

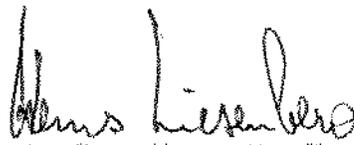
UM METODO PARA RECONHECIMENTO DE FIGURAS

ESBOCADAS A MAO

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida pelo Sr. Williams Santos e aprovada pela Comissão Julgadora.

*Exempl.*

Campinas, 03 de fevereiro de 1990



Prof. Dr. Hans K. E. Niesenber  
orientador

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

Sa59m

12396/BC

UNICAMP

## INTRODUCAO

Um aspecto muito importante na criação de ferramentas computacionais é a melhoria da comunicação homem-máquina. A qualidade das facilidades nesta comunicação é fundamental, principalmente quando há necessidade de entradas gráficas. O progresso acentuado e a conseqüente diminuição no custo do "hardware" e "software" gráficos contribuíram para o alto uso das saídas gráficas e sua gradual implementação nas interfaces homem-máquina.

O desenvolvimento do desenho a mão como técnica de entrada pode ser uma possibilidade para o sistema suportar os modelos mentais dos objetos gráficos imaginados pelo usuário. O usuário teria a oportunidade de esboçar suas idéias, sem se preocupar com as restrições impostas pelos sistemas habituais de entradas, e o sistema o ajudaria a transformar o seu modelo em uma representação "correta" para o computador.

O objetivo deste trabalho é a definição de um método para o reconhecimento de figuras esboçadas a mão, bem como a sua implementação em um protótipo que funcionará como uma interface de entrada direta de dados gráficos. O protótipo receberá como entrada um esboço a mão, que deverá ser composto por subfiguras padrões, assim como em um diagrama de circuito lógico ou em um fluxograma. A saída do protótipo será uma descrição das subfiguras componentes do desenho de entrada, indicando como elas estão conectadas entre si.

Este trabalho está dividido em duas partes. A primeira parte consiste em uma introdução ao assunto (capítulo 1) e à descrição do método de reconhecimento proposto (capítulo 2), sem se preocupar com os detalhes de implementação. A segunda parte do trabalho trata especificamente da implementação do método proposto, além de trazer uma análise dos resultados obtidos.

# INDICE

## PARTE I: DESCRICAO DO METODO

### CAPITULO 1: RECONHECIMENTO DE PADROES

1.1-	INTRODUÇÃO .....	1
1.2-	FATORES HUMANOS NA INTERAÇÃO HOMEM-MÁQUINA .....	4
1.3-	RECONHECIMENTO DE CARACTERES .....	7
1.3.1-	pesquisas importantes em reconhecimento de caracteres	13
1.4-	RECONHECIMENTO DE FIGURAS .....	14
1.4.1-	aproximação de figuras .....	15
1.4.2-	descrição de figuras .....	16
1.4.3-	segmentação de figuras .....	17
1.5-	MÉTODOS DE RECONHECIMENTO DE PADRÕES .....	18
1.5.1-	casamento de padrão .....	22
1.5.2-	métodos decisão-teórico .....	23
1.5.3-	métodos sintático .....	25
1.6-	COMPARAÇÃO ENTRE OS MÉTODOS .....	26
1.7-	MÉTODO PROPOSTO .....	28

## CAPITULO 2: METODO DE RECONHECIMENTO

2.1- FIGURAS DE ENTRADA .....	30
2.2- BIBLIOTECA DE PADRÕES .....	33
2.3- SEGMENTAÇÃO DA FIGURA DE ENTRADA .....	36
2.4- RECONHECIMENTO DOS PADRÕES SIMPLES .....	41
2.5- RECONHECIMENTO DOS PADRÕES COMPOSTOS .....	44
2.6- DETERMINAÇÃO DAS INTERCONEXÕES .....	47
2.7- USO DAS REGRAS DE CONEXÕES .....	48
2.8- RECONHECIMENTO DE CARACTERES .....	50

## PARTE II: IMPLEMENTAÇÃO DO RECONHECEDOR DE FIGURAS ESBOCADAS A MÃO

### CAPITULO 3: BIBLIOTECA DE PADRÕES

3.1- DESCRIÇÃO .....	53
3.2- CONSTRUÇÃO DAS BIBLIOTECAS .....	60
3.3- LEITURA DAS BIBLIOTECAS .....	63

## CAPITULO 4: PROTOTIPO IMPLEMENTADO

4.1- ENTRADA E SEGMENTAÇÃO DO DESENHO .....	65
4.2- MÉTODO DE SEGMENTAÇÃO UTILIZADO .....	66
4.2.1- método 01 .....	66
4.2.2- método 02 .....	69
4.3- LISTA DOS SEGMENTOS .....	72
4.4- QUANTIZAÇÃO DA DIREÇÃO .....	73
4.5- RECONHECIMENTO DAS FIGURAS SIMPLES .....	73
4.5.1- figuras fechadas .....	75
4.5.2- figuras abertas .....	78
4.6- RECONHECIMENTO DAS FIGURAS COMPOSTAS .....	79
4.7- DETERMINAÇÃO DAS INTERCONEXÕES .....	80
4.8- RECONHECIMENTO DE CARACTERES .....	84

## CAPITULO 5: CONCLUSOES

5.1- RESULTADOS .....	88
5.2- SUGESTÕES PARA FUTURAS MELHORIAS NO PROTÓTIPO .....	91
5.3- CONCLUSÕES .....	95

BIBLIOGRAFIA .....	97
--------------------	----

## PARTE I

### DESCRICAO DO METODO

# CAPITULO 01

## RECONHECIMENTO DE PADROES

### 11- INTRODUCAO:

A qualidade das facilidades de comunicação homem-máquina em ferramentas computacionais é um aspecto que vem crescendo em importância. Estas facilidades não afetam diretamente a funcionabilidade das ferramentas de software, mas, se cuidadosamente projetadas, permitem, entre outras, o uso das ferramentas por um grupo de usuários não especialistas em informática, o aumento da produtividade destes usuários e eventualmente eliminam fontes de erros.

Um dos problemas que ocorre frequentemente nesta área é que o diálogo entre usuário e máquina se processa em níveis qualitativamente distintos na entrada e na saída. As saídas, em muitos casos, são bastante sofisticadas e baseiam-se na capacidade humana em reconhecer padrões com certa eficiência e rapidez. Em vez de listas enormes de resultados numéricos, por exemplo, utilizam-se gráficos em três dimensões. No caso da entrada de dados esta operação se processa de forma menos natural na maioria dos casos. O usuário é que precisa se adaptar às restrições impostas pela interface. O uso de teclado e menus na concepção de esquemáticos, por exemplo, são pouco naturais neste tipo de atividade.

Este desnível qualitativo entre a entrada e a saída é um dos principais obstáculos para as atividades de interação [HOS82]. Um desses

obstáculos encontrados pelo usuário durante uma interação gráfica é a contínua mudança de dispositivos requerida durante a entrada dos dados, que, em muitas vezes, é inevitável, uma vez que o uso de teclados é mais apropriado nos casos onde entradas alfanuméricas são essenciais, enquanto que o uso do mesmo nas entradas gráficas não é muito eficaz.

Para entradas exclusivamente em forma de texto, com nenhuma outra atividade para o usuário, o uso do teclado é quase sempre mais rápido que qualquer outro dispositivo de entrada manual de dados, pois um texto consiste de uma sequência de caracteres sem informação posicional, ou seja, um novo caracter é colocado sempre à direita (exceto no final de uma linha) do caracter fornecido imediatamente anterior. Entretanto, o número de aplicações no qual as entradas são puramente textuais é muito menor do que se imagina [WAR85]. O número de teclas que afetam o posicionamento dos caracteres em vários sistemas é uma indicação de quão inadequado é o uso dos teclados em entradas gráficas.

Sistemas gráficos que requerem uma troca de dispositivos entre entrada de textos e entradas gráficas implicam na alternância das mãos do usuário em pelo menos dois dispositivos, quebrando assim sua atenção, pois além de ocupar-se da aplicação em si, o usuário deve ficar atento ao gerenciamento dos dispositivos de entrada. Com isto, o nível de interação entre o usuário e a máquina fica comprometido e esta troca constante entre dispositivos aumenta a possibilidade de ocorrerem erros.

Um elemento importante para um sistema CAD é um subsistema de entrada para aquisição direta de dados [SHI87]. A interação homem-máquina baseada em símbolos gráficos e desenhos parece ser vantajosa nesta área,

pois relações e processos complexos são mantidos claros, distintos e geralmente compreensíveis. O progresso acentuado e a conseqüente diminuição no custo do hardware e dos pacotes de software gráfico contribuíram para o uso cada vez mais elevado das saídas gráficas e sua gradual implementação nas interfaces homem-máquina [HOS82].

O aparecimento de dispositivos como a caneta óptica ou o "mouse", com os quais é possível apontar um ponto qualquer em uma superfície, facilitou a interação através de figuras.

Desenhos são usados largamente na atividade de projeto, pois eles são a linguagem na qual um projetista expressa seu modelo conceitual e define as formas, dimensões, funções e restrições. Estes desenhos possuem regras de construção e algumas vezes, símbolos padrões que, quando combinados com a habilidade humana de associação e inferência, possibilitam uma interpretação correta de seu significado. Desta forma, o uso de um padrão gráfico ou um sistema de reconhecimento de símbolos parece ser um modo mais conveniente para adaptar a interface à maneira de como as pessoas se comunicam. As aplicações relacionadas com o reconhecimento de símbolos gráficos mostram um esforço comum em adaptar as técnicas de entrada de dados ao modo convencional de escrita ou desenho usando lápis e papel. O objetivo é a independência de teclados e dispositivos similares, e a diminuição das permanentes trocas entre diferentes dispositivos durante a entrada gráfica. Entretanto alcançar esta independência é uma tarefa complexa.

Algumas idéias interessantes referente a técnicas de entrada de dados foram desenvolvidas e implementadas nos últimos anos. A maioria

desses desenvolvimentos são soluções para problemas particulares e não podem ser generalizadas [JAN85]. Para conseguir ferramentas mais gerais, é necessário considerar os mecanismos psicológicos utilizados pelo usuário.

## 12- FATORES HUMANOS NA INTERACAO HOMEM-MAQUINA:

Uma barreira geral para o projeto de sistemas computacionais é a falta de conhecimento das necessidades e características dos usuários. Uma experiência muito comum é o sistema não atender às características únicas de uma dada tarefa ou requerer que essa tarefa seja realizada de um modo não natural para o usuário.

O nível mais alto da ação desejada pelo usuário é chamado modelo mental ou intenção. Normalmente, o modelo mental pode ser visto como uma interpretação do usuário para uma determinada tarefa. Um modelo mental do usuário pode ser um determinante no rendimento de um sistema e também um meio de entender porque erros particulares ocorrem. A habilidade de uma pessoa utilizar um determinado sistema está relacionado com a sua particular concepção do sistema e o entendimento deste relacionamento pode facilitar o projeto de futuros sistemas [KEL86]. Para que um modelo possa ser concretizado, é necessário transformá-lo em instruções para o computador. Um método de desenvolvimento de linguagens interativas emprega a estruturação do modelo do usuário em primitivas peculiares ao tipo de tarefa envolvida. O trabalho mental requerido para transformar um modelo visual em instruções para o sistema pode ser dividido em duas etapas:

- 1- a transformação do modelo visual do usuário em primitivas (nível mental);

2- a transformação destas primitivas em instruções ao computador (nível de comando).

A primeira transformação acima é um processo não formalizado, via de regra, dada a sua complexidade intrínseca. A decomposição de modelos visuais não é evidente, uma vez que esta decomposição varia de pessoa para pessoa, isto se elas decompõem suas intenções.

Se os seres humanos decompõem seus modelos visuais através da formação de primitivas mentais, seria uma boa estratégia de projeto analisá-las e desenvolver um conjunto de instruções que reproduzam estas primitivas. Mas isto pode ser perigoso. Uma vez que não parece existir uma maneira simples de formar primitivas para todos os fins, o conjunto de instruções desenvolvidas forçaria alguns usuários a trabalhar de uma forma não usual, tendo de adaptar seus modelos mentais de acordo com essas primitivas existentes. Como não existe uma "melhor maneira" de se executar uma tarefa, ou um conjunto de primitivas, de forma suficientemente geral capaz de compor o modelo do usuário segundo seu ponto de vista, este método não funcionaria muito satisfatoriamente [JAN85]. Será preferível que o próprio usuário estenda o conjunto de instruções de acordo com suas necessidades e hábitos.

Existem alguns sistemas de entrada de dados gráficos que deixam os usuários trabalharem de uma forma muito natural. Por exemplo, sistemas como o APPLICON'S BRAVO! habilita o usuário a selecionar comandos do sistema através de símbolos gráficos, simples o suficiente para serem fornecidos via mesa digitalizadora. Estes símbolos são interpretados como um comando ou uma série de comandos a partir dos quais informações

geométricas podem ser extraídas. Um sistema CAD que usa técnicas de reconhecimento "on-line" para processar esboços feitos a mão é o GEOMAP, desenvolvido por Hosake e Kimura [HOS82]. O sistema recebe como entrada um desenho rascunhado em uma folha de papel colocada sobre uma mesa digitalizadora. A folha está dividida em 4 regiões. Três delas são usadas para esboçar as projeções do desenho nos planos cartesianos. O desenho final é composto a partir dessas projeções e de informações adicionais escritas nos desenhos. O sistema CASUS [JAN85] é semelhante a este. Os exemplos mostrados acima possuem uma característica em comum: utilizam o esboço a mão como entrada, tentando simular o comportamento natural das pessoas quando escrevendo em uma folha de papel para descrever ou explicar um problema técnico.

Usando o esboço a mão como uma técnica de entrada, o usuário desenvolve sua intenção até um certo estágio, sem se preocupar com as restrições do sistema, tal como o conjunto de instruções existentes. O desenvolvimento do desenho a mão como técnica de entrada de dados pode ser uma possibilidade para suportar os modelos mentais dos objetos gráficos imaginados pelo usuário. Freqüentemente, um primeiro desenho está incompleto, não tem fronteiras bem definidas, está inconsistente, ambíguo, etc. O usuário teria a oportunidade de esboçar suas idéias e o sistema o ajudaria a transformar o seu modelo em uma representação "correta" para o computador. O esboço a mão pode suportar a formação de novas idéias e deixa o usuário em seu próprio ambiente de manipulação de um desenho.

A técnica de esboço a mão permite que o usuário manipule diretamente o desenho. Não é necessário que ele selecione um elemento do desenho de entrada e digite algum número para que seja realizada alguma

operação sobre o desenho, diminuindo assim, as oportunidades de executar operações erradas.

Para que um sistema possa aceitar um esboço a mão como entrada de dados, ele deve ser capaz de reconhecê-lo e extrair os dados desejados a partir desse esboço. Os dados extraídos dos desenhos dependem do método utilizado para o reconhecimento. Sendo assim, a parte principal do subsistema de entrada de dados deve ser um reconhecedor de padrões, capaz de reconhecer os objetos gráficos (desenhos). Seria aconselhável que o reconhecedor também reconhecesse caracteres, pois eles poderão ser usados para fornecer informações adicionais, indispensáveis em determinadas ocasiões.

O objetivo deste trabalho é a definição de um método para o reconhecimento de figuras esboçadas a mão, bem como a sua implementação em um protótipo que funcionará como uma interface de entrada direta de dados gráficos. Este protótipo receberá como entrada um esboço a mão, que deverá ser composto por subfiguras padrões, assim como em um diagrama de circuito lógico ou em um fluxograma. O protótipo fornecerá como saída uma descrição das subfiguras componentes do desenho de entrada, indicando como elas estão conectadas entre si.

### **1.3- RECONHECIMENTO DE CARACTERES:**

Durante os últimos 30 anos houve um considerável aumento no interesse em problemas relacionados com o reconhecimento de padrões. Este interesse criou uma necessidade crescente de métodos teóricos, além de software e hardware adequados para o uso em projetos de aplicativos nesta

área [FU-80]. As aplicações do reconhecimento de padrão incluem reconhecimento de caracteres, detecção de alvo, análise de sinais, diagnose médica, entre outros.

Em combinação com técnicas de reconhecimento de caracteres, técnicas de reconhecimento de figuras podem ser aplicadas em áreas tais como cópias automáticas de documentos escritos a mão contendo diagramas de bloco e programação automática a partir de um fluxograma esboçado a mão.

Na elaboração de um diagrama lógico de um circuito eletrônico composto por portas lógicas e suas conexões, por exemplo, identificadores dos elementos componentes e de algumas interconexões costumam ser acrescentados aos diagramas. Geralmente essas informações adicionais são compostas de caracteres. Elas são acrescentadas para uma melhor definição do significado do desenho, sendo indispensáveis em certos casos. Conseqüentemente, o caracter é um elemento indispensável no desenho [ISH87].

Problemas relacionado com a replicação de funções humanas são freqüentemente atrativos para os pesquisadores. Neste caso, a simulação da leitura humana tem sido objeto de intensas pesquisas nas últimas três décadas. Tamanho esforço foi dispendido não somente por ser este um problema desafiante, mas também pelo fato dele fornecer subsídios para o processamento automático de grandes volumes de dados. Por outro lado, se a máquina é suficientemente sofisticada e confiável, ela abrirá novos horizontes de aplicações.

O reconhecimento de caracteres foi um dos grandes responsáveis

pelo desenvolvimento da análise de imagens e reconhecimento de padrões. Embora este problema pareça muito simples à primeira vista, usualmente existem dificuldades adicionais que complicam e conduzem a uma solução específica [GAI78]. Dentre essas dificuldades adicionais temos as distorções possíveis, especialmente no caso do reconhecimento de caracteres escritos a mão. A figura 1.1 [MAN86] mostra as diferentes áreas abrangidas pelo termo reconhecimento de caracteres.

A ordem de dificuldade do problema de reconhecimento de caracteres, começando do mais difícil, é: reconhecimento da escrita, caracteres escritos a mão, reconhecimento "on-line" de caracteres que fornece informações sobre a ordem de entrada de traços, assim como a imagem do carácter, reconhecimento de caracteres de formatos fixos e, finalmente, o reconhecimento de caracteres magnéticos ou mecânicos.

O reconhecimento de caracteres avançou a partir do uso de esquemas primitivos para o reconhecimento de números impressos por máquinas até a aplicações de técnicas sofisticadas para o reconhecimento de uma grande variedade de caracteres e símbolos escritos a mão. O problema de reconhecimento da escrita é de grande interesse e desafio para os pesquisadores, pois mesmo o ser humano, que possui os leitores ópticos mais treinados - os olhos, erram em cerca de 4% quando lendo na ausência de contexto [SUE80].

Os erros na leitura são causados por uma variação infinita de formatos, resultados do hábito de escrita, estilo, educação, região de origem, ambiente social, humor, saúde e outras condições do escritor/leitor, bem como de outros fatores, tais como o instrumento e

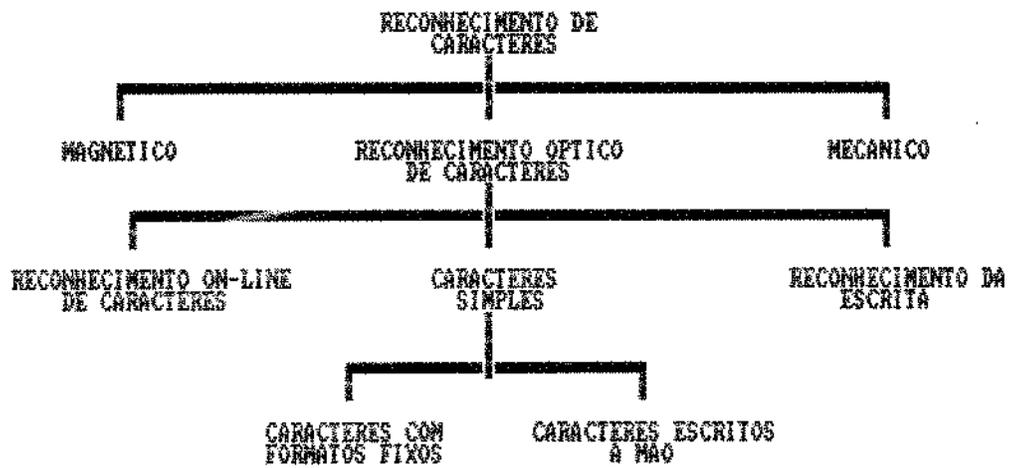


FIGURA 1.1

superfície de escrita, métodos de esquadramento e algoritmos de reconhecimento [MAN86].

Para reduzir as variações na forma, aumentando assim a taxa de reconhecimento, comitês em vários países desenvolveram padrões de escrita a mão. Os usuários eram instruídos para escreverem tão perto dos modelos fornecidos quanto possível. Em alguns destes padrões desenvolvidos, segmentos extras foram introduzidos para reduzir as possibilidades de interpretações ambíguas.

Existem inúmeras maneiras de reconhecer caracteres, assim como existem uma infinidade de maneiras de escrevê-los. As técnicas de reconhecimento variam largamente de acordo com as características escolhidas, o modo de extração destas características e o esquema de classificação usado. Os resultados ilustram a importância da qualidade dos dados e a possibilidade de alterações das restrições para encontrar uma taxa de reconhecimento maior.

Alguns sistemas de reconhecimento de caracteres escritos a mão são treináveis. Existem dois tipos de treinos possíveis:

1- os usuários são treinados a escrever no estilo que o dispositivo pode reconhecer [ITO78];

2- o dispositivo é treinado através de exemplos fornecidos pelos usuários mostrando como eles escrevem [HIR85][ISH87][HUA86][NEV79].

No primeiro caso, os usuários devem adaptar-se ao estilo particular reconhecido pelo dispositivo de leitura. A medida de sucesso

depende da habilidade dos usuários em aprender este estilo de escrita e continuar a usá-lo (para sempre). No segundo caso, o sistema reconhece os caracteres pela comparação de cada forma escrita com os exemplos que foram previamente fornecidos. Estes sistemas são baseados nas seguintes premissas:

1- as formas extraídas pelo sistema serão diferentes se os usuários os visualizarem de modo diferente e serão iguais se os usuários os virem como tais;

2- os exemplos dados pelos usuários formam uma amostragem completa de como eles escrevem cada caracter;

3- os exemplos são típicos da escrita do usuário;

4- a escrita do usuário não mudará.

Na realidade é muito difícil obter exemplos representativos. Geralmente o usuário escreve com mais cuidado durante a entrada dos exemplos. Além do mais, o estilo de cada um pode variar durante algum período de tempo. Sendo assim, estes sistemas não são muito adequados, embora quando utiliza-se um reconhecedor treinável e a forma dos padrões é especificada com ajuda de segmentos extras para diferenciar caracteres facilmente confundíveis, uma taxa de reconhecimento consideravelmente alta é obtida, desde que as premissas acima sejam observadas.

Embora as máquinas de reconhecimento difiram entre si enormemente no que diz respeito ao desempenho, elas não são muito satisfatórias [SUE80]. Isto é evidente pela rejeição esporádica ou reconhecimento errôneo de caracteres que podem ser facilmente reconhecidos por seres humanos. Por outro lado, essas máquinas, ocasionalmente, reconhecem vários caracteres cuja forma está tão distorcida, que não são facilmente reconhecíveis por

olhos humanos. Em suma, o desempenho das máquinas presentes é algo inconsistente e irregular. Convém observar que a distinção final e interpretação inteligente pelos seres humanos da imensa variedade de estilos de escrita manual de caracteres são adquiridos através de um prolongado processo de aprendizagem. A extração da essência de tal conhecimento e de sua inferência em um computador diminuirá a distância entre a "inteligência" humana e a da máquina no tocante ao reconhecimento de caracteres.

Embora as pesquisas relacionadas com o reconhecimento de padrões já vêm se desenvolvendo há vários anos, novos temas estão surgindo. Neste ponto, os autores vislumbram quatro novas ênfases e direções que são cruciais ao futuro desenvolvimento do tema [SUE80]:

1- desenvolvimento de um conjunto padrão de caracteres que pode ser confortavelmente usado pelo homem para produzir exemplos facilmente reconhecíveis pelas máquinas;

2- pesquisas mais fundamentais e psicológicas para o melhor entendimento do processo complicado da percepção e formação de caracteres escritos, incluindo o estudo de várias formas geradas a partir de cada símbolo;

3- seleção sistemática de características distintivas e combinação ótima de técnicas de reconhecimento para fortalecer o poder discriminante da máquina;

4- a aplicação de informações gramaticais, lingüísticas e contextuais para resolução de ambiguidades entre caracteres parecidos.

Entretanto, o caminho a ser seguido em todas as direcções, deve levar em consideração as propriedades da máquina e fatores humanos.

### 1.3.1- Pesquisas importantes em reconhecimento de caracteres:

As origens do reconhecimento de caracteres pode ser encontrada em 1870 quando Carey inventou a "retina scanner" que é um sistema de transmissão de imagens que utilizava um mosaico de fotocélulas, e mais tarde, em 1890, quando Nipkow inventou o esquadrinhador sequencial, que foi um grande avanço rumo à televisão moderna e às máquinas de leitura. A versão moderna dos reconhecedores ópticos de caracteres (OCR) apareceu em meados dos anos 40 com o desenvolvimento do computador digital. Pela primeira vez, um OCR foi concebido como um método de processamento de dados com aplicação ao mundo comercial [MAN86].

Uma das primeiras tentativas em reconhecimento de caracteres foi feita por uma grupo liderado por Grimsdale em 1958, onde em um trabalho pioneiro eles descrevem um método onde o padrão de entrada é varrido por um esquadrinhador tipo CRT (cathode ray tube) e, subseqüentemente, um computador digital extrai as características básicas do padrão. Estes valores eram comparados com outros armazenados no computador durante o processo de reconhecimento. Em outro trabalho do mesmo período, Bledsoe e Brawning descreveram um método um pouco mais geral utilizando um fotomosaico de 10x15 fotocélulas.

No início dos anos 60, um grupo do MIT sob liderança de Murray Eden apresentou a idéia de que todos os caracteres latinos podem ser gerados a partir de um subconjunto de quatro segmentos. Este conceito deu

origem a um grande número de pesquisas e foi conhecido como o método "análise por síntese". Segundo [MAN86], a importância maior do trabalho de Eden foi que ele provou formalmente que todos os caracteres escritos a mão podem ser formados por um número finito de traços (segmentos). Esta noção foi posteriormente usada em todos os métodos de reconhecimento de padrão sintático aplicado em reconhecimento de caracteres.

Em 1966, Casey e Nagy da IBM apresentaram um dos primeiros esforços no reconhecimento de caracteres Chineses. Em 1968 eles introduziram um método não convencional para o reconhecimento, a "máquina de leitura autônoma", uma vez que nenhum treinamento prévio ou informação a priori com relação ao caracter era necessária, mas somente o conhecimento das frequências de pares de caracteres da linguagem.

No início dos anos 70, um grupo de pesquisadores liderados por Parks produziu um método de extração de características para construir uma lista de propriedades topológicas e um método hierarquicamente estruturado para reconhecimento. Em 1972 Tou e Gonzalez introduziram um esquema em dois estágios onde no primeiro, as medidas eram tomadas por meio de uma grade de medidas para separar as classes dos padrões em vários subgrupos e, no segundo estágio, um número de características especiais eram extraídas. Nos meados dos anos 70, Pavlidis, fazendo uso deste algoritmo "divide-e-une", produziu aproximações poligonais dos caracteres.

#### 14- RECONHECIMENTO DE FIGURAS:

#### 1.4.1- Aproximação de figuras:

Existem ocasiões que, ao examinar uma figura, estamos interessados somente em extrair uma descrição do que ela representa. A descrição desejada pode ser simplesmente uma classificação da figura em um pequeno conjunto de classes pré-especificadas; neste caso, isto pode ser realizado pela medida de várias propriedades da figura como um todo. Por outro lado, a descrição pode envolver propriedades e relações entre objetos que aparecem na figura. Para obter tal descrição, torna-se necessário explicitar a localização dos objetos componentes da figuras e extrair suas propriedades e relacionamentos.

Duas ferramentas básicas podem ser usadas na classificação e reconhecimento de figuras: amostragem (*sampling*) e quantificação. No processo de amostragem são retirados alguns pontos da figura de entrada e através de interpolação, uma função analiticamente simples pode ser extraída para descrever a função. No processo de quantificação é permitido que uma figura possua apenas um conjunto finito de valores em um ponto (níveis de cinza), substituindo o valor real de cada ponto pelo nível de quantificação mais próximo.

Um outro método que pode ser aplicado para aproximar uma figura é a expansão em série, utilizando apenas os primeiros termos da série como uma aproximação. Neste método, tenta-se encontrar uma função que descreva o contorno da figura. Uma vez que, em geral, é difícil obter uma função analítica que descreva exatamente este contorno, utiliza-se expansões em séries para tal descrição.

#### 1.4.2- Descrição de figuras:

Um problema que surge comumente na área de reconhecimento de padrões, consiste em determinar o quanto duas figuras se parecem ou encontrar uma parte de uma figura que case com outra figura, particularmente quando as figuras são altamente padronizadas. Entretanto, um dos problemas centrais no campo de processamento de figuras por computador é a descrição de figuras. Em geral, uma descrição consiste de um conjunto de declarações verdadeiras sobre a figura em alguma linguagem apropriada. Algumas vezes, entretanto, necessitamos somente associar a figura a uma determinada classe, onde por uma classe de figuras entende-se um conjunto de figuras que tenham propriedades, atributos, forma ou outras características em comum. O problema geral da classificação automática de padrões é encontrar propriedades que mapeiem o conjunto de figuras em um conjunto de classes. Uma propriedade de uma figura pode ser, por exemplo, um número real obtido pela aplicação de alguma função à figura. A decisão de quais propriedades são comuns em uma determinada classe dependem dos aspectos que a caracteriza e que, se não únicos, possuem medidas ou grandezas diferentes. Por vezes a definição de uma classe não é vinculada a uma única propriedade ou característica, mas sim a um conjunto delas.

Dado um conjunto de propriedades, é, em princípio, possível calcular o grau no qual cada propriedade contribui para a decisão na classificação. Se este cálculo for impraticável, pode-se avaliar a eficiência das propriedades empiricamente pela implementação ou simulação do sistema de reconhecimento.

O problema de selecionar um bom conjunto de propriedades é de maior relevância no reconhecimento de padrões, pois este conjunto pode definir com maior precisão, ou menor margem de erro, um determinado padrão. Em vários sistemas, as propriedades parecem ter sido selecionadas por seus tratamentos matemáticos simples ou por facilidade na implementação, do que pelas suas adequabilidades para a tarefa de classificação a que se destinam. Em outros casos, as propriedades têm sido designadas na tentativa de simular a fisiologia do percurso visual (ou seja, como os seres vivos identificam os objetos que estão à sua volta) ou para modelar a percepção visual. Por outro lado, vários problemas na classificação de padrões têm soluções simples baseadas em operações de pré-processamento ou aplicações de funções matemáticas às figuras, que quase certamente não carregam nenhuma semelhança com aquelas usadas pelos organismos vivos.

O melhor método é usar o conhecimento em relação à estrutura dos padrões e as definições das classes como um guia na escolha das propriedades. Por exemplo, se os padrões são desenhados a mão, certos tipos de propriedades são evidentemente mais apropriadas que outras, ou então, dois padrões devem pertencer à mesma classe se eles diferirem somente na posição, orientação ou tamanho, ou se eles puderem ser obtidos a partir da outra por uma transformação qualquer. Pode-se utilizar algumas propriedades métricas ou topológicas para classificação das figuras.

#### 1.4.3- Segmentação de Figuras:

Vários problemas de reconhecimento de padrões envolvem mais que simplesmente a designação de figuras em classes; eles requerem uma

descrição da figura em termos de fragmentos, onde o número de descrições é tão grande que torna impraticável considerar cada descrição como definindo uma classe. Tipicamente, uma descrição se refere a vários subconjuntos de fragmentos da figura e especifica propriedades destes subconjuntos. Para encontrar tal descrição, um sistema de reconhecimento automático deve ser capaz de fragmentar figuras (segmentação). Entretanto, não existe nenhum método universal para segmentação de uma figura [ROS73].

Existem alguns métodos possíveis para selecionar os pontos de quebra nos quais as bordas de uma figura podem ser segmentadas. Entre eles, podemos citar: posições extremas, pontos de inflexão, pontos de curvatura máxima, características de contorno ou casamento de contorno com padrões.

Qualquer figura é completamente determinada pela especificação de suas bordas como um conjunto de curvas dirigidas. Desta forma, uma figura pode ser descrita pelas equações destas curvas [ROS73].

## 15- METODOS DE RECONHECIMENTO DE PADROES:

Os métodos aplicados no reconhecimento de caracteres não são diferentes daqueles aplicados em qualquer problema de reconhecimento de padrões [MAN86]. As metodologias podem ser sumarizadas como se segue:

- (i) comparação global ponto a ponto [ISH87];
- (ii) transformações globais: Fourier, momento de inércia, ... [GRA72];
- (iii) extração das propriedades locais: pontos finais, ângulos pronunciados, quinas, etc. Estas propriedades podem ser extraídas por meio

de máscaras [IKE81][CHUA86];

(iv) busca através de certas linhas: as linhas podem formar retângulos ou ser simplesmente verticais ou horizontais [TOU72];

(v) análise por meio de curvaturas: análise geométrica [NEV79][YAM80];

(vi) métodos estruturais: decomposição do caracter em elementos constituintes, descrição topológica ou redução do caracter em um grafo [YHAB1][MUR86].

Os algoritmos de reconhecimento de figuras esboçadas a mão podem ser classificados em dois grupos: OFFLINE, que reconhecem figuras que já estejam prontas, e ONLINE, que reconhecem as figuras quando desenhadas em uma mesa digitalizadora, por exemplo. Algumas implementações impõem várias restrições ao esboço do desenho, como, por exemplo, o usuário ter de indicar a segmentação entre símbolos e ter de desenhar cada símbolo em uma determinada ordem e com um número pré-definido de traços. Estas restrições torna o processo de desenho da figura muito enfadonho.

O reconhecimento de padrões trabalha principalmente com a descrição e análise de medidas retiradas de processos físicos ou mentais. Para fornecer uma descrição efetiva e eficiente dos padrões, um pré-processamento geralmente é requerido para remover ruídos e redundâncias nas medidas. Então, um conjunto de características e relações entre estas medidas são extraídas para a representação dos padrões. A análise (classificação e/ou descrição) dos padrões é realizado com base nesta representação.

No reconhecimento online, as figuras são representadas por linhas sucessivas. Por esta razão não é necessário um pré-processamento para extração de contornos, processos estes que são muito caros em termos de

tempo de computação, além de serem imperfeitos.

Para determinar um conjunto adequado de características e suas relações para a representação de padrões, de tal forma a proporcionar uma alta taxa de reconhecimento, é necessário uma análise cuidadosa dos padrões em estudo. Conhecimento sobre as características e as estruturas dos padrões deve ser amplamente utilizado. Deste ponto de vista, o estudo do reconhecimento de padrão inclui a análise de suas características e o projeto de sistemas de reconhecimento.

Existem vários métodos matemáticos propostos para solucionar os problemas de reconhecimento. Estes métodos podem ser agrupados em duas classes: decisão-teórico ou discriminante, e sintático ou estrutural. De um ponto de vista mais geral, esses métodos podem ser discutidos, como feito abaixo, dentro da mesma estrutura em termos de representação dos padrões e análise estrutural e/ou tomada de decisão (decision-making). O subproblema de representação envolve primariamente a seleção da representação. O subproblema de tomada de decisão (e/ou análise estrutural) envolve primariamente a seleção de medida de similaridade.

Um método intuitivo usado nos primeiros desenvolvimentos é o método de casamento de padrões (*matching*). O processo de decisão é o simples casamento da entrada desconhecida com cada padrão, e o critério usado reflete diretamente a semelhança entre os dois (comparação ponto a ponto). A diferença entre o método decisão-teórico e o método sintático aparece logo nos tipos de atributos que são procurados nas figuras para o reconhecimento. Enquanto a abordagem decisão-teórica envolve-se com atributos essencialmente numéricos e globais das figuras, como por exemplo, a sua área e a dispersão de seus pontos em relação ao seu centro de massa, a abordagem estrutural ou sintática procura extrair elementos dessas

figuras que expressem a sua estrutura, tais como ângulos, curvas e arcos. Essa manipulação de diferentes atributos pelas duas abordagens traz consigo algoritmos matemáticos de classificação bastante distintos. Enquanto a abordagem decisão-teórica usa algoritmos matemáticos de classificação dos padrões, a abordagem estrutural ocupa-se de algoritmos baseados na teoria formal de linguagens [RUI89].

No método decisão-teórico, um padrão é representado por  $N$  características ou por um vetor característico  $N$ -dimensional e o processo de decisão é baseado na medida de similaridade que, geralmente, é expressa em termos de uma medida de desvio, por uma função de probabilidade ou por uma função de distribuição. No método sintático, um padrão é representado como uma cadeia, uma árvore ou um grafo de padrões primitivos e suas relações. Este método traça uma analogia entre a estrutura dos padrões e a sintaxe de uma linguagem, e o processo de decisão geralmente é um procedimento de análise (parsing).

A tabela 1.1 resume os métodos sob um ponto de vista geral, enquanto que a tabela 1.2 fornece os pontos positivos e negativos dos métodos com respeito à implementação e à tolerância às distorções.

As tabelas 1.1 e 1.2 apresentam as diferentes técnicas usadas para o reconhecimento de caracteres e compara seus méritos baseados nos seguintes critérios:

- 1) sensibilidade à deformação da imagem: ruído, distorção, variação no estilo, translação e rotação da imagem, etc;
- 2) implementação prática da técnica: construção de máscaras (para casamento), velocidade, complexidade da implementação, independência de outras técnicas auxiliares;

## CLASSIFICACAO DOS METODOS DE RECONHECIMENTO DE PADROES

METODO	RECONHECIMENTO	PROCESSO DE DECISAO
casamento de padrao	dados crus	casamento direto
decisao-teorico (discrimin.)	vetor de caracteristicas	funcao discriminante, distancia minima, probabilidade maxima, etc...
simatico (estrutural)	string, arvore, grafo	derivacao (parsing) casamento de grafos

**TABELA 1.1**

## METODOS DE RECONHECIMENTO

CRITERIO	TOLERANCIA A DISTORCAO					FACILIDADE DE IMPLEMENTACAO			
	RUIDO	DISTORCAO	ESTILO	TRANS-LACAO	ROTA-CAO	MASCAS	VELOCIDADES	COMPLEXIDADE	INDEPENDENCIA
C A R A C T E R I S T I C A S      G L O B A I S									
CASAMENTO DE PADRAO	+	0	-	-	-	+	-	0	-
TRANSFORMACAO	-	+	+	+	+	0	-	-	0
D I S T R I B U I C A O      D E      P O N T O S									
ESQUEMAMENTO	-	0	-	-	0	-	+	+	-
MOMENTOS	0	0	-	+	+	0	-	0	-
N-TUPLA	0	-	0	-	0	-	+	+	0
CARACT. LOCI	-	+	+	+	0	0	+	+	-
CRUZAMENTO E DISTANC.	-	+	+	+	0	-	+	+	-
CARACT. GEOMETRICAS E TOPOLOGICAS	-	+	+	+	0	-	+	0	+

ONDE:  
 - : dificil implementacao ou pouco tolerante;  
 0 : medio;  
 + : facil implementacao ou muito tolerante.

**TABELA 1.2**

### 1.5.1- Casamento de Padrão:

Este método utiliza-se das características globais do padrão para realizar o reconhecimento. Tais características não refletem qualquer propriedade local, geométrica ou topológica do esboço.

O método consiste em observar as características do estado (ligado/desligado) de todos os pontos dentro do retângulo envolvente do carácter. Ele simplesmente mede a semelhança do carácter de entrada e as referências armazenadas. Este método requer uma alta cardinalidade do conjunto de características. Para representar adequadamente um carácter escrito a mão, por exemplo, na forma digital, pelo menos 500 bits (matriz 20 x 25) de memória são necessários. Esta forma de reconhecimento é um processo muito caro em termos de esforço computacional. Estes fatores, aliados à falta de equipamento digitalizadores, forçaram vários pioneiros a trabalharem sobre um número limitado de amostras. Seus resultados não foram nada conclusivos. Felizmente, alguns bancos de dados de caracteres escritos a mão estão disponíveis e têm estimulado uma pesquisa maior neste campo. Como exemplos de banco de dados de caracteres implementados, onde o reconhecimento é realizado usando o casamento de padrões, citamos [SUE80]:

1- Munson: cada carácter é armazenado em uma matriz 24x24. Este banco contém 12740 exemplos de caracteres utilizados na linguagem FORTRAN (64 caracteres). A taxa de reconhecimento é de 85% em média.

2- Highleyman: cada carácter é armazenado em uma matriz 12x12,

contendo um total de 1800 exemplos de letras e 500 exemplos de números para 36 caracteres alfanuméricos. A taxa de reconhecimento de 70%.

3- outros: Mori, Krause, Suen - cada um com mais de 100000 exemplos de caracteres alfanuméricos.

Um dos problemas em usar os próprios pontos como características (casamento de padrão) é a alta dimensão do vetor de características resultante. Podemos reduzir a cardinalidade deste vetor através da extração de características globais, que podem ser obtidas facilmente por transformações e expansões em séries dos dados de entrada.

Em termos de implementação, estas características são essencialmente inafetadas por pequenas variações locais e possuem baixa sensibilidade aos ruídos. Além do mais, expansões e transformações fornecem alguma liberdade em relação a rotação e translação. Entretanto, elas têm alta sensibilidade à distorção e variação no estilo. Sob o ponto de vista prático, é fácil construir máscaras para estas técnicas, mas a velocidade é baixa e geralmente são necessárias técnicas complementares ou características adicionais para que reconhecedores baseados neste esquema de reconhecimento mantenham um desempenho satisfatório.

#### 1.5.2- Métodos Decisão-Teórico:

Um diagrama de um sistema baseado em Métodos Decisão-Teórico pode ser visto na figura 1.2. A fase de pré-processamento geralmente é tratada na área de processamento de sinais e imagem. Nossa discussão está limitada

à extração e seleção das características.

A extração das características relevantes para a classificação é um problema que depende dos padrões e do número de classes sob estudo. Nesta abordagem os objetos são caracterizados por medidas quantitativas de seus atributos e dos aspectos que se julgar mais significativos para a discriminação desses objetos. Esta abordagem procura determinar, baseada nestes atributos, as medidas que descrevem um padrão específico, com menor erro possível, dentro de uma determinada classe. Esta classificação é feita formulando-se regras ou funções de decisão aplicadas às grandezas dos atributos medidos. Em [RUI89] temos alguns exemplos de atributos empregados nessa abordagem.

Métodos gerais para a extração de características podem ser considerados como pertencentes ao método das transformações especiais ou das medidas de desvio. O propósito do método de transformações especiais é de transformar o espaço característico original em um espaço de dimensão menor para a representação e/ou discriminação de classe. Tanto as transformações lineares, como as não-lineares foram sugeridas. Fourier, Walsh-Hadamard e Haar para geração de características, e a expansão de Karhunen-Loeve e o método das componentes principais para redução da dimensão do espaço característico são algumas transformações frequentemente utilizadas [FU-80].

O principal objetivo da seleção das características é extrair um subconjunto de  $L$  características a partir de um conjunto dado de  $N$  características ( $L < N$ ) sem uma degradação significativa do desempenho do sistema de reconhecimento. Infelizmente, um cálculo direto da probabilidade

## METODOS DECISAO-TEORICO

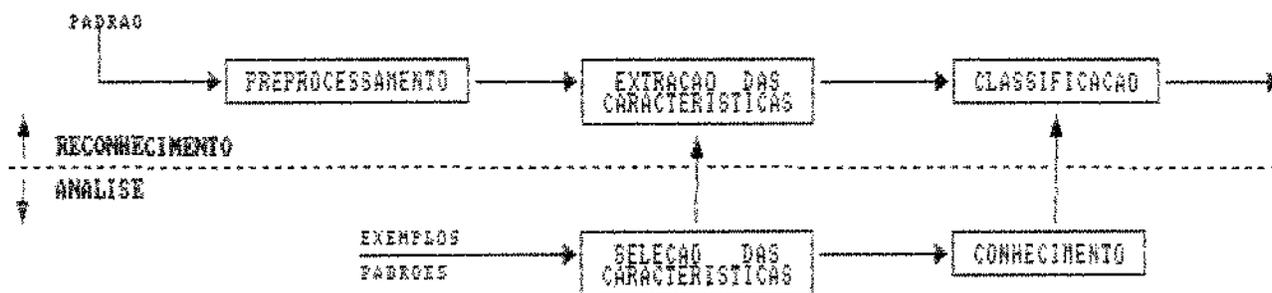


FIGURA 1.2

---

## METODOS SINTATICOS

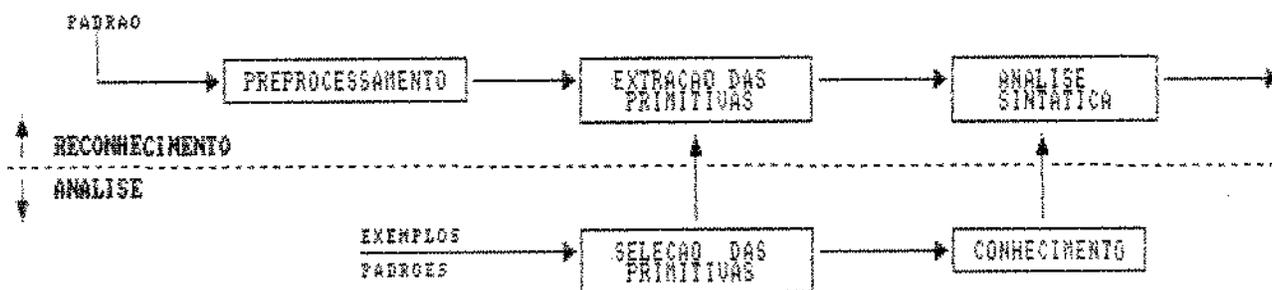


FIGURA 1.3

de reconhecimento é freqüentemente impossível ou impraticável, parcialmente devido à falta de expressões analíticas gerais, pois as existentes são simples demais para serem consideradas. Uma solução é encontrar um critério indireto para guiar a seleção das características.

O método mais comum é definir uma medida de desvio, que está relacionada à maior e/ou menor probabilidade de não reconhecimento para a seleção da característica. Isto é, o melhor subconjunto é selecionado de forma a maximizar a medida pré-especificada. Em [KAR74] e [RUI89] temos alguns exemplos de medidas de desvios que poderiam ser utilizados por este método.

### 1.5.3- Métodos Sintáticos:

Um diagrama de um sistema baseado em Métodos Sintáticos pode ser visto na figura 1.3. Neste método, um padrão é representado por uma sentença na linguagem especificada por uma gramática.

O método sintático busca o reconhecimento de objetos através de uma descrição de seus elementos mais complexos em termos de elementos mais simples, dada sua forma ou aspecto físico. Neste caso não são os atributos do objeto que determinam sua classificação, como no método decisão-teórico, mas sim a estrutura de todo o padrão é que alimentará o processo classificatório.

Segmentos são considerados boas primitivas (componentes básicos de um padrão) para desenhos feitos a mão, assim como os fonemas para a

fala. Entretanto, nenhum dos dois pode ser facilmente extraído pela máquina. O problema de segmentação ainda é objeto de estudo. Até o momento não existe uma solução geral para o problema da seleção da primitiva.

Após a seleção das primitivas, o próximo passo é a construção da gramática que irá definir uma linguagem para descrever os padrões sob estudo. A seleção de uma gramática particular é afetada pelas primitivas selecionadas e pelo poder de descrição da gramática e eficiência na análise. Seria ideal que existisse uma máquina de inferência gramatical que pudesse inferir uma gramática ou descrição estrutural a partir de um conjunto de padrões dado (esta máquina só existe para casos muito especiais).

Devido aos ruídos, variações nas medidas padrões, erros de segmentação e extração de primitivas temos ambiguidades nas linguagens de descrição dos padrões.

## 16- COMPARACAO ENTRE OS METODOS:

A forma mais simples de reconhecimento é, possivelmente, o casamento de padrões. Baseado em um critério de semelhança, o padrão de entrada é classificado como um padrão definido na biblioteca. Agora, se uma descrição completa do padrão é requerida para o reconhecimento, uma análise sintática é necessária. Entre estes dois extremos, existem vários métodos. Por exemplo, uma série de testes pode ser projetada para testar a ocorrência ou não de certos sub-padrões (primitivas) ou certas combinações entre elas. O procedimento de análise sintática (parsing) para o

reconhecimento, geralmente não é determinístico, aumentando a ineficiência computacional.

O desenvolvimento entre a teoria e a prática em reconhecimento de padrões não esteve muito balanceado nos últimos anos. Vários resultados teóricos, especialmente em conexão com o método decisão-teórico, foram publicados. Aplicações práticas foram gradualmente enfatizadas durante as duas últimas décadas, particularmente nas áreas médicas, industrial e monitoramento remoto. A maioria dos resultados práticos são considerados não conclusivos e necessitam de refinamentos [FU-80].

Comparado com o reconhecimento decisão-teórico, o reconhecimento sintático é uma área de pesquisa mais recente. Quando os padrões são complexos e o número de classes é muito grande, seria vantajoso descrever cada padrão em termos de suas componentes e considerar além da classificação, a descrição dos padrões. A utilidade prática do método sintático depende da nossa habilidade em reconhecer as primitivas simples e representar suas relações por composição de operações.

No método decisão-teórico ainda procura-se por uma extração de características efetiva e eficiente e por técnicas de seleção. A complexidade computacional dos sistemas de reconhecimento de padrões poderia ser um objeto interessante para investigações. No método sintático, o problema de extração e seleção de primitivas necessita de maiores atenções. Uma seleção apropriada da gramática padrão afeta diretamente a complexidade ou a eficiência do resultado. Algoritmos de inferência gramaticais que sejam computacionalmente possíveis ainda estão por vir.

Em aplicações práticas, é possível que o método de decisão-teórico e o método sintático sejam usados juntos. O primeiro método seria aplicado para a extração ou reconhecimento das primitivas, enquanto que o segundo método seria usado para a análise estrutural e reconhecimento dos padrões. Quando as primitivas não puderem ser extraídas com alta confiabilidade, analisadores sintáticos com correção de erro podem ser usados para o reconhecimento de padrões distorcidos.

## 1.7- METODO PROPOSTO:

Como já foi dito, os métodos aplicados no reconhecimento de caracteres não são diferentes daqueles aplicados em qualquer problema de reconhecimento de padrões. O método que será proposto no próximo capítulo para o reconhecimento de figuras esboçadas a mão, objeto desta dissertação, utilizará algumas propriedades dos métodos vistos anteriormente. Ele está baseado na extração dos segmentos componentes de cada subfigura padrão do desenho de entrada e dependendo do posicionamento entre eles, e de algumas de suas propriedades geométricas, o desenho será classificado (reconhecido). Do ponto de vista de classificação, o método pode ser considerado como sendo um método sintático, embora não utilize explicitamente uma análise sintática para o reconhecimento, nem uma gramática para definição dos padrões.

O método foi concebido na tentativa de simular o reconhecimento feito pelos seres humanos baseando-se exclusivamente na forma e disposição dos componentes do desenho para o seu reconhecimento. Desta forma, nenhuma aplicação de transformações ao desenho de entrada se fará necessário e,

como a aquisição dos dados será realizada diretamente, a partir de um esboço feito pelo usuário, não necessitaremos de uma etapa prévia de preprocessamento da imagem, como detecção de bordas ou quantificação de níveis de cinza, por exemplo.

O método está baseado nas características geométricas do desenho, sendo tolerante à distorção, à variação no estilo e à translação. Durante a concepção do método, não nos preocupamos muito com a complexidade de implementação que porventura o método viesse a ter, nem com a velocidade de reconhecimento. O principal objetivo foi o de propor um método de fácil entendimento que fosse independente de outras técnicas para manter um índice razoável de reconhecimento, de fácil expansão e que simulasse o reconhecimento utilizado pelos humanos.

### METODO DE RECONHECIMENTO

#### 2.1- FIGURAS DE ENTRADA:

Alguns métodos convencionais *on-line* de reconhecimento de figuras esboçadas a mão impõem várias restrições na entrada do desenho, tais como:

- 1- indicar os pontos de segmentação entre os vários símbolos;
- 2- cada símbolo dever ser traçado com um número pré-determinado de segmentos (ou riscos) e/ou em uma ordem pré-estabelecida.

Estas restrições resultam em um procedimento de entrada muito rude, totalmente artificial sob o ponto de vista do usuário, constituindo-se em um sistema não-amigável.

Neste capítulo descreveremos um método *on-line* para o reconhecimento de figuras esboçadas a mão. Este método está sendo proposto para o reconhecimento de figuras formadas por um conjunto de subfiguras padrões ligadas entre si, tais como fluxogramas, circuitos digitais ou elétricos, redes de Petri, diagramas representando um modelo ER em banco de dados ou qualquer figura formada por padrões definidos pelo usuário. O conjunto de padrões que compõe a figura de entrada deve ser finito, onde os elementos possuem formatos bem definidos e "distintos" entre si. O reconhecimento da figura de entrada consistirá no reconhecimento dos padrões componentes e na determinação do modo no qual eles estão conectados.

A entrada das figuras a serem reconhecidas poderá ser realizada por uma mesa digitalizadora, *mouse*, caneta óptica, teclado ou por qualquer outro dispositivo semelhante que suporte uma entrada bidimensional, ou seja, que possam seleccionar algum ponto na tela. Internamente esta figura será representada por uma seqüência de pontos coordenados.

O método foi proposto de tal forma que a entrada fosse realizada de um modo mais natural possível, onde nenhuma restrição estivesse presente - é como se a figura estivesse sendo desenhada sobre uma folha de papel, utilizando-se um lápis. Além de permitir uma liberdade total durante a fase de entrada da figura, o método tenta simular o comportamento natural das pessoas para o reconhecimento dos padrões. Ao que parece, os seres humanos utilizam apenas o formato dos padrões e um conhecimento prévio sobre o tipo das figuras para reconhecê-las. Ou seja, sabendo o que cada padrão representa, quais os padrões que podem formar a figura e a forma geométrica desses padrões, somos capazes de distinguir, um a um, todos os padrões utilizados para compor o desenho, desde que este tenha sido esboçado com o devido cuidado, isto é, com uma distorção razoavelmente pequena. Portanto, basear-nos-emos nas formas geométricas dos desenhos e em algumas propriedades topológicas para o seu reconhecimento. Nenhuma outra informação ou técnica adicional será utilizada.

Para reconhecer a figura desenhada é necessário o reconhecimento dos padrões componentes (denominados subfigura componentes ou simplesmente subfiguras). Portanto, devemos ser capazes de extrair estes padrões a partir do desenho de entrada. O processo de extração das partes componentes de um desenho é chamado de segmentação - segmentar uma figura significa decompor esta figura em partes componentes mais elementares. A figura de

entrada será decomposta em subfiguras, que por sua vez, serão decompostas em segmentos. Para permitir uma liberdade maior durante a entrada da figura, serão observadas as seguintes regras :

- 1- nenhuma indicação da segmentação entre subfiguras e segmentos é dada;
- 2- tamanhos e posições dos símbolos são arbitrários (formato livre);

Uma vez que os seres humanos podem reconhecer figuras que não estejam muito bem desenhadas, o método deve ter pouca sensibilidade a distorções do desenho, ou seja, ele deve ser capaz de reconhecer subfiguras com contornos distorcidos. A capacidade de tolerar distorções varia de pessoa para pessoa. Dependendo do montante de informações sobre a semântica dos padrões, uma pessoa poderá reconhecer mais precisamente os padrões do que outras com uma experiência menor. Mesmo assim, em muitos casos, temos de recorrer à adivinhação para o reconhecimento de algumas subfiguras, o que nos pode levar a erros de reconhecimento.

Como o método não pode adivinhar a intenção do usuário, seria interessante aproveitar a semântica das subfiguras para o reconhecimento. Isto pode ser feito através da introdução de regras de formação das figuras. Estas regras indicarão quais subfiguras podem estar conectadas entre si, o número máximo e mínimo de conexões por subfigura, além de informações geométricas adicionais. Mais tarde veremos como estas regras poderão ser utilizadas no reconhecimento. Um esquema do método pode ser visto na figura 2.1.

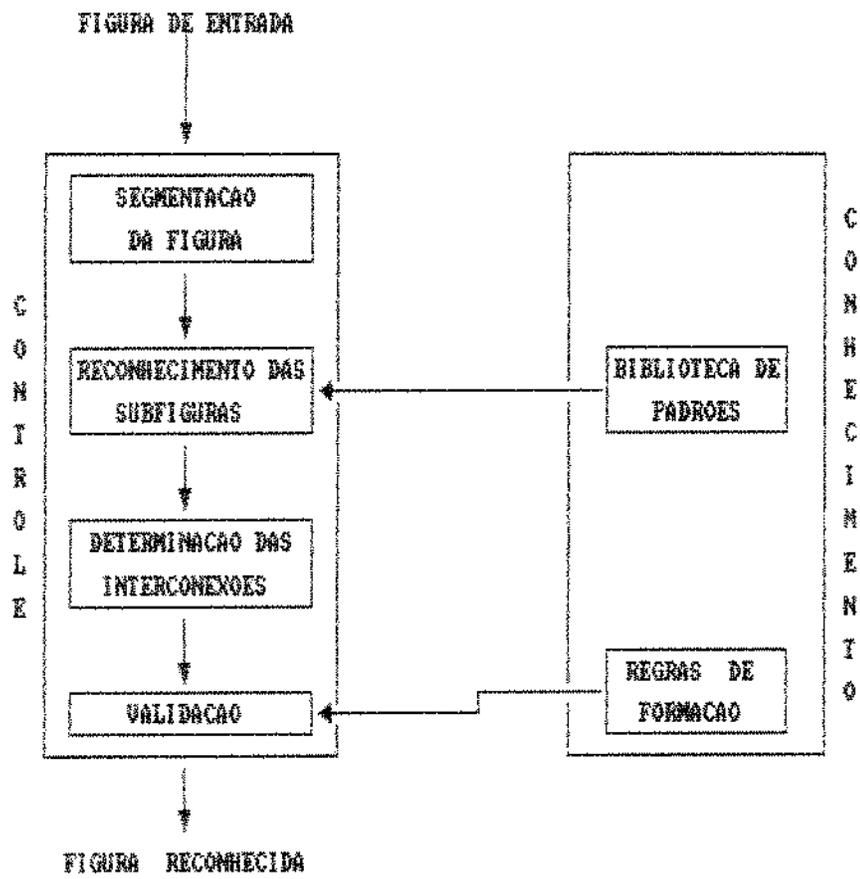


FIGURA 2.1

## 2.2- BIBLIOTECA DE PADORES:

O método utilizado para o reconhecimento foi proposto de tal forma que ele seja implementável em um microcomputador e de fácil entendimento. Como os recursos existentes em um microcomputador são muito limitados, não podemos utilizar o casamento de padrões para reconhecer a figura de entrada, uma vez que isto exige uma biblioteca muito grande, onde teríamos dezenas de exemplos armazenados para um mesmo padrão. Uma outra desvantagem é que necessitaríamos de uma etapa prévia de "aprendizado" da máquina ou do usuário. Além disso, como foi visto no capítulo anterior, é muito difícil obter exemplos significativos destes padrões para compor a biblioteca, a tolerância a distorções é pequena e o tempo para o reconhecimento é grande, comparado a outras técnicas existentes.

Para que o método seja de fácil compreensão, é desejável que o desenho de entrada não sofra nenhuma transformação matemática e que nenhuma outra técnica auxiliar seja utilizada para reconhecer as figuras.

Uma vez que pretendemos reconhecer vários tipos de padrões, devem existir informações que possibilitem este reconhecimento. Estes dados estarão armazenados em uma biblioteca de padrões com as seguintes características:

1- todo padrão que pode compor uma figura de entrada será descrito apenas uma vez na biblioteca;

2- cada padrão será descrito pelo tipo de seus componentes (segmentos) e posição relativa entre eles;

- 3- não existirá nenhuma informação relativa ao tamanho ou ordem na qual os segmentos componentes do padrão deverão ser desenhados;
- 4- esta biblioteca será comum a todos os usuários;
- 5- existirá uma biblioteca para cada tipo de aplicação.

Na biblioteca, os dados referentes aos padrões serão:

a) padrões simples:

- 1- nome do padrão;
- 2- número de segmentos componentes;
- 3- tipo de segmentos;
- 4- direção e sentido (se arco) desses segmentos;
- 5- propriedades geométricas entre os segmentos componentes;
- 6- descrição dos pontos de E/S.

b) padrões compostos:

- 1- nome do padrão;
- 2- nomes dos padrões componentes;
- 3- propriedades entre os componentes;
- 4- descrição dos pontos de E/S.

No caso de padrões compostos, um padrão componente poderá ser um padrão simples ou um padrão composto.

Como cada padrão só será descrito uma única vez e será comum a todos os usuários, temos que o tamanho da biblioteca será reduzido. Este fato é fundamental, pois, uma vez que este método foi proposto para ser implementável em um micro-computador, não temos uma grande quantidade de memória disponível. Tendo um tamanho reduzido, a biblioteca de padrões

poderá permanecer na memória do computador durante o processo de reconhecimento, aumentando assim, sua eficiência temporal.

Caso seja necessário, um padrão poderá ser descrito mais de uma vez na biblioteca, já que o sistema não faz nenhum teste para garantir a unicidade dos padrões nela descritos. A necessidade de descrição múltipla de um padrão poderá surgir na definição dos padrões compostos, pois estes poderão ser decompostos, de maneiras diferentes, em vários subpadrões. Como o reconhecimento basear-se-á somente nos dados cadastrados na biblioteca, cada padrão composto deverá ser descrito de tantas maneiras quantas possíveis for sua decomposição, sob pena de não funcionamento do reconhecedor, caso esta descrição não for completa.

Mesmo com esta multiplicidade de definições para alguns padrões, o tamanho da biblioteca permanecerá reduzido, pois a cardinalidade do conjunto formado pelas diferentes maneiras de decomposição dos padrões é pequena e nem todos terão esta multiplicidade de descrição.

Os dados contidos nesta biblioteca refletem o método de reconhecimento utilizado, pois não existem restrições quanto ao tamanho ou ordem na qual a figura foi desenhada e as informações nela contida são puramente geométricas, referindo-se exclusivamente ao formato dos padrões. Adicionalmente, esta biblioteca conterá dados referentes a algumas propriedades que os segmentos componentes dos padrões devem apresentar e dados sobre os locais utilizados para conexão dos padrões (pontos de E/S). Estas informações serão úteis para a discriminação entre padrões com formatos semelhantes, que não podem ser diferenciados apenas com a descrição de seus segmentos componentes.

A grande vantagem em se utilizar uma biblioteca para o reconhecimento é que o usuário pode expandir os tipos de figuras reconhecíveis criando apenas novas bibliotecas ou modificando as bibliotecas já existentes. Já que cada padrão é descrito unicamente pelo seu formato, a construção ou alteração das bibliotecas é uma tarefa relativamente fácil, como será visto no próximo capítulo.

Os dados contidos na biblioteca são suficientes para descreverem univocamente um padrão que possa compor a figura de entrada. Como o reconhecimento está baseado nos segmentos que formam cada subfigura, uma descrição desses segmentos é suficiente para descrever os padrões. Caso exista dúvida no reconhecimento baseado unicamente nesta descrição dos componentes, as propriedades cadastradas na biblioteca referentes aos segmentos componentes, devem eliminar esta incerteza. Desta forma, já que os segmentos e suas propriedades que se julgarem relevantes ao reconhecimento estão completamente definidos pelos dados contidos na biblioteca, não necessitaremos de informações adicionais para realizar o reconhecimento. A construção da biblioteca foi baseada nos dados que julgamos ser relevantes para o reconhecimento dos padrões, refletindo diretamente o método proposto. A utilização deste conjunto de dados, não descarta a existência de outros dados que podem ajudar no reconhecimento.

### **2.3- SEGMENTAÇÃO DA FIGURA DE ENTRADA:**

Como a biblioteca de padrões descreve os segmentos que compõe a subfigura, é necessário que o desenho de entrada seja segmentado, ou seja,

devemos extrair os elementos fundamentais que compõe o desenho, que no nosso caso, esses elementos são segmentos. Uma vez que somente os segmentos obtidos na fase de segmentação serão utilizados para o reconhecimento, devemos garantir que este processo seja o mais preciso possível. Como as figuras são esboçadas a mão, a segmentação deve ser tolerante a ruídos e distorções. A fase de segmentação é crítica, pois, para que o funcionamento do reconhecedor proposto seja satisfatório, devemos garantir um processo de segmentação eficiente, uma vez que o reconhecimento da figura de entrada será feito utilizando-se exclusivamente do tipo e posicionamento desses segmentos.

Entretanto, é muito difícil obter um método de segmentação eficiente, pois não existe um método que seja geral [ROSS73]. Além disso, como as figuras serão esboçadas a mão, o método utilizado deve ser tolerante às distorções da figura de entrada para evitar que ocorram erros de segmentação. Existem vários métodos de segmentação citados na literatura, mas a maioria deles são eficazes quase que exclusivamente para a aplicação alvo para os quais eles foram criados. Se aplicarmos um método de segmentação pouco tolerante a distorções em uma figura esboçada a mão, muito provavelmente este método irá extrair da figura mais segmentos do que deveria, uma vez que um segmento de reta esboçado a mão, por exemplo, poderá ser segmentado em vários outros segmentos devido à não linearidade do segmento esboçado. O problema em utilizar um processo de segmentação semelhante ao descrito anteriormente, consiste no excesso de segmentos que serão obtidos, ou seja, um segmento componente da figura poderá ser segmentado em vários outros.

Uma possível solução para o problema da segmentação seria a realização de uma etapa de averiguação do resultado do método utilizado, onde, nesta etapa, haveria uma fusão dos segmentos extras que foram obtidos devido à distorção do desenho. Desta forma, os segmentos que foram quebrados em vários outros serão recompostos, corrigindo-se assim, alguns dos erros decorrentes da pouca tolerância à distorção do processo de segmentação utilizado.

Para o caso das subfiguras que compõem o desenho de entrada serem todos fechados, poder-se-ia fazer esta fusão dos segmentos somente após encontrar alguma subfigura componente, pois, sabendo-se que os segmentos dados formam uma subfigura, poderíamos abstrair dos detalhes de cada segmento e preocupar-nos com o conjunto de segmentos como um todo. Seria como observar a subfigura de uma distância segura, onde, baseado no formato desta subfigura, a fusão dos segmentos seria facilitada.

Após a figura ser segmentada, cada segmento extraído será aproximado por um segmento de reta ou arco de circunferência. Com esta aproximação, o processo de reconhecimento é simplificado, pois teremos de comparar apenas dois tipos de segmentos durante a busca na biblioteca de padrões. Esta aproximação não restringe a aplicabilidade do método, pois quase a totalidade das figuras formadas por padrões nas aplicações para as quais se destina o reconhecedor proposto, possuem componentes que são formadas por segmentos de retas ou arcos de circunferência ou que podem ser aproximados por eles sem maiores problemas. Além do mais, caso não houvesse esta aproximação e tentássemos reconhecer uma variedade maior de curvas, o reconhecimento seria muito prejudicado, pois em um esboço a mão onde todos os desenhos estão distorcidos, diferenciar curvas como hipérbolas,

parábolas ou elipsóides é uma tarefa muito difícil e provavelmente necessitaríamos de técnicas auxiliares para garantir um grau de reconhecimento aceitável.

Como consequência desta aproximação dos segmentos componentes, temos uma tolerância à distorção já embutida no método. A biblioteca de padrões, usada para o reconhecimento, contém as informações geométricas e topológicas de cada padrão. Como vimos, ela descreve os segmentos componentes de cada padrão, qual o tipo desses segmentos e qual o posicionamento relativo entre eles. Desta forma, ao se desenhar um segmento qualquer, estaremos interessados apenas em sua forma e direção, não interessando o seu tamanho ou abertura, por exemplo. Desejamos saber apenas se este segmento é um segmento de reta ou um arco de circunferência, independentemente se ele está bem desenhado ou não - a distorção deste segmento é irrelevante para o reconhecimento.

Após a segmentação da figura de entrada e da aproximação desses segmentos por um segmento de reta ou arco de circunferência, temos de ajustar a direção dos segmentos, uma vez que estes segmentos poderão ter apenas algumas direções possíveis. Fazemos um ajuste na direção de um segmento caso a orientação deste não esteja coincidindo com uma das orientações permitidas. A quantização da direção é feita para simplificar o processo de casamento deste segmento com algum segmento da figura padrão, sendo este um artifício muito usado em sistemas de reconhecimento de padrão.

A direção de um segmento é dada pela direção do segmento de reta que une os seus pontos extremos. Com a quantização da direção, introduzimos

uma tolerância na direção dos segmentos esboçados na entrada, pois segmentos que estejam próximos de uma determinada direção são considerados como que possuindo aquela direção específica. O número de direções possíveis pode variar de acordo com a implementação. Caso o número de direções seja muito pequeno, o número de padrões possíveis para representação fica prejudicado devido à esta restrição. Caso o número de direções seja muito elevado, a tolerância na direção dos segmentos fica pequena, pois segmentos com direções próximas podem ser considerados distintos. O número de direções deve ser tal que concilie a tolerância à distorção com o poder de descrição dos padrões. Caso seja necessário, as direções possíveis não precisam estar uniformemente distribuídas no plano. Isto pode ser útil quando, por exemplo, deseja-se uma tolerância maior em segmentos diagonais e uma melhor discriminação em segmentos horizontais ou verticais. Neste caso, poder-se-ia concentrar as direções possíveis nas proximidades dos eixos cartesianos. Em todo caso, esta escolha vai depender da aplicação alvo desejada.

Através das aproximações feitas até agora, além de uma razoável tolerância a distorções, a quantidade de dados manipuláveis ficou reduzida, pois estamos tratando apenas com segmentos de retas ou arcos de circunferências em algumas direções específicas, não interessando dados extras tais como o tamanho ou ordem na qual estes segmentos foram desenhados. Com isto, o processo de identificação fica facilitado, já que os detalhes de cada segmento são totalmente irrelevantes.

Os processos de segmentação, a aproximação por um segmento de reta ou arco e a quantização da direção são independentes, podem ser realizados simultaneamente durante a entrada do desenho, diminuindo assim o

tempo de reconhecimento.

## 2.4- RECONHECIMENTO DOS PADROES SIMPLES:

Após a figura de entrada ter sido segmentada, temos uma sequência de segmentos descrevendo esta figura. Como não existe nenhuma restrição de como a figura foi desenhada, os segmentos que compõem um determinado padrão estarão espalhados nesta sequência. Além do mais, pode ser que um segmento esteja quebrado, isto é, podem existir dois ou mais segmentos de entrada representando um único segmento componente de um padrão. Neste caso, antes da tentativa de reconhecimento é necessário percorrer a sequência de entrada e fundir os segmentos fragmentados. Na descrição do reconhecimento dada a seguir, supomos que não existem segmentos fragmentados e que não existe nenhuma subfigura componente da figura de entrada que não esteja cadastrada na biblioteca de padrões.

O método para reconhecimento de figuras simples descrito abaixo é ineficiente em termos de implementação, mas de uso geral. Como veremos adiante, na implementação do modelo foi feita uma restrição na entrada do desenho, visando uma melhoria na velocidade de reconhecimento.

Dado um segmento da sequência de entrada, procuramos na biblioteca por todos os padrões que são formados por um único segmento. O segmento dado será comparado com o segmento que compõe cada um dos padrões selecionados. Caso os dados deste segmento seja compatível com os dados do segmento componente de um determinado padrão, ou seja, eles são do mesmo tipo, possuem direções e aberturas (se arco) compatíveis, então este

segmento é considerado como sendo o padrão analisado. Caso não haja compatibilidade, temos que o segmento dado não forma nenhum padrão.

Neste caso, procuramos na sequência de entrada por outro segmento que esteja conectado ao segmento já selecionado e testamos se existe algum padrão que possa ser formado pelo último segmento selecionado (dois segmentos estão conectados se eles possuírem uma extremidade em comum). Caso exista algum, este segmento será retirado da sequência de entrada e substituído pelo padrão encontrado. Caso o segundo segmento selecionado não forme nenhum padrão, testamos se existe algum padrão que possa ser formado pelos dois segmentos. Se existir, esses segmentos são substituídos na sequência de entrada pelo padrão correspondente.

Caso esses segmentos não formarem nenhum padrão, procuramos na sequência de entrada por um terceiro segmento conectado a um dos dois segmentos já selecionados e verificamos se existe algum padrão de um, dois ou três segmentos que possa ser formado pelos segmentos dados. Se existir, os segmentos são substituídos pelo padrão correspondente na sequência de entrada. O processo acima se repete até que se acabem os segmentos que de alguma forma estão conectados ao primeiro segmento dado ou então que se encontre um padrão que tenha o primeiro segmento como segmento componente. Se não existir nenhum padrão que possa ser formado utilizando o primeiro segmento, ele será considerado como sendo um conector e será substituído na sequência de entrada por um conector.

Então, seleciona-se o próximo segmento da sequência de entrada e repete-se todo o processo descrito acima, até que todos os segmentos sejam substituídos por padrões ou por conectores.

Observamos que esta procura por um padrão é muito ineficiente, pois a cada segmento selecionado, toda a biblioteca de padrões é analisada. Isto é necessário devido à liberdade total existente na entrada da figura. Seria aconselhável colocar alguma restrição na entrada de figura que facilite a procura por um padrão e que não retrinja demasiadamente a liberdade durante a fase de desenho da figura.

Para possibilitar a correção de erros que porventura venha a ocorrer nesta fase é aconselhável que, ao invés de substituir os segmentos pelos padrões correspondentes na seqüência de entrada, seja construído uma seqüência de padrões a partir dos padrões que vão sendo encontrados, mantendo a seqüência dos segmentos de entrada intacta, indicando os segmentos que foram utilizados para contruir um determinado padrão na seqüência contruída. O processo de reconhecimento de figuras simples pode ser esquematizado conforme o algoritmo mostrado a seguir.

```
enquanto existirem segmentos que não formam fig. reconhecida faça
  selecionar próximo segmento de entrada;
  comparar segmento selecionado com figuras padrão de 1 segmento;
  se encontrou figura
    então
      colocar padrão reconhecido na seqüência de figuras já
        reconhecidas;
    senão
      enquanto existirem segmentos selecionados e não encontrou
        figura faça
          selecionar próximo segmento;
```

```

repetir
    comparar segmentos selecionados com figuras padrão
        que possam ser formadas por algum subconjunto
        dos segmentos;
    até que encontre figura ou acabe subconjuntos;
    se encontrou figura
        então
            colocar padrão reconhecido na seqüência de
                figuras já reconhecidas;
        fim se;
    fim enquanto;
    se existirem segmentos selecionados
        então
            considerar estes segmentos como conectores;
        fim se;
    fim se;
fim enquanto;

```

## 2.5- RECONHECIMENTO DOS PADROES COMPOSTOS:

Após a etapa de procura pelos padrões simples tiver acabado, temos uma seqüência de padrões e conectores descrevendo a figura de entrada. Esta seqüência será utilizada para a busca dos padrões compostos que porventura existirem na figura.

A procura pelos padrões simples foi dirigida pela seqüência dos segmentos de entrada, ou seja, a partir de um novo segmento selecionado,

procurávamos na biblioteca por algum padrão que pudesse ser formado pelos segmentos já selecionados. A procura pelos padrões compostos será dirigida pela biblioteca de padrões. A partir de um padrão composto existente na biblioteca, testamos se existe possibilidade desse padrão existir na seqüência de padrões e conectores montada na etapa anterior.

Durante esta busca, percorremos a biblioteca de padrões até que seja encontrado um padrão composto. A partir deste padrão, selecionamos todos os padrões da seqüência de padrões já reconhecidos que possam compor o padrão composto selecionado. Verificamos a seguir se existe(m) alguma(s) combinação(ões) desses padrões que satisfaça(m) todas as propriedades do padrão composto. Caso exista(m) alguma desta(s) combinação(ões), os padrões e conectores correspondentes serão substituídos pelo padrão composto selecionado. Se não existir uma combinação adequada, assumiremos que o padrão composto selecionado não existe na figura de entrada. A seguir, selecionamos o próximo padrão composto na biblioteca e o processo acima é repetido até toda que a biblioteca seja averiguada.

Se no processo acima algum padrão composto foi encontrado, percorreremos mais uma vez toda a biblioteca, procurando por padrões compostos, conforme descrito acima. Esta busca deve se repetir pois podem existir padrões compostos formados por outros padrões compostos. Toda vez que encontramos um padrão composto, devemos verificar se ele pode ser um componente de um outro padrão e para tanto, devemos repetir a busca na biblioteca, utilizando agora a seqüência de padrões que foi modificada na etapa anterior. Este processo deve se repetir até que, ao percorrer toda a biblioteca, nenhum novo padrão composto seja encontrado na seqüência de padrões. Isto garante que não existem mais padrões compostos na figura de

entrada que estejam cadastrados na biblioteca de padrões, caso a segmentação tenha sido feita corretamente, pois a sequência de padrões não foi alterada nesta última passagem. Logo, se nenhum padrão foi encontrado, certamente também não será caso percorrêssemos novamente a sequência, uma vez que esta é a mesma da passagem anterior.

O fato de não encontrar um novo padrão composto, não garante que tal padrão não exista. Pode acontecer que existam padrões compostos que não foram completamente definidos na biblioteca de padrões, ou seja, nem todas as possibilidades de decomposição desses padrões foram descritas. Caso isto aconteça, será conveniente a alteração da biblioteca, incluindo esta nova possibilidade de formação deste padrão composto.

O processo de reconhecimento de figuras compostas pode ser esquematizado conforme o algoritmo mostrado a seguir.

repita

para cada figura composta cadastrada na biblioteca faça

selecionar todas subfiguras já reconhecidas que possam formar o  
padrão composto;

se existir algum subconjunto das subfiguras selecionadas que  
satisfaça as propriedades da figura composta

então

substituir estas subfiguras pelo padrão composto;

fim se;

fim para;

até que nenhuma figura composta seja reconhecida;

## 2.6- DETERMINAÇÃO DAS INTERCONEXÕES:

Uma vez reconhecidos todos os padrões que compõe a figura de entrada, resta descobrir como eles estão conectados. A biblioteca de padrões possui todas as informações sobre os pontos de entrada e saída que podem existir em cada padrão. Qualquer ligação entre os padrões deve ser realizada por intermédio desses pontos. Utilizaremos a sequência dos padrões e conectores montada na etapa anterior para descobrir o modo no qual as subfiguras estão conectadas. Como duas subfiguras só podem estar ligadas por meio de um conector, procuramos pelos conectores na sequência dos padrões reconhecidos. Para cada conector devemos ter ao menos uma subfigura ligada a ele. Através das posições das subfiguras e dos conectores desenhados, podemos descobrir quem está conectado com quem.

Após determinar todas as conexões, fazemos uso dos pontos de E/S definidos para cada padrão da biblioteca e determinamos quais são os pontos utilizados por um determinado conector para ligar duas subfiguras quaisquer. Desta forma, saberemos que um ponto de saída de um determinado padrão está conectado a um determinado ponto de entrada de outro padrão.

Durante a etapa de identificação dos pontos de E/S, podemos fazer um teste de validação parcial do reconhecimento de toda a figura. Caso for encontrado dois pontos de entrada (ou saída) conectados entre si, ou então algum conector cujo ponto de ligação não foi identificado, temos uma indicação que algum erro ocorreu na fase de reconhecimento. Como este erro é localizado, saberemos exatamente quais padrões foram reconhecidos erroneamente e de alguma maneira, pode-se tentar recuperar deste erro. Como

será visto na próxima seção, as regras de conexão (ou formação) da figura também podem indicar se alguma subfigura foi reconhecida indevidamente. O método utilizado para recuperar-se do erro pode ser aplicado em ambos os casos.

## 2.7- USO DAS REGRAS DE CONEXOES:

Geralmente, figuras formadas por padrões possuem alguma semântica embutida, de tal forma que a figura não pode ser desenhada ao acaso, ou seja, os padrões não podem ser conectados aleatoriamente. Existem certas regras (regras de conexão ou formação) que indicam quais ligações são válidas na figura, determinando de quantas ligações pode participar um determinado padrão, com quem ele pode estar conectado e de que forma. As regras de formação podem ser registradas no dicionário de padrões como regras "se-então".

Para os fluxogramas, por exemplo, segundo [MUR86], existem 52 regras de formação divididas em quatro grupos:

1º grupo: os pontos finais de um mesmo conector não podem estar em um mesmo padrão;

2º grupo: o número mínimo de conectores ligados a determinados padrões não pode exceder um certo valor;

3º grupo: o número máximo de conectores ligados a determinados padrões não pode exceder um certo valor;

4º grupo: os tamanhos de determinados padrões devem ser maiores ou menores que o tamanho médio dos demais.

Para os circuitos digitais, segundo o mesmo artigo anteriormente citado, existem sete regras de formação, como por exemplo, se o número de conectores ligados a uma porta NOT maior que 2, então ERRO. O número e tipo de regras de conexão varia de acordo com o tipo de figura a ser reconhecida.

Assumindo que a figura de entrada é bem formada, ou seja, todas as suas subfiguras padrões componentes estão definidas na biblioteca e elas não violam nenhuma regra de formação, se durante a verificação das regras for encontrado alguma violação, o erro é devido à má segmentação da figura do desenho. Ao ser detectado algum erro, existem duas possibilidades: ou abortamos o processo de reconhecimento ou tentamos corrigir o erro. Abortar o processo é uma tarefa trivial e não merece maiores considerações. Corrigir o erro pode ser muito trabalhoso, embora não seja necessariamente complicado.

Uma vez que o erro é localizado, sabemos exatamente qual (ou quais) subfigura foi responsável pela violação à alguma regra de formação, podemos identificar qual regra foi violada. A partir dessas subfiguras, podemos descobrir quais os segmentos responsáveis por suas formações, já que a seqüência de segmentos de entrada não foi destruída durante a construção da seqüência de padrões reconhecidos. Portanto, a partir desses segmentos, tenta-se encontrar um outro padrão que possa ser formado por eles, procurando-se por um padrão cujo desvio existente entre ele e a subfigura desenhada seja o menor possível. A partir daí, substitui-se o padrão responsável pelo erro pelo novo padrão reconhecido. Este processo se repete até que não exista nenhuma violação das regras de conexão, ou então,

até que não exista mais nenhuma figura possível de ser substituída (neste caso, houve um erro na formação do desenho de entrada).

Para evitar que o processo acima se repita indefinidamente, pode-se determinar um desvio máximo entre o desenhado e o padrão de tal forma que, se o desvio ultrapassar este limiar, o processo é abortado. É necessário ter um mecanismo de parada, pois certamente algum dia alguém submeterá uma figura não muito bem formada para o reconhecimento. Para substituir um erro, teremos de substituir um ou mais segmentos que foi reconhecido erroneamente. Para economizar trabalho, pode-se durante a fase de segmentação aproximar todos os segmentos por um segmento de reta ou um arco de circunferência e considerar que o segmento desenhado é aquele cujo desvio for menor. A outra aproximação feita no segmento fica como uma segunda opção de reconhecimento e será utilizada nos casos de erro de segmentação. Ao encontrar um erro, substitui-se um segmento componente do padrão que causou o erro de tal forma que a soma dos desvios do padrão seja mínima, e então, tenta-se encontrar um outro padrão que possa ser formado pelo novo conjunto de segmentos. Este é um processo iterativo que pode parar normalmente quando o desvio ultrapassar o limiar estabelecido ou quando esgotar o número de combinações possíveis das substituições dos segmentos componentes.

## 2.8- RECONHECIMENTO DE CARACTERES:

Embora o método descrito neste capítulo não tenha sido proposto com a finalidade de reconhecer caracteres, acreditamos que ele seja aplicável nestes casos. Para tanto, os caracteres possuirão uma biblioteca,

assim como todos os tipos de figuras. Aliás, o tratamento que deve ser dado aos caracteres é o mesmo que é dado para qualquer outro tipo de figura, sendo que neste caso, os padrões são as letras e números. Para que os caracteres sejam reconhecidos junto com figuras, devemos ter de indicar explicitamente se estamos desenhando uma figura ou um caracter, pois eles serão armazenados em seqüências distintas. Assim como na medida em que vamos desenhando uma figura, os segmentos componentes vão compondo uma seqüência de entrada, os segmentos que compõe o caracter vão formando uma seqüência. Um dos motivos destas seqüências serem distintas é para evitar que se tente reconhecer uma figura como um caracter e vice-versa, diminuindo assim o tempo de reconhecimento. Um outro motivo é que pode-se fazer algumas otimizações dependendo do tipo de figura desenhada e estas otimizações podem não ser aplicáveis para o reconhecimento de caracteres.

Dependendo da conveniência, pode-se limitar o conjunto de caracteres a ser reconhecido. A diferença que pode existir entre reconhecer figuras e caracteres é que as propriedades dos segmentos componentes sejam diferentes; entretanto, acreditamos que o método não necessite sofrer nenhuma alteração. Tanto os caracteres quanto as subfiguras estão sujeitos a distorções, variações no estilo e ruídos, onde a tolerância a estas distorções dependerá da implementação. Convém lembrar que o método descrito não pode adivinhar a intenção do usuário. Portanto, as figuras e caracteres reconhecíveis devem ser desenhadas com cuidado, dentro do limite do bom senso.

## PARTE II

IMPLEMENTACAO DO RECONHECEDOR DE FIGURAS

ESBOCADAS A MAO

A partir de agora daremos atenção à implementação do método descrito no capítulo anterior. O reconhecedor implementado é composto basicamente de três módulos: entrada, reconhecimento e validação.

Durante a etapa de entrada é feita a segmentação do desenho. O esboço de entrada é decomposto em seus componentes mais elementares (segmentos) que serão utilizados para o reconhecimento de todo o desenho. A etapa de reconhecimento consiste essencialmente na comparação das subfiguras componentes do desenho de entrada com os padrões definidos em uma biblioteca (biblioteca de padrões). Todos os padrões que poderão compor os desenhos devem estar obrigatoriamente definidos nesta biblioteca. A partir das subfiguras já reconhecidas, determina-se como elas estão conectadas entre si. Por subfiguras reconhecidas entende-se as subfiguras componentes do desenho de entrada que já foram associadas a algum padrão definido na biblioteca de padrões. Após a determinação das interconexões entre as subfiguras, é feita uma validação do resultado da etapa de reconhecimento, onde determinaremos se existe alguma conexão inválida entre as subfiguras, a partir de regras de conexão (ou formação) definidas pelo usuário. Um diagrama de blocos mostrando os módulos do reconhecedor pode ser vista na figura 3.1.

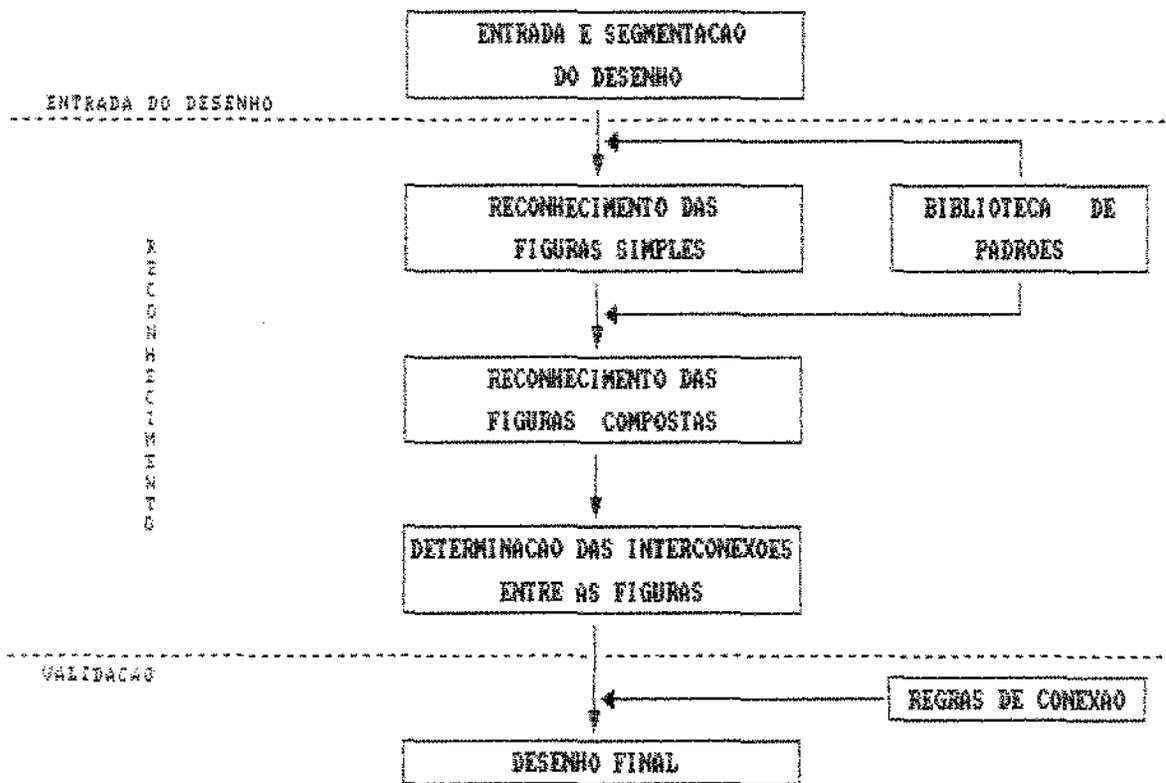


FIGURA 3.1

## CAPITULO 03

### BIBLIOTECA DE PADROES

#### 3.1- DESCRICAO:

A biblioteca de padrões é um arquivo que contém as definições de todos os padrões que poderão ser utilizados para compor o desenho de entrada. Como utilizaremos apenas o conteúdo desta biblioteca para reconhecer o desenho de entrada, ela deve armazenar todos os dados relevantes para o reconhecimento, conforme descrito no capítulo anterior.

Como foi visto, o método utilizado para o reconhecimento baseia-se no formato dos segmentos componentes das subfiguras e em suas posições, pois segundo [ROS73], temos que toda figura é completamente determinada pela especificação de suas bordas como um conjunto de curvas dirigidas; desta forma ela pode ser descrita pelas equações destas curvas. Uma vez que o formato dos padrões serão decompostos em segmentos de retas e arcos de circunferência, não necessitaremos de muitos dados para identificar univocamente cada segmento componente. Para identificar um segmento de reta, basta especificarmos as coordenadas dos seus pontos extremos. No caso de um arco de circunferência, temos ainda de especificar as coordenadas do seu centro e o sentido (horário ou antihorário) no qual devemos percorrer o arco a partir de um extremo para atingirmos o outro. Para que se possa identificar corretamente um padrão, não basta apenas descrever seus segmentos componentes; devemos ainda conhecer a ordem na qual devemos conectá-los, de tal forma a reproduzir este padrão. Como na biblioteca de padrões cada subfigura está definida apenas pela descrição

dos seus segmentos componentes, uma forma de determinarmos a ordem na qual eles devem ser conectados, é definindo cada segmento componente na ordem correta no qual eles deveriam aparecer, caso percorrêssemos a subfigura padrão em um determinado sentido. Portanto, a partir dos dados contidos na biblioteca devemos ser capazes de identificar com exatidão cada segmento que compõe o padrão e, partindo de um segmento qualquer e seguindo a ordem na qual estes segmentos foram especificados, identificaremos de forma única o padrão definido.

Existirá uma biblioteca para cada tipo de aplicação. Assim, se desejarmos reconhecer novas figuras formadas por novos padrões, necessitamos somente construir uma nova biblioteca contendo a definição desses padrões ou acrescentar estas definições à uma biblioteca já existente, não sendo necessário alterar a estrutura do reconhecedor. Isto o torna bastante flexível.

Para cada padrão que possa compor a figura de entrada, a biblioteca conterá os seguintes dados:

1. nome do padrão;
2. indicador se o padrão é simples ou composto;
3. número de componentes que formam o padrão;
4. número de pontos de entrada e saída utilizado para conexões;
5. descrição de cada componente que forma o padrão;
6. propriedades destas componentes;
7. descrição do pontos de entrada e saída ( pinos de E/S );
8. dados para impressão do padrão;

Como foi visto no capítulo anterior, os dados contidos na

biblioteca são suficientes para descreverem univocamente os padrões que poderão compor a figura de entrada.

A busca por um padrão simples durante o reconhecimento será baseada nos segmentos componentes da subfigura a ser reconhecida. Para cada padrão simples definido na biblioteca, temos o número de segmentos utilizados para compor este padrão e a sua descrição. Os segmentos componentes de um determinado padrão estão definidos de tal forma que cada segmento está conectado ao próximo, ou seja, a extremidade final de um segmento coincide com a extremidade inicial do próximo segmento componente do padrão definido na biblioteca. Para cada segmento, temos os seguintes dados definindo-os:

1) tipo do segmento: segmento de reta ou arco de circunferência.

2) direção mínima e máxima que o segmento pode ter. A direção de um segmento foi definida como sendo o ângulo formado com a horizontal pelo segmento de reta que une a extremidade inicial do segmento com a extremidade final, onde este ângulo é medido no sentido anti-horário. Portanto, para que um segmento seja considerado como um componente de um padrão, sua direção deve estar entre as direções limites, dentre outras condições.

3) se o segmento for um arco de circunferência, temos também a abertura máxima que o arco pode ter e o sentido no qual ele deve ser desenhado. A abertura de um arco foi definida como sendo o ângulo formado entre os segmentos de reta que unem o centro aos extremos inicial e final do arco. Portanto, para que um arco seja considerado como componente do padrão, sua abertura deve ser menor ou igual à abertura máxima permitida.

Uma vez que a direção e a abertura do segmento estão dentro de certos valores, para que um segmento componente de uma subfigura desenhada seja considerado como componente de um padrão, estes dados devem estar dentro dos limites impostos, aumentando assim a tolerância à distorções, pois não é necessário um ajuste perfeito entre o segmento desenhado e o segmento do padrão.

Um outro detalhe existente na descrição dos segmentos componentes dos padrões é a ausência de informações referentes ao tamanho e localização dos segmentos. Desta forma, qualquer segmento que satisfaça as características descritas na biblioteca, não importando o seu tamanho ou localização, pode ser um componente de um padrão. A ausência dessas informações simplesmente reflete o método utilizado para o reconhecimento, já que este foi proposto de forma a ser independente do tamanho e posição das subfiguras componentes do desenho de entrada.

Embora não existam informações relacionadas diretamente com o tamanho dos componentes dos padrões, podem existir informações relacionando o tamanho desses componentes, indicando que um deve ser maior, menor ou igual a outro, não interessando o tamanho real deles.

No caso da busca por padrões compostos, as componentes serão outros padrões simples ou compostos, ao invés de segmentos. Para cada padrão composto definido na biblioteca, temos o número de componentes que forma o padrão e o nome dos padrões componentes. Não temos na biblioteca qualquer descrição mais detalhada desses componentes. Para saber se os padrões que podem compor algum padrão composto realmente definem este

padrão, é feita uma análise do posicionamento relativo entre eles e de algumas propriedades que estes componentes devem satisfazer.

Para que um conjunto de componentes seja considerado formadores de um determinado padrão (simplex ou composto), além de satisfazer a descrição dos componentes, este conjunto deve satisfazer as propriedades que porventura existam entre as componentes formadoras do padrão. Estas propriedades, que estão cadastradas na biblioteca, para cada padrão, se referem a algumas características geométricas e topológicas que se julgarem relevantes para a discriminação entre padrões cujas descrições das componentes não são suficientes para garantir o seu reconhecimento correto. A princípio, o usuário pode definir qualquer tipo de propriedade, entretanto o protótipo implementado consegue distinguir somente as seguintes propriedades, relacionadas ao tamanho ou posicionamento dos componentes:

1) Propriedades dos Segmentos: dados dois segmentos  $seg1$  e  $seg2$ , temos:

- a) tamanho:
  - $seg1$  igual a  $seg2$
  - $seg1$  menor que  $seg2$
  - $seg1$  maior que  $seg2$
  - $seg1$  diferente de  $seg2$
  
- b) posicionamento:
  - $seg1$  paralelo a  $seg2$
  - $seg1$  perpendicular a  $seg2$
  - $seg1$  é horizontal
  - $seg1$  é vertical

2) Propriedades das Subfiguras: dados duas figuras fig1 e fig2, temos:

- a) tamanho:
- fig1 igual a fig2
  - fig1 menor que fig2
  - fig1 maior que fig2
  - fig1 diferente de fig2
- b) posicionamento:
- fig1 acima de fig2
  - fig1 abaixo de fig2
  - fig1 à esquerda de fig2
  - fig1 à direita de fig2
  - fig1 dentro de fig2
  - fig1 parcialmente dentro de fig2

Se as componentes analisadas satisfazem a descrição do padrão e as propriedades de suas componentes, então este conjunto será considerado como formador do padrão e será representado pelo padrão em questão.

Após o reconhecimento das subfiguras componentes do desenho de entrada, devemos descobrir o modo na qual elas estão interligadas entre si. Como em uma figura formada por padrões, estes padrões não podem estar conectados de qualquer forma, ou seja, existem pontos onde estes padrões podem ser conectados, sendo que a biblioteca deve conter uma descrição destes pontos que podem ser utilizados para conexão (pontos de E/S). Para cada padrão, temos cadastrado na biblioteca o número de pontos de E/S que podem existir e uma descrição destes pontos. Esta descrição consiste no tipo do ponto de conexão (entrada ou saída), local (esquerda, direita, parte superior ou inferior do padrão), nome (rótulo) do ponto e posição do ponto de conexão em relação ao local onde ele se encontra, isto é, se ele é

o primeiro, segundo, ..., n-ésimo ponto do lado esquerdo, direito, etc.

Assim que os pontos de conexão forem localizados nas subfiguras componentes do desenho de entrada, a partir do nome da subfigura e da posição destes pontos, conseguimos discriminar através de quais pontos dois padrões estão conectados.

A biblioteca também possui dados que permitem imprimir um desenho do padrão (dados de impressão). A partir do ponto inicial de um segmento, temos cadastrado na biblioteca o deslocamento que deve ser realizado nas duas coordenadas cartesianas para atingirmos a outra extremidade do segmento. Se este segmento for um arco de circunferência, temos também o posicionamento do centro deste arco. Os deslocamentos são dados como se o retângulo envolvente do padrão possuísse tamanho unitário, e o ponto inicial de referência é o canto inferior esquerdo deste retângulo. Após o reconhecimento do desenho de entrada, sabemos onde estão localizados todas as suas subfiguras componentes e o tamanho do retângulo envolvente de cada uma. Para uma subfigura dada, tomando como referência o canto inferior esquerdo do seu retângulo envolvente, podemos encontrar a localização dos extremos de cada segmento componente desta subfigura utilizando os dados de impressão do padrão e o tamanho do retângulo envolvente da subfigura. Com isto, podemos fazer um desenho melhorado do padrão, na escala na qual foi desenhada a subfigura. Assim, podemos obter um desenho melhorado de toda figura de entrada como um resultado visual do reconhecimento. Obviamente, temos de fazer alguns ajustes no posicionamento e tamanho dos padrões componentes da figura de entrada, de modo que eles fiquem alinhados e com tamanhos proporcionais em relação aos demais.

O conteúdo da biblioteca de padrões será carregado para a memória antes da entrada do desenho, durante a etapa de configuração do sistema. Nesta etapa o usuário escolhe o tipo de figura que ele irá desenhar e o dispositivo de entrada utilizado para desenhá-lo. Estes dados permanecerão na memória até que se escolha outro tipo de figuras ou então que se encerre a execução do programa, mas em um determinado instante, só existirá uma biblioteca na memória para o tipo específico do desenho escolhido.

### 3.2- CONSTRUCAO DAS BIBLIOTECAS:

As bibliotecas são construídas separadamente através da entrada direta dos dados pelo usuário, por um programa específico chamado de *construtor de bibliotecas*. Este programa montará todos os arquivos que formam a biblioteca de padrões. Para cada tipo de figura existirão 6 arquivos cujos conteúdos são:

1. definição dos padrões;
2. segmentos componentes dos padrões simples;
3. figuras componentes dos padrões compostos;
4. propriedades das componentes dos padrões;
5. pontos de E/S do padrão;
6. impressão.

Estes arquivos contém os dados a serem utilizados para na descrição dos padrões durante a etapa de reconhecimento, como descrito na seção anterior.

Não existe nenhuma restrição na entrada dos padrões. Eles podem ser definidos em qualquer ordem. Entretanto, durante a entrada dos segmentos que compõe um padrão simples, deve ser obedecido o seguinte critério: pode-se começar a definição por qualquer segmento, mas a partir daí, deve-se definir os outros segmentos percorrendo o padrão no sentido anti-horário ou no sentido horário de forma que dois segmentos definidos consecutivamente estejam conectados entre si, ou seja, eles têm uma extremidade em comum. Uma vez começado a definição, este sentido não poderá ser alterado. O motivo desta restrição baseia-se em [ROS79] onde, como já foi dito anteriormente, toda figura é completamente determinada pela especificação de suas bordas como um conjunto de curvas direcionadas. Logo, o padrão poderá ser reconstruído se percorrermos a definição de suas componentes em uma seqüência específica. Esta é uma premissa básica adotada pelo reconhecedor. Portanto, se ela não for seguida, o processo de reconhecimento não funcionará.

Se considerarmos os extremos dos segmentos que definem o padrão como nós de um grafo e os segmentos componentes como arcos deste grafo, ao definir este padrão, devemos descrever um percurso que passa por todos os arcos do grafo, uma única vez. Portanto, a tarefa de definição do padrão pode ser reduzida a um problema de teoria dos grafos de encontrar um "percurso de Euler" no grafo representado pelo padrão. Desta forma, um padrão qualquer pode ter apenas dois nós de grau ímpar, a saber, o primeiro nó e o último nó do percurso de definição do padrão. Caso não seja possível encontrar este percurso, o padrão deve obrigatoriamente ser considerado composto, onde seus componentes devem satisfazer a descrição acima, ou seja, deve existir um percurso que passa por todos os seus segmentos componentes, uma única vez (sem repetição).

Esta é uma boa técnica que pode ser usada para decidir se um determinado padrão é composto ou simples. Um outro fato que podemos analisar para tomar esta decisão, ocorre quando todos os padrões que podem ser utilizados para formar a figura de entrada são fechados, isto é, possuem uma área bem definida. Neste caso, sempre que tivermos um subconjunto dos segmentos que compõem um determinado padrão definindo uma área fechada, o padrão deverá ser considerado composto, onde este subconjunto de segmentos será um subpadrão componente.

Um cuidado que devemos ter ao definir figuras compostas, é evitar uma definição recursiva de um padrão, ou seja, que um subpadrão de um padrão composto tenha como componente o próprio padrão composto. Fazendo uma analogia com linguagens de programação, considerando os subpadrões como procedimentos/funções desta linguagem e a descrição desses padrões como suas chamadas de ativação, devemos evitar que haja recursividade nos procedimentos, quer seja ela direta ou indireta.

Uma vez que o sistema não averigua tais possibilidades, fica por conta e risco do usuário o cumprimento das restrições expostas acima, sob pena de não funcionamento adequado do reconhecedor.

A entrada dos dados para a construção da biblioteca será feita diretamente pelo usuário, onde ele fornecerá o nome do padrão, número de componentes e a descrição detalhada de todas as componentes, das propriedades entre elas, dos pontos de E/S utilizados para conexão e dos dados utilizados para a impressão. Pela maneira na qual a biblioteca deve ser construída, os segmentos componentes de cada padrão estão definidos em

uma seqüência pré-definida de tal forma que dois segmentos consecutivos quaisquer do padrão estejam ligados um ao outro na composição do padrão.

### 3.3- LEITURA DAS BIBLIOTECAS:

Antes de se realizar o reconhecimento das figuras, as definições dos padrões serão carregadas para a memória em forma de uma lista duplamente ligada, ordenada pelo número de componentes do padrão. Existirá uma lista índice que apontará para o primeiro padrão de N componentes. Uma vez que o reconhecimento será baseado no número de componentes da subfigura, através desta lista índice poderemos ter acesso direto aos padrões com o mesmo número de componentes da subfigura, sem a necessidade de percorrer toda a lista de definição dos padrões.

Não será necessário ordenar a lista de padrões durante a leitura do arquivo de definição, pois o construtor das bibliotecas já montará este arquivo ordenado, independentemente da seqüência na qual os padrões foram definidos durante sua montagem. A montagem da lista dos padrões e da lista índice será feita simultaneamente com a leitura dos arquivos de definições. Ao final da leitura, teremos montado a estrutura mostrada na figura 3.2, onde o conteúdo das listas estão mostrados na figura 3.3.

O conteúdo dos arquivos montados pelo construtor de bibliotecas contém exatamente os mesmos campos das listas correspondentes, exceto o arquivo dos padrões, que contém os dados mostrados na figura 3.4.

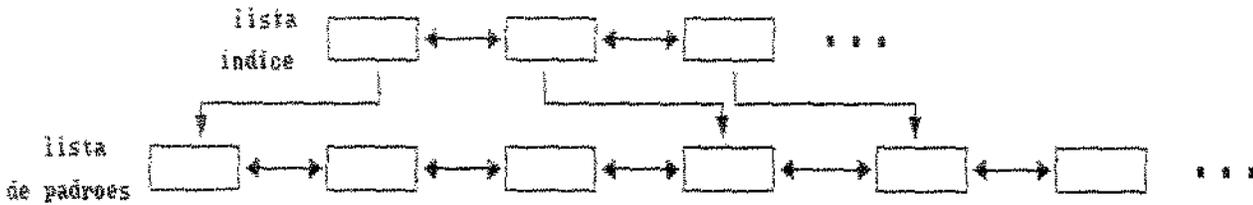
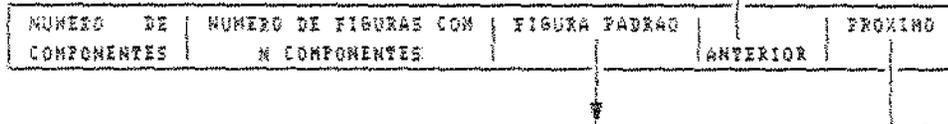
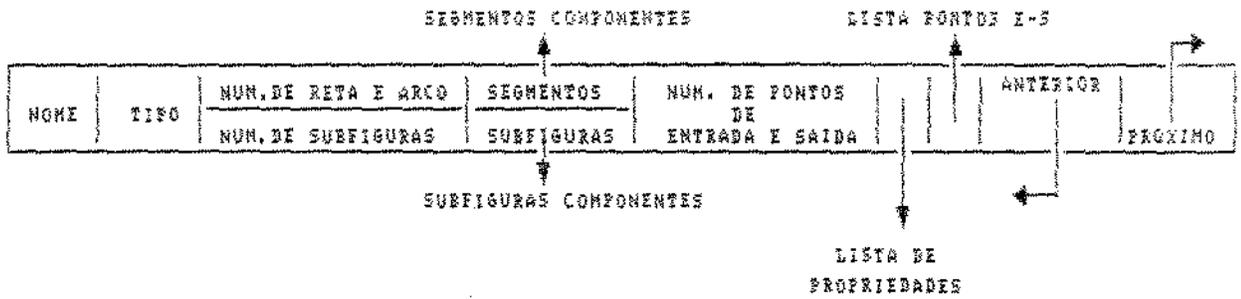


FIGURA 3.2

lista indice:



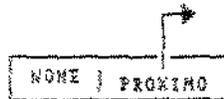
lista de padroes:



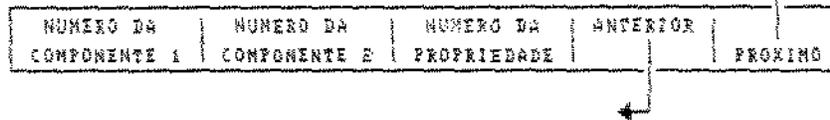
lista dos segmentos :  
componentes



lista das subfiguras :  
componentes



lista de propriedades:



lista dos pontos :  
de E/S

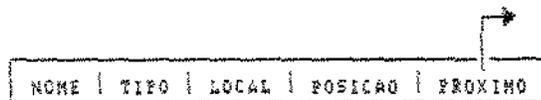


FIGURA 3.3

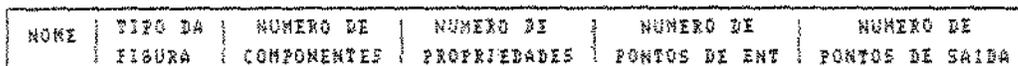


FIGURA 3.4

Este arquivo guiará a construção da lista de padrões e demais listas. Após ler um registro deste arquivo, os campos correspondentes da lista padrão que estão presentes também no registro são preenchidos. A partir dos valores dos números de componentes (NC) e tipo da figura, é lido os próximos NC registros do arquivo dos segmentos ou figuras componentes do padrão, dependendo se a figura é simples ou composta. Durante a leitura dos NC registros, as listas correspondentes são montadas e um apontador para o primeiro elemento destas listas é colocado no campo correspondente da lista de padrão. O mesmo processo se aplica para a montagem da lista de propriedades e pontos de E/S. Após o elemento da lista padrão estar completamente montado, verifica-se se o elemento anterior desta lista possui NC componentes. Caso o número de componentes do elemento anterior da lista seja diferente de NC, um novo elemento da lista índice é criado, apontando para o elemento padrão que foi montado.

### PROTOTIPO IMPLEMENTADO

#### 4.1- ENTRADA E SEGMENTACAO DO DESENHO:

Os segmentos que compõem o desenho de entrada poderão ser desenhados em qualquer ordem, sendo que a única restrição existente é que uma vez começado a entrada de algum segmento de um determinado padrão componente do desenho, este padrão deverá ser todo desenhado antes que se inicie a entrada de outro. Esta restrição foi colocada de modo a facilitar o reconhecimento e, principalmente, torná-lo mais rápido, como será visto posteriormente. O desenho poderá ser desenhado através de uma mesa digitalizadora ou via teclado. Como existe a restrição de que os segmentos que compõe uma subfigura padrão devem ser consecutivos, não poderemos utilizar um esquadrinhador óptico (*scanner*) na entrada, uma vez que esta restrição poderia ser violada devido ao método de varredura deste instrumento.

Durante a entrada, o desenho será representado por uma seqüência de pontos. Esta seqüência é construída à medida que o desenho está sendo feito, pois isto preserva a ordem na qual os segmentos que compõem cada subfigura padrão foi desenhado. A preservação desta ordem é importante devido à restrição imposta durante a entrada, que é fundamental no processo de reconhecimento. Se esta ordem fosse alterada, esta restrição não teria sentido e não poderíamos retirar nenhum proveito deste fato. Após a entrada de todo o desenho utilizaremos esta seqüência de pontos para determinar os

segmentos componentes das subfiguras (segmentação). Em princípio, a fase de segmentação poderia ser realizada simultaneamente com a entrada dos pontos, entretanto decidimos não proceder desta forma, pois isto dificultaria alterações do desenho que porventura vierem a ser realizadas durante a entrada.

A etapa de segmentação é de fundamental importância para o reconhecimento do desenho de entrada, pois, caso ele seja segmentado de forma errada, certamente o resultado do reconhecimento será falho. Acreditamos que, caso o desenho seja bem formado, ou seja, todos os padrões que o compõe estão cadastrados na biblioteca de padrões, e o processo de extração dos segmentos componentes seja preciso, o reconhecedor será capaz de reconhecer corretamente quase a totalidade das figuras de entrada, uma vez que o reconhecimento está baseado unicamente na posição e no formato dos segmentos que compõem os padrões.

## **4.2- MÉTODO DE SEGMENTAÇÃO UTILIZADO:**

Durante o desenvolvimento do reconhecedor foram implementados dois métodos para segmentação do desenho.

### **4.2.1- MÉTODO 01:**

Neste método, a segmentação foi guiada pela busca dos pontos que indicariam o início e final de um determinado segmento (pontos característicos ou pontos críticos). O processo de segmentação consiste em determinar se o próximo ponto da seqüência de pontos de entrada pertence ao mesmo segmento que os pontos anteriores. Um ponto característico indica que

os próximos pontos pertencem a um novo segmento.

Um ponto característico pode ser determinado por:

1. mudança acentuada na direção em relação aos pontos anteriores;
2. mudança na concavidade (ponto de inflexão);
3. distância razoável em relação ao último ponto (novo segmento). Isto indica uma movimentação do cursor sem que tenha entrado nenhum ponto. Aliás, sempre que isto ocorrer, teremos um ponto crítico, pois houve um posicionamento do cursor para o início de um novo segmento.

Assim que um ponto característico é encontrado, todos os pontos existentes entre este ponto e o último ponto característico encontrado, serão substituídos por um segmento de reta ou arco de circunferência, dependendo da disposição espacial dos pontos. Após o segmento ser identificado, os pontos utilizados serão descartados e este novo segmento será colocado em uma lista de segmentos.

Dado dois pontos característicos, que serão os pontos extremos do segmento, e a sequência de pontos entre eles, o segmento formado por estes pontos será aproximado, ou por um segmento de reta que une os pontos característicos, ou por um arco de circunferência que passa pelos pontos característicos e pelo ponto médio entre eles. Esta aproximação é feita calculando-se a equação analítica das duas curvas. A partir destas equações, calcula-se o desvio existente entre o segmento desenhado e as curvas aproximadas. Este desvio é feito ponto a ponto e é dado pela distância euclidiana entre o ponto desenhado e os respectivos pontos situados na curva. O desvio total será a soma destas distâncias e o

segmento desenhado será aproximado pela curva cujo desvio for menor. Caso os desvios forem iguais, o segmento é tido como sendo uma reta, pois é muito mais provável que uma reta mal desenhada se pareça com uma curva, do que uma curva mal desenhada se pareça com uma reta. Entretanto, isto só ocorrerá em casos onde o desenho foi muito mal desenhado. Embora o reconhecedor seja tolerante a distorções, espera-se que o usuário trabalhe dentro de um certo limite de tolerância.

Entretanto, este método não funcionou adequadamente, uma vez que o desenho de entrada é esboçado a mão, dificultando em muito a obtenção dos pontos característicos, pois, devido à distorção dos segmentos, um segmento de reta ou arco de circunferência desenhado eram segmentados indevidamente em vários outros.

Na tentativa de corrigir este problema, logo após a segmentação da figura e os ajustes realizados em seus formatos, realizou-se uma etapa de fusão dos segmentos que porventura estivessem quebrados. Desta forma, caso algum segmento componente de um padrão tenha sido segmentado em vários outros, ou então, durante a entrada, o segmento não foi desenhado como sendo um único segmento, esta etapa se encarregaria de fundir estes segmentos em um único componente. Após a execução desta etapa, teremos uma confiabilidade maior no resultado do processo de segmentação, mas, mesmo assim, poderão ainda continuar existindo segmentos quebrados na figura. Esta rotina de fusão dos segmentos também faz o ajuste nas quinas das subfiguras.

Para evitar um trabalho maior na etapa de fusão dos segmentos que porventura estejam quebrados foi feita uma modificação no método, de forma

a torná-lo menos sensível a distorções. A alteração realizada consistia em desprezar as pequenas variações na direção dos segmentos desenhados. Através desta modificação, segmentos de reta com pequenas distorções eram segmentados corretamente, enquanto que os arcos continuavam a ser segmentados em vários outros segmentos. Após outras modificações e algumas melhorias na etapa de fusão dos segmentos resultantes, tivemos de abandonar o método de segmentação inicialmente proposto, uma vez que ele estava distorcido em relação à concepção original, sensivelmente mais complicado, com resultados nada animadores, já que o objetivo inicial de construir um método simples e de fácil entendimento não foi alcançado. Dentre as possíveis soluções, decidimos implementar um novo método para realizar a segmentação.

#### 4.2.2- MÉTODO 02:

Na esperança de conseguir um método de segmentação eficiente e tolerante a distorções, decidimos implementar um método de segmentação já existente, com algumas alterações que o tornasse pouco sensível às distorções.

O método implementado foi o método de Hung & Kaswand [RUI89]. Este método "baseia-se na codificação das linha pelo código da cadeia obedecendo à vizinhança mostrada na figura 4.1, e estabelece três outras grandezas relativas a este código" (maiores detalhes sobre o método pode ser encontrado em [RUI89]). Através dessas três grandezas e pelo uso de certas regras definidas sobre elas, pode-se determinar os pontos característicos de uma linha.

Na versão implementada do método, temos 16 direções possíveis no código da cadeia, conforme mostrado na figura 4.2. As definições das grandezas e regras no método original de Huang não foram alteradas. Após a determinação dos pontos críticos, a sequência de pontos entre eles é analisada de modo a determinar se o segmento por eles definido pode ser aproximado por um segmento de reta ou arco de circunferência.

Assim como no método de segmentação descrito anteriormente, os segmentos encontrados são analisados por uma rotina de fusão, que unirá segmentos quebrados. Embora o resultado obtido utilizando este método tenha sido melhor que o resultado obtido pelo primeiro método de segmentação, constatamos que ocorreram os mesmos problemas, principalmente na segmentação de arcos, embora eles tenham ocorrido em menor quantidade. A melhoria em relação ao primeiro método de segmentação foi significativa, mas este método ainda necessita de melhorias, pois ele possui certa sensibilidade a distorções.

Como pode ser observado, o método de segmentação implementado está aquém do razoável. Poder-se-ia desenvolver um método mais eficaz e tolerante a distorções, que pudesse segmentar a figura de entrada com uma boa margem de segurança. Entretanto, como foi visto no capítulo 2, é muito difícil obter um método de segmentação eficaz e, por mais que nos esforçássemos em implementar algum, certamente ele não seria totalmente seguro. Como a eficiência do método de reconhecimento depende da segmentação, seria de grande importância o desenvolvimento de um processo de segmentação seguro. Entretanto, o objetivo deste trabalho não é o de produzir um algoritmo de segmentação altamente tolerante a distorções, e

sim, mostrar que o método de reconhecimento proposto é confiável e computacionalmente viável. Além do mais, o desenvolvimento de tal método de segmentação exigiria esforço e dedicação maior ao assunto, o que foge do propósito inicial do trabalho. Devido aos erros de segmentação produzidos pelos métodos implementados, a taxa de reconhecimento do protótipo implementado seria baixa, o que esconderia a viabilidade e confiabilidade do método de reconhecimento proposto.

Portanto, para efeito de testes do método de reconhecimento, a criação da lista dos segmentos componentes das subfiguras padrões formadoras do desenho de entrada foi realizada manualmente, onde a figura a ser reconhecida foi entrada já segmentada. O desenvolvimento de um método de segmentação tolerante a distorções do desenho fica como tema para pesquisas futuras.

Uma possível saída para o problema da segmentação poderia ser um melhoramento da etapa de fusão dos segmentos encontrados, etapa esta a ser realizada após a segmentação da figura. Caso esta etapa seja eficiente, os erros encontrados durante a segmentação poder ser corrigidos a tempo. Com isto, muito trabalho poderia ser poupado, pois o desenvolvimento de um algoritmo eficiente para esta etapa parece ser mais fácil que o desenvolvimento de um algoritmo eficiente para segmentação. Sendo assim, a figura de entrada poderia ser segmentada por um método com relativa tolerância a distorções e os erros de segmentação seriam corrigidos nesta etapa de fusão dos segmentos encontrados. O algoritmo de fusão poderá utilizar os formatos dos segmentos próximos como ajuda no processo; ajuda esta que não pode ser obtida através da seqüência dos pontos de entrada, uma vez que o posicionamento dos pontos próximos nesta seqüência não

fornecem muitas informações. Poderíamos também tentar corrigir os erros de segmentação através de um conhecimento prévio do tipo de padrões que poderão compor a figura de entrada. Os erros de segmentação podem ser corrigidos pela fusão de certos segmentos, uma vez que o problema encontrado nesta etapa não foi a falta de segmentos extraídos, mas sim, o excesso deles, ou seja, todos os segmentos componentes de uma determinada subfigura padrão são extraídos, entretanto, alguns deles eram segmentados em vários outros.

### 4.3- LISTA DOS SEGMENTOS:

Após a segmentação da figura, a seqüência de pontos de entrada terá sido substituída por uma lista de segmentos, na qual será baseada todo o processo de reconhecimento. Através da lista de segmentos temos acesso direto aos dados relevantes para o reconhecimento, uma vez que este se baseia no formato e na posição relativa entre os segmentos que compõem os padrões. Além disso, durante o processo de criação da lista de segmentos, as distorções advindas da entrada do desenho foram eliminadas. Portanto, esta lista nos fornece uma descrição sucinta de todo o desenho. As informações diretamente manipuláveis a partir da lista de segmentos são muito mais significativas que aquelas existentes na lista de pontos.

Esta lista foi implementada como uma lista duplamente ligada, onde cada elemento contém os dados mostrados na figura 4.3, onde:

- flag : indica se este segmento está conectado ao segmento posterior;
- tipo : indica se este é um segmento de reta ou um arco de circunf;

desvio : desvio total em relação ao segmento desenhado;

direção: direção do segmento de reta que une os pontos extremos;

pto.inicial: coordenadas do ponto inicial;

pto.final : coordenadas do ponto final;

se segmento for um arco de circunferência:

centro : coordenadas do centro do arco;

sentido: indica em qual sentido o arco foi desenhado;

outro: outra possibilidade de reconhecimento do segmento. Esta segunda possibilidade de reconhecimento de um determinado segmento poderá ser utilizado caso ocorra algum erro na etapa de reconhecimento, conforme descrito no capítulo anterior.

#### 4.4- QUANTIZACAO DA DIRECAO:

Além da aproximação no formato do segmento desenhado em relação aquele armazenado na lista de segmentos, existe uma outra aproximação que é realizada nos segmentos da lista: uma aproximação realizada na direção do segmento. Existirão apenas 16 direções possíveis para os segmentos, a saber: 0, 22, 45, 67, 90, 112, 135, 157, 180, 225, 247, 270, 292, 315, 337 e 360 graus. Quando um segmento é identificado, a sua direção é aproximada para uma das 16 possíveis. Através desta quantização na direção, o processo de reconhecimento dos padrões fica facilitado, pois segmentos com direções próximas serão considerados como possuindo a mesma direção.

#### 4.5- RECONHECIMENTO DAS FIGURAS SIMPLES:

O reconhecimento de figuras simples se processa através da busca na lista de segmentos por uma seqüência que satisfaça a definição de alguma figura padrão armazenada na biblioteca de padrões.

Antes da busca ser iniciada, os segmentos resultantes da etapa de segmentação passou por um processo de fusão, de tal forma que os segmentos que estavam quebrados foram unidos. Este processo também ajustou as quinas mal formadas. Existem três casos onde as quinas são ajustadas, conforme a figura 4.4.

O reconhecimento das figuras é baseado no número de segmentos componentes dessas figuras e no posicionamento entre eles. Logo, se a segmentação foi feita de modo errada, o reconhecimento falhará. Temos como premissa que a figura de entrada foi construída adequadamente, onde todas as subfiguras padrões utilizadas em sua composição estão definidas na biblioteca padrão correspondente. Logo, qualquer erro que ocorra durante a fase de reconhecimento é tido como um erro de segmentação. À medida que as subfiguras vão sendo reconhecidas, uma lista de figuras já reconhecidas é formada. Os dados contidos em cada elemento desta lista estão mostrados na figura 4.5, onde:

- 1) os dados referentes ao retângulo envolvente servem para indicar o tamanho e posicionamento da subfigura componente. Estes dados serão utilizados na determinação das interconexões entre as subfiguras;

- 2) o nome da subfigura indica a qual padrão da biblioteca ela foi associada;

- 3) o número da subfigura indica a ordem na qual elas foram descritas. Estes números serão mostrados no final do reconhecimento para indicar as

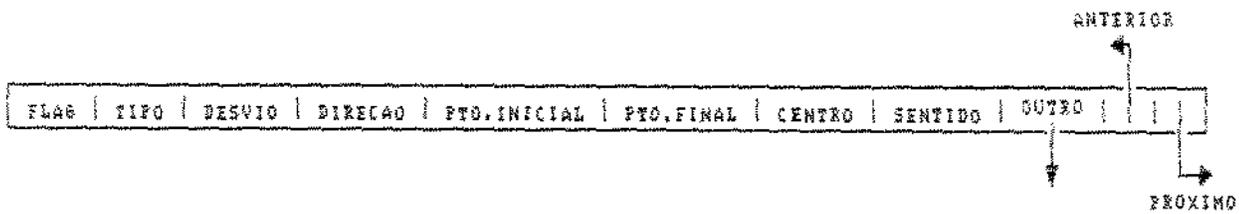


FIGURA 4.3

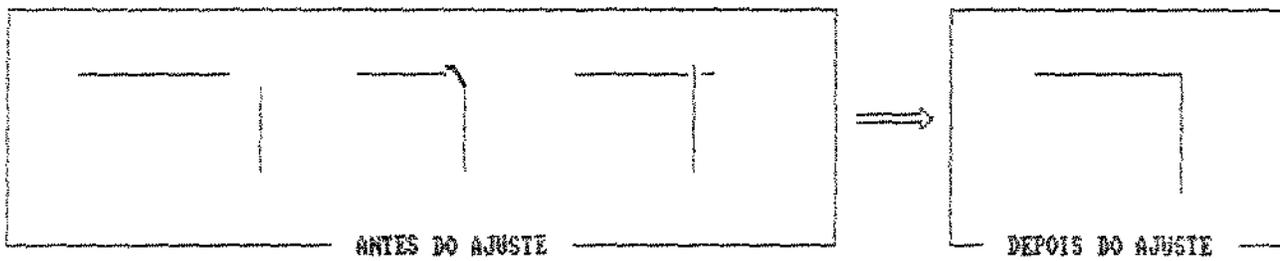


FIGURA 4.4

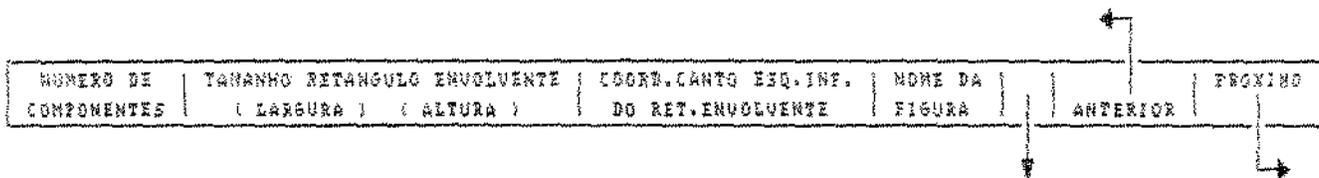


FIGURA 4.5

interconexões entre as subfiguras componentes do desenho de entrada;

4) o apontador para a lista de segmentos indica qual foi o primeiro segmento utilizado para compor o desenho. Os demais segmentos componentes da subfigura estão logo após este segmento. Através destes segmentos poderemos compará-los com os segmentos componentes dos padrões cadastrados na biblioteca de forma a reconhecer a subfigura.

O método utilizado na busca dos segmentos e na comparação com os padrões depende do tipo de padrões que estão sendo usados para compor o desenho de entrada. O desenho poderá ser formado por dois tipos de padrões: fechados e abertos. Para cada um desses tipos, o método de busca será diferente.

#### 4.5.1- Figuras Fechadas:

No capítulo anterior descrevemos um algoritmo para a busca das figuras componentes, onde, a cada novo segmento analisado da lista de segmentos, tínhamos de testar se este segmento, junto com os demais já selecionados, definiam um padrão. Quando todos os padrões que poderão formar a figura de entrada são fechados, ou seja, possuem uma área delimitada e bem definida, não necessitaremos testar se cada novo segmento lido na lista de segmentos define um padrão, junto com os demais, pois somente será necessário verificar se um determinado conjunto de segmentos forma um padrão, se estes segmentos definirem uma figura fechada.

Dado um segmento da lista de segmentos, verifica-se se ele define uma figura fechada. Se definir, vamos na lista de padrões e comparamos este

segmento com todos os padrões formados por apenas um segmento. Se a figura estiver bem formada e a segmentação foi feita corretamente, deve existir algum padrão cuja descrição seja compatível com o segmento dado. Então assume-se que o segmento define este padrão e um novo elemento na lista de figuras reconhecidas é inserido, correspondendo ao padrão encontrado. Se nenhum padrão definido na biblioteca que satisfaça a descrição do segmento selecionado foi encontrado, houve então um erro durante a segmentação da figura; o processamento é cancelado e uma mensagem indicando que houve um erro é mostrada. Caso o segmento dado não defina uma figura fechada, selecionamos o próximo segmento e testamos se este segmento e o anterior formam uma figura. Caso eles definam uma figura fechada, o padrão correspondente é inserido na lista de figuras reconhecidas. Caso contrário, seleciona-se o próximo segmento da lista e testa-se se os segmentos já selecionados definem uma figura fechada.

Este processo é repetido até que se encontre uma figura fechada, ou que se acabem os segmentos. Uma vez que existe a restrição na entrada da figura que os segmentos componentes de uma subfigura padrão devem estar juntos, ou seja, não existe nenhum segmento não pertencente a esta subfigura entre eles, a busca por uma figura fechada é feita a partir do último segmento selecionado. Ao encontrar uma figura, a ordem dos segmentos que definem esta figura é alterada de modo que dois segmentos, pertencentes à lista de segmentos, consecutivos que compõem o padrão estejam conectados.

Isto é necessário, pois os segmentos que compõem os padrões cadastrados na biblioteca são descritos na ordem em que eles apareceriam caso percorrêssemos o padrão em um determinado sentido, ou seja, dois segmentos componentes que foram descritos consecutivamente na definição de

um padrão estão conectados. Sendo assim, podemos comparar se o conjunto de segmentos seleccionados formam um determinado padrão obedecendo a ordem em que eles foram definidos na biblioteca. Os segmentos que formam a figura fechada são comparados com os segmentos do primeiro padrão definido na biblioteca que possui o mesmo número de segmentos.

A comparação dos segmentos formadores da subfigura fechada com os segmentos componentes de um determinado padrão é feita um a um, obedecendo a ordem na qual os primeiros estão posicionados na lista de segmentos. Dado o primeiro segmento formador da subfigura, procura-se por um segmento componente do padrão cuja definição seja compatível com a descrição do segmento dado. Ao encontrar este segmento, comparamos, um a um, se as descrições dos outros segmentos formadores da subfigura são compatíveis com os segmentos do padrão, segundo a ordem na qual estes últimos foram descritos (ou no sentido contrário). Caso os formatos e propriedades dos segmentos seleccionados sejam compatíveis com aqueles do padrão, a figura foi reconhecida.

Caso contrário, a comparação acima é repetida com o próximo padrão com o mesmo número de componentes. Esta comparação se repete até que a figura seja reconhecida ou que se acabem os padrões com este número de segmentos componentes. Neste caso, uma mensagem de erro é mostrada e o reconhecimento é cancelado. A figura reconhecida é colocada em uma lista de figura. Os segmentos seleccionados que não foram utilizados para formar a subfigura são colocados nesta lista como conectores.

Todo o processo descrito acima é repetido até que os segmentos encontrados na etapa de segmentação acabem.

#### 4.5.2- Figuras Abertas:

Quando os padrões que poderão formar a figura são abertos, a busca na lista de padrões é realizada a cada novo segmento lido da lista de segmentos conforme algoritmo descrito no capítulo anterior. A comparação entre os segmentos selecionados e os segmentos componentes de um determinado padrão é feita de modo análogo ao caso de figuras fechadas. Entretanto, o momento de realizar esta comparação é que é diferente.

Dado um segmento, procura-se na lista padrão se existe alguma figura de um único segmento que possa ser definida pelo segmento dado. Caso não exista, selecionamos o próximo segmento da lista e testamos se este segmento pode formar uma figura. Se não formar, verificamos se existe alguma figura padrão que possa ser formada pelos dois segmentos selecionados. A busca continua, pegando-se o próximo segmento da lista de segmentos e procurando por uma figura padrão de 1, 2 ou 3 componentes que possa ser formada pelos segmentos dados. Observamos que a cada segmento novo que é lido, aumenta-se consideravelmente o número de comparações com os padrões definidos. Se tivermos selecionado  $N$  segmentos, devemos testar se existe algum padrão de 1, 2, 3, ...,  $N$  componentes que possa ser formado por estes segmentos. Com isto, o tempo de reconhecimento será consideravelmente alto. Figuras fechadas também podem ser reconhecidas por este método. Entretanto, figuras abertas não podem ser reconhecidas pelo método anterior.

Assim como na busca por figuras fechadas, a busca começa a partir do último segmento selecionado e todas as figuras reconhecidas são colocadas em uma lista. Os segmentos restantes que não foram utilizados para formar um padrão são tidos como sendo conectores.

#### 4.6- RECONHECIMENTO DAS FIGURAS COMPOSTAS:

Após o reconhecimento das figuras simples e dos conectores, usaremos a lista de figuras criada nesta etapa para procurar pelas figuras compostas que possam existir no desenho de entrada. Esta busca basear-se-á nas figuras já reconhecidas e no posicionamento entre elas, pois as componentes das figuras compostas são outras figuras, ao invés de segmentos, como na busca por figuras simples.

Para cada figura composta definida na lista padrão, testamos se na lista das figuras já reconhecidas existem todas as componentes da figura composta. Caso existam, todas as figuras que possam ser componentes da figura composta são selecionadas e é verificado se existe alguma combinação das figuras selecionadas que satisfaça as propriedades existentes na definição da figura composta, de tal forma que ela possa formar esta figura. Se existir esta combinação, todas as figuras componentes são retiradas da lista de figuras reconhecidas e é colocado um novo elemento correspondendo à figura composta encontrada. Esta busca é realizada para todas as figuras compostas cadastrada na biblioteca de padrões.

O processo acima é repetido até que, ao percorrer todas as figuras compostas cadastradas na biblioteca, nenhuma mais seja encontrada

no desenho de entrada. O motivo desta repetição da busca em toda a biblioteca é devido à possibilidade de uma figura composta poder ser formada por outra figura composta. Logo, pode acontecer de na primeira procura pelas figuras compostas cadastradas, seja encontrado figuras compostas que são componentes de outras. Como estas componentes ainda não existiam na lista de figuras já reconhecidas, a busca pela figura composta formada por outras também compostas, falharia. Ao percorrer todas as figuras compostas cadastradas e nenhuma for encontrada no desenho de entrada, temos a garantia que nenhuma outra figura será encontrada, pois a lista de figuras reconhecidas não foi alterada desde a última averiguação, e como nenhuma outra figura composta foi encontrada, o mesmo acontecerá se procurarmos mais uma vez, já que esta busca é baseada nos elementos que constam na lista de figuras já reconhecidas. Por restrição de implementação, uma figura composta pode ter no máximo 20 componentes.

#### **4.7- DETERMINACAO DAS INTERCONEXOES:**

Após reconhecer todas as subfiguras padrões que compõem o desenho de entrada, temos de determinar como elas estão conectadas entre si. Isto será feito a partir da lista das figuras reconhecidas criada na etapa anterior. Cada elemento desta lista representa uma figura padrão ou então um conector, onde temos, entre outras informações, a posição ocupada pela figura. Baseado na posição das figuras e dos conectores, poderemos determinar quais figuras estão conectadas com quais, e de que forma elas estão conectadas, isto é, através de quais pontos de E/S.

Em alguns tipos de figuras, existirão cruzamentos entre os conectores sem entretanto eles estarem interconectados. Isto ocorre, por exemplo, nos diagramas de circuito digitais, onde, por estética ou por falta de espaço, os segmentos utilizados para ligação dos componentes do circuito se interceptam, sem entretanto, haver uma conexão física entre eles. No protótipo implementado, o cruzamento de conectores não é considerado para efeito de interconexão entre os componentes.

Durante a determinação das interconexões serão criadas duas listas. A primeira, denominada lista completa de conexões, indicará simplesmente qual subfigura está conectada a um determinado conector, não interessando de que modo estão ligados. Nesta lista, as ligações serão do tipo conector-não conector e conector-conector. Observamos que ligações diretas do tipo não conector-não conector não podem existir, pois duas figuras só poderão ser ligadas por conectores. Esta lista será montada baseada nas posições das figuras e conectores reconhecidos e cada elemento da lista conterá seguintes dados, conforme mostrado na figura 4.6:

1) nome: nomes dos padrões associados às duas figuras que estão conectadas. Como só temos conexão do tipo conector-não\_conector e conector-conector, o nome "cnct" indicará que a figura é um conector;

2) num. da figura: número associado a cada subfigura componente ou conector, na mesma ordem no qual eles foram desenhados. Este número identifica univocamente cada elemento componente do desenho de entrada;

3) coordenada do ponto de conexão entre os dois componentes da conexão. Esta coordenada será utilizada para determinação do ponto de E/S utilizado para a ligação das subfiguras;

4) local: se uma das componentes desta conexão for uma subfigura padrão, este campo indica se a conexão ocorre no lado direito, esquerdo, na

parte superior ou inferior desta subfigura. Este valor também será utilizado para determinação do ponto de E/S onde ocorre a ligação das subfiguras;

5) fig.1 e fig.2: apontadores para lista de figuras reconhecidas, descrevendo os elementos que estão conectados entre si.

A construção desta lista não exige muito esforço computacional, uma vez que ela é construída baseada nas posições das subfiguras padrões e dos conectores. Entretanto, as informações contidas nesta lista estão incompletas, pois não sabemos como as subfiguras estão conectadas, ou seja, através de quais pontos de E/S ocorrem as ligações. Além disso, para determinarmos se duas subfiguras estão conectadas, devemos verificar se existe um conector ligando-as ou então se existem conectores ligados entre si de tal forma que eles definam um percurso entre as duas figuras. Portanto, embora esta lista contenha todas as interconexões existentes na figura de entrada, os dados estão incompletos, além de existirem dados irrelevantes que apenas complicam a descrição dessas interconexões.

Devido aos problemas existentes na lista completa de interconexões, devemos eliminar os dados irrelevantes e colocar na lista de conexões informações sobre os pontos de E/S onde as conexões ocorrem. Desta forma, é criada a segunda lista de conexões, a partir da lista completa, simplificada, onde os elementos pertencentes à lista contém as seguintes informações, conforme mostrado na figura 4.7:

1) os 6 primeiros campos são os mesmos da lista completa de conexões descritos anteriormente;

2) tipo pto conex.: indica se os pontos dos componentes da conexão são pontos de entrada ou de saída;

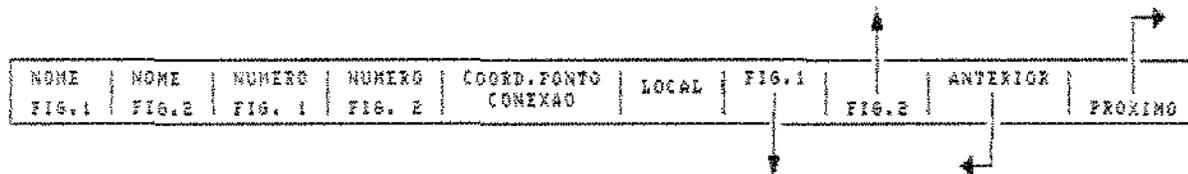


FIGURA 4.6

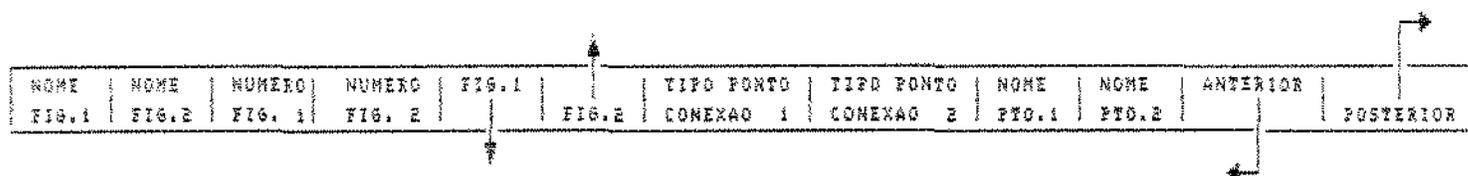


FIGURA 4.7

3) nome pto.: rótulo associado aos pontos utilizados para conexão das subfiguras 1 e 2.

A lista simplificada conterá apenas ligações do tipo não conector-não conector, de tal forma que saberemos diretamente qual subfigura está conectada com qual. Isto é realizado a partir da lista completa de conexões eliminando os conectores e considerando que as figuras estão ligadas diretamente. Após determinar quais figuras estão conectadas, devemos identificar através de quais pontos de E/S estas conexões são feitas. Para realizar a identificação desses pontos, consultaremos a biblioteca de padrões, onde cada ponto de E/S possível está definido. Na lista de conexão que está sendo criada, a partir das coordenadas do local onde é feita a conexão e das coordenadas da figura componente, podemos determinar em qual lado a conexão está localizada e em qual colocação (ou seja, se ela é a primeira, segunda, ..., n-ésima conexão neste lado). A partir desses dados e da descrição dos pontos de E/S existentes na biblioteca, facilmente determinamos seu nome (rótulo) do ponto e tipo (de entrada ou de saída). Caso ocorra alguma conexão ilegal, isto é, uma conexão localizada em algum ponto não definido pela biblioteca, uma mensagem de erro indicando o problema é emitida e o processo de reconhecimento é encerrado.

Através desta lista podemos realizar alguns testes de validação das figuras reconhecidas, como a verificação do número total e posição dos pontos de entrada e saída, ou determinar se existe alguma subfigura isolada, ou seja, se existe subfigura que não está conectada a nenhuma outra.

No protótipo implementado, não é necessário que todos os pontos de E/S definidos na biblioteca sejam utilizados. Poderão existir pontos que não estejam ligados a nada. Isto permite uma flexibilidade maior na definição dos padrões, pois os padrões que possuem número de pontos variáveis são definidos apenas uma vez, descrevendo todos os pontos de E/S que podem aparecer na figura. Caso exista figuras que possuem várias combinações de pontos de E/S que não podem ocorrer simultaneamente, pode-se definir a figura mais de uma vez na biblioteca, com os mesmos componentes, mas com os pontos de conexão diferentes.

O resultado final do reconhecimento será esta lista simplificada de conexões contendo as figuras reconhecidas com seus respectivos pontos de E/S utilizados para as interconexões. Uma vez determinado cada subfigura padrão componente da figura de entrada e como elas estão conectadas, todo o desenho terá sido reconhecido, necessitando apenas, talvez, de um resultado gráfico mostrando um desenho do esboço de entrada com contornos bem definidos.

#### 4.8- RECONHECIMENTO DE CARACTERES:

Muitas vezes necessitamos de colocar informações adicionais em um desenho de modo a torná-lo mais consistente. Geralmente essas informações são palavras ou números especificando o tipo ou função de um determinado componente do desenho, por exemplo. Sem estas informações extras, o desenho pode perder muito em seu significado. Portanto, em muitos casos, é fundamental que um sistema de reconhecimento de padrões também reconheça caracteres.

No protótipo que foi implementado, não foi feita nenhum tratamento especial para reconhecer caracteres. Embora não nos preocupamos com o reconhecimento de caracteres, acreditamos que o método se aplica neste caso, pois caracteres são um tipo especial de padrão, onde cada letra ou número possui um formato bem definido. Abaixo descrevemos como o reconhecimento de caracteres poderia ser implementado, considerando que o desenho de entrada poderia ser composto por figuras padrões e caracteres.

O processo utilizado para o reconhecimento de caracteres não é diferente do processo utilizado para reconhecer outras figuras. Os caracteres poderão ser reconhecidos como um conjunto isolado de padrões, onde o usuário deverá indicar explicitamente o momento no qual ele está entrando com caracteres ou com subfiguras componentes do desenho. Isto se deve por dois motivos: ganho temporal e os procedimentos utilizados para o reconhecimento das subfiguras e dos caracteres podem ser diferentes.

Como os caracteres são padrões diferentes daqueles que estão sendo utilizados para compor o desenho de entrada, eles são definidos em uma biblioteca separada. Sendo assim, se o usuário indicar que ele está entrando com um caracter, saberemos em qual biblioteca buscar um padrão definido que possa casar com o caracter desenhado. Com isto, o tempo gasto para o reconhecimento diminui, pois o número de comparações entre a subfigura desenhada e os padrões definidos na biblioteca é menor, uma vez que não precisaremos utilizar as duas bibliotecas para realizar esta tarefa.

Um outro problema que surgiria caso não fosse indicado esta

separação na entrada, estaria no fato dos caracteres não serem padrões fechados, enquanto que os padrões utilizados para compor o desenho de entrada podem ser todos padrões fechados. Como foi visto anteriormente, fizemos uma otimização no reconhecimento quando os padrões componentes forem todos fechados. O tempo gasto para o reconhecimento de figuras fechadas é menor, uma vez que os procedimentos usados para reconhecer figuras abertas e figuras fechadas são diferentes e no caso de figuras fechadas, o número de comparações com a biblioteca é consideravelmente menor. Se a separação entre a entrada de caracteres e subfiguras não fosse feita, não poderíamos utilizar esta otimização e portanto, necessitaríamos de reconhecer todas as subfiguras como se elas fossem padrões abertos, aumentando consideravelmente o tempo de reconhecimento, além do que, o número de padrões a serem testados seria somado aos padrões existentes na biblioteca de definição dos caracteres, piorando ainda mais o rendimento temporal do reconhecedor.

O reconhecimento de caracteres seria feito de maneira análoga ao reconhecimento das subfiguras componentes do desenho de entrada. Como sabemos a cada instante o tipo de padrão que está sendo desenhado, montaremos uma lista de segmentos separada para os caracteres. A construção desta lista é feita do mesmo modo que na lista dos segmentos componentes das subfiguras e ela será utilizada para o reconhecimento dos caracteres, assim como no caso das subfiguras. Fazendo uma analogia com o reconhecimento das subfiguras, para os caracteres, o processo de reconhecimento encerra após todos os caracteres *simples* serem reconhecidos, pois não existem caracteres compostos nem é necessário determinar as interconexões entre os caracteres, já que eles não estão conectados a nenhum outro. Logo, o resultado do reconhecimento dos caracteres será uma

lista indicando quais caracteres foram escritos e as suas localizações e tamanhos, de forma que possamos fazer um desenho melhorado do esboço de entrada como um resultado visual do reconhecimento.

## CONCLUSOES

### 5.1- RESULTADOS:

Os testes realizados no reconhecedor foram feitos em duas etapas. Na primeira etapa analisamos o reconhecimento de figuras isoladas com o objetivo de testar a independência do método em relação à ordem na qual os componentes da figura são desenhados. Na segunda etapa os desenhos de entrada foram compostos por subfiguras padrões conectadas, onde o objetivo foi o teste do reconhecimento dos pontos de conexões utilizados para as interconexões entre as subfiguras.

Como exemplo dos testes, as figuras 5.1 a 5.14 mostram os exemplos que foram submetidas para o reconhecimento. Cada segmento componente das figuras estão numeradas, indicando a ordem na qual eles foram desenhados, e as setas indicam a seqüência de entrada dos pontos que compõem estes segmentos.

A tabela 5.1 indica o tempo gasto para o reconhecimento das figuras mostradas anteriormente. Nesta tabela, observamos que o tempo gasto no reconhecimento de figuras que possuem arcos de circunferência como componentes foi consideravelmente maior que o tempo gasto no reconhecimento de figuras cujos componentes são todos segmentos de reta. Isto ocorre devido ao algoritmo utilizado para encontrar os padrões, que, a cada novo segmento analisado, testa se este segmento está conectado aos segmentos

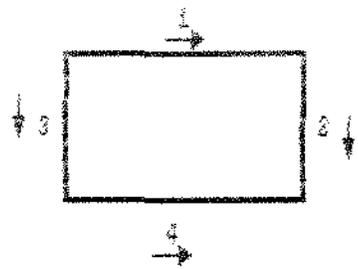


FIG-5.1

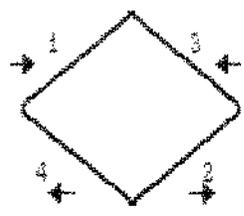


FIG-5.2

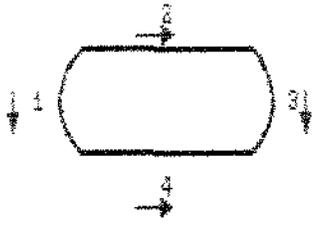


FIG-5.3

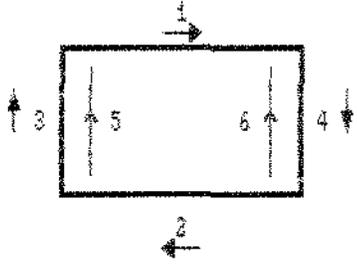


FIG-5.4

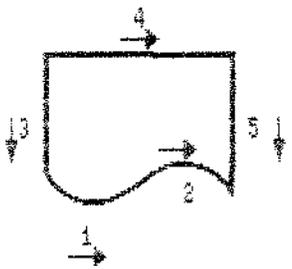


FIG-5.5

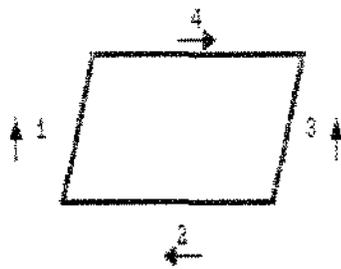


FIG-5.6

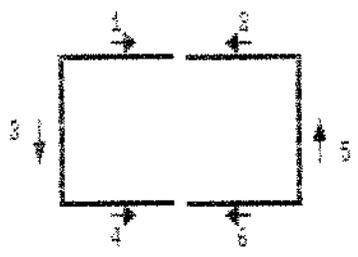


FIG-5.7

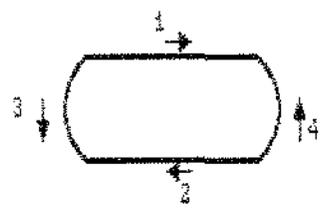


FIG-5.8

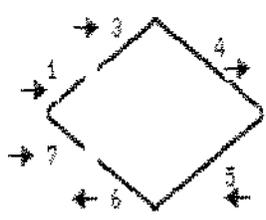


FIG-5.9

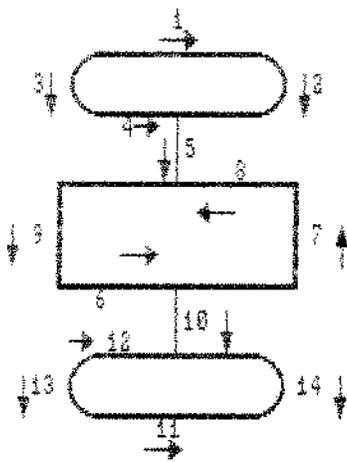


FIG-5.10

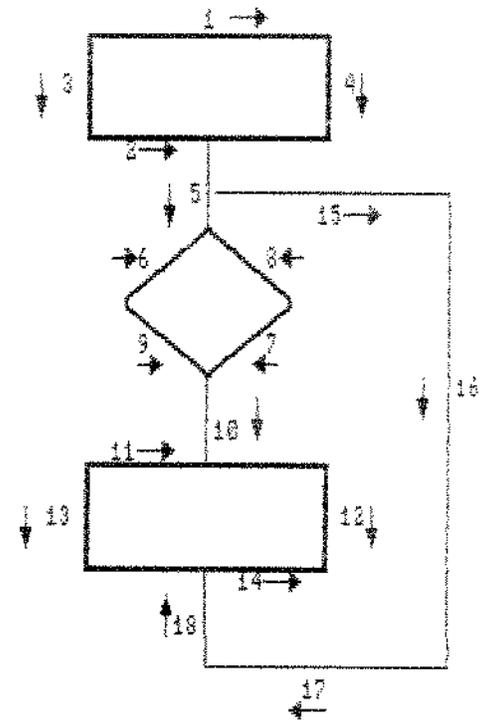


FIG-5.11

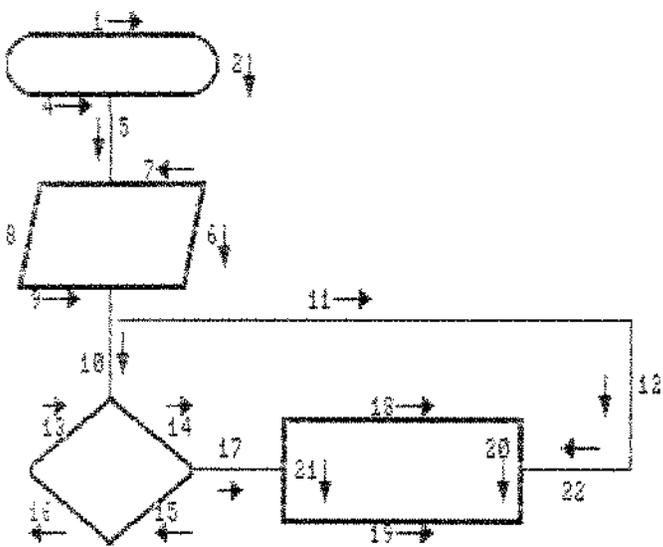


FIG-5.12

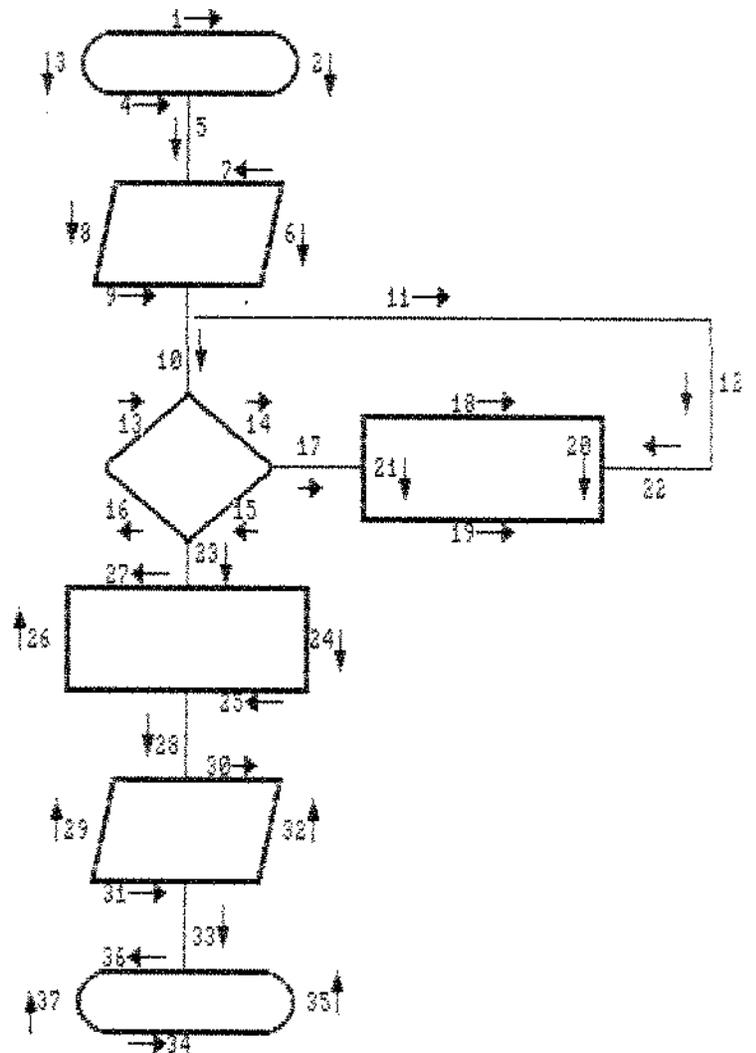


FIG-5.13

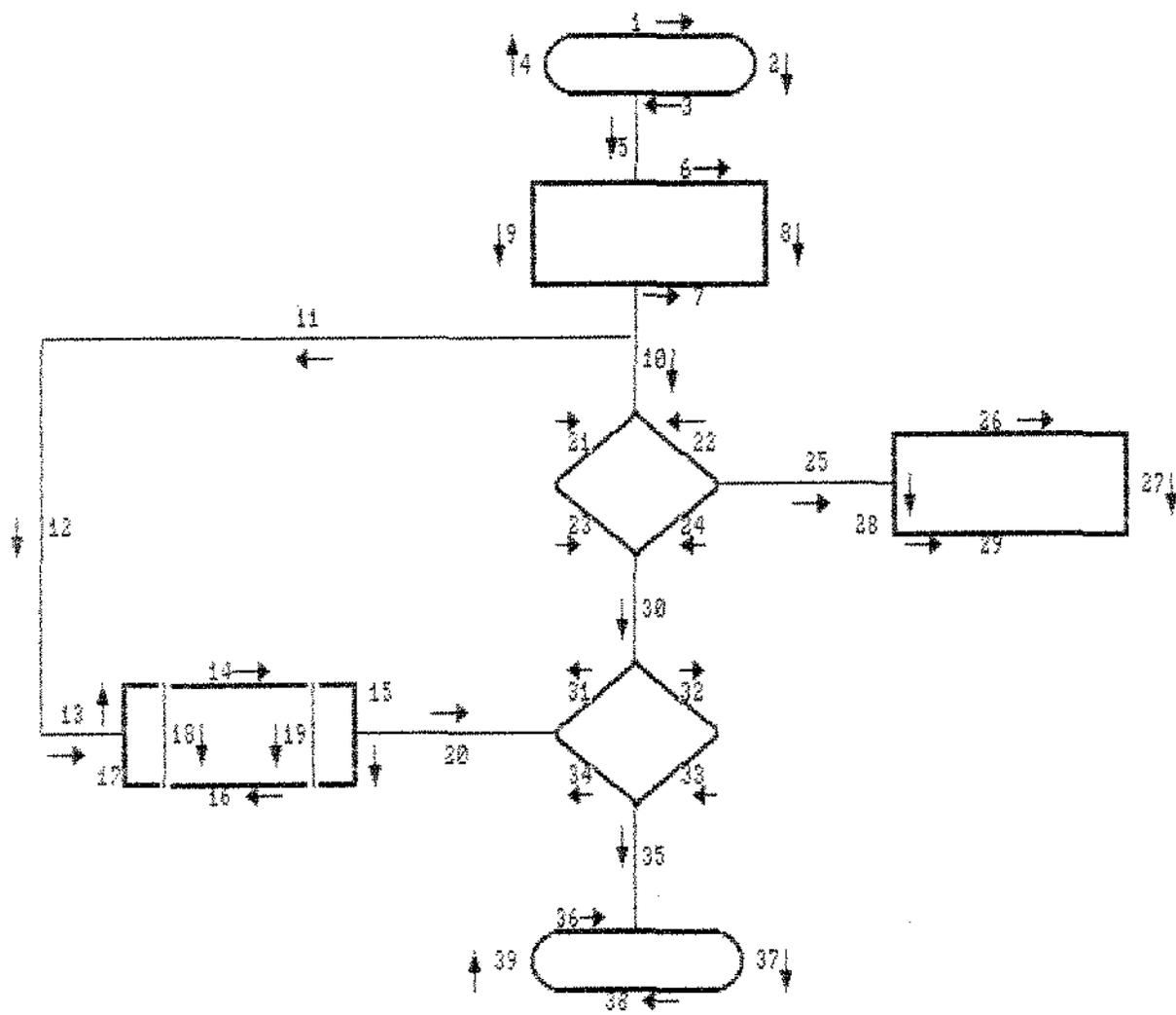


FIG-5.14

FIGURA	TEMPO GASTO (seg)	
	FIG. ABERTA	FIG. FECHADA
5.1	0.52	0.65
5.2	0.53	0.66
5.3	2.78	0.99
5.4	0.81	1.00
5.5	2.27	1.13
5.6	0.76	0.78
5.7	1.43	1.20
5.8	1.99	0.99
5.9	1.49	1.38
5.10	8.40	5.59
5.11	4.05	4.14
5.12	9.90	9.05
5.13	20.55	14.49
5.14	46.27	24.15

TABELA 5.1 : tempo gasto para reconhecimento

anteriormente selecionados. Este teste é realizado através do cálculo da interseção deste segmento com os demais, utilizando suas equações analíticas para encontrar o ponto de interseção. Este cálculo é consideravelmente mais demorado quando temos arcos envolvidos, pois suas equações são do segundo grau, envolvendo o uso de algumas funções matemáticas complexas computacionalmente, como o cálculo de raiz quadrada, o que não ocorre quando somente segmentos de reta estão envolvidos no cálculo.

Um outro detalhe que podemos observar na tabela 5.1 é que, embora o algoritmo utilizado para o reconhecimento de figuras abertas seja, em média, visivelmente mais lento que o utilizado para o reconhecimento de figuras fechadas, existem casos em que o tempo gasto no reconhecimento de figuras abertas é menor que o caso de figuras fechadas. Isto ocorre quando o padrão a ser reconhecido está localizado no início da biblioteca de padrões. Como o reconhecimento das figuras abertas é realizado pela comparação direta com os padrões definidos na biblioteca (no caso de figuras fechadas, a comparação com os padrões definidos na biblioteca só é feita quando já encontramos uma subfigura fechada), se o padrão procurado seja um dos primeiros existentes na biblioteca, o número de comparações será pequeno, diminuindo assim o tempo gasto no reconhecimento. Em compensação, se o padrão procurado esteja definido no final da biblioteca, o tempo gasto no reconhecimento será elevado. Isto pode ser visto claramente na tabela 5.1, observando o tempo gasto no reconhecimento das figuras 5.2 e 5.3 que, embora tenham o mesmo número de segmentos componentes, o reconhecimento da primeira figura é consideravelmente mais rápido.

Entretanto, como já era esperado, à medida que as figuras de entrada ficam mais complexas, o tempo gasto no reconhecimento de figuras abertas vai aumentando consideravelmente, tornando este processo bem mais lento que o reconhecimento das figuras fechadas, como pode ser visto na tabela 5.1, compensando assim a distinção feita entre o reconhecimento das figuras abertas e das figuras fechadas.

Os testes foram realizados em um micro-computador tipo IBM-PC/XT.

Como pode ser observado na tabela anterior, o tempo gasto no reconhecimento não foi muito grande. Acreditamos que o desempenho do protótipo implementado poderá ser melhorado, uma vez que durante a sua implementação não nos preocupamos em otimizar nenhuma função e nem o código fonte gerado (o protótipo foi implementado em linguagem C). O programa fonte possui aproximadamente 10.000 linhas de código, enquanto o executável gerado ocupa em torno de 110Kbytes em disco. O programa fonte encontra-se disponível no DCC-UNICAMP.

Embora o protótipo implementado não possua um processo de segmentação altamente tolerante à distorções, acreditamos que o desempenho obtido não será prejudicado caso esta etapa venha a ser implementada, pois esta etapa poderá ser realizado durante a entrada do desenho.

## 5.2- SUGESTOES PARA FUTURAS MELHORIAS NO PROTOTIPO:

A implementação do reconhecedor está incompleta. Embora a parte referente ao método de reconhecimento estar totalmente implementada, o módulo de segmentação precisa ser melhorado, pois o atual é pouco tolerante a distorções, principalmente quando trabalhamos com arcos. Como a segmentação é um processo fundamental que determina diretamente a taxa de reconhecimento e, ao que parece, é muito trabalhoso obter e implementar um método de segmentação altamente tolerante a distorções, uma possível solução para o problema seria a utilização de um método de segmentação com uma tolerância razoável a pequenas distorções e, a partir dos segmentos resultantes da aplicação deste método, processá-los de modo que os

segmentos que foram quebrados em vários outros sejam devidamente unidos. Observamos durante a implementação que o problema enfrentado na segmentação era o excesso de segmentos obtidos, e não, a falta deles, ou seja, todos os segmentos componentes de uma determinada subfigura foram encontrados; entretanto, vários desses segmentos eram quebrados em dois ou mais outros.

Uma outra deficiência existente na implementação está na apresentação dos resultados. Seria extremamente útil se tivéssemos como saída um desenho melhorado da figura de entrada, com os contornos muito bem formados, sem as deformações existentes na figura de entrada. A dificuldade em construir esta saída estaria nos ajustes necessários nas posições e tamanhos de cada subfigura e na centralização dos conectores em relação a cada subfigura componente de uma conexão. Em um desenho final teríamos de fazer um ajuste no posicionamento das subfiguras, de modo que figuras localizadas abaixo de outras, fiquem na mesma coluna, evitando assim, que elas permaneçam espalhadas no desenho. Teríamos também de centralizar os conectores participantes de uma determinada conexão, além de fazer um ajuste nos tamanhos das subfiguras padrões, para que elas fiquem com dimensões compatíveis uma em relação à outra. Creio que o processo de ajustes para a composição final do desenho, embora trabalhoso, não apresentaria maiores problemas.

Embora não seja necessário, poder-se-ia melhorar a entrada do desenho, permitindo que este ocupe mais de uma tela no monitor, já que do modo que está implementado o protótipo, o desenho de entrada pode ocupar apenas uma tela. Com isto, seria interessante um gerenciamento de janelas, onde em uma mesma tela, poderíamos observar partes distintas do desenho que porventura não viesse a caber em uma única tela; estas janelas poderiam

trabalhar com escalas distintas. Uma outra facilidade que o protótipo poderia ter é a possibilidade de alteração do desenho durante a sua entrada. Como a segmentação é realizada junto com a entrada, ao alterar uma parte do desenho, devemos descobrir quais segmentos foram gerados nesta parte específica do desenho e substituí-los por outros. Poder-seria também permitir que um desenho, já reconhecido ou não, seja composto com a aplicação que está sendo desenvolvida no momento, ou seja, em um determinado momento da entrada do desenho, o usuário poderia recuperar algum desenho armazenado em disco e inseri-lo na aplicação que está sendo desenhada.

Através dessas facilidades adicionais, o usuário poderia ter um nível de interação maior com a máquina. Entretanto, como o objetivo do trabalho não foi a construção de um editor gráfico associado ao reconhecedor, deixamos aqui apenas uma sugestão de melhoria a ser realizada, quem sabe, em um futuro próximo.

Uma outra melhoria que poderia ser inserida no protótipo, seria a possibilidade de associar um sentido aos conectores. Em um fluxograma ou rede de petri, por exemplo, o sentido dos conectores determina um "fluxo dos acontecimentos" segundo a semântica associada a estas figuras. Desta forma, poder-se-ia fazer uma simulação da evolução do sistema representado pela figura de entrada. Como entrada receberíamos um esboço da figura, que poderia conter alguns comandos, valores ou expressões associadas aos padrões, em alguma sintaxe pré-definida. A partir deste esboço, o reconhecedor de figuras extrairia os padrões componentes do desenho e indicaria como eles estão conectados. Poderíamos representar os sentidos associados aos conectores, pela seqüência em que foram fornecidos os pontos

de um dedo conector, através de uma indicação explícita, ou então através do desenho pela colocação de setas, as quais o protótipo deveria reconhecer.

Após o reconhecimento da figura de entrada e dos valores, comandos ou expressões associadas aos padrões, poderíamos utilizar o significado desses padrões para descrever o comportamento dinâmico do sistema representado pela figura. Nos circuitos digitais ou elétricos, por exemplo, poderíamos calcular os valores de algumas grandezas físicas em pontos especificados, ou então, o estado de um determinado sistema, a partir de valores iniciais das variáveis envolvidas. Em um fluxograma poderíamos ter como saída um programa escrito em alguma linguagem que representasse o esboço dado. Acredito que seria de grande importância obter o comportamento de um sistema a partir de um esboço representativo deste sistema.

Finalmente, devemos ainda implementar o funcionamento das regras de conexão entre símbolos, tal como descrito no capítulo 02. Como já foi visto, através do uso destas regras poderíamos identificar alguns erros de segmentação e tentar corrigi-los. Do ponto de vista de implementação e funcionamento do reconhedor, substituiríamos alguns dos segmentos componentes das subfiguras que violassem alguma dessas regras de conexão, de tal forma que tivéssemos novas subfiguras de formatos semelhantes às anteriores, mas que não violassem nenhuma das regras de conexão definidas. Os segmentos a serem substituídos seriam escolhidos de tal forma que o desvio total em relação ao padrão cadastrado seja o menor possível. Caso após substituir os segmentos houver ainda alguma regra de conexão violada, temos duas alternativas: cancelar o reconhecimento, indicando onde o erro

ocorreu, ou substituir as subfiguras responsáveis pelo erro por outras de formatos semelhantes e cujas regras de conexão fossem satisfeitas. Baseado no artigo [MUR86], no qual o método proposto foi inspirado, acreditamos que a taxa de reconhecimento aumentaria, pois, além de poder substituir as subfiguras reconhecidas erroneamente, teremos a oportunidade de validar o reconhecimento. Entretanto, como haveria mais processamento, o tempo médio gasto para o reconhecimento aumentaria. Caso o reconhecimento tenha sido realizado corretamente, o processamento extra seria simplesmente alguns testes, o que não degradaria sensivelmente o tempo gasto para o reconhecimento. Um processamento maior somente seria necessário quando houvesse erro no reconhecimento. Como é preferível esperar um pouco mais por um resultado confiável ao invés de obter um reconhecimento errado, a relação custo-benefício seria positiva.

### 5.3- CONCLUSOES:

O objetivo deste trabalho foi a construção de um método para o reconhecimento de figuras esboçadas a mão. Estas figuras devem ser compostas por subfiguras padrões interconectadas, cujos formatos sejam bem definidos e distintos entre si. Através dos resultados obtidos pelo protótipo implementado, acreditamos que o método proposto mostrou-se viável em termos computacionais, pois o tempo gasto para o reconhecimento não foi elevado, o método é de fácil entendimento, confiável, e também, sua implementação pode ser realizada em um microcomputador, tornando-o bastante acessível.

O grande problema do método proposto está em obter um método de segmentação da figura de entrada que seja altamente tolerante a distorções. Este método é de fundamental importância, já que o reconhecimento da figura de entrada está baseado nos segmentos componentes de cada subfigura padrão que forma o desenho. Através dos resultados obtidos, temos que a taxa de reconhecimento será próxima de 100% caso a segmentação da figura de entrada seja feita corretamente, o que mostra a confiabilidade do método proposto.

Com a implementação de algumas das melhorias no protótipo sugeridas na seção anterior e outras otimizações no código fonte do programa, com a finalidade de aumentar a velocidade de reconhecimento, o protótipo poderá servir como um sistema de entrada direta de dados gráficos. Esta entrada seria realizada como se o usuário estivesse utilizando um lápis e papel. Como saída, ele obterá um desenho melhorado do esboço de entrada, além de poder realizar testes de simulação sobre o modelo representado. Desta forma, teríamos um alto nível de interação usuário-máquina, onde os processos intermediários desde a entrada do esboço e o resultado final da aplicação seriam transparentes ao usuário.

## BIBLIOGRAFIA

[FU-80] KING-SUN FU

Recent Developments in Pattern Recognition

IEEE Transactions on Computers, vol C-29, n910, p 845-854,

Oct. 1980

[GAI78] G. GAILLAT

A Simple Learning Decision Algorithm for Character Recognition and  
Pattern Classification

Pattern Recognition, vol 10, p 99-104, 1978

[G0080] G. GOOS & J. HARTMANIS (editors)

Lecture Notes in Computer Science, vol 89, cap 03

Darmstadt, Sept 1980, ed. Springer Verlag

[GRA72] G. H. GRANLUND

Fourier Preprocessing for Hand Print Character Recognition

IEEE Transactions on Computers, p 195-201, Feb 1972

[HAR83] STEVEN HARRINGTON

Computer Graphics - A Programming Approach

McGraw-Hill Book Comp, 1983

[HIR85] P. HIRANVANICHAKORN, T. AGUI & M. NAKAJIMA

A Recognition Method of Handprinted Thai Characters by Local Features

Transactions of the IECE of Japan, vol E68, n92, p 83-90, Feb. 1985

[HOS82] NAMONU HOSAKA & FUMIHIKO KIMURA

Using Handwriting Action to Construct Models of Engineering Objects  
Computer, vol 15, n<sup>o</sup>11, p 35-47, Nov. 1982

[HUA86] JUN S. HUANG & KEREN CHUANG

Heuristic Approach to Handwritten Numeral Recognition  
Pattern Recognition, vol 19, n<sup>o</sup>1, p 15-19, 1986

[IKE81] IKEDA, YAMAMURA, MITAMURA, FUJIWARA, TOMINANGA & KIYONO

On Line Recognition of Handwritten Characters Utilizing Positional  
and Stroke Vector Sequences  
Pattern Recognition, vol 13, n<sup>o</sup>3, p 191-206, 1981

[ITO78] M. R. ITO & T. L. CHUI

On-line Computer Recognition of Proposed Standard ANSI(USASI)  
Handprinted Characters  
Pattern Recognition, vol 10, p 341-349, 1978

[ISH87] M. ISHII, M. IWASAKI & M. YAMADA

A Character Recognition System with Learning Dictionary for Handwritten Drawings  
System and Computers um Japan, vol 18, n<sup>o</sup>2, p 65-76, 1987

[JAN85] JANSEN, NULLMEIER & ROEDINGER

Handsketching as a Human Factor Aspect in Graphical Interaction  
Computer & Graphics, vol 9, n<sup>o</sup>3, p 195-210, 1985

[KEL86] W. A. KELLOG & T. J. BREEN

IBM Research Report n913942, Sep 1986

[LIN78] WEN C. LIN & LANE H. PUN

Machine Recognition and Plotting of Hand-sketched Line Figures

IEEE Transactions on System, Man and Cybernetics, vol SMC-8,

n91, Jan 1978

[MAN86] J. MANTAS

An Overview of Character Recognition Methodologies

Pattern Recognition, vol 19, n96, p 425-430, 1986

[MUR86] MURASE & WAKAHARA

On Line Hand-Sketching Figure Recognition

Pattern Recognition, vol 19, n92, p 147-160, 1986

[NEV79] ARTHUR J. NEVINS

An Orientation Free Study of Handprinted Characters

Pattern Recognition, vol 11, p 155-164, 1979

[NEW79] WILLIAM M. NEWMAN & ROBERT F. SPROULL

Principles of Interactive Computer Graphics

McGraw-Hill Book Comp, 1979

[PFA82] G. PFAFF, H. KUHLMANN & H. HANUSA

Constructing User Interfaces Based on Logical Input Devices

Computer, vol 15, n911, p 62-68, Nov 1982

[R0573] AZRIEL ROSENFELD

Picture Processing by Computer

Academic Press, 3rd printing, 1973

[RUI89] EVANDU EDUARDO SERON RUIZ

Comparação de técnicas e métodos para visão computacional em  
ambientes industriais

Tese de Mestrado, UNICAMP, FEE

Maio, 1989

[SHI87] S. SHIMADA, S. KAKUMOTO & M. EJIRI

A Recognition Algorithm of Dashed and Chained Lines for Automatic  
Inputing of Drawings

System and Computers in Japan, vol 18, nº6, p 25-37, 1987

[SPR85] R. SPROULL, W. SUTHERLAND & M. ULLNER

Device-independent Graphics

McGraw-Hill Book Comp, 1985, cap 1-3,10

[SUE80] CHING Y. SUEN, MARC BERTHOLD & SHUNI MORI

Automatic Recognition of Handprinted Characters - The State of the  
Art

Proceedings of IEEE, vol 68, nº4, p 469-487, Apr 1980

[TOU72] J. T. TOU & R. C. GONZALEZ

Recognition of Handwritten Characters by Topological Feature  
Extraction and Multilevel Categorization

IEEE Transactions on Computers, p 776-785, Jul 1972

[WAR85] J. R. WARD & B. BLESSER

Interactive Recognition of Handprinted Characters for Computer Input  
IEEE - Computer Graphics & Application, vol 5, n29, p 24-37, 1985

[YAM80] KAZUHIKI YAMAMOTO & SHUNJI MORI

Recognition of Handprinted Characters by an Outermost Point Method  
Pattern Recognition, vol 12, n24, p 229-236, 1980

[YHAB81] E. F. YHAP & E. C. GREANIAS

An On-line Chinese Character Recognition System  
IBM J. Res. Develop, vol 25, n23, p 187-194, May 1981