO		
	Acertos	Erros	Rejeições
0.7500	18	4	0
0.8250	18	2	2
0.9000	9	0	13

Tabela 15 - Resultados (b) para a experiência 23 (76%)

LIMIAR MÍNIMO	RECONHECIMENTO		
	Acertos	Erros	Rejeições
0.7500	18	4	0
0.8250	18	4	0
0.9000	9	2	11

Tabela 16 - Resultados (b) para a experiência 23 (81%)

LIMIAR MÍNIMO	RECONHECIMENTO		
	Acertos	Erros	Rejeições
0.7500	18	4	0
0.8250	16	2	4
0.9000	6	2	14

Tabela 17 - Resultados (b) para a experiência 23 (86%)

LIMIAR MÍNIMO	RECONHECIMENTO		
	Acertos	Erros	Rejeições
0.7500	4	4	14
0.8250	4	4	14
0.9000	3	4	15

Tabela 18 - Resultados (b) para a experiência 23 (91%)

PREDICADO RECONHECER/2 COM PRIMEIRA INSTÂNCIA

```
reconhecer(a,maiusculo) :-
  (reta_do_artefato(X1,Y1,Z1,W1)
  ;
  X1 = 10,
  Y1 = 20,
  Z1 = 20,
  W1 = 25,
  assert(ausencia_de_primitiva(a,maiusculo,g2))
  ),
  primitiva(g2,X1,Y1,Z1,W1),
  (retract(reta_do_artefato(X1,Y1,Z1,W1)),
  asserta(copia_da_reta(X1,Y1,Z1,W1))
  );true
),
(reta_do_artefato(X2,Y2,Z2,W2)
;
X2 = 20,
Y2 = 15,
Z2 = 10,
W2 = 20,
assert(ausencia_de_primitiva(a,maiusculo,d2))
),
primitiva(d2,X2,Y2,Z2,W2),
(retract(reta_do_artefato(X2,Y2,Z2,W2)),
asserta(copia_da_reta(X2,Y2,Z2,W2))
);true
),
(concatenacao(a,maiusculo,$+$,X2,Y2,Z2,W2,X1,Y1,Z1,W1,XR1,YR1,ZR1,WR1)
;
XR1 = 20,
YR1 = 15,
ZR1 = 20,
WR1 = 25,
assert(concatenacao_impossivel(a,maiusculo,$+$))
),
(reta_do_artefato(X3,Y3,Z3,W3)
;
X3 = 20,
Y3 = 15,
Z3 = 20,
W3 = 25,
assert(ausencia_de_primitiva(a,maiusculo,h2))
),
primitiva(h2,X3,Y3,Z3,W3),
(retract(reta_do_artefato(X3,Y3,Z3,W3)),
asserta(copia_da_reta(X3,Y3,Z3,W3))
);true
),
```

```

Concatenacao(a, maiusculo, $*$, XR1, YR1, ZR1, WR1, X3, Y3, Z3, W3, XR2, YR2, ZR2, WR2)
;
XR2 = 20,
YR2 = 15,
ZR2 = 20,
WR2 = 25,
assert(concatenacao_impossivel(a, maiusculo, $*$))
),
(reta_do_artefato(X4, Y4, Z4, W4)
;
X4 = 20,
Y4 = 25,
Z4 = 30,
W4 = 30,
assert(ausencia_de_primitiva(a, maiusculo, g2))
),
primitiva(g2, X4, Y4, Z4, W4),
(retract(reta_do_artefato(X4, Y4, Z4, W4)),
asserta(copia_da_reta(X4, Y4, Z4, W4))
;true
),
Concatenacao(a, maiusculo, $+$, XR2, YR2, ZR2, WR2, X4, Y4, Z4, W4, XR3, YR3, ZR3, WR3)
;
XR3 = 20,
YR3 = 15,
ZR3 = 30,
WR3 = 30,
assert(concatenacao_impossivel(a, maiusculo, $+))
),
(reta_do_artefato(X5, Y5, Z5, W5)
;
X5 = 30,
Y5 = 10,
Z5 = 20,
W5 = 15,
assert(ausencia_de_primitiva(a, maiusculo, d2))
),
primitiva(d2, X5, Y5, Z5, W5),
(retract(reta_do_artefato(X5, Y5, Z5, W5)),
asserta(copia_da_reta(X5, Y5, Z5, W5))
;true
),
Concatenacao(a, maiusculo, $+$, X5, Y5, Z5, W5, XR3, YR3, ZR3, WR3, XR4, YR4, ZR4, WR4)
;
XR4 = 30,
YR4 = 10,
ZR4 = 30,
WR4 = 30,
assert(concatenacao_impossivel(a, maiusculo, $+))
),
assert(vetor_final(a, maiusculo, XR4, YR4, ZR4, WR4))

```

PREDICADO RECONHECER/2 COM MELHOR INSTÂNCIA

```
reconhecer(a, maiusculo) :-
  (pegar_a_melhor(20,10,1,20,d2,_,X1,Y1,Z1,W1)
   ;X1 = 20,Y1 = 10,Z1 = 1,W1 = 20,
   assert(ausencia_de_primitiva(a,maiusculo,d2))),
  (retract(reta_do_artefato(X1,Y1,Z1,W1));true),
  (pegar_a_melhor(X1,Y1,Z1,W1,g2,$+$,X2,Y2,Z2,W2)
   ;X2 = 1,Y2 = 20,Z2 = 20,W2 = 30,
   assert(ausencia_de_primitiva(a,maiusculo,g2))),
  (retract(reta_do_artefato(X2,Y2,Z2,W2));true),
  (concatenacao(a,maiusculo,$+$,X1,Y1,Z1,W1,X2,Y2,Z2,W2,XR1,YR1,ZR1,WR1)
   ;XR1 = 20,YR1 = 10,ZR1 = 20,WR1 = 30,
   assert(concatenacao_impossivel(a,maiusculo,$+$))),
  (pegar_a_melhor(XR1,YR1,ZR1,WR1,h2,$*$,X3,Y3,Z3,W3)
   ;X3 = 20,Y3 = 10,Z3 = 20,W3 = 30,
   assert(ausencia_de_primitiva(a,maiusculo,h2))),
  (retract(reta_do_artefato(X3,Y3,Z3,W3));true),
  (concatenacao(a,maiusculo,$*$,XR1,YR1,ZR1,WR1,X3,Y3,Z3,W3,XR2,YR2,ZR2,WR2)
   ;XR2 = 20,YR2 = 10,ZR2 = 20,WR2 = 30,
   assert(concatenacao_impossivel(a,maiusculo,$*$))),
  (pegar_a_melhor(XR2,YR2,ZR2,WR2,d2,$+$,X4,Y4,Z4,W4)
   ;X4 = 40,Y4 = 1,Z4 = 20,W4 = 10,
   assert(ausencia_de_primitiva(a,maiusculo,d2))),
  (retract(reta_do_artefato(X4,Y4,Z4,W4));true),
  (concatenacao(a,maiusculo,$+$,X4,Y4,Z4,W4,XR2,YR2,ZR2,WR2,XR3,YR3,ZR3,WR3)
   ;XR3 = 40,YR3 = 1,ZR3 = 20,WR3 = 30,
   assert(concatenacao_impossivel(a,maiusculo,$+$))),
  (pegar_a_melhor(XR3,YR3,ZR3,WR3,g2,$+$,X5,Y5,Z5,W5)
   ;X5 = 20,Y5 = 30,Z5 = 40,W5 = 40,
   assert(ausencia_de_primitiva(a,maiusculo,g2))),
  (retract(reta_do_artefato(X5,Y5,Z5,W5));true),
  (concatenacao(a,maiusculo,$+$,XR3,YR3,ZR3,WR3,X5,Y5,Z5,W5,XR4,YR4,ZR4,WR4)
   ;XR4 = 40,YR4 = 1,ZR4 = 40,WR4 = 40,
   assert(concatenacao_impossivel(a,maiusculo,$+$))).
```

```

%-- fase 1:
%-- calculo das poligonais pequenas:

muitas_poligs(TAM_MIN,OPCAO) :-
    nl,
    reta_do_artefato(X,Y,Z,W),
    [!
        abs(X-Z,XZ),
        abs(Y-W,YW),
        RESULT is sqrt(XZ*XZ + YW*YW),
        ifthen(RESULT =< TAM_MIN, (assert(poligonal(X,Y,Z,W)),
            retract(reta_do_artefato(X,Y,Z,W))
                ;
            true)
        ],
    fail.

```

```

%-- fase 2:
%-- chamada do predicado analise_poligonal:

muitas_poligs(TAM_MIN,OPCAO) :-
    (analise_poligonal(OPCAO)
    ;
    true
    ).

```

- fase 3:
- analise da sequencialidade das poligonais pequenas:
- aqui deve-se ver quais as poligonais que se seguem e que estao
- muito pequenas para serem deixadas na imagem, ok ?

```

analise_poligonal(OPCAO) :-
poligonal(X1,Y1,Z1,W1),
[!
  (poligonal(Z1,W1,Z2,W2),
  assertz(poligonal(X1,Y1,Z2,W2)),
  retract(poligonal(X1,Y1,Z1,W1)),
  retract(poligonal(Z1,W1,Z2,W2))
  ;
  reta_do_artefato(Z1,W1,Z2,W2),
  assertz(poligonal(X1,Y1,Z2,W2)),
  retract(poligonal(X1,Y1,Z1,W1)),
  retract(reta_do_artefato(Z1,W1,Z2,W2))
  ;
  reta_do_artefato(Z2,W2,Z1,W1),
  assertz(poligonal(X1,Y1,Z2,W2)),
  retract(poligonal(X1,Y1,Z1,W1)),
  retract(reta_do_artefato(Z2,W2,Z1,W1))
  ),
true
!],
fail.

```

- fase 4:
- substituicao de poligonal por reta_do_artefato:

```

analise_poligonal($s$) :- %-- SIM, substitua poligonal por reta_do_artefato
poligonal(X,Y,Z,W),
[!
  assertz(reta_do_artefato(X,Y,Z,W)),
  retract(poligonal(X,Y,Z,W))
!],
fail.

```

```

analise_poligonal($n$) :- %-- NAO, deletar sumariamente poligonal/4
abolish(poligonal/4).

```

- fase 5:
- fim !

```

analise_poligonal(OPCAO) :-
true.

```


Dados	Data	11
Procura	Search	11
Estrutura	Structure	11
Análise sintática	Syntactic parsing	12
Linguagem de Descrição de Padrões	Pattern Description Language	12, 73
Casamento	Matching	16, 52
Agrupamentos	Clusters	41
Baseados em sinal	Signal-based	50
Máscara, gabarito	Template	52
Nebulosidade	Fuzziness	70
Gramática nebulosa	Fuzzy grammar	70