

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

RECONHECIMENTO E LOCALIZAÇÃO DE OBJETOS BI-DIMENSIONAIS  
PARCIALMENTE VISÍVEIS

POR : ROBERTO HIROSHI HIGA

ORIENTADOR : MÁRCIO LUIZ DE ANDRADE NETTO

Este exemplar corresponde à relação final da tese  
defendida por Roberto Hiroshi Higa

aprovada pela Comissão  
Julgadora em 19 12 1990

  
Orientador

Dissertação apresentada à Faculdade de Engenharia Elétrica -  
FEE - UNICAMP, como parte dos requisitos exigidos para obtenção do  
título de MESTRE EM ENGENHARIA ELÉTRICA

## AGRADECIMENTOS

- Ao professor Márcio Luiz de Andrade Netto, pela orientação ao longo do desenvolvimento deste trabalho.
  
- Aos meus ex-professores Sérgio de Mello Schneider e Viktor Bojarczuk, pelo incentivo inicial.
  
- Ao CNPq, CAPES e FAEP, pelo apoio financeiro.
  
- Aos amigos José Luiz, Marcélia e Hsieh, meus companheiros de estrada,
  
- Aos companheiros de moradia Sérgio, Raimes e Ely.
  
- A todos os meus familiares e
  
- A todos aqueles que direta ou indiretamente contribuíram para a execução deste trabalho.

Dedico este trabalho aos meus

familiares e, em especial,

à minha mãe Massaé.

## RESUMO

Este trabalho refere-se aos problemas envolvidos na construção de um sistema capaz de reconhecer e localizar objetos bi-dimensionais parcialmente visíveis e dispostos de forma aleatória.

Para isto, são consideradas as técnicas mais correntemente utilizadas para a construção de um sistema sistema de visão computacional. Define-se um domínio de aplicação e, então, descreve-se a definição e implementação de um sistema capaz de realizar tal tarefa.

Os resultados obtidos são apresentados e os problemas encontrados, discutidos.





**CAPÍTULO 5 - DISCUSSÃO E CONCLUSÃO**

5.1. Análise.....78  
5.2. Sugestões para posterior desenvolvimento.....79  
5.3. Considerações finais.....81

**REFERÊNCIAS BIBLIOGRÁFICAS.....83**

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 - APLICAÇÃO DE VISÃO COMPUTACIONAL EM AUTOMAÇÃO INDUSTRIAL

De uma forma geral, a automatização de processos tem por objetivo liberar o trabalhador humano de tarefas excessivamente repetitivas e, ainda, substituí-lo em tarefas que envolvam risco à sua saúde. Particularmente em processos produtivos, busca-se índices de produtividade mais elevados, ou seja, um nível de produção maior sem uma elevação de custos excessiva. Entretanto, muitas das tarefas envolvidas nestes processos são de grande complexidade e envolvem habilidades de interação com o ambiente, características do ser humano. Isto sugere que, para se obter um grau de automatização cada vez maior, faz-se necessário o desenvolvimento de máquinas dotadas de uma maior capacidade de interação com o ambiente. Assim, é cada vez mais frequente a utilização de tecnologia de sensoriamento com o objetivo de proporcionar uma melhor interação destas máquinas com o ambiente [GONZALEZ-82].

Dentre os sentidos dos quais o ser humano é dotado, aquele que tem despertado maior interesse é o sentido de visão. Isto fez com que a disciplina que busca investigar formas de se dotar máquinas com o sentido visual, conhecida como visão computacional, tenha experimentado um grande desenvolvimento nos últimos anos. Áreas de aplicação como reconhecimento de caracteres, processamento de imagens médicas, sensoriamento remoto e outras têm sido exaustivamente estudadas e sistemas têm sido propostos.

Em aplicações industriais, a utilização de sistemas de visão computacional pode ser dividida em dois grandes grupos [ROSEN-78] :

- 1) Sistemas de visão para inspeção e
- 2) Sistemas de visão para manipulador controlado por sensor (robô industrial).

Os sistemas de visão computacional aplicados à inspeção podem, ainda, ser divididos em inspeção quantitativa e qualitativa. A inspeção quantitativa é necessária quando as dimensões das peças examinadas são críticas e devem ser rigorosamente mantidas dentro de um limite de tolerância. Este tipo de inspeção é utilizado em aplicações como a verificação de características-chave de uma peça e verificação do desgaste de ferramentas. Já na inspeção qualitativa, tenta-se emular o ser humano quando este inspeciona visualmente uma peça sem o auxílio de instrumentos de medida. Exemplos deste tipo de inspeção são a

verificação da correta disposição de um conjunto de peças e a localização de defeitos de fabricação diversos.

No caso de sistemas de visão para manipulador controlado por sensor, o sistema de visão é suposto ser um componente de sensoriamento integrado a um sistema maior de manipulação. Ele visa permitir ao robô industrial um ajuste-fino no posicionamento e orientação de seu efetuator com o objetivo de manipular peças imprecisamente posicionadas. Sua complexidade depende do tipo de tarefa a ser executada e pode variar desde a manipulação de objetos previamente separados até a identificação e localização de peças aleatoriamente dispostas em uma caixa ("bin-picking").

## 12 - CONSIDERAÇÕES SOBRE VISÃO COMPUTACIONAL PARA ROBÔS

Segundo Alan Pugh [PUGH-83], as pesquisas por robôs mais modernos têm se caracterizado pelo esforço em dotá-los de uma capacidade sensorial maior, capaz de habilitá-los a realizar tarefas de elevada complexidade tal como a de montagem, uma aplicação a qual os robôs anteriores, raramente, estiveram associados. Dentro deste quadro, o sistema de visão computacional aparece como o principal dispositivo sensorial a eles associados.

Para serem utilizados por robôs, sistemas de visão computacional devem procurar satisfazer às seguintes condições [CHIN-86] :

- 1) Velocidade: A velocidade na aquisição e processamento da imagem deve ser compatível com a velocidade de execução da tarefa em curso,
- 2) Acuidade: A acuidade na localização das peças presentes na cena deve ser compatível com as necessidades da aplicação e
- 3) Flexibilidade: Variações nas dimensões físicas das peças devem ser suficientemente toleradas e o controle sobre o ambiente sob o qual a tarefa é executada deve ser o mínimo possível.

Os ambientes industriais, por sua vez, apresentam algumas características inerentes ao seu domínio:

- 1) O número de peças em uma dada aplicação, usualmente, é limitado,
- 2) Peças podem ter suas dimensões, tolerâncias e características particulares especificadas com precisão,
- 3) Peças apresentam características capazes de distingui-las das demais e que são encontradas em diferentes tipos.
- 4) Em cenas contendo múltiplas peças, existem várias configurações possíveis.

Tais restrições e propriedades são amplamente exploradas, sendo incorporadas ao próprio projeto do sistema, ou seja, o sistema é construído a partir de um modelamento do domínio de aplicação e do processamento de aquisição da imagem. Disto resulta, por parte dos sistemas de visão, uma forte dependência da aplicação a que se destinam. Tais sistemas são ditos baseados em modelos.

Chin e Dyer [CHIN-86] caracterizaram como as principais funções de um sistema de visão baseado em modelos a extração de características, o modelamento dos objetos e a estratégia de reconhecimento (fig. 1.1). Assim, a construção de um sistema de visão deste tipo consiste, basicamente, em responder às seguintes questões:

- 1) Quais características seriam extraídas da imagem para descrever adequadamente propriedades físicas e suas inter-relações em uma cena ?
- 2) O que constitui uma representação adequada destas características e suas inter-relações para se caracterizar uma classe ?
- 3) Como seria a estratégia de reconhecimento para reconhecer objetos em cenas complexas ?

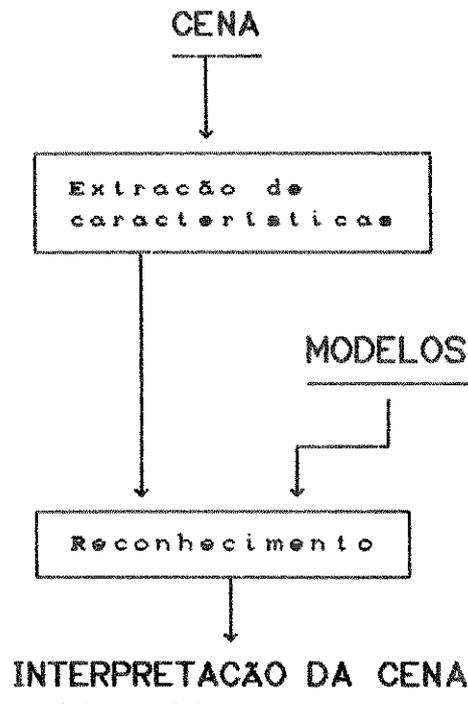


Figura 1.1 - Sistema de visão baseado em modelos

### 13 - OBJETIVOS DO TRABALHO

Ao longo deste trabalho, considera-se o problema de um robô retirando peças de uma caixa ou "Bin-picking" (fig. 1.2). Especificamente, são tratados os problemas envolvidos na construção de seu sistema visual, ou seja, consideradas algumas restrições e propriedades com respeito ao domínio de aplicação e procedimento de aquisição da imagem, tenta-se fornecer respostas para as questões acima levantadas.

Um sistema visual como dispositivo sensor de um robô é útil na medida em que permitir a este manipular com cenas complexas onde diversos objetos aparecem. Assim, define-se como sendo as tarefas deste sistema visual:

- ▶ Identificar as peças presentes na cena,
- ▶ Determinar a localização de cada peça identificada e
- ▶ Determinar as relações de sobreposição entre cada uma delas

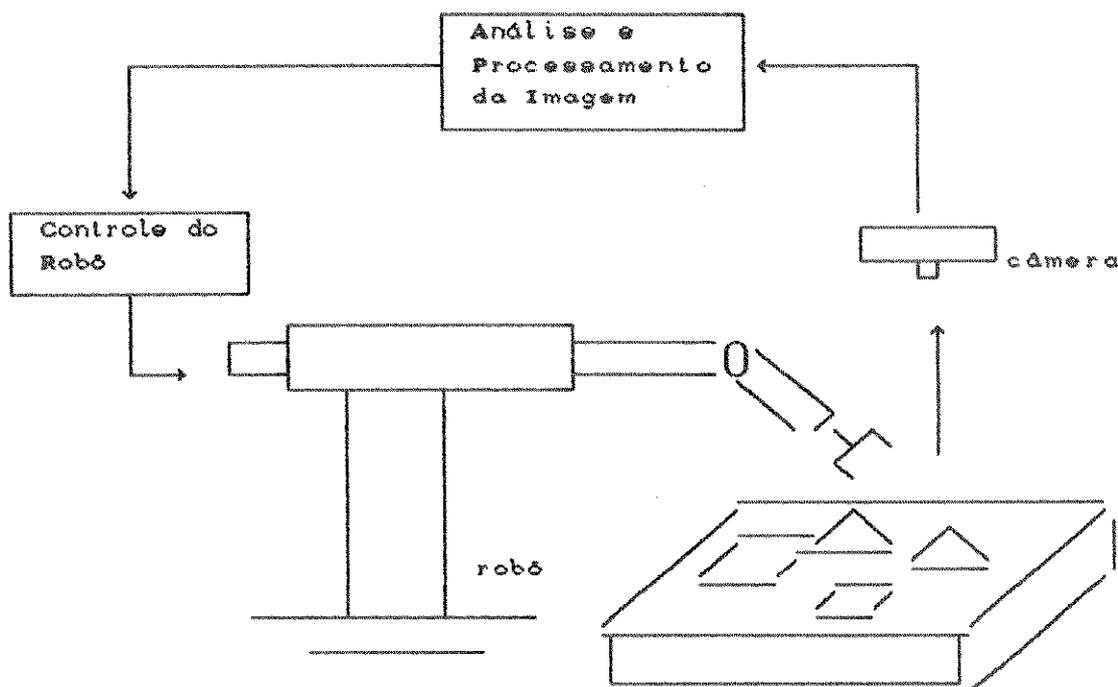


Figura 1.2 - O Problema "Bin-Picking"

Na implementação resultante são utilizadas imagens sintéticas e algumas condições práticas, exigidas de um sistema visual para robôs, não são satisfeitas. Entende-se, aqui, que tal implementação não pretende ter, ainda, um caráter prático, mas

sim, identificar os problemas existentes e servir como uma primeira aproximação na busca por melhores soluções.

Os capítulos subsequentes deste texto estão distribuídos da seguinte forma:

- ▶ Capítulo 2: são apresentados os módulos lógicos que compõem um sistema de visão e para cada um são apresentadas alternativas de abordagem.
- ▶ Capítulo 3: baseado nas técnicas expostas no capítulo 2, é discutida a concepção e construção de um sistema que execute as tarefas acima propostas.
- ▶ Capítulo 4: são apresentados alguns resultados experimentais.
- ▶ Capítulo 5: o sistema e os resultados obtidos são analisados.

## CAPÍTULO 2

# TÉCNICAS PARA CONSTRUÇÃO DE UM SISTEMA DE VISÃO COMPUTACIONAL

## 2.0 GENERALIDADES

Este capítulo tem como objetivo apresentar os módulos lógicos que compõem um sistema de visão computacional e de que forma esses módulos se inter-relacionam. Para cada módulo apresentado, são discutidas alternativas de abordagens, enfatizando aquelas que se referem ao presente sistema.

Assume-se que as imagens já passaram por uma fase de pré-processamento que, seguindo-se à sua aquisição e digitalização, procura diminuir ruídos e possíveis distorções causadas pela iluminação. Assim, o sistema de visão computacional é considerado, a partir do ponto em que este inicia a busca por elementos significativos para a interpretação das cenas propostas.

## 2.1 - SEGMENTAÇÃO

Segmentar uma imagem, aqui, significa dividi-la em suas partes constituintes com o objetivo de extrair elementos que possuam algum significado para o observador.

Em sistemas de análise automática de imagens, o processo de segmentação desempenha um papel crucial, pois ele é o primeiro passo a ser executado e os elementos por ele extraídos são utilizados em processos subsequentes como descrição e reconhecimento. Se erros acontecem, estes podem se propagar e levar o sistema a uma decisão errônea.

Ao longo dos anos, muitas técnicas para segmentar imagens têm sido propostas, podendo ser classificadas em três categorias [FU-81] :

- 1) Utilização de limiar,
- 2) Detecção de bordas e
- 3) Extração de regiões

A maioria delas possui uma natureza "ad hoc", não existindo nenhum algoritmo geral e que possa ser adotado como padrão [FU-81 e ROSENFELD-82]. Além disso, muito pouco é conhecido sobre como medir-se o erro em um algoritmo de segmentação [YASNOFF-77], o que torna difícil estabelecer um critério de

comparação entre algoritmos.

Isto tudo faz com que, quando da construção de um sistema de análise automática de imagens, a escolha de um ou outro algoritmo de segmentação fique a critério do projetista, de acordo com as necessidades de seu sistema.

## 2.11 - SEGMENTAÇÃO POR LIMIAR

Neste método de segmentação, os elementos procurados são regiões da imagem com um certo grau de homogeneidade, em geral, correspondentes a objetos presentes na cena. Ele baseia-se no prévio conhecimento de que diferentes elementos são caracterizados por níveis de cinza distintos e tendem, assim, a formar histogramas de distribuição de níveis de cinza onde cada elemento é caracterizado por um pico, separado dos demais por um vale. Por exemplo, a figura 2.1 mostra um histograma típico de uma imagem onde as regiões correspondentes a objetos são caracterizadas por um pico ( $m_1$  ou  $m_2$ ) e as correspondentes ao fundo da imagem pelo outro ( $m_2$  ou  $m_1$ ). A segmentação desta imagem consistiria em considerar os pixels, cujos valores são inferiores a um limiar  $T$  como pertencentes ao fundo da imagem (ou a objetos presentes na cena) e os pixels, cujos valores excedem  $T$  como pertencentes aos objetos presentes na cena (ou ao fundo da imagem).

No pixels

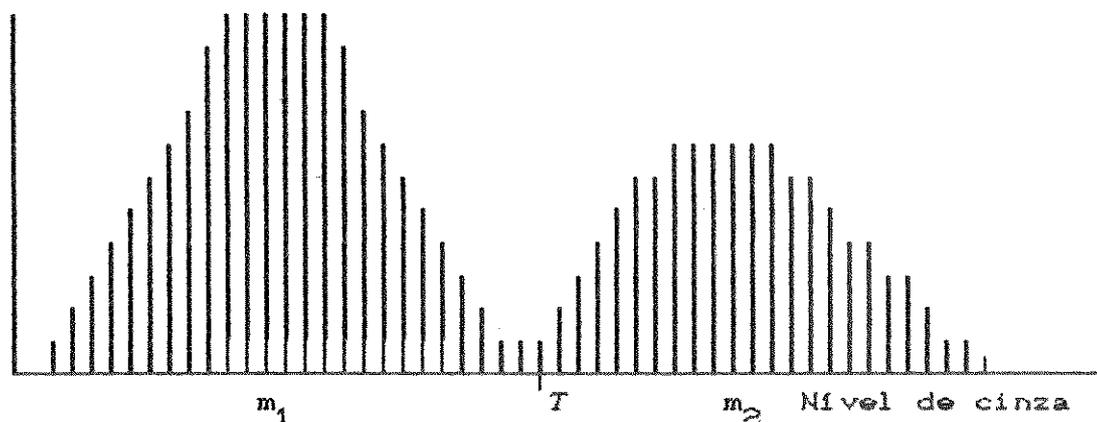


Figura 2.1 - Exemplo de histograma.

No caso mais geral (multimodal), a técnica do limiar pode ser, matematicamente, descrita da seguinte forma [FU-81, WESZKA-78] :

$$s(x, y) = k \text{ se } T_{k-1} \leq f(x, y) < T_k$$

$$p/k = 1, 2, 3, \dots$$

onde  $(x, y)$  são as coordenadas do pixel,  $f(x, y)$  e  $s(x, y)$  são os níveis de cinza da imagem e a função característica da imagem segmentada, respectivamente;  $T_0, \dots, T_m$  são os valores dos limiares e  $m$  é o número de rótulos, correspondentes a picos distintos no histograma de níveis de cinza, atribuídos à imagem segmentada. Um operador de limiar  $T$  pode ser visto como um teste envolvendo uma função  $T$  da forma:

$$T(x, y, N(x, y), f(x, y))$$

onde  $N(x, y)$  denota alguma propriedade local do ponto  $(x, y)$ . Quando  $T$  depende apenas de  $f(x, y)$ , o limiar é chamado global. Se  $T$  depende de ambos,  $f(x, y)$  e  $N(x, y)$ , ele é dito ser local e se ele depende de  $f(x, y)$ ,  $N(x, y)$  e das coordenadas espaciais  $x$  e  $y$ , ele é dito ser dinâmico [FU-81, WESZKA-78 e GONZALEZ-87].

No caso restrito em que o histograma é caracterizado apenas por dois picos (bi-modal), o método é descrito da seguinte forma:

$$s(x, y) = \begin{cases} 1 & \text{se } f(x, y) \geq T \\ e \\ 0 & \text{se } f(x, y) < T \end{cases}$$

Ressalta-se que, neste caso, o método apresenta um desempenho muito superior e sua aplicação é muito mais confiável.

## 2.1.1 - ESCOLHA DE LIMIARES

Um fator decisivo para o bom desempenho do método é a escolha do(s) limiar(es). Assim, esta escolha constitui o principal problema a ser resolvido quando de sua utilização.

Quando o histograma dos níveis de cinza é fortemente bi-modal, com dois picos comparáveis em amplitude e separados por um vale marcadamente profundo, a escolha do limiar torna-se relativamente fácil. Tal fato pode ser observado no seguinte procedimento para determinação do limiar (global) [ROSENFELD-82]: sendo o histograma denotado por  $p(z)$ , encontra-se dois máximos locais  $p(z_1)$  e  $p(z_j)$  relativamente separados. Seja  $p(z_k)$  um ponto entre  $p(z_1)$  e  $p(z_j)$  tal que  $p(z)$  assumo seu valor mínimo. Então, se  $p(z_k)/[\min(p(z_1), p(z_j))]$  é suficientemente pequeno,  $p(z_k)$  é muito menor que ambos,  $p(z_1)$  e  $p(z_j)$ , caracterizando um vale significativo entre os picos e  $z_k$  pode ser tomado como um limiar útil na segmentação da imagem.

Quando o histograma não se enquadra nas exigências do procedimento de segmentação acima, então, outras considerações devem ser observadas para viabilizar a aplicação da técnica. Algumas possibilidades são apresentadas abaixo:

- a) Se alguma informação extra a respeito da distribuição dos níveis de cinza é conhecida, esta pode ser utilizada no processo de escolha do limiar. Por exemplo, se a percentagem da área ocupada por objetos é conhecida e corresponde aos níveis de maior (ou menor) amplitude, pode-se escolher o limiar de forma a separar tal percentagem de níveis de cinza.
- b) Se sub-populações de níveis de cinza correspondentes a regiões diferentes se sobrepõem, dificultando a determinação de um limiar capaz de separá-los, pode-se fazer uso de um algoritmo de "suavização" (vide [ROSENFELD-82 e GONZALEZ-86]). Apesar de tais algoritmos tenderem a "borrar" as bordas das regiões, eles modificam a imagem de tal forma que o histograma dos níveis de cinza correspondente possui características mais favoráveis ao algoritmo de segmentação acima colocado.
- c) Se não é possível obter-se um limiar global que seja efetivo sobre toda a imagem, pode-se tentar a utilização de limiares locais. Por exemplo, divide-se a imagem original em blocos e determina-se o histograma de níveis de cinza. Espera-se, com isto, obter-se histogramas adequados a um processo de segmentação trivial. Por fim, procede-se a interpretação global das sub-imagens segmentadas.
- d) Wezka [WEZSKA-74] propôs submeter a imagem a um operador de detecção de elementos de borda (vide seção 2.1.2), mais especificamente o operador laplaciano e, então, utilizar apenas os pontos que apresentam uma alta resposta ao operador na construção do histograma. Como apenas os pontos próximos às bordas dos objetos apresentam alta resposta ao operador laplaciano, aproximadamente a mesma quantidade de pontos serão claros e escuros, produzindo um histograma simétrico com o vale entre os picos realçado, o que facilita a determinação de um limiar.

## 2.12 - SEGMENTAÇÃO POR DETECÇÃO DE BORDAS

Outra forma de se segmentar uma imagem é detectando-se as bordas das regiões nela presentes. Isto é possível se a cena representada pela imagem é composta por objetos homogêneos, cujos limites correspondem aos limites entre regiões de níveis de cinza constantes na imagem.

Basicamente, o processo de segmentação consiste de dois sub-processos. O primeiro deles faz uso de operadores cuja finalidade é detectar variações bruscas nos níveis de cinza da imagem, ou seja, elementos de borda, produzindo uma nova

representação da imagem. O segundo utiliza os elementos de borda detectados pelo processo anterior e junta-os, para obter bordas significativas.

A figura 2.2 mostra formas típicas que um elemento de borda assume.

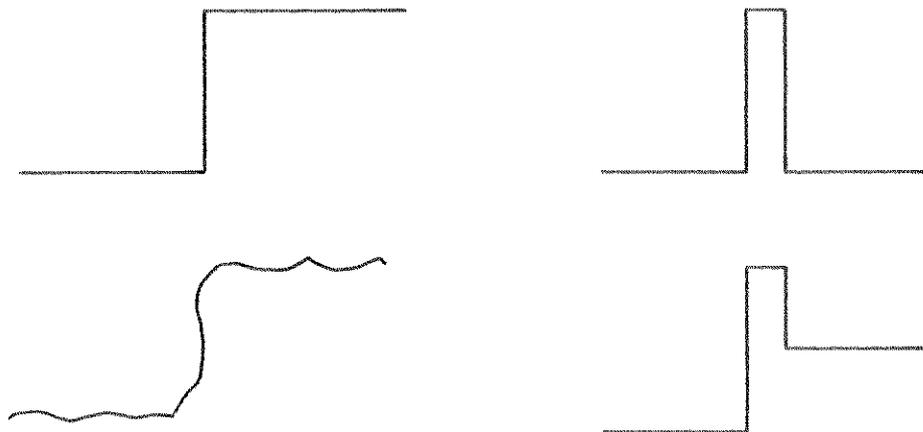


Figura 2.2 - Formas típicas de uma borda (modificado de [ROSENFELD-82]).

### 2.1.2.1 - DETECÇÃO DE ELEMENTOS DE BORDA

Um operador que detecta elementos de borda é um operador matemático (ou seu equivalente computacional) aplicado a uma pequena área da imagem com a finalidade de detectar variações bruscas na função imagem (níveis de cinza) na área considerada. O fato de diferentes operadores apresentarem melhor desempenho em diferentes aplicações propiciou a proposição de vários deles, que hoje abundam a literatura. Davis [DAVIS-75], Rosenfeld [ROSENFELD-82] e Gonzalez e Wintz [GONZALEZ-86] apresentam uma boa introdução a alguns destes operadores.

Operadores de detecção de elementos de borda podem ser classificados da seguinte forma [BALLARD-82]:

- a) que aproximam operadores derivativos,
- b) que utilizam múltiplas máscaras em diferentes orientações e
- c) que aproximam intensidades locais por modelos paramétricos de elementos de borda.

## A.1 - O GRADIENTE

Dado uma função  $f(x,y)$ , o gradiente de  $f$  nas coordenadas  $(x,y)$  é definido como o vetor:

$$\nabla f(x,y) = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

Este vetor apresenta importantes propriedades:

- i) Ele aponta na direção, dada por  $\text{tg}^{-1} \left[ \frac{\partial f / \partial y}{\partial f / \partial x} \right]$  de máxima taxa de crescimento e
- ii) Sua magnitude, dada por  $\sqrt{(\partial f / \partial x)^2 + (\partial f / \partial y)^2}$ , fornece o valor desta taxa em unidades de distância. Para fins de eficiência computacional, pode-se substituir a expressão acima por  $|\partial f / \partial x| + |\partial f / \partial y|$ .

Para funções discretas, que é o caso de imagens digitais, as derivadas são aproximadas por diferenças e a magnitude do vetor gradiente,  $G[\nabla f(x,y)]$ , toma a seguinte forma:

$$G[\nabla f(x,y)] = \sqrt{[f(x,y) - f(x+1,y)]^2 + [f(x,y) - f(x,y+1)]^2}$$

Note que as diferenças medidas por este operador não são simétricas com relação a  $(x,y)$ . Para evitar este fato, pode-se definir o operador de forma diferente, por exemplo, calculando-se a diferença entre pixels vizinhos na vertical e horizontal.

$$G[\nabla f(x,y)] = \sqrt{[f(x+1,y) - f(x-1,y)]^2 + [f(x,y+1) - f(x,y-1)]^2}$$

Muitas variantes para este operador têm sido propostas. O operador medindo as diferenças nas diagonais ( $45^\circ$  e  $135^\circ$ ), definido como

$$G[\nabla f(x,y)] = \sqrt{[f(x+1,y+1) - f(x,y)]^2 + [f(x,y+1) - f(x+1,y)]^2}$$

é conhecido como operador de Roberts.

Outra abordagem conhecida, baseada na utilização de um número de pixels maior na vizinhança do pixel  $(x,y)$  de uma forma ponderada, é o operador de Sobel (fig. 2.3), dado por

$$S_x = [f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)] - \\ [f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1)]$$

$$S_y = [f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)]$$

$$G[\nabla f(x, y)] = \sqrt{S_x^2 + S_y^2}$$

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figura 2.3 - Operador de Sobel.

Todos estes operadores são locais e podem ser, assim, aplicados em paralelo, o que é tido como uma vantagem. Entretanto, o fato de considerarem apenas informações locais em detrimento de informações globais a respeito das bordas, faz com que eles sejam igualmente sensíveis a um vestígio de borda ou a um ruído presente na imagem.

Uma vez escolhida a aproximação para o operador gradiente, para se obter uma imagem gradiente  $g(x)$  mais significativa, em geral, considera-se um limiar  $T$  e procede-se conforme uma das abordagens abaixo [GONZALEZ-86]:

- i) A abordagem mais simples consiste em fazer  $g(x, y)$  igual ao valor da magnitude do vetor gradiente de  $f(x, y)$ , ou seja,

$$g(x, y) = G[\nabla f(x, y)]$$

- ii) Outra forma é considerar apenas os valores cujo módulo do gradiente seja maior que um certo limiar  $T$ , mantendo os pontos restantes iguais à imagem original.

$$g(x, y) = \begin{cases} G[\nabla f(x, y)] & \text{se } G[\nabla f(x, y)] \geq T \\ f(x, y) & \text{de outra forma} \end{cases}$$

- iii) Pode-se, ainda, atribuir às bordas um valor fixo  $L_g$  com o objetivo de enfatizá-las.

$$g(x, y) = \begin{cases} L_g & \text{se } G[\nabla f(x, y)] \geq T \\ f(x, y) & \text{de outra forma} \end{cases}$$

- iv) Quando se quer observar a variação dos níveis de cinza nas bordas sem interferência do restante da imagem, pode-se considerar o valor da magnitude do gradiente e atribuir ao

restante um valor fixo  $L_b$ .

$$g(x, y) = \begin{cases} G[\nabla f(x, y)] & \text{se } G[\nabla f(x, y)] \geq T \\ L_b & \text{de outra forma} \end{cases}$$

v) Finalmente, quando apenas as localizações das bordas interessam, utiliza-se

$$g(x, y) = \begin{cases} L_g & \text{se } G[\nabla f(x, y)] \geq T \\ L_b & \text{de outra forma} \end{cases}$$

## A.2 - O LAPLACIANO

Dada uma função  $f(x, y)$ , o laplaciano é um operador derivativo de segunda ordem definido da seguinte forma:

$$L[f(x, y)] = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Para funções discretas, ele é aproximado por:

$$L[f(x, y)] = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

A figura 2.4 ilustra tal operador.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Figura 2.4 - Máscara correspondente ao operador laplaciano.

Ainda que o operador laplaciano responda bem a variações de intensidade, este operador é raramente utilizado (ele próprio) para detecção de elementos de borda. Isto se deve ao fato de ele, como um operador derivativo de segunda ordem, ser inadmissivelmente sensível a ruídos [GONZALEZ-86]. Assim, ele é relegado a uma função secundária, servindo como um detector que verifica se um dado pixel está do lado claro ou escuro de uma dada borda.

## B - MÚTIPLAS MÁSCARAS DE CONVOLUÇÃO

Definindo-se máscaras de convolução (fig. 2.5) tal que cada uma delas responda mais fortemente a elementos de borda em uma dada orientação, sua utilização consiste em submeter cada ponto da imagem a elas. Considera-se que um elemento de borda foi encontrado se a maior resposta às máscaras de convolução é superior a um dado limiar T, adotando-se como sua orientação a associada com a máscara de máxima resposta [ROSENFELD-82 e ROBINSON-77].

Direção do Gradiente	Máscara de Prewitt	Máscara de Kirsch	Máscara Nível 3	Máscara Nível 5
Norte	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
Noroeste	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$
Oeste	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$
Sudoeste	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$
Sul	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Sudeste	$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$
Leste	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Nordeste	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$

Figura 2.5 - Exemplos de máscaras de convolução (modificado de [ROBINSON-77]).

## C - O OPERADOR DE HUECKEL

A idéia por trás do operador de Hueckel é modelar um elemento de borda na imagem por meio de uma borda ideal que melhor lhe aproxime. Para isto, o operador é aplicado sobre um domínio circular  $\mathcal{D}$  (fig. 2.6) e considera-se uma borda ideal sobre este domínio, descrita como

$$\mathcal{F}(x, y, c, s, \rho, d, b) = \begin{cases} b & \text{se } c \cdot x + s \cdot y \leq \rho \\ b + d & \text{se } c \cdot x + s \cdot y > \rho \end{cases}$$

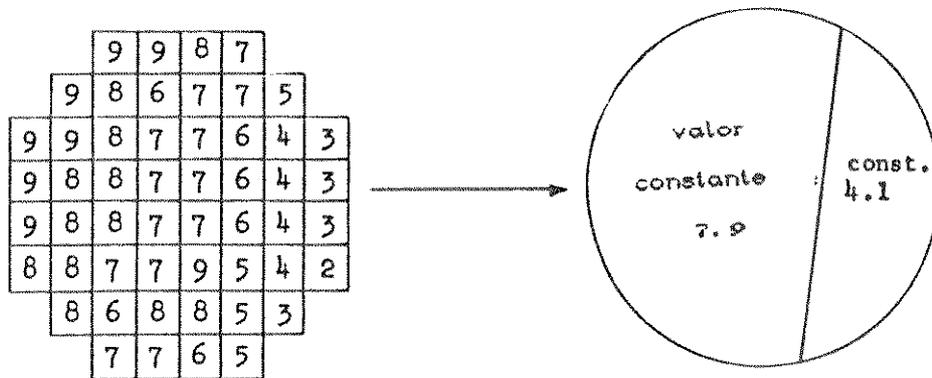


Figura 2.6 - Bordas no domínio da imagem e modelada (modificado de [HUECKEL-71]).

Se os elementos de borda da imagem, a cada ponto, são denotados por  $\varepsilon(x, y)$ , a meta do operador é selecionar os parâmetros  $(c, s, \rho, d, b)$  tal que

$$\Delta(c, s, \rho, d, b) = \int_{\mathcal{D}} [\varepsilon(x, y) - \mathcal{F}(x, y, c, s, \rho, d, b)]^2 dx dy$$

seja mínimo e inferior a um dado limiar.

Para simplificar os cálculos realizados pelo operador, Hueckel [HUECKEL-71] considerou os espectros de Fourier, em coordenadas polares, de  $\varepsilon$  e  $\mathcal{F}$ , cujos componentes são dados, respectivamente, por

$$\alpha_i = \int_{\mathcal{D}} \mathcal{X}_i(x, y) \cdot \varepsilon(x, y) dx dy \quad e$$

$$f_1(c, s, \rho, d, b) = \int_D \mathcal{X}_1(x, y) \cdot \mathcal{Y}(x, y, c, s, \rho, d, b) dx dy$$

e ponderando cada componente espectral por um fator  $w_1$ , o problema de minimização anterior toma a seguinte forma:

$$\Delta'(c, s, \rho, d, b) = \sum_{i=0}^{\infty} (w_i \cdot [\alpha_i - f_i(c, s, \rho, d, b)])^2$$

Hueckel [HUECKEL-71] apresentou a solução analítica para o operador da seguinte forma:

Definindo:	$w_0 = 1 = 1.000$ $w_1 = (4/3)^{1/2} = 1.155$ $w_2 = 1 = 1.000$ $w_3 = 1 = 1.000$	$w_4 = (4/3)^{1/2} = 1.155$ $w_5 = (4/3)^{1/2} = 1.144$ $w_6 = (4/5)^{1/2} = 0.895$ $w_7 = (4/5)^{1/2} = 0.895$ $w_i = 0$ para todo $i \geq 8$ .
------------	--	--

$$r^2 = x^2 + y^2, \quad Q(r) = \sqrt{1 - r^2}$$

$$H_0(x, y) = (5/6\pi) \cdot Q(r) \cdot (1 + 2r^2),$$

$$H_1(x, y) = (8/27\pi)^{1/2} \cdot Q(r) \cdot (5r^2 - 2),$$

$$H_2(x, y) = (3/\pi)^{1/2} \cdot Q(r) \cdot x \cdot (4r^2 - 1),$$

$$H_3(x, y) = (3/\pi)^{1/2} \cdot Q(r) \cdot y \cdot (4r^2 - 1),$$

$$H_4(x, y) = (3/\pi)^{1/2} \cdot Q(r) \cdot x \cdot (3 - 4r^2),$$

$$H_5(x, y) = (3/\pi)^{1/2} \cdot Q(r) \cdot y \cdot (3 - 4r^2),$$

$$H_6(x, y) = (32/3\pi)^{1/2} \cdot Q(r) \cdot (x^2 - y^2),$$

$$H_7(x, y) = (32/3\pi)^{1/2} \cdot Q(r) \cdot 2xy;$$

$$\mathcal{X}_0 = (24\pi/25)^{1/2} \cdot H_0,$$

$$\mathcal{X}_1 = (9/2)^{1/2} \cdot H_1,$$

$$\mathcal{X}_2 = H_2 + H_4,$$

$$\mathcal{X}_3 = H_3 + H_5,$$

$$\mathcal{X}_4 = (3/2)H_6,$$

$$\mathcal{X}_5 = (3/2)H_7,$$

$$\mathcal{X}_6 = (5/4)^{1/2}(H_2 - H_4),$$

$$\mathcal{X}_7 = (5/4)^{1/2}(H_3 - H_5).$$

$$\begin{aligned}
a_i &= \int_D H_i(x, y) \epsilon(x, y) dx dy & i = 0, \dots, 7. \\
e_0(c, s) &= a_2 c + a_3 s, \\
e_1(c, s) &= a_4 c + a_5 s, \\
e_2(c, s) &= a_1 + a_6(c^2 - s^2) + a_7 2sc, \\
U(c, s) &= [e_0(c, s) / |e_0(c, s)|] \cdot [e_1^2(c, s) + e_2^2(c, s)]^{1/2}, \\
A(c, s) &= e_0(c, s) + U(c, s).
\end{aligned}$$

pode-se reduzir o problema de minimização de  $\Delta'$  ao problema de extremização (encontrar o máximo ou mínimo) de  $A(c, s)$  e neste ponto

$$\begin{aligned}
\rho &= e_2(c, s) / (\sqrt{2} [U(c, s) + e_1(c, s)]), \\
d &= 4A(c, s) / [\sqrt{3\pi} (1 - \rho^2)^2 (1 + 2\rho^2)] \quad e \\
b &= a_0 - d(4 + \rho[3 + \rho(2 + \rho)]) (1 - \rho^2)^2 / 8
\end{aligned}$$

que é a solução do operador.

Posteriormente Hueckel [HUECKEL-73] estendeu este operador para detectar linhas e bordas conjuntamente.

## 2.1.2.2 - DETERMINAÇÃO DE BORDAS

Os operadores de detecção de elementos de borda, acima discutidos, respondem a descontinuidades nos níveis de cinza. Se as únicas descontinuidades presentes na imagem são as bordas dos objetos, então, seria de se esperar que a simples aplicação dos operadores acima fosse suficiente para determinação destas bordas. Entretanto, isto é inviável quando os seguintes fatores são considerados:

- i) A presença de ruído, iluminação não uniforme e outros fatores que introduzem descontinuidades e deturpam a imagem.
- ii) Os operadores de detecção de elementos de borda respondem, também, a pixels na vizinhança da borda, não conseguindo caracterizar um único pixel como elemento de borda.

Sendo desejável que as bordas encontradas sejam contínuas e tenham a espessura de um único pixel, algoritmos de detecção de elementos de borda são seguidos por procedimentos visando a interligação dos elementos de borda de forma a produzir bordas significativas.

Abaixo, são apresentadas algumas técnicas de análise de elementos de borda para determinação de bordas.

- a) Análise local.
- b) Análise global via transformação de Hough.
- c) Análise global via métodos grafo-teóricos.

## A - ANÁLISE LOCAL

Uma forma muito simples de estabelecer as ligações entre bordas consiste em analisar as características dos pixels em uma pequena vizinhança do pixel  $(x,y)$  e ligar os pontos considerados similares, segundo um dado critério.

As duas propriedades utilizadas para determinar-se a similaridade ou não entre elementos de borda neste tipo de análise são [GONZALEZ-86]:

- i) a magnitude da resposta do operador que produziu o elemento de borda e
- ii) a orientação do elemento de borda.

Em geral, considera-se que dois elementos de borda são similares com relação a uma das propriedades se o módulo da diferença entre os valores assumidos por cada um para tal propriedade não excede um dado limiar. Assim, se  $G$  denota a magnitude do elemento de borda e  $\alpha$  sua orientação, dois elementos de borda localizados nos pontos  $(x,y)$  e  $(x',y')$  são similares com relação à magnitude se

$$| G[\nabla f(x,y)] - G[\nabla f(x',y')] | \leq T \quad e$$

similares com relação à orientação se

$$| \alpha(x,y) - \alpha(x',y') | \leq A$$

onde  $T$  e  $A$  são limiares. Para medir a similaridade entre dois elementos de borda pode-se considerar as propriedades acima em separado ou em conjunto.

## B - ANÁLISE GLOBAL VIA TRANSFORMAÇÃO DE HOUGH

A transformação de Hough é aplicável à detecção de bordas em situações onde, embora pouco se conheça sobre as localizações das bordas, suas formas podem ser descritas como uma curva paramétrica. Apresenta como vantagens a robustez a ruídos, tolerância no reconhecimento de formas ligeiramente distorcidas e o seu potencial de implementação em paralelo.

## B.1 - A TRANSFORMAÇÃO DE HOUGH

A idéia fundamental da transformação de Hough é, a partir de um padrão espacialmente disperso no espaço da imagem, produzir padrões espacialmente compactos no espaço de possíveis parâmetros, convertendo o difícil problema de detecção global (no espaço da imagem) em um problema de detecção de picos, de solução mais fácil (no espaço de parâmetros).

Duda e Hart [DUDA-72] fornecem uma boa introdução ao método, considerando o problema de detectar linhas retas em uma imagem. Para fins de ilustração considera-se o mesmo problema.

Uma reta pode ser descrita pela distância separando-a da origem e pelo ângulo entre a reta normal à ela e o eixo de coordenadas  $x$  (fig. 2.7-a), denotados por  $\rho$  e  $\theta$ , respectivamente. Sua equação correspondente é dada por

$$\rho = x \cdot \cos\theta + y \cdot \text{sen}\theta$$

e se  $\theta$  é restrito ao intervalo  $[0, \Pi]$ , então, esta forma paramétrica para uma reta é única, ou seja, a toda reta no espaço  $x$ - $y$  corresponde um único ponto no espaço  $\rho$ - $\theta$ .

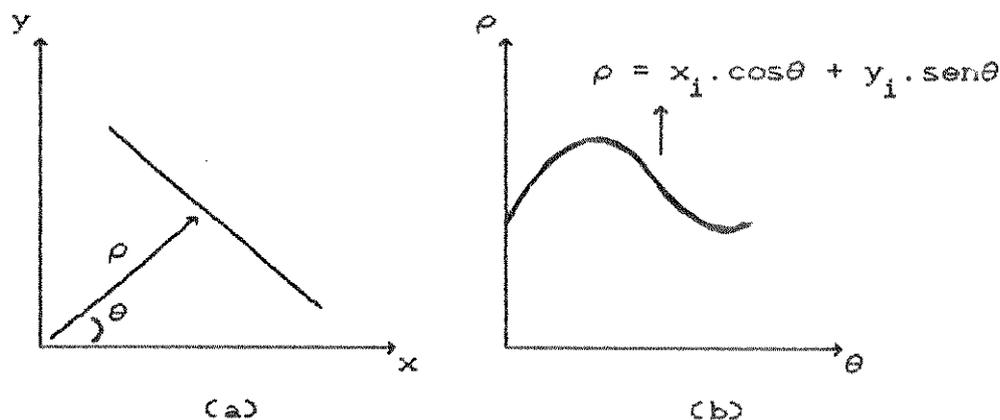


Figura 2.6 - Transformação de Hough. (a) Parâmetros. (b) Espaço de parâmetros.

Dado um conjunto de pontos  $\{(x_i, y_i)\}_{i=1}^n$ , com o objetivo de determinar-se o conjunto de retas que melhor se ajuste a eles, procede-se a transformação dos pontos no espaço da imagem para curvas senoidais no espaço dos parâmetros (fig. 2.6-b), definidas por

$$\rho = x_i \cdot \cos\theta + y_i \cdot \text{sen}\theta$$

A equação acima mostra que a transformação de Hough pode ser interpretada como um mapeamento de um para muitos (de um ponto no espaço da imagem para muitos possíveis valores dos parâmetros), ou seja, cada ponto no espaço da imagem dá origem, no espaço dos parâmetros, a todas as retas que passam por ele. Assim, no espaço dos parâmetros, curvas correspondendo a pontos colineares no espaço da imagem possuem um ponto de intersecção comum  $(\rho_0, \theta_0)$  (fig. 2.7).

Então, procedendo-se a transformação de Hough para todos os pontos no espaço da imagem, devido à propriedade acima, pontos colineares, no espaço da imagem, dão origem a pontos, no espaço dos parâmetros, onde diversas senoides se interceptam, definindo a equação da reta procurada.

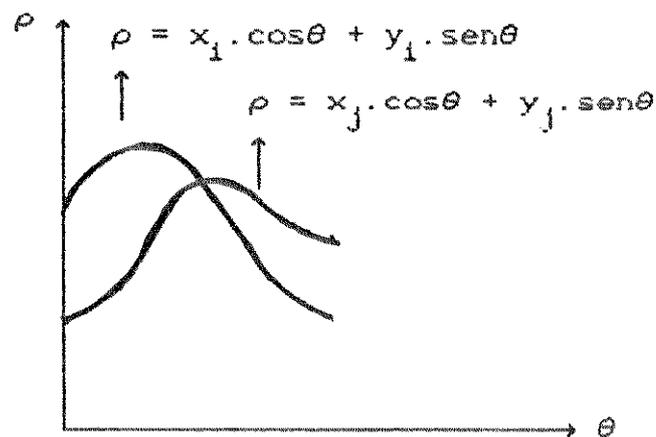


Figura 2.7 - Determinação da reta no espaço de parâmetros.

Embora, aqui, a transformação de Hough tenha sido apresentada num contexto de detecção de bordas descritas por equações paramétricas de retas, deve ser ressaltado que, ao longo dos anos, estudos têm sido desenvolvidos para estender seus conceitos e sua aplicabilidade. Kimme et. al. [KIMME-75] descreveram um procedimento para detectar circunferências e arcos aproximadamente circulares; Stockman e Agrawala [STOCKMAN-77] mostraram a correspondência entre a transformação de Hough e "template matching" (vide [ROSENFELD-82]) e Ballard [BALLARD-81], Davis [DAVIS-82] e Casasent e Krishnapuram [CASASENT-87] propuseram diferentes generalizações para a transformação de Hough, com o objetivo de detectar formas arbitrárias. Uma extensa revisão sobre o assunto pode ser encontrada em [ILLINGWORTH-88].

## C - ANÁLISE GLOBAL VIA MÉTODOS GRAFO-TEÓRICOS

Aqui, o conjunto de elementos de borda encontrado é interpretado como um grafo, com cada elemento de borda correspondendo a um nó. Se a magnitude de cada elemento de borda é denotado por  $s(x)$  e sua orientação por  $\phi(x)$ , dois nós  $x_i$  e  $x_j$  são ligados por um arco se os elementos de bordas são tais que:

- i) Suas orientações  $\phi(x_i)$  e  $\phi(x_j)$  estão aproximadamente alinhadas e
- ii) Suas magnitudes  $s(x_i)$  e  $s(x_j)$  excedam a um dado limiar T.

Martelli [MARTELLI-72, MARTELLI-76] foi o primeiro a propôr a utilização de técnicas de busca heurísticas para a determinação do percurso de mínimo custo correspondentes a uma borda.

Supondo que:

- i) O percurso seguiria contornos que são direcionados de  $x_A$  para  $x_B$ .
- ii) Existe um método disponível para gerar os nós sucessores de um dado nó e
- iii) Existe uma função de avaliação  $f(x_j) = s(x_i) + h(x_j)$ , que é uma estimativa para o percurso de menor custo entre  $x_A$  e  $x_B$  passando por  $x_j$ .

o algoritmo de busca é colocado da seguinte forma [BALLARD-82]:

- 1) Gere os nós sucessores do nó inicial (Coloque-os em uma lista chamada OPEN com ponteiros de retorno para o nó inicial).
- 2) Remova o nó  $x_i$  de mínimo  $f$  de OPEN. Se  $x_i = x_B$ , então pare. O caminho entre o nó  $x_i$  e o nó inicial é o caminho ótimo. Se OPEN está vazio acuse falha.
- 3) Em caso contrário, gere nós sucessores de  $x_i$ , colocando-os em OPEN com ponteiros para  $x_i$ . vá para o passo 2.

Se  $h^*(x_j)$  o custo ótimo real, se  $h(x_j) > h^*(x_j)$  o algoritmo é mais eficiente, mas pode não encontrar o melhor percurso. Já se  $h(x_j) \leq h^*(x_j)$  a busca resultará, sempre, no melhor percurso uma vez que a seguinte condição seja satisfeita:

Se para quaisquer dois nós  $x_i$  e  $x_j$ ,  $k(x_i, x_j)$  é o mínimo custo tomado de  $x_i$  para  $x_j$  (se possível), então

$$k(x_i, x_j) \geq h^-(x_i) - h^-(x_j)$$

Ballard e Brown [BALLARD-82] sugerem algumas funções de avaliação independentes da aplicação. Entretanto, ressalta-se que boas funções de avaliação podem ter componentes tanto dependentes quanto não dependentes da aplicação.

## 2.13 - SEGMENTAÇÃO POR EXTRAÇÃO DE REGIÕES

Nesta abordagem, o problema é o dual ao de encontrar bordas entre regiões de níveis de cinza diferentes, ou seja, o que se deseja é encontrar as próprias regiões presentes na imagem.

O procedimento é baseado na homogeneidade entre pixels pertencentes a uma mesma região e, com base em alguma propriedade que a meça, identificá-los e agrupá-los.

Tal problema pode ser definido da seguinte forma [PLAVIDIS-77, ZUCKER-76 e HOROWITZ-76]: Seja  $P(x)$  um predicado que assume os valores "verdadeiro" e "falso", dependendo de o conjunto de pixels  $x$  ser considerado homogêneo ou não, segundo algum critério. Seja a região compreendendo toda a imagem denotada por  $R$ , então o processo de segmentação pode ser visto como um processo que particiona  $R$  em  $n$  sub-regiões,  $R_1, \dots, R_n$  tal que

- 1)  $\bigcup_{i=1}^n R_i = R$
- 2)  $R_i$  é uma região conectada,  $i = 1, \dots, n$
- 3)  $R_i \cap R_j = \emptyset$  para todo  $i$  e  $j$ ,  $i \neq j$
- 4)  $P(R_i) = \text{verdadeiro}$  para  $i = 1, \dots, n$
- 5)  $P(R_i \cup R_j) = \text{falso}$  para todo  $i \neq j$

onde  $\emptyset$  é o conjunto vazio.

Plavidis [PLAVIDIS-77] classifica as estratégias de segmentação em:

- a) Partição de regiões
- b) Agrupamento de regiões
- c) Partição e agrupamento de regiões

## A - PARTIÇÃO DE REGIÕES

Este método tende a requerer um tempo de processamento maior, quando comparado com os outros dois. Entretanto, apresenta a vantagem de permitir estimativas mais confiáveis dos predicados definidos sobre uma grande área [PLAVIDIS-77].

Considera-se, inicialmente, toda a imagem como uma única região que é, então, submetida a sucessivas partições até que as condições acima sejam satisfeitas. A forma como as partições são feitas é pré-determinada, em geral, consistindo de partições em quadrantes e da utilização de uma estrutura de representação conhecida como quadtree (vide seção 2.3).

Devido a esta forma fixa das partições, ao final deste processo, pode-se ter algumas regiões adjacentes com características idênticas. Para contornar tal situação, segue-se um processo de agrupamento de tais regiões.

## B - AGRUPAMENTO DE REGIÕES

Neste caso, o processo é iniciado ou a nível de pixel ou considerando uma segmentação grosseira (com regiões altamente homogêneas) e consiste em agrupar estas sub-regiões em regiões maiores. Para cada par de regiões adjacentes mede-se a similaridade entre ambas, agrupando as altamente similares. Repete-se tal processo até que mais nenhum agrupamento seja admissível.

Outra forma de colocação do problema consiste em considerar pontos iniciais (sementes) a partir dos quais, por agregação de outros pixels, as regiões são geradas.

Esta abordagem, embora apresente algumas dificuldades como a escolha das regiões iniciais (ou sementes) e a seleção de uma propriedade adequada para medir a similaridade entre as regiões [GONZALEZ-86], têm sido amplamente mais utilizada que a partição de regiões [PLAVIDIS-77].

## C - PARTIÇÃO E AGRUPAMENTO DE REGIÕES

Este esquema foi proposto por Horowitz e Plavidis [HOROWITZ-76] e consiste em uma combinação dos dois métodos anteriores, com o intuito de beneficiar-se dos pontos positivos de ambos.

O processo inicia-se com uma segmentação inicial que é, então, submetida a sucessivas partições ou agrupamentos,

até que as condições acima colocadas (propriedades formais da segmentação) sejam satisfeitas. Em geral, a implementação deste algoritmo, da mesma forma que o caso de partição de regiões, faz uso da quadtree.

Plavidis [PLAVIDIS-77] descreve um algoritmo geral para tal abordagem da seguinte forma:

1. Faça  $S = \text{verdadeiro}$
2. Enquanto ( $S = \text{verdadeiro}$ ) execute o bloco 2.2  
bloco 2.2
  1. Faça  $S = \text{falso}$
  2. Para cada nó  $j$  da quadtree avalie  $P(x_j)$ .  
Se falso faça  $S = \text{verdadeiro}$ , particione  $x_j$  e atualize a quadtreefim do bloco 2.2
3. Faça  $M = \text{verdadeiro}$
4. Enquanto ( $M = \text{verdadeiro}$ ) execute o bloco 4.4  
bloco 4.4
  1. Faça  $M = \text{falso}$
  2. Para cada nó  $j$  da quadtree e para cada um de seus nós adjacentes, digamos  $k$ , execute o bloco 4.5.  
bloco 4.5
    1. Se  $P(x_j \cup x_k)$  é verdadeiro, agrupe as duas regiões, faça  $M = \text{verdadeiro}$  e atualize a quadtree.fim do bloco 4.5fim do bloco 4.4.

## 2.2 - REPRESENTAÇÃO E DESCRIÇÃO

Após segmentar uma imagem, utilizando métodos tais como os discutidos na seção anterior, é, em geral, útil "representar" e "descrever" o conjunto de pixels resultante em uma forma mais interessante (mais compacta e de mais fácil manipulação) para posterior processamento.

Muitos esquemas de representação e descritores para as regiões são avaliáveis na literatura, sendo que muitos deles podem ser encontrados em [ROSENFELD-82 e BALLARD-82].

### 2.2.1 - ESQUEMAS DE REPRESENTAÇÃO

Similarmente aos processos de segmentação, que produzem regiões segmentadas ou considerando as similaridades (limiar e extração de regiões) ou as dissimilaridades (detecção de bordas) entre pixels que compõem a imagem, têm-se, basicamente, duas opções sobre a qual basear a representação de uma região:

- a) em suas características externas (bordas) ou
- b) em suas características internas (pixels compreendendo a região)

### A.1 - CÓDIGO DA CADEIA

O código da cadeia é constituído por uma sequência de segmentos de reta de comprimento e orientação especificados e, em geral, é baseado em um esquema de conectividade (conectividade 4 ou 8) onde a orientação de cada segmento é codificada utilizando-se um esquema de numeração (fig. 2.8).



Figura 2.8 - Representação pelo código da cadeia.

Pode-se gerar o código, anotando-se o ponto inicial e, seguindo a borda no sentido horário (ou anti-horário). A sequência de números correspondentes à direção de cada segmento conectando cada par de pixels constitui o código procurado.

Este tipo de representação resulta em um código muito sensível a pequenos distúrbios devido a ruídos e segmentação imperfeita. Uma alternativa para tal problema consiste em proceder uma re-amostragem da borda diminuindo-se a resolução da imagem [GONZALEZ-87].

O código pode, ainda, ser feito independente do ponto inicial da cadeia para contornos fechados. Isto é obtido escolhendo-se a sequência resultante tal que esta resulte no número inteiro de mínima magnitude possível.

Finalmente, para obter invariância com relação a rotação, pode-se calcular a "derivada" do código que consiste, simplesmente, em outra sequência de números indicando a orientação relativa dos segmentos do código original.

## A.2 - POLILINHAS (APROXIMAÇÃO POLIGONAL)

Uma polilinha é uma concatenação de segmentos de reta, aproximando uma dada curva com um nível de acuidade arbitrário e que, em geral, é representado por uma lista de pontos (se o primeiro ponto da lista coincide com o último, têm-se a representação de uma curva fechada).

De modo geral, algoritmos de aproximação poligonal envolvem considerações semelhantes às do processo de segmentação [BALLARD-82]. Assim, tais algoritmos podem ser classificados de acordo com a política que os norteiam em:

- a) Agrupamento: pontos ao longo do contorno são agrupados até que um dado critério de erro (ex: mínimo erro quadrático) seja satisfeito. Anota-se os parâmetros da linha, assume-se o erro igual a zero e repete-se o processo. Ao final, a intersecção entre linhas adjacentes determinam os vértices da poligonal.
- b) Explicitação: o contorno é sucessivamente subdividido em duas partes até que um dado critério seja satisfeito (ex: que a maior perpendicular entre o contorno e as linhas da poligonal que o aproximam não excedam um dado limiar).
- c) Explicitação e Agrupamento: Como no caso da segmentação, trata-se de uma combinação entre as duas primeiras, com o objetivo de buscar um melhor desempenho.

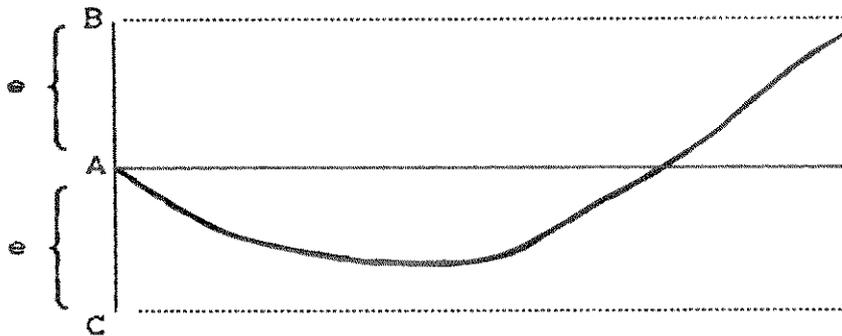


Figura 2.9 - Ilustração do algoritmo de aproximação poligonal (modificado de [PLAVIDIS-77]).

A título de ilustração, apresenta-se a seguir, um algoritmo de aproximação poligonal utilizando uma política de agrupamento [PLAVIDIS-77]. Dado um máximo erro admissível  $e$ , um ponto A por onde a linha aproximando a curva passa, define-se dois pontos B e C a uma distância  $e$  de A (fig. 2.9). Então, o algoritmo tenta encontrar o maior segmento tal que a curva não ultrapasse os limites determinados pelas paralelas passando por B e C.

1. Faça  $i_s = 1$ ,  $j = 0$ .
2. Enquanto  $i_s$  é menor que  $N$ , execute o bloco 21
  - Início do Bloco 21
    1. Faça  $B = A + e$  e  $C = A - e$
    2. Execute o bloco 31
      - Início do Bloco 31
        1. Encontre as linhas  $L_A$  e  $L_B$  juntando  $B$  e  $C$  com  $f(i)$ .  
Se  $i = i_s + 1$ , defina as linhas  $L_{acima}$  e  $L_{abaixo}$  iguais a  $L_B$  e  $L_C$ , respectivamente.  
De outra forma execute o bloco 32.
        - Início do Bloco 32
          1. Se  $L_B$  está acima de  $L_{acima}$ , substitua  $L_{acima}$  por  $L_B$ .
          2. Se  $L_C$  está abaixo de  $L_{abaixo}$ , substitua  $L_{abaixo}$  por  $L_C$ .
          3. Se o ângulo entre  $L_{acima}$  e  $L_{abaixo}$  é positivo abandone os blocos 32 e 31.
        - fim do Bloco 32
      - fim do Bloco 31
    3. Incremente  $j$  e faça  $z(j) = i - 1$ . Faça  $u(j)$  o ponto médio entre  $L_{acima}$  e  $L_{abaixo}$ . Faça  $i_s = i$
  - fim do Bloco 21.

Maiores detalhes sobre algoritmos para os vários casos, podem ser encontrados em [PLAVIDIS-77].

## B.1 - ESQUELETO DE UMA REGIÃO

Conhecido como esqueletização, transformação de eixo médio e transformação de eixo simétrico, o afinamento é um processo o qual busca reduzir a largura de um objeto, assumido de forma alongada, a um único pixel, o esqueleto (fig. 2.10).

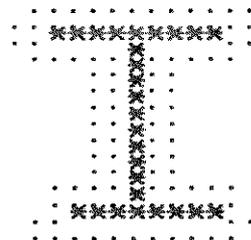


Figura 2.10 - Exemplo de objeto representado por seu esqueleto.

Este tipo de representação têm uma importância fundamental para a análise de imagens onde a forma pode ser completamente determinada por linhas. Por exemplo, no reconhecimento de caracteres, submeter a imagem a um algoritmo de afinamento, além de promover uma redução no volume de dados e aumentar a eficiência do sistema, reduz o efeito de tipos de fontes diferentes [SMITH-87]. Em alguns casos, no procedimento de segmentação, os elementos de borda são submetidos a um algoritmo de afinamento, como uma alternativa ao processo de detecção de bordas [SUZUKI-87 e FU-81].

Ainda que muitos algoritmos sejam avaliáveis na literatura [NACCACHE-84, GOVINDAN-87 e SUZUKI-87], a título de ilustração, apresenta-se, a seguir, o algoritmo proposto por Zhang e Suen [ZHANG-84].

Como os demais, o algoritmo de Zhang e Suen procura iterativamente apagar pontos de uma região, considerando as seguintes restrições:

- 1) Não remover pontos terminais,
- 2) Não quebrar a conectividade e
- 3) Não causar excessiva erosão da região.

Considerando uma imagem binária (pixels pertencentes ao objeto possuem valor 1 e ao fundo valor 0), o algoritmo, basicamente, consiste de duas sub-iterações aplicadas aos pontos de contorno de uma região (considera-se como pontos de contorno de uma região pixels com valor 1 e tendo no mínimo um 8-vizinho de valor 0).

P <sub>0</sub>	P <sub>2</sub>	P <sub>3</sub>
P <sub>8</sub>	P <sub>1</sub>	P <sub>4</sub>
P <sub>7</sub>	P <sub>6</sub>	P <sub>5</sub>

Figura 2.11 - Nomenclatura para vizinhança (modificado de [ZHANG-84]).

Assumindo-se a vizinhança de um ponto conforme mostrado pela figura 2.11, a primeira sub-iteração coloca uma marca para remoção num ponto de contorno p<sub>1</sub> se as seguintes condições são satisfeitas:

$$(a) \quad 2 \leq N(p_1) \leq 6$$

$$(b) \quad S(p_1) = 1$$

$$(c) \quad p_2 \cdot p_4 \cdot p_6 = 0$$

$$(d) \quad p_4 \cdot p_6 \cdot p_8 = 0$$

onde:  $N(p_1)$  é o número de vizinhos diferentes de zero de  $p_1$ , ou seja,  $N(p_1) = p_2 + p_3 + \dots + p_8 + p_9$  e

$S(p_1)$  é o número de transições 0-1 na sequência  $p_2, p_3, \dots, p_8, p_9$ .

Na segunda sub-iteração, as condições (a) e (b) são mantidas e as condições (c) e (d) são modificadas para

$$(c) \quad p_2 \cdot p_4 \cdot p_8 = 0$$

$$(d) \quad p_2 \cdot p_6 \cdot p_8 = 0$$

A figura 2.12 mostra o fluxograma correspondente ao algoritmo. A imagem original é armazenada na matriz  $IT$  e ao contador  $C$  é atribuído o valor 0. A matriz  $M$  é uma matriz auxiliar das mesmas dimensões de  $IT$ .

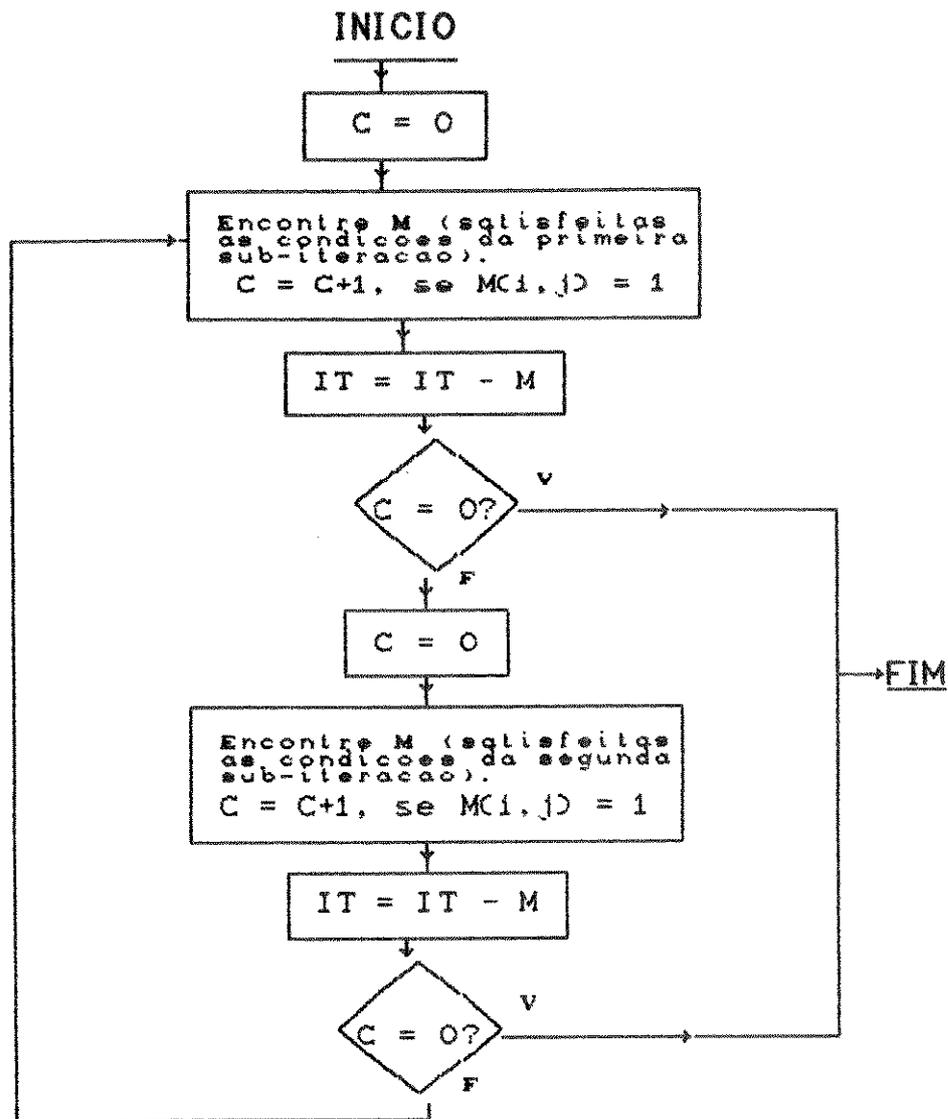


Figura 2.12 - Algoritmo de esqueletização (modificado de [ZHANG-84]).

## B.2 - QUADTREES

Quadtree é uma forma de representação de uma região baseada em árvores de grau 4 (cada nó da árvore possui 4 descendentes). O nó raiz representa toda a imagem. Se toda a imagem possui um único valor de pixel, o nó raiz estará rotulado com este valor. Em caso contrário, ele possuirá 4 descendentes, correspondendo aos quadrantes da imagem original, que serão rotulados de maneira análoga.

A partir de sua definição, pode-se, facilmente, obter um algoritmo para gerar a representação em quadtree de uma

imagem [BALLARD-82]:

```
Quadtree(integer PIRAMIDE; integer X, Y, NÍVEL);
início
  NovoNo(P);
  Tipo(P) := PIRAMIDE(INDX, Y, NÍVEL);
  se Tipo(P) = 0 ou 1 então Retorne(P)
  caso contrário
    início
      SW(P) := Quadtree(PIRAMIDE, 2*X, 2*Y, NÍVEL+1);
      SE(P) := Quadtree(PIRAMIDE, 2*X+1, 2*Y, NÍVEL+1);
      NW(P) := Quadtree(PIRAMIDE, 2*X, 2*Y+1, NÍVEL+1);
      NE(P) := Quadtree(PIRAMIDE, 2*X+1, 2*Y+1, NÍVEL+1);
      Retorne(P);
    fim;
  fim;
fim;
```

onde: SW, SE, NW e NE denotam os descendentes de um nó da quadtree, PIRAMIDE a estrutura de quadtree e IND é uma função de indexação que, dados o nível na quadtree e as coordenadas X e Y, extrai o valor apropriado para a região.

Muitas operações tais como o cálculo da área da região representada pela quadtree são simples e elegantes [BALLARD-82], mas tal representação apresenta sérias desvantagens tal como ser variante com o deslocamento [ROSENFELD-82].

## 2.2.2 - DESCRITORES

Uma vez as regiões componentes da imagem em análise tendo sido extraídas e adequadamente representadas, faz-se necessário caracterizar cada uma delas, onde entendemos como caracterização o processo de medir algumas das propriedades (geométricas, topológicas, estatísticas, etc) destas regiões.

Da mesma forma que a política utilizada no processo de segmentação influencia a escolha de um ou outro esquema de representação, podendo tornar alguns mais adequados que outros, o esquema de representação utilizado pode implicar em facilidades na obtenção de alguns descritores e dificuldades na obtenção de outros.

Embora seja possível proceder transformações entre as várias formas de representação [ROSENFELD-82], é conveniente que a escolha do esquema de representação leve em consideração quais descritores serão utilizados, bem como qualquer posterior processamento.

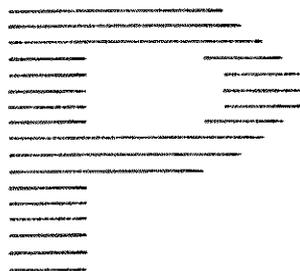
Descrevem-se, a seguir, alguns destes descritores.

## A - O NÚMERO DE EULER

O número de Euler é uma propriedade topológica que define um conjunto de objetos que são equivalentes, independentemente de deformações, ou seja, ele descreve a conectividade de uma região e não a sua forma [BALLARD-82].

Para um objeto bi-dimensional constituído de regiões conectadas, onde cada uma delas pode ter buracos, o número de Euler é definido como (fig. 2.13)

$$(\text{número de regiões conectadas}) - (\text{número de buracos})$$



$$E = 1 - 1 = 0$$

Figura 2.13 - Número de Euler de uma figura.

Muitos métodos simples para a sua obtenção são disponíveis e, geralmente, baseiam-se na contagem do número de padrões de variados tipos na imagem [ROSENFELD-82].

Considerando os padrões mostrados na tabela 2.1 e utilizando-se 4-conectividade para os componentes conectados  $S$  e 8-conectividade para  $\bar{S}$ , tem-se que o número de Euler é dado por

$$1/4(v - t + 2d)$$

e no caso inverso

$$1/4(v - t - 2d)$$

Padrões				No de padrões na figura	
1 0	0 1	0 0	ou	0 0	v
0 0	0 0	1 0	ou	0 1	
1 0	ou	0 1			d
0 1	ou	1 0			
0 1	1 0	1 1	ou	1 1	t
1 1	1 1	0 1	ou	1 0	

Tabela 2.1 - Cálculo do número de Euler [ROSENFELD-82].

## B - MOMENTOS

O momento  $(i, j)$  de uma imagem  $f$  é definida como

$$\sum_x \sum_y x^i \cdot y^j \cdot f(x, y)$$

Considerando  $f$  como uma função de distribuição de massa sobre o plano da imagem, pode-se atribuir interpretações físicas aos momentos. Assim  $m_{00}$  é a massa total da imagem,  $m_{02}$  e  $m_{20}$  são os momentos de inércia de  $f$ , com relação aos eixos  $x$  e  $y$ , respectivamente, etc.

Fatos como a razão entre dois momentos onde,  $i + j$  são idênticos, serem invariantes a escala fazem dos momentos descritores importantes na caracterização de regiões.

## C - ÂNGULOS

Ângulos são propriedades locais e inerentemente ligados à descrição de bordas. Eles podem assumir uma importância muito grande, pois são invariantes à translação, rotação e escalamento das regiões que descrevem.

Considerando, por exemplo, a borda de uma região descrita pelo código da cadeia, pode-se caracterizar um ângulo no ponto  $P$  como a média das inclinações de cada segmento do código, considerando uma sequência de  $K$  segmentos centrados no ponto  $P$  [ROSENFELD-82] (fig 2.14).

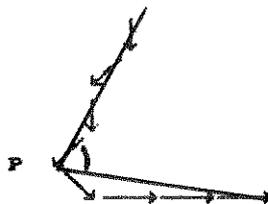


Figura 2.14 - Exemplo de cálculo de ângulo.

## D - CIRCULARIDADE

Circularidade ou "compactness" é um descritor

relacionado à forma da região e é definido como  $p^2/A$ , onde  $p$  é o perímetro da região e  $A$  é sua área.

Este descritor assume o valor  $4\pi$  para uma região com a forma de um disco e aumenta à medida que sua forma torna-se alongada ou irregular ou, ainda, se suas bordas tornam-se onduladas [ROSENFELD-82]. Ele é invariante à translação, rotação e escalamento.

## 2.3 - MODELOS E RECONHECIMENTO

Os métodos para representação e descrição discutidos na seção anterior referem-se à representação e descrição de uma imagem específica. Para se obter a interpretação dessa imagem, é necessário que o sistema possua uma descrição interna das classes de objetos conhecidos por ele. Esta descrição é referenciada como modelo [ROSENFELD-82] e está intimamente relacionada à estratégia utilizada pelo sistema no processo de reconhecimento dos objetos presentes na cena.

Aqui, consideram-se os seguintes modelos, bem como os procedimentos de reconhecimento a eles associados:

- a) Modelos de decisão teórica
- b) Modelos sintáticos
- c) Modelos grafo-relacionais

A opção por um ou outro modelo depende das exigências do sistema em curso e é parte fundamental de sua definição, pois além de, como citado, haver uma forte relação entre modelos e estratégias de reconhecimento, tal opção terá influência tanto na escolha do esquema de representação e descrição de uma imagem quanto no processo de segmentação utilizado.

### 2.3.1 - ABORDAGEM PELO MÉTODO DE DECISÃO TEÓRICA

Nesta abordagem, apenas as características globais dos objetos são consideradas, desprezando-se seus aspectos estruturais.

#### 2.3.1.1 - MODELOS

Considere um espaço  $n$ -dimensional onde a cada dimensão corresponde um descritor global de um objeto (por exemplo

circularidade, momentos de ordem n, etc). Neste espaço, um objeto é representado pelo vetor

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

onde cada elemento do vetor representa um descritor global.

Se o conjunto de descritores associados a este espaço é capaz de caracterizar as várias classes de objetos conhecidos pelo sistema, então, dois vetores correspondentes a classes de objetos distintos, neste espaço, estarão "distantes" um do outro, enquanto dois vetores correspondentes a objetos de uma mesma classe apresentarão uma tendência a estarem próximos (fig. 2.15).

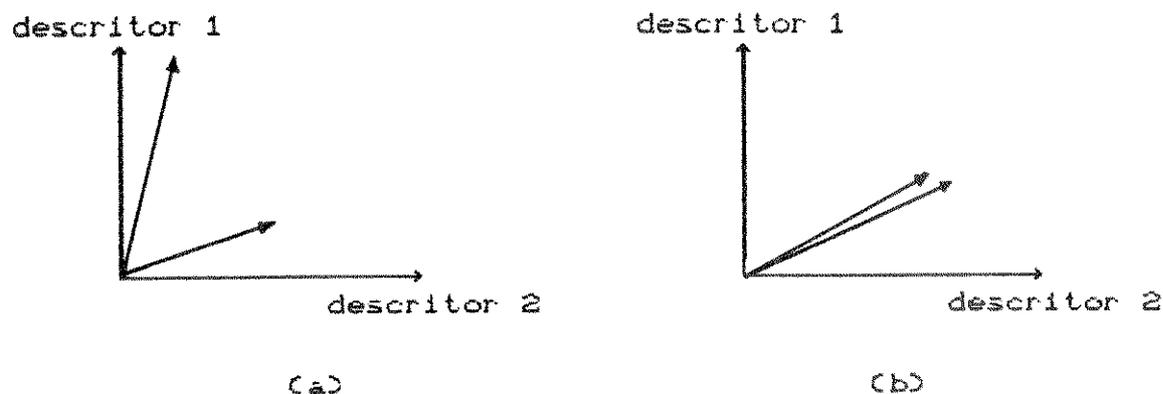


Figura 2.15 - Espaço de descritores. (a) Objetos pertencentes a classes diferentes. (b) Objetos pertencentes à mesma classe.

### 2.3.12 - CLASSIFICAÇÃO

Com base no acima exposto, o problema de classificação corresponde a assinalar a cada possível vetor no espaço de descritores uma classe apropriada. Isto corresponde a dividir o espaço de descritores em regiões mutuamente excludentes onde cada região corresponde a uma classe.

Matematicamente, tal problema de classificação pode ser formulado da seguinte forma. Sejam  $w_1, \dots, w_n$  designadores das n possíveis classes de objetos a serem reconhecidos e seja X um vetor de descritores (obtidos a partir da imagem de entrada). Então, para todo X pertencente a  $w_1$ , têm-se:

$$D_i(X) > D_j(X) \text{ para } i = 1, \dots, n; \quad i \neq j$$

onde  $D$  é chamado função discriminante. Se a função discriminante considera conhecidos a função densidade de probabilidade do vetor de descritores  $X$ ,  $P(X/w_i)$  bem como a probabilidade de ocorrência de  $w_i$ ,  $P(w_i)$ , o classificador é dito ser paramétrico e seu objetivo é, com base nas informações disponíveis, proceder a tarefa de classificação de tal maneira a minimizar a probabilidade de uma classificação errônea. Dentro desta categoria destaca-se o classificador bayessiano que baseia-se na implementação da regra de decisão de Bayes. De outra forma, se considera-se como conhecidos a forma da função discriminante, o método de decisão é dito ser não paramétrico. Maiores detalhes podem ser encontrados em [DUDA-73].

Abaixo, descrevem-se dois tipos de funções discriminantes não paramétricas, que definem esquemas de classificação distintos.

### 2.3.1.2.1 - FUNÇÃO DISCRIMINANTE LINEAR

Neste caso, a função discriminante é uma combinação linear dos componentes do vetor  $X$ ,  $x_1, x_2, \dots, x_n$ .

$$D_i(X) = \sum_{k=1}^n \alpha_{ik} \cdot x_k + \alpha_{i,n+1} \quad i = 1, \dots, m$$

onde  $\alpha$ 's são ditos serem os pesos da função e  $m$  é o número de classes possíveis.

A equação acima, é a equação de um hiperplano no espaço de descritores e o limite de decisão entre duas regiões associadas às classes  $w_i$  e  $w_j$  pode ser colocado da seguinte forma:

$$D_i(X) - D_j(X) = \sum_{k=1}^n \alpha'_k \cdot x_k + \alpha'_{n+1} = 0$$

onde  $\alpha'_k = \alpha_{ik} - \alpha_{jk}$  e  $\alpha'_{n+1} = \alpha_{in+1} - \alpha_{jn+1}$ .

Então, se  $D_i(X) - D_j(X) > 0$  para todo  $i \neq j$ , a entrada é classificada como pertencente à classe  $w_i$ .

### 2.3.1.2.2 - CLASSIFICADOR DE MÍNIMA DISTÂNCIA

Aquí, considera-se um conjunto de vetores de

referência no espaço de descritores  $R_1, \dots, R_m$  onde cada  $R_i$  está associado a uma classe  $w_i$ .

O classificador de mínima distância classifica uma dada entrada  $X$  como pertencente à classe  $w_i$  se:

$$|X - R_i| < |X - R_j| \quad \text{p/ todo } i, j \quad i \neq j$$

A distância  $|X - R_i|$ , em geral, é a distância euclideana, dada por

$$|X - R_i| = \sqrt{(X - R_i)^T (X - R_i)}$$

## 2.3.2 - ABORDAGEM SINTÁTICA

A abordagem sintática, diferentemente da abordagem pelo método de decisão teórica, que privilegia a análise de características globais do objeto, privilegia características locais, ou seja, procura descrever os objetos pelas suas partes constituintes.

A idéia que norteia esta abordagem é representar cada classe de objetos a serem reconhecidos por uma gramática. Dessa forma, os objetos de uma dada classe serão os elementos da linguagem gerada por tal gramática.

Para se manipular com modelos matemáticos de linguagens e métodos utilizados na sua geração e processamento, faz-se uso de elementos da teoria de linguagens formais. Para maiores detalhes, pode-se consultar o livro de Gonzalez e Thomason [GONZALEZ-78].

### 2.3.2.1 - MODELOS

Uma gramática é, formalmente, definida como a quádrupla  $G = (N, \Sigma, P, S)$  onde:

- $N$  é um conjunto finito de não-terminais ou variáveis;
- $\Sigma$  é um conjunto finito de terminais ou constantes;
- $P$  é um conjunto finito de regras de produção e
- $S$  em  $N$  é o símbolo inicial

tal que, os conjuntos  $N$  e  $\Sigma$  são disjuntos ( $N \cap \Sigma = \emptyset$ ) e o conjunto  $P$  de regras de produção seja constituído por regras da forma  $\alpha \rightarrow \beta$  onde  $\alpha$  é uma concatenação de elementos do alfabeto  $G (N \cup \Sigma)$  com pelo menos um símbolo terminal e  $\beta$  é uma concatenação de elementos do alfabeto.

Quando não existe qualquer restrição nas formas sob as quais as regras de produção são apresentadas, a gramática é dita ser irrestrita (tipo 0). Entretanto, se tais restrições são consideradas, uma conveniente hierarquia é obtida. Tal hierarquia é denominada hierarquia de Chomsky [GONZALEZ-78]:

- (1) Gramáticas sensitivas a contexto (tipo 1): são gramáticas onde a forma das regras de produção possui a forma  $\theta A \delta \rightarrow \theta \rho \delta$ .  $\theta$  e  $\rho$  são concatenações de elementos do alfabeto da gramática,  $\rho$  é uma concatenação destes mesmos elementos a exceção da cadeia vazia, denotada por  $\lambda$ , e  $A$  é um não-terminal.
- (2) Gramáticas livres de contexto (tipo 2): são gramáticas onde as regras de produção são da forma  $A \rightarrow \alpha$  onde  $A$  é um não-terminal e  $\alpha$  é uma concatenação de elementos do alfabeto da gramática, a exceção de  $\lambda$ .
- (3) Gramáticas regulares (tipo 3) : são gramáticas com regras de produção da forma  $A \rightarrow aB$  ou  $A \rightarrow a$  onde  $A$  e  $B$  são não-terminais e  $a$  é terminal.

Note-se que toda gramática regular é livre de contexto, toda livre de contexto é sensitiva a contexto e toda sensitiva a contexto é irrestrita. Por convenção, uma linguagem é classificada de acordo com o tipo da mais restritiva gramática que a gerou.

Outro parâmetro de classificação de uma gramática é quanto a ambiguidade. Uma gramática é dita ser ambígua se no mínimo uma sentença (padrão) na linguagem gerada por tal gramática leva a duas ou mais estruturas de derivação, ou seja, a partir do símbolo inicial  $S$ , pela aplicação de duas ou mais seqüências distintas de regras de produção, a sentença pode ser reconhecida.

Como um exemplo de representação, considere o problema de reconhecer cromossomos sub-medianos e telocêntricos [GONZALEZ-78]. Uma gramática  $G = (N, \Sigma, P, S)$  livre de contexto descreve as classes de objetos a serem reconhecidos. A figura 2.16 mostra os padrões primitivos associados a elementos terminais da gramática  $G$ .

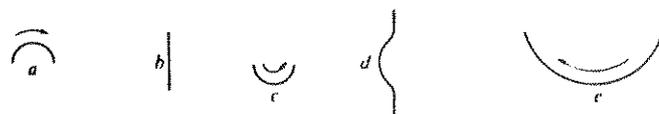


Figura 2.16 - Elementos primitivos da gramática  $G$  [GONZALEZ-78].

Os cromossomos são representados por uma cadeia composta por elementos terminais correspondentes a uma varredura no sentido horário através do seu contorno. O conjunto de

elementos terminais é dado por  $N = \{ S, S_1, S_2, A, B, C, D, E, F \}$  onde  $S$  é o símbolo inicial. Finalmente, o conjunto de regras de produção  $P$  é composto pelas seguintes regras:

$S \rightarrow S_1$	$B \rightarrow e$	$S \rightarrow S_2$	$C \rightarrow bC$
$S_1 \rightarrow AA$	$C \rightarrow Cb$	$S_2 \rightarrow BA$	$C \rightarrow b$
$A \rightarrow CA$	$C \rightarrow d$	$A \rightarrow AC$	$D \rightarrow bD$
$A \rightarrow DE$	$D \rightarrow Db$	$A \rightarrow FD$	$D \rightarrow a$
$B \rightarrow bB$	$E \rightarrow cD$	$B \rightarrow Bb$	$F \rightarrow Dc$

As duas regras de produção  $S \rightarrow S_1$  e  $S \rightarrow S_2$  indicam que a classe de cromossomos descrita por  $G$  consiste de duas sub-classes, ou seja, sub-mediana e telocêntrica.

### 2.3.2.2 - RECONHECIMENTO

Reconhecer se um dado padrão pertence ou não a uma classe  $C_1$  equivale ao problema de verificar se a cadeia de elementos terminais correspondentes ao padrão pertence ou não à linguagem especificada pela gramática associada à classe  $C_1$ . Tal processo é conhecido como análise sintática.

Da teoria das linguagens formais, têm-se que tal reconhecimento pode ser obtido por meio de modelos matemáticos de máquinas de computação, chamadas autômatos. A cada tipo de gramática, segundo a hierarquia de Chomsky, corresponde um autômato:

- Tipo 0: máquina de Turing
- Tipo 1: autômato "linear bounded"
- Tipo 2: autômato "push-down"
- Tipo 3: autômato finito

Entretanto, pode-se proceder a análise sintática de uma cadeia, considerando-se algoritmos que podem ser ou não implementados em forma de autômatos. Tal espécie de algoritmos são conhecidos como "parsing algorithms" e, basicamente, estão, implícita ou explicitamente, associados a duas possíveis abordagens:

- a) "Top-down": Dados a gramática  $G$  e a sentença  $X$  em  $L(G)$ , começando do símbolo inicial  $S$ , tenta-se derivar  $X$  pela sucessiva aplicação de regras de produção em  $P$ .
- b) "Bottom-up": Dados a gramática  $G$  e a sentença  $X$  em  $L(G)$ , começando de  $X$ , pela aplicação das regras de produção no sentido inverso, tenta-se reduzir  $X$  ao símbolo inicial  $S$ .

um exemplo de derivação de uma sentença associada a um cromossomo telocêntrico, começando do símbolo inicial S, pode ser dado por (fig. 2.17):

$S \Rightarrow S_2 \Rightarrow BA \Rightarrow BbA \Rightarrow eba \Rightarrow ebDE \Rightarrow ebaE \Rightarrow ebacD \Rightarrow ebacDb \Rightarrow ebacab$

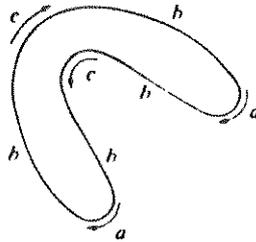


Figura 2.17 - Cromossomo telocêntrico (GONZALEZ-78).

### 2.3.3 - ABORDAGEM GRAFO-RELACIONAL

Nesta abordagem, como na abordagem sintática, os objetos também são descritos estruturalmente. Entretanto, diferentemente daquela, onde as classes são modeladas por gramáticas, nesta as classes são modeladas por grafos (grafos de referência).

A imagem de entrada origina uma representação semelhante ao grafo de referência. Assim, o problema de reconhecimento pode ser caracterizado como um problema de "casamento entre grafos". Tais problemas de representação e reconhecimento são ilustrados pela figura 2.18 e podem ser tratados pela utilização de elementos da teoria dos grafos.

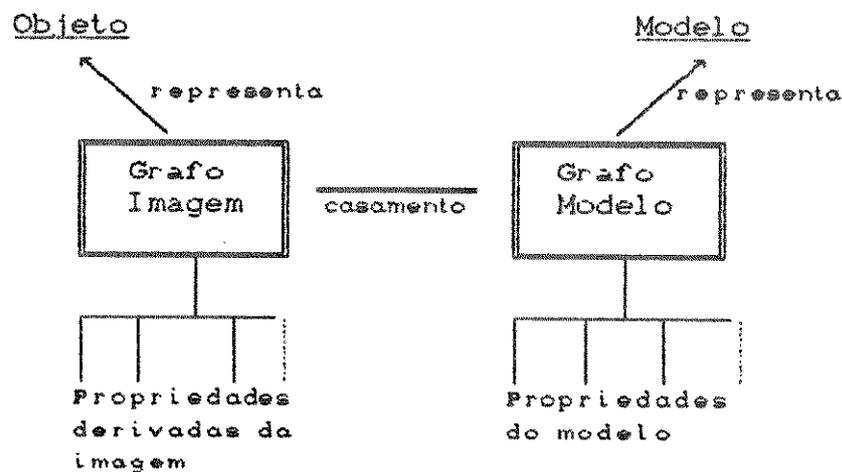


Figura 2.18 - Abordagem grafo-relacional (modificado de [BALLARD-82]).

### 2.3.3.1 - MODELOS

Um grafo finito  $G(V,E)$  consiste em um conjunto finito de nós  $V = \langle v_1, v_2, \dots \rangle$  e um conjunto de arcos  $E = \langle e_1, e_2, \dots \rangle$ , onde a cada arco corresponde um par de nós. Se o arco  $(v,w)$  corresponde a um arco  $e$ , então,  $e$  é dito ser incidente sobre os nós  $v$  e  $w$ . Se todo par de nós  $(v,w)$  associado a um arco  $e$  é um par ordenado, então, o grafo  $G$  que o contém é dito ser direcionado ou um dígrafo e o arco  $e$  é dito ser direcionado de  $v$  para  $w$ .

Várias formas de representação para grafos são possíveis. Sua escolha é importante na medida em que a facilidade de implementação, bem como a eficiência de um algoritmo de manipulação de grafo depende da própria escolha de representação.

Dentre as várias formas de representação, citam-se duas:

- a) **Matriz de adjacência:** A matriz de adjacência  $A$  é uma matriz  $n \times n$  onde  $n$  é o número de nós do grafo representado. Um elemento  $a_{ij}$  de  $A$  é igual a 1 se existe um arco do nó  $i$  para o nó  $j$ . De outra forma  $a_{ij}$  é zero.
- b) **Estrutura de adjacência:** O grafo é representado por uma lista contendo todos os seus nós. A cada nó  $\alpha$ , associa-se uma lista de nós sucessores  $\beta$ , indicando um arco de  $\alpha$  para  $\beta$ .

Para construir a representação grafo-relacional de um objeto, associa-se características locais do objeto como ângulos, linhas ou regiões aos nós e as relações entre tais características locais com os arcos do grafo. Cada nó deve ser descrito por descritores e propriedades referentes aos elementos aos quais está associado. Os arcos podem ser rotulados e associados a um tipo específico de relação, sendo direcionados quando a relação é assimétrica.

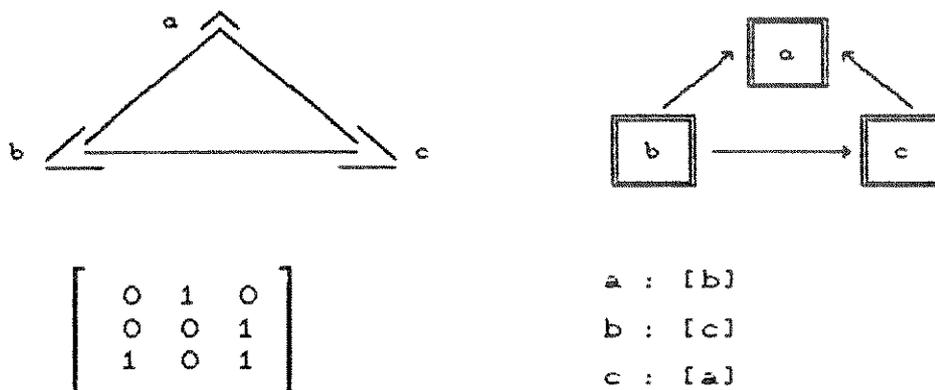


Figura 2.19 - Representação grafo-relacional.

Como um exemplo, a figura 2.10 mostra a imagem de um triângulo, um possível grafo de representação (nós correspondem aos ângulos e os arcos à relação de sucessão ao percorrer-se o seu contorno no sentido anti-horário) e sua representação, segundo os esquemas apresentados.

### 2.3.3.2 - RECONHECIMENTO

Como já colocado, adotando-se o modelo grafo-relacional, o problema de reconhecimento apresenta-se como um problema de casamento entre grafos. Ballard e Brown [BALLARD-82] dividem o problema nos seguintes casos:

- a) **Isomorfismo entre grafos:** Dados dois grafos  $(V_1, E_1)$  e  $(V_2, E_2)$ , encontrar um mapeamento bijetor  $f$  entre  $V_1$  e  $V_2$  tal que, para cada  $v_1$  e  $v_2 \in V_1$  e  $V_2$ ,  $f(v_1) = v_2$  e para cada nó de  $E_1$  conectado a qualquer par de nós  $v_1$  e  $v_1' \in V_1$ , existe um nó de  $E_2$  conectando  $f(v_1)$  e  $f(v_1')$ .
- b) **Isomorfismo entre grafo e sub-grafo:** Encontrar isomorfismos entre um grafo  $(V_1, E_1)$  e um sub-grafo de outro grafo  $(V_2, E_2)$ .
- c) **Isomorfismo entre sub-grafos:** Encontrar isomorfismos entre sub-grafos de um grafo  $(V_1, E_1)$  e sub-grafos de outro grafo  $(V_2, E_2)$ .
- d) **Casamento inexato:** Uma versão computacional diferente de uma versão "pura" de isomorfismo entre grafos.

Algoritmos para os vários casos são disponíveis na literatura. Berztiss [BERZTISS-73] apresenta uma solução para o isomorfismo entre dígrafos e Ullman [ULLMAN-76] explora duas soluções para o problema de isomorfismo entre grafo e sub-grafo.

#### A - "BACKTRACK SEARCH"

De um modo geral, algoritmos para determinação de isomorfismos entre grafos envolvem um processo de busca exaustiva. "Backtrack" é uma tal estratégia de busca, onde tenta-se, a partir de uma solução parcial, estendê-la sucessivamente, até que se encontre uma solução geral do problema ou esgotem-se as possibilidades de busca.

Abaixo, apresenta-se um algoritmo que, utilizando-se da estratégia de "Backtrack", encontra todos os isomorfismos entre dois grafos  $X = (V_x, E_x)$  e  $Y = (V_y, E_y)$ , onde  $V_x =$

$V_y = \langle 1, 2, \dots, m \rangle$ .

```
Isomorfismo(S, K);
início
  se  $S = V_y$  então notifique-isomorfismo(F)
  caso contrário
    para todo  $V \in (V_y - S)$  faça
      se casamento(K, V) então
        início
          F(K) := V;
          Isomorfismo(S U  $\langle V \rangle$ , K+1);
        fim;
    fim;
fim;
```

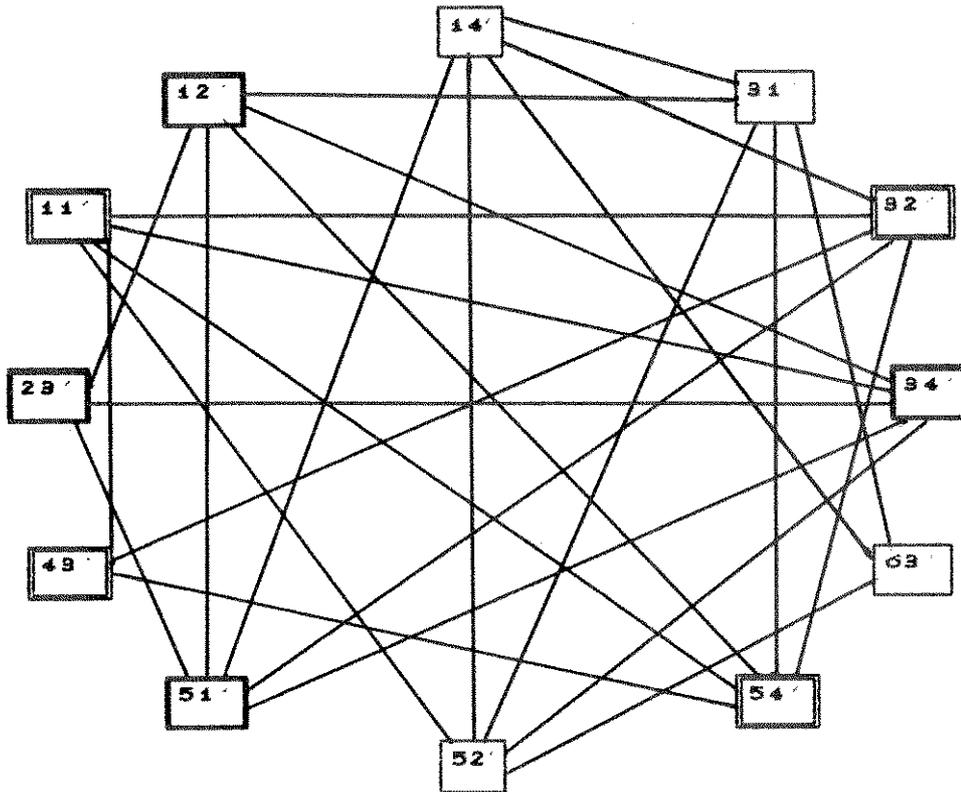
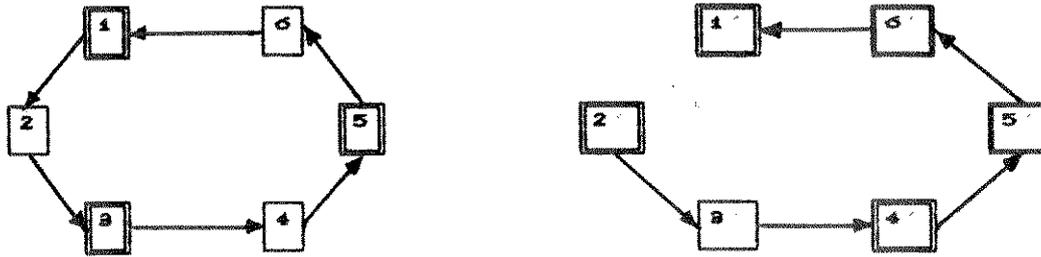
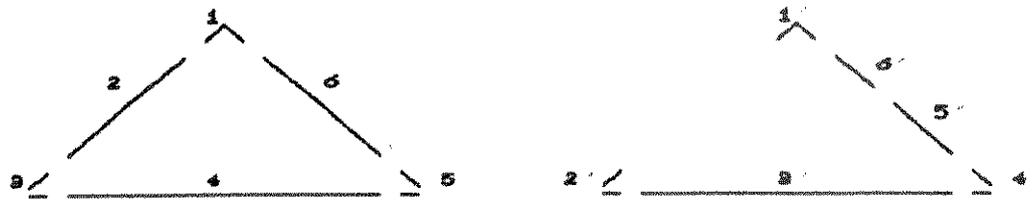
O algoritmo é invocado como Isomorfismo(0,1), notifique-isomorfismo imprime ou salva F (estrutura global com a solução corrente) e casamento(K,V) testa se  $V \in V_y$  corresponde a  $K \in V_x$  sob o isomorfismo parcial F.

## B - CLIQUES

Uma das formas de se abordar o problema de isomorfismo entre sub-grafos é por meio da utilização de grafos de associação. Um grafo de associação é uma estrutura relacional construída a partir de outras duas estruturas relacionais a serem "casadas" (fig 2.20). Para construí-la [AMBLER-75], considere duas estruturas relacionais  $(V_1, P, R)$  e  $(V_2, P, R)$  onde  $V$  denota um conjunto de elementos,  $P$  propriedades destes elementos e  $R$  relações entre eles. Para cada  $v_1 \in V_1$  e  $v_2 \in V_2$  constrói-se um nó de  $G$  rotulado  $(v_1, v_2)$  se  $\forall p \in P p(v_1) \leftrightarrow p(v_2)$ . Para dois nós rotulados  $(v_1, v_2)$  e  $(v_1', v_2')$  de  $G$ , existirá um arco entre eles se  $\forall r \in R r(v_1, v_1') \leftrightarrow r(v_2, v_2')$ . Tal relação reflete uma relação de compatibilidade entre as associações correspondentes aos nós.

Então, para estabelecer o casamento (isomorfismo) entre dois sub-grafos, procura-se sub-grafos em  $G$  totalmente conectados. Tal tipo de sub-grafo é conhecido como clique e refere-se ao maior conjunto de associações totalmente compatíveis entre si (fig 2.20).

A importância do conceito de cliques e grafos de associação é permitirem trazer determinados tipos de problemas de um domínio um tanto quanto "ad hoc" para um domínio clássico (teoria dos grafos). Por exemplo, o problema de determinar o melhor casamento entre duas estruturas relacionais pode ser tratado como o problema de determinar o clique máximo do grafo de associação das duas estruturas a serem "casadas".



Prop. ângulo     
  Prop. segmento     
  $\longrightarrow$  Rel. próximo

Figura 2.20 - Exemplo de grafo de associação e maximal clique (modificado de [BALLARD-82]).

Algoritmos para a determinação de cliques podem ser encontrados em [BRON-73, AMBLER-75 e REINGOLD-77]. Abaixo, apresentamos o algoritmo recursivo proposto por Ambler et al [AMBLER-75].

```

Cliques(X,Y) :=
  se nenhum nó em Y - X está conectado
  todos os elementos de X
  então < X >
  caso contrário
    Cliques(X U { y },Y) U Cliques(X,Y - { y })
  onde y é conectado a todos os elementos de X.
  
```

O algoritmo é invocado com Cliques( $\emptyset$ , todos os nós do grafo) e retorna todos os cliques que incluem X e estão incluídos em Y.

## 2.4 - SUMÁRIO

Foram apresentados os módulos lógicos que compõem um sistema de visão computacional e para cada um, algumas alternativas de abordagens foram expostas.

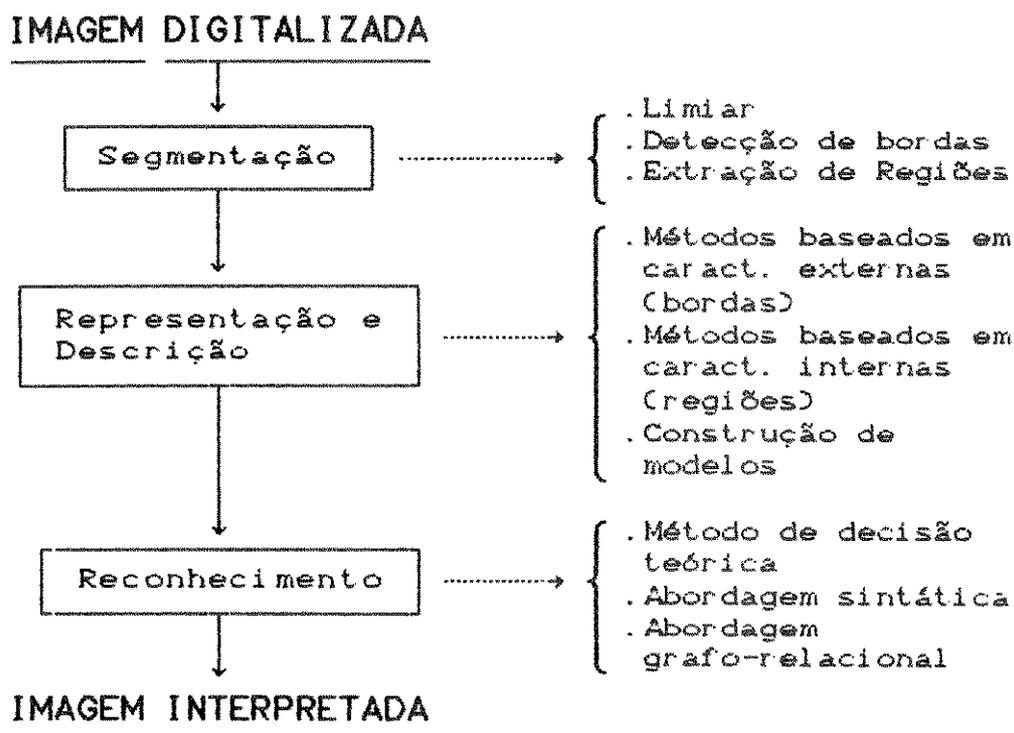


Figura 2.21 - Sistema de visão computacional.

Obviamente, tal exposição não é completa, mesmo porque seria impossível abranger toda a diversidade de propostas existentes. Entretanto, procurou-se deixar claro a forma como o sistema de visão computacional é, aqui, assumido (fig. 2.21), bem como as abordagens apresentadas.

Note-se, também, que o processo de aquisição de modelos (ou aprendizado) não foi explorado neste capítulo, sendo exposto, como utilizado pelo sistema, no capítulo que se segue.

## CAPÍTULO 3

# DEFINIÇÃO E IMPLEMENTAÇÃO DE UM SISTEMA DE VISÃO COMPUTACIONAL

### 3.0 - GENERALIDADES

Neste capítulo, com base no exposto no capítulo anterior, são discutidas a definição e a implementação de um sistema de visão computacional capaz de reconhecer e localizar objetos parcialmente visíveis.

São definidos um conjunto de hipóteses simplificadoras no processo de aquisição da imagem e um domínio de aplicação (conjunto de objetos os quais são passíveis de serem reconhecidos) sobre o qual o sistema deve atuar. Então, cada um dos módulos que o compõem é discutido. Por fim, são considerados alguns aspectos de implementação.

### 3.1 - DEFINIÇÃO DO SISTEMA

Basicamente, a arquitetura do sistema em discussão é da forma como exposta no capítulo 2, acrescido de informações que lhe são particulares, mais especificamente:

- i) Restrições quanto ao domínio de aplicação e
- ii) Capacidade de lidar com cenas onde os objetos apareçam posicionados de forma aleatória, podendo estar, inclusive, uns sobrepostos a outros.

Suas principais funções são:

- 1) Identificação dos objetos presentes na cena: Ainda que tais objetos não estejam totalmente visíveis, com outros objetos sobrepostos a eles, o sistema deve ser capaz de, se possível, discerni-los e classificá-los como pertencentes a uma das classes de objetos conhecidos.
- 2) Determinação do posicionamento de cada objeto identificado: Para que o sistema manipulador (robô), ao qual o presente sistema visa auxiliar, possa manipular os objetos identificados, é necessário que ele saiba onde e como tais objetos estão posicionados sobre o plano de trabalho. Assim, o sistema de visão deve fornecer a posição e a orientação de cada objeto identificado.
- 3) Determinação das relações de sobreposição entre os objetos

identificados: Ainda com o intuito de fornecer maiores informações sobre como os objetos identificados estão dispostos sobre o plano de trabalho, o sistema de visão deve determinar as relações de sobreposição existentes entre eles.

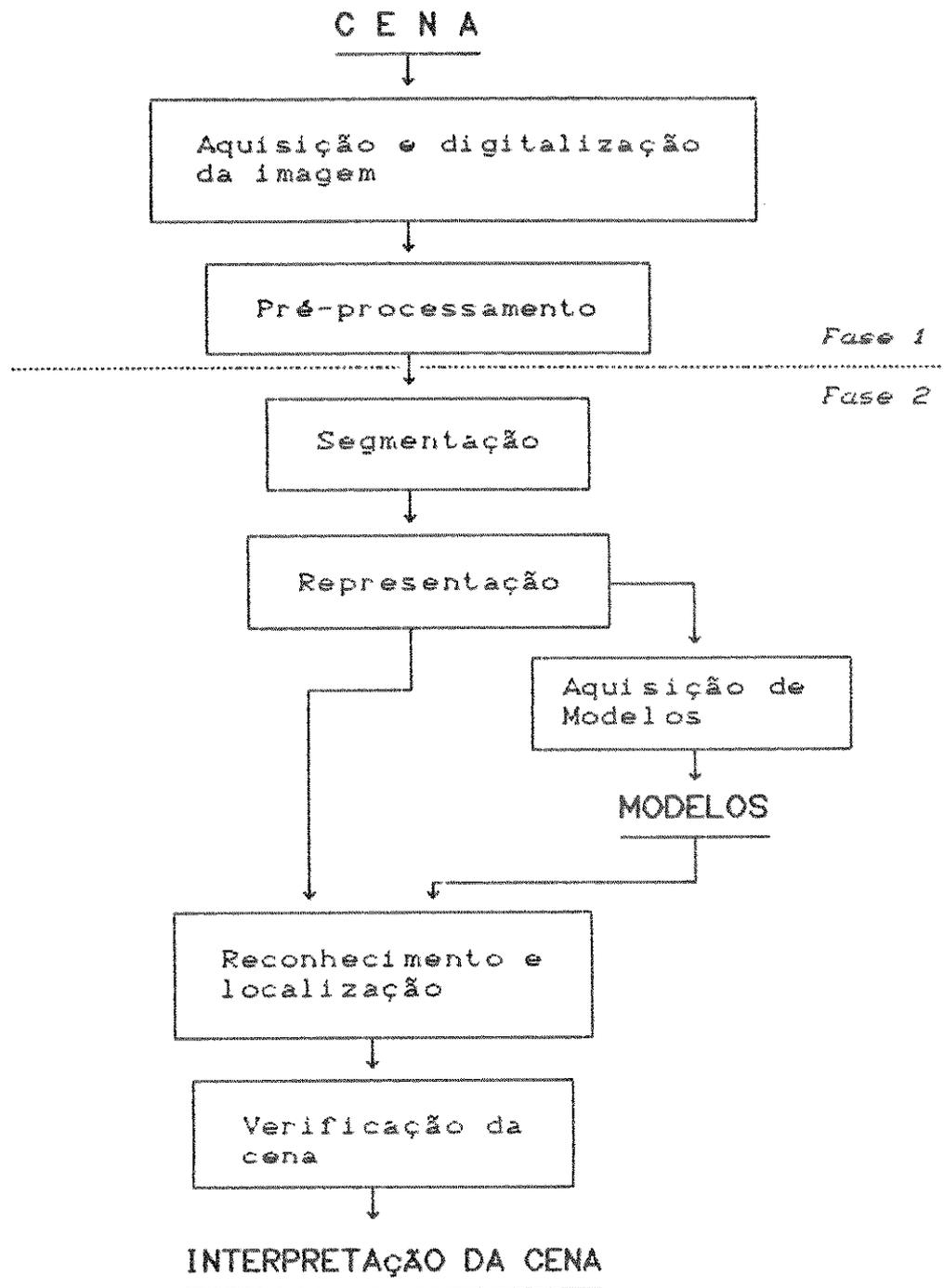


Figura 3.1 - Esquema do Sistema de Visão Computacional.

A figura 3.1 apresenta uma representação esquemática da organização de seus principais componentes. A parte do sistema denominada como Fase 1 refere-se aos processos de aquisição e digitalização da imagem e de pré-processamento visando a eliminação de ruídos e possíveis distorções causadas pela iluminação. Tais processos são assumidos definidos e implementados, não sendo discutidos neste texto.

Sendo o objetivo do sistema um pouco mais que, simplesmente, reconhecer os objetos presentes em uma cena, para se fazer referência às suas funções como um todo, utiliza-se o termo "interpretação da cena".

### 3.1.1 - RESTRIÇÕES

Tanto os objetos que compõem a cena a ser interpretada quanto a forma de aquisição da imagem estão sujeitos a um certo número de restrições. Uma vez que tais restrições definem o domínio de aplicação do sistema, tendo influência direta sobre o modelo de representação interna, sua consideração é muito importante durante as fases de projeto e implementação.

A imposição de restrições pode ter por objetivo obter tanto uma maior eficiência computacional quanto uma maior facilidade na elaboração das representações internas dos objetos conhecidos.

Em aplicações industriais, tal imposição é facilitada por dois fatores:

- 1) **Facilidade de controle ambiental:** Embora, seja desejável que o controle sobre o ambiente seja o mínimo possível, com o sistema sendo flexível o suficiente para lidar com condições desfavoráveis, o ambiente industrial é próprio para tal.
- 2) **Natureza dos objetos é conhecida:** Sendo a aplicação conhecida, o conjunto de objetos reconhecíveis é, também, conhecido e, em geral, suas formas são facilmente descritas.

No presente sistema, consideram-se as seguintes restrições:

#### ▶ Com relação à aquisição da imagem

- 1) As imagens são adquiridas com uma definição de 128x128.
- 2) A faixa de níveis de cinza é de 16 possíveis valores.
- 3) Sua aquisição é suposta adquirida de um ambiente controlado com um nível de ruído bastante reduzido.

► E com relação à natureza dos objetos reconhecíveis

- 1) Os objetos são supostos inerentemente bi-dimensionais (2-D).
- 2) Os objetos são supostos ter formas poligonais e são caracterizados exclusivamente por seu contorno.

Supondo objetos de formas triangulares e retangulares em vários tamanhos como os objetos conhecidos pelo sistema, uma possível cena a ser interpretada poderia ser a mostrada pela figura 3.2.

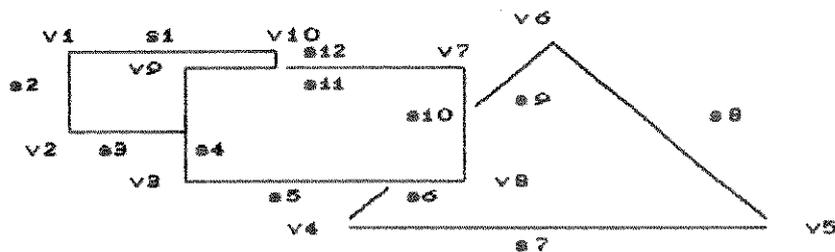


Figura 3.2 - Exemplo de cena a ser interpretada.

Ainda que muitas das idéias empregadas na definição do sistema sejam independentes das restrições acima, sua implementação faz extenso uso delas, tornando o sistema particular a este domínio de aplicação.

### 3.1.2 - MÓDULO DE SEGMENTAÇÃO

O fato de o domínio de aplicação do sistema ser constituído por objetos de forma poligonal, caracterizados apenas pelo seu contorno, sugere que o problema de segmentação seja abordado pelo método de detecção de bordas.

O operador de detecção de elementos de borda escolhido é o operador de aproximação do gradiente de Sobel (vide seção 2.1.2.1). Como não há uma concordância sobre uma abordagem superior às demais, com cada uma delas tendo um desempenho superior em determinadas aplicações [BALLARD-82], tal operador foi escolhido por sua simplicidade e facilidade de implementação.

Para reduzir a quantidade de dados a ser manipulado e facilitar o subsequente procedimento de determinação de bordas, a imagem gradiente (os elementos de borda encontrados) são submetidos a um procedimento de afinamento ou esqueletização (vide seção 2.2.1). Isto, então, produz uma imagem gradiente composta

por elementos de borda com largura máxima de um pixel, dispensando a necessidade da utilização de um processo de determinação de bordas (vide seção 2.1.2.2).

Dessa forma, a informação obtida na saída do módulo de segmentação são as bordas ou contornos dos objetos presentes na cena. A figura 3.3 ilustra, esquematicamente, a organização do módulo de segmentação.

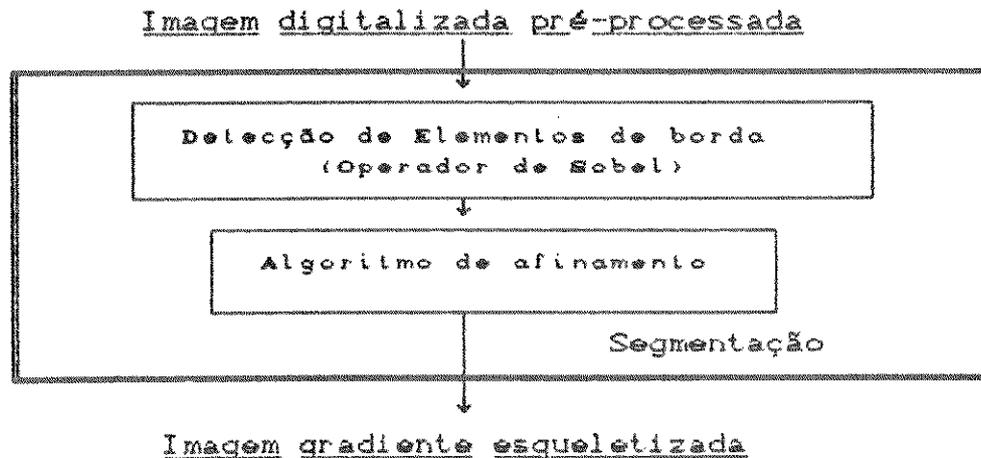


Figura 3.3 - Módulo de segmentação.

### 3.13 - MÓDULO DE REPRESENTAÇÃO

Como citado na seção 2.2, este módulo tem por objetivo obter uma representação mais adequada para o sub-sequente processo de reconhecimento da imagem obtida do módulo de segmentação, ainda, a nível de pixels.

Para isto, inicialmente, submete-se a imagem segmentada a um procedimento de aproximação poligonal, utilizando o algoritmo exposto na seção 2.1.2.a.2. Uma vez que a estratégia de reconhecimento adotada requer a representação dos objetos por meio de grafos (vide seção 3.1.5), este módulo converte a representação da imagem, da aproximação poligonal, para uma representação em forma de grafo. Tal grafo, representando os objetos presentes na imagem, é denominado grafo-imagem.

Por fim, o grafo imagem é sub-dividido em sub-grafos representativos dos objetos que compõem a cena. Tal partição do grafo-imagem constitui-se no primeiro levantamento das relações de sobreposição entre os objetos.

A figura 3.4 ilustra, esquematicamente, a organização deste módulo.

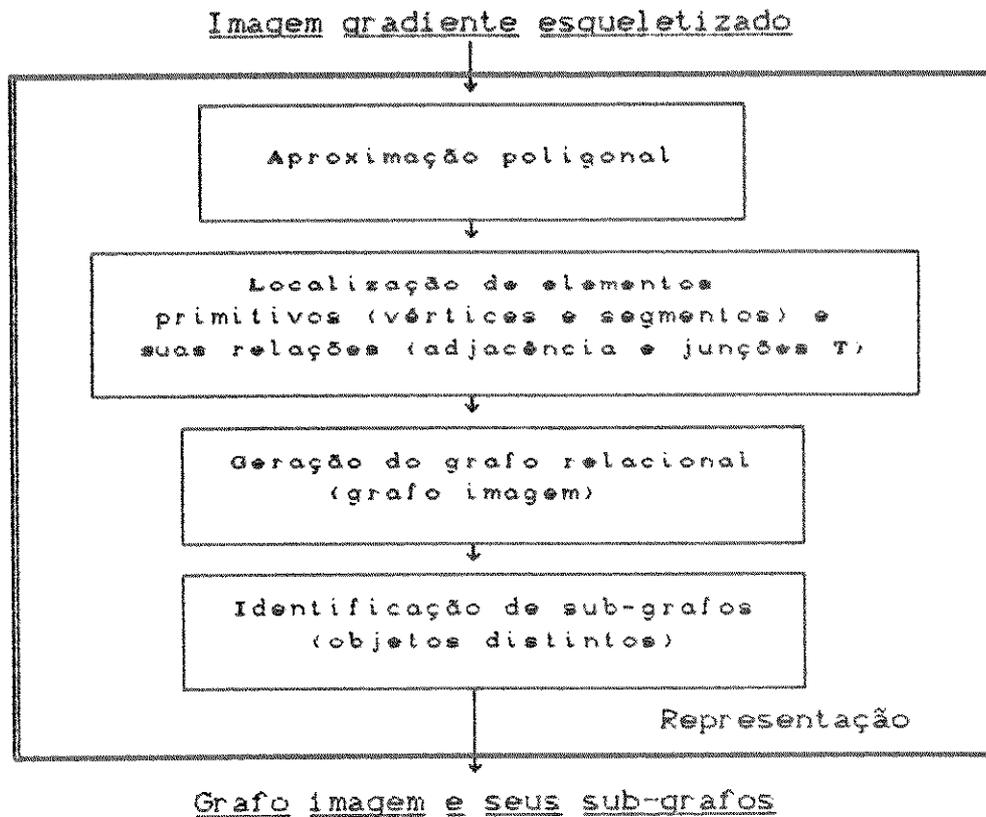


Figura 3.4 - Módulo de Representação.

### 3.1.3.1 - JUNÇÕES T

Denomina-se de junção *T* à configuração gerada pelo encontro de dois segmentos da forma ilustrada pela figura 3.5. Em geral, tal junção aparece quando um objeto está sobreposto a outro, sendo assim, um indício para a determinação das relações de sobreposição. No caso da figura 3.5, o objeto A obstrui parcialmente o objeto B e os elementos *a* e *a'* são a parte superior das respectivas junções *T* enquanto os elementos *b* e *b'* são a parte inferior.

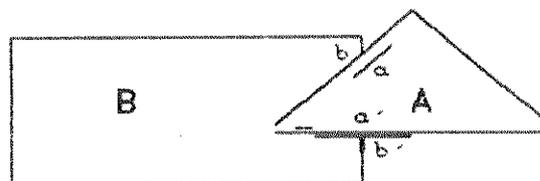


Figura 3.5 - Configuração onde aparecem junções *T*.

Entretanto, casos "degenerativos", também, podem dar origem a junções T. A figura 3.6 ilustra algumas possíveis configurações de objetos não sobrepostos que originam junções T. No caso (a) dois objetos de forma retangular têm um de seus segmentos se tocando totalmente e no caso (b) um objeto de forma triangular e outro de forma retangular têm seus segmentos tocando-se apenas parcialmente. Em virtude da ocorrência de tais casos, as junções T são consideradas apenas como indícios de sobreposição, estando sujeitos a uma posterior confirmação.

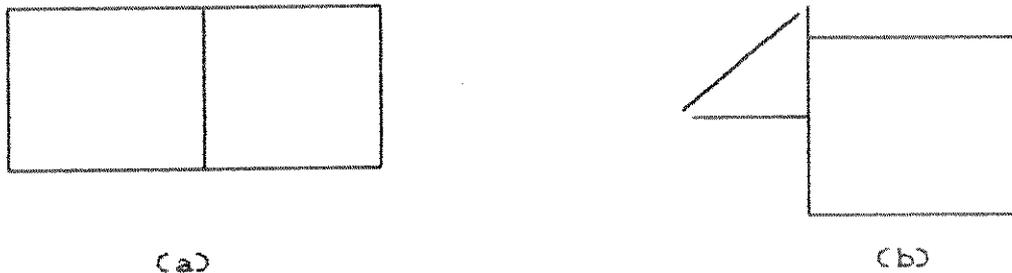


Figura 3.6 - Junções T : Casos degenerativos.

### 3.1.3.2 - O GRAFO RELACIONAL

A utilização de grafos relacionais para descrever objetos têm por objetivo refletir seus aspectos estruturais com os nós correspondendo a elementos primitivos e os arcos a relações entre eles.

No presente sistema, os elementos primitivos considerados para descrever os objetos são as partes que determinam a sua forma, mais especificamente, os elementos que compõem o seu contorno. Assim, um grafo de descrição possui a seguinte correspondência com o objeto que ele representa:

nó do grafo	↔	vértice do objeto (ângulo) ou segmento de reta (lado)
	e	
arco do grafo	↔	relação de adjacência

onde cada nó do grafo é caracterizado por um conjunto de propriedades relacionadas ao seu elemento primitivo associado (Tab. 3.1).

Dois tipos de arcos são utilizados na representação de uma cena. Um deles é bi-direcional e refere-se a uma relação de adjacência entre elementos primitivos. O outro é um arco direcionado, utilizado para representar junções T. Se o arco é direcionado de um elemento A para um elemento B, o sistema

interpreta-o como descrevendo uma junção *T* onde o elemento *A* corresponde à parte superior da junção e o elemento *B* à parte inferior. Isto significa que o objeto correspondente ao elemento *A* pode estar obstruindo parte do objeto correspondente ao elemento *B*.

Tipo de nó	Propriedade
ângulo	abertura posição orientação
segmento	comprimento

Tabela 3.1 - Propriedades associadas aos elementos primitivos utilizados na descrição de objetos.

Como um exemplo, a figura 3.7 ilustra o grafo imagem correspondente à representação da cena mostrada na figura 3.2.

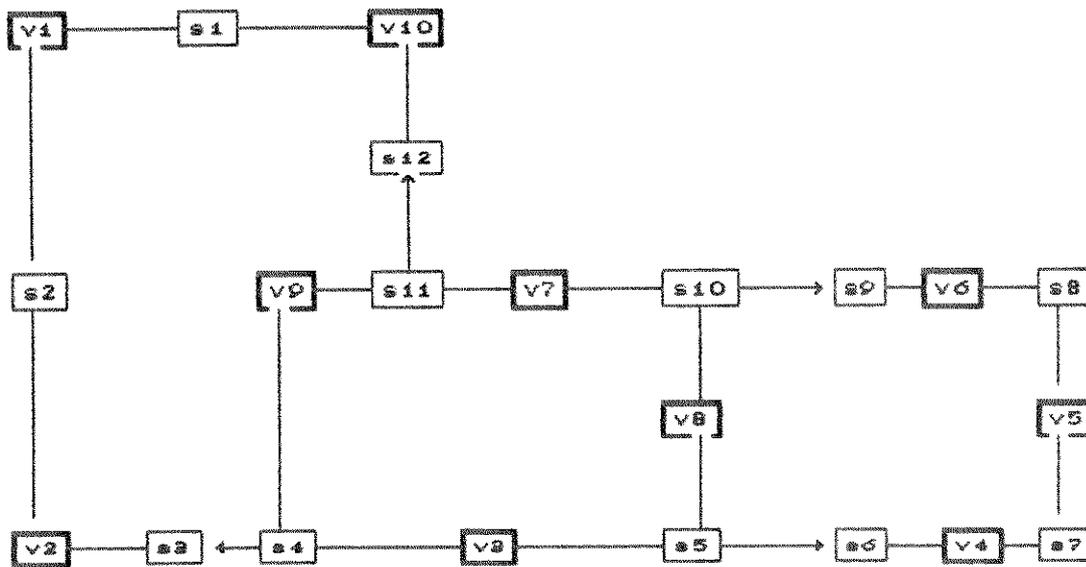


Figura 3.7 - Grafo imagem representativo da cena mostrada na figura 3.2.

Observe que o grafo imagem pode ser interpretado como composto por diversos sub-grafos interligados por arcos correspondentes a junções *T*, onde o único tipo de arco é o arco de

adjacência.

### 3.1.4 - MÓDULO DE AQUISIÇÃO DE MODELOS

Este módulo refere-se a um processamento "off-line" destinado a proporcionar ao sistema a aquisição de modelos correspondentes às classes de objetos conhecidos e está em correspondência com o que se denomina processo de aprendizagem ou treinamento.

Numa abordagem por espaço de descritores utilizando, por exemplo, uma função discriminante linear, este módulo corresponderia a, utilizando amostras de padrões conhecidos, determinar os valores dos pesos da função discriminante (os  $\alpha$ 's), tal que proporcionassem a melhor classificação [DUDA-73] e em uma abordagem sintática, corresponderia a inferir as gramáticas descritoras das classes, a partir de amostras de padrões, num processo denominado inferência gramatical [GONZALEZ-78]. No caso da abordagem grafo-relacional, o processo de inferir modelos a partir de exemplos poderia envolver procedimentos de grande complexidade tais como o aprendizado de descrições estruturais [WINSTON-78].

Em qualquer dos casos acima, a simples definição e implementação deste módulo mereceria um estudo a parte. Assim, tal módulo é projetado da forma a mais simples possível e capaz de atender às necessidades do presente sistema.

#### 3.1.4.1 - O MODELO UTILIZADO

Cada classe conhecida é representada por um seu exemplar, sendo sua forma de representação idêntica àquela derivada do processo de representação, com a ressalva de que o único tipo de arco existente é o de adjacência.

Um fato importante a se notar é a adoção de um sistema de coordenadas, dito sistema de coordenadas do modelo e sob o qual o modelo é descrito. A razão de se adotar um sistema de coordenadas para descrição do modelo refere-se à tarefa de localização do objeto (determinação de sua posição e orientação) (vide seção 3.1.5).

A adoção do sistema de coordenadas é tal que sua origem coincide com o centro de massa do objeto considerado, sendo a direção dos seus eixos adotadas de forma arbitrária. A figura 3.8 mostra um possível objeto e o seu eixo de referência associado.

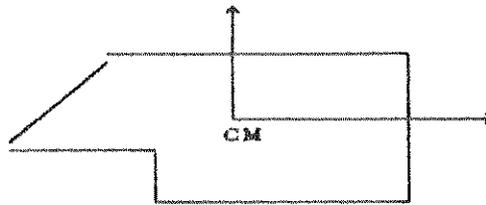


Figura 3.8 - Exemplo de um modelo e seu eixo de referência.

Assim, o processo de modelamento consiste em apresentar o objeto que se quer adquirir e, a partir do grafo imagem obtido do módulo de representação (obviamente o sub-processo que determina seus sub-grafos constituintes não é executado), determinar o centro de massa do objeto. Então, adotar um sistema de coordenadas tal que sua origem coincida com o centro de massa do objeto. As direções dos eixos, uma vez que podem ser escolhidos de forma arbitrária, são feitos coincidentes com as direções do sistema de coordenadas da imagem. A figura 3.9 ilustra, esquematicamente, a organização deste módulo.

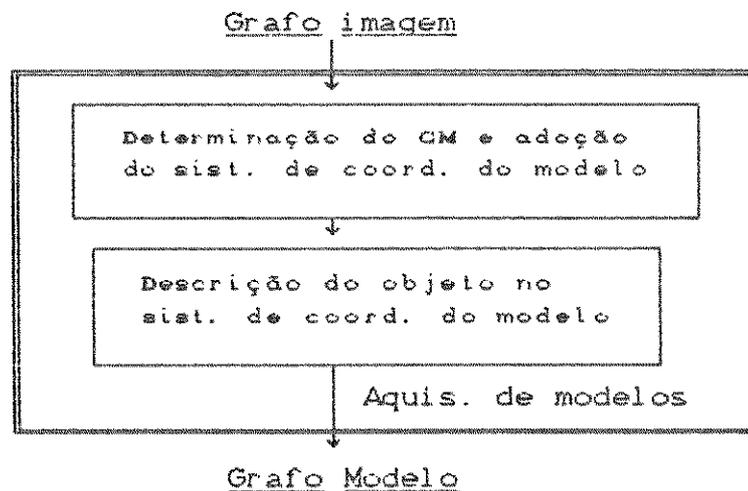


Figura 3.9 - Módulo de aquisição de modelos.

### 3.15 - MÓDULO DE RECONHECIMENTO E LOCALIZAÇÃO DE OBJETOS

Um importante fator a se considerar na definição deste módulo refere-se às características das cenas com as quais o sistema trabalha, marcadamente, o fato de ser permitido a um objeto obstruir outro parcialmente. Obviamente, a estratégia de reconhecimento não pode fazer uso de descritores globais, o que, de imediato, inviabiliza a utilização do método de decisão

teórica. O método sintático poderia ser empregado, alterando-se a gramática descritora das classes para que aceitasse cadeias imperfeitas (a parte referente a um possível símbolo terminal estaria oculto), mas isto poderia levar a gramáticas excessivamente complexas [GONZALEZ-78]. O método que parece ser o mais apropriado e elegante para lidar com esta situação é o método grafo-relacional. Desde que o modelo contém tanto características locais como a forma como estas se relacionam, representadas como grafo, o estabelecimento da correspondência modelo-imagem (reconhecimento) dependerá de ambas as informações, com cada característica local fornecendo um vestígio para identificação do objeto parcialmente visível.

Outra importante consideração a se fazer neste módulo refere-se ao fato de que as mesmas restrições espaciais que são utilizadas pelo sistema no processo de reconhecimento determinam a escala, a orientação e o posicionamento do objeto em relação à referência. Isto significa que as tarefas deste módulo são realizadas de forma simultânea, ou seja, o mesmo procedimento que executa a tarefa de reconhecimento determina, simultaneamente, a escala, a orientação e o posicionamento do objeto reconhecido.

### 3.1.5.1 - RECONHECIMENTO

Um objeto é considerado como pertencente a uma classe se ele possui a mesma forma que o exemplar tomado como referência, de modo que a única variação permitida dentro de uma classe é a escala. Uma vez que ambos modelo e imagem são representados como grafos, a abordagem grafo-relacional, de uma forma simples, consiste em confrontar ambos os grafos, um sub-grafo imagem e um grafo modelo, em busca de uma correspondência (vide seção 2.3.3). Entretanto, algumas alterações nesta estratégia simples são necessárias para lidar com as seguintes possibilidades:

- 1) Identificação dúbia: Os vestígios fornecidos pelo sub-grafo considerado são compatíveis com dois ou mais objetos. Isto acontece quando duas ou mais classes referem-se a objetos que têm uma parte de suas formas idênticas, sendo justamente tal parte visível na cena.
- 2) Sub-grafos correspondentes a um mesmo objeto: Um objeto se sobrepõe a outro de forma que este não apresente todas as suas características visíveis agrupadas em um único sub-grafo.

A figura 3.10 ilustra tais casos. No caso (a) se objetos com os formatos semelhantes às letras E e F fazem parte do domínio de aplicação, no arranjo da cena, um objeto de formato retangular pode se sobrepôr a um outro, impedindo que se determine sua identidade (E ou F). No caso (b) o grafo imagem correspondente a cena apresentará dois sub-grafos distintos correspondentes a um mesmo objeto. No primeiro caso é necessário que o sistema, se possível, resolva a ambiguidade e no segundo, deve-se evitar a

dupla identificação de um mesmo objeto (o sistema associa dois ou mais sub-grafos distintos a um objeto, mas ambos pertencem à uma mesma classe, possuem a mesma escala e orientação e estão posicionados no mesmo local).

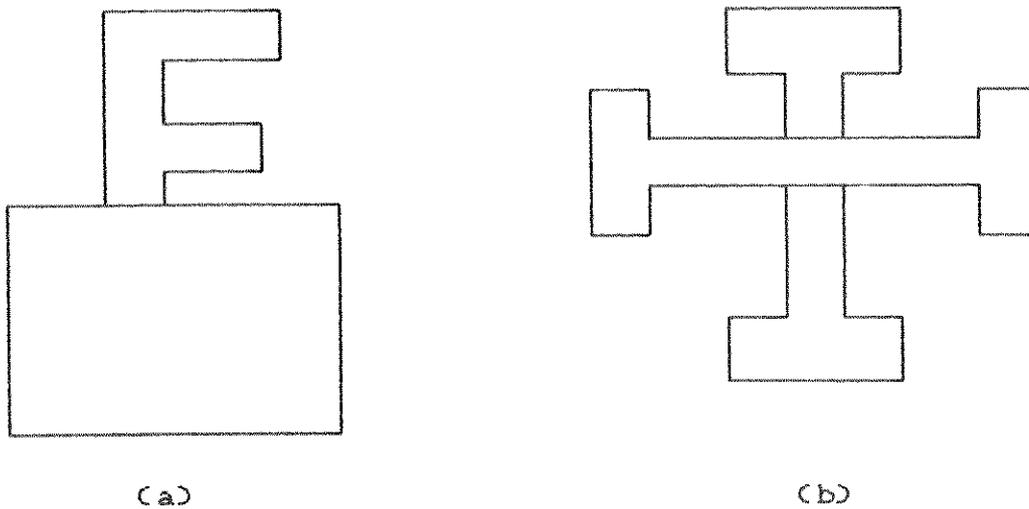


Figura 3.10 - Possíveis configurações de objetos.

Para lidar com tais possibilidades, a estratégia de reconhecimento utilizada é como mostrada na figura 3.11.

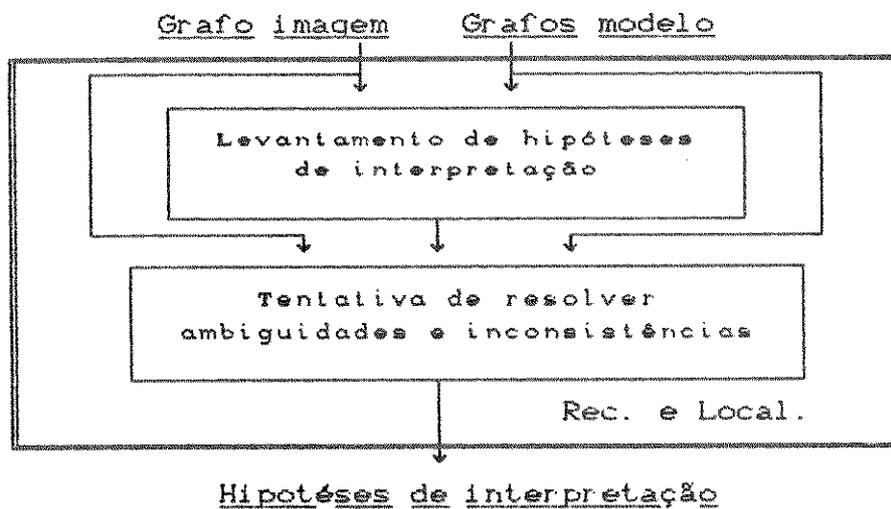


Figura 3.11 - Módulo de Reconhecimento e Localização.

### 3.15.11 - LEVANTAMENTO DE HIPÓTESES DE INTERPRETAÇÃO

Inicialmente, os sub-grafos constituintes do grafo imagem são diferenciados entre sub-grafos significativos e não significativos. Considera-se como significativos aqueles que possuem um número de nós  $\geq 5$  e como não significativos aqueles com número de nós  $< 5$ .

Para proceder o levantamento de hipóteses, considera-se apenas os sub-grafos significativos. Cada um dos sub-grafos do grafo imagem é confrontado com cada um dos grafos modelos. Para cada sub-grafo imagem (considerado como um grafo), procura-se uma correspondência com um sub-grafo do grafo modelo, ou seja, procura-se um isomorfismo entre sub-grafos (vide seção 2.3.3). Considera-se um isomorfismo bem sucedido se no mínimo 60% do grafo imagem está em correspondência com o grafo modelo.

Ao final do procedimento, cada um dos sub-grafos imagem, encontra-se numa das seguintes situações:

- 1) Não existe nenhum isomorfismo bem-sucedido com nenhum dos grafos-modelo.
- 2) Existe um único sub-grafo modelo tal que o isomorfismo é bem-sucedido.
- 3) Existem mais de um sub-grafo modelo tal que o isomorfismo é bem sucedido.

No primeiro caso, não se pode associar o sub-grafo imagem a qualquer das classes de modelos conhecidos. Isto pode indicar que um objeto não-pertencente ao domínio de aplicação está presente na cena ou que aconteceu um erro em processos anteriores de extração de características (segmentação e representação). Em ambos os casos, o objeto correspondente ao sub-grafo é rejeitado, não sendo considerado na interpretação da cena. No segundo caso, o sub-grafo é considerado reconhecido, pertencente à classe correspondente ao grafo modelo e as transformações de escalamento, rotação e translação associados ao isomorfismo como indicadores de sua escala, orientação e posição (vide seção 3.1.4.1.2). Finalmente, o último caso refere-se à identificação dúbia. Todas as classes associadas ao grafos modelos que resultam em isomorfismo bem-sucedido são consideradas hipóteses viáveis para interpretação dos objetos, com informações de escala, orientação e posição. A figura 3.12 ilustra tal situação. Os grafos imagem  $G\text{-img}_1$  e  $G\text{-img}_n$  possuem mais de uma hipótese de interpretação e o grafo imagem  $G\text{-img}_2$  possui uma única interpretação possível, sendo reconhecido como um objeto da classe associada ao grafo modelo  $G\text{-mod}_3$ .

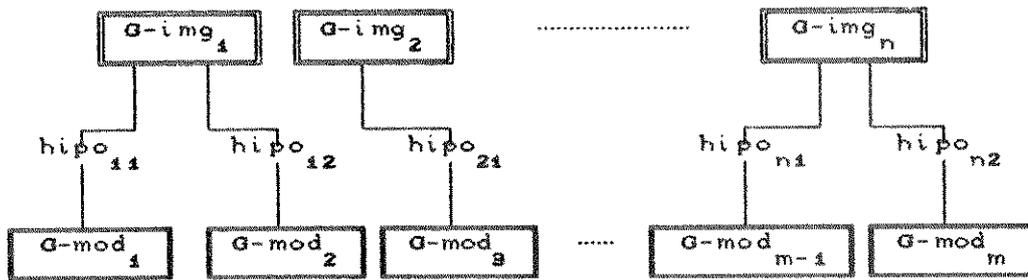


Figura 3.12 - Diversas interpretações possíveis para um objeto.

### 3.15.12 - RESOLUÇÃO DE AMBIGUIDADES E INCONSISTÊNCIAS

Uma vez consideradas as hipóteses iniciais, faz-se necessário a verificação dos casos de identificação dúbia e sub-grafos correspondentes a um mesmo objeto. Para isto, utiliza-se os sub-grafos considerados não significativos para a verificação das hipóteses num processo de extensão das evidências existentes.

Para cada uma das hipóteses de cada um dos sub-grafos não resolvidos (com mais de uma hipótese de interpretação), verifica-se a existência de sub-grafos não significativos capazes de aumentar as evidências de tais hipóteses. Isto significa verificar o isomorfismo entre o sub-grafo do grafo modelo não considerado na obtenção da hipótese e os sub-grafos não significativos, mantidas as restrições de escala, rotação e translação. A figura 3.13 ilustra uma possível situação final após o término do procedimento de extensão de hipóteses. Os sub-grafos imagens com índices maiores que  $n$  são não significativos e estão associados a interpretações em que sua consideração é compatível com a hipótese inicial.

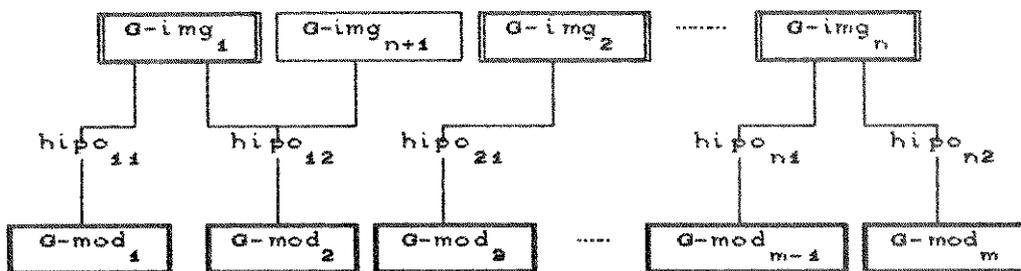


Figura 3.13 - Hipóteses extendidas.

Cada possível combinação das hipóteses de interpretação dos sub-grafos imagem gera uma hipótese de interpretação da cena. Para cada uma destas hipóteses, verifica-se se existem sub-grafos diferentes, identificados como um mesmo tipo de objeto e com as mesmas transformações de escala, rotação e translação. Em caso afirmativo, elimina-se uma das interpretações, o que soluciona o problema de sub-grafos distintos correspondendo a um mesmo objeto serem interpretados independentemente.

Para decidir entre as possíveis hipóteses de interpretação da cena, considera-se como um critério comparativo para tais hipóteses, o número de vestígios explicados por ela, ou seja, o número de nós do grafo imagem envolvido na interpretação. Assim, o sistema adota como sendo a interpretação final da cena aquela dentre as hipóteses que apresentar um número maior de vestígios explicados.

### 3.15.2 - LOCALIZAÇÃO E DETERMINAÇÃO DE ESCALA

Considerando que o estabelecimento de um isomorfismo entre dois grafos implica em uma correspondência biunívoca entre os nós de ambos os grafos e lembrando que das propriedades descritoras dos elementos primitivos associados aos nós de tais grafos (vide seção 3.1.3.2), apenas a abertura dos ângulos é invariante às transformações de escalamento, rotação e translação, tal procedimento de isomorfismo, necessariamente, leva a imposições de transformações que sustentem tal correspondência.

Lembrando que o grafo imagem descreve um objeto no sistema de coordenadas da imagem e que o grafo modelo descreve um objeto (o objeto tomado como referência) no sistema de coordenadas do modelo, fazendo o sistema de coordenadas do modelo coincidente com o sistema de coordenadas da imagem é equivalente a, agora, descrever os dois objetos, imagem e modelo, com relação a um único sistema de coordenadas (fig. 3.14). Assim, ao considerar a transformação associada ao isomorfismo, observa-se que esta é equivalente a uma transformação que leva do objeto modelo para o objeto imagem. Tal transformação pode, também, ser entendida como a descrição de um sistema de coordenadas do objeto (equivalente ao sistema de coordenadas do modelo) com relação ao sistema de coordenadas do modelo. Uma vez que este coincide com o sistema de coordenadas da imagem, a transformação de translação é equivalente à localização do objeto no plano de trabalho. As transformações de escala e rotação determinam a escala e orientação do objeto imagem com relação ao objeto modelo.

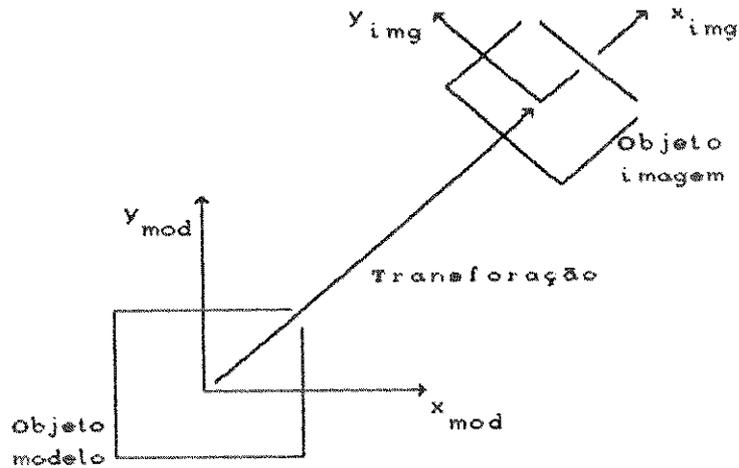


Figura 3.14 - Localização de um objeto no plano de trabalho.

### 3.16 - VERIFICAÇÃO DA CENA

Este módulo, simplesmente, verifica as relações de sobreposição entre os objetos já identificados para a (cada uma das possíveis) interpretação (ões) da cena.

Tal processo consiste na verificação de todas as junções  $T$ . Considera-se que tais junções ocorrem devido a apenas duas situações, sendo qualquer outra possibilidade considerada rara e não sendo, aqui, tratada:

- 1) Um objeto se sobrepõe a outro e apenas parte de um dos lados de sua forma poligonal é visível.
- 2) Dois objetos estão colocados lado a lado e, embora existam junções  $T$ , todos os lados das formas poligonais são visíveis.

Como cada um dos segmentos que formam a junção  $T$  está associado a um objeto, a tarefa de verificação de sobreposição entre objetos consiste na verificação das relações de escala do segmento suspeito interrompido com relação ao seu correspondente modelo. Se tal relação é consideravelmente inferior à relação de escala da transformação associada ao isomorfismo, assume-se a relação de sobreposição. Em caso contrário, tal suposição é desprezada, assumindo-se que os objetos estão dispostos lado a lado. A figura 3.15 ilustra a estrutura do módulo de verificação da cena.

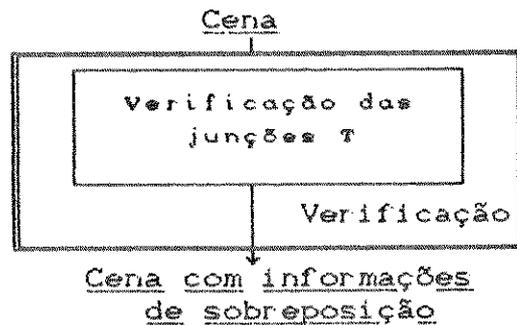


Figura 3.15 - Módulo de Verificação da cena.

### 3.2 - ALGUNS ASPECTOS DE IMPLEMENTAÇÃO

O programa referente ao presente sistema é implementado a maior parte em Prolog com uma pequena parte em C (Tab. 3.2). A razão da utilização de Prolog como linguagem de desenvolvimento básica deveu-se às facilidades por ele oferecidas no que se refere à manipulação de relações (grafos são descritos por relações) e facilidade de programação. Uma vez que o objetivo desta implementação visa a construção de um protótipo, critérios como eficiência e compatibilidade com outros sistemas não são, aqui, considerados. A parte desenvolvida em C refere-se à parte do sistema de mais baixo nível, onde a manipulação de imagens (níveis de cinza e gradiente) é uma constante e envolve a utilização de estruturas "array".

Desenvolvimento em C	Desenvolvimento em Prolog
Segmentação Representação (parte)	Representação (parte) Aquisição de modelos Reconhec. e Localização Verificação da cena

Tabela 3.2 - Implementação.

#### 3.2.1 - MÓDULO DE SEGMENTAÇÃO

A implementação do operador de Sobel é direta como exposto na seção 2.1.2.1 e o algoritmo de afinamento utilizado é o proposto por Zhang e Suen [ZHANG-84] e exposto seção 2.2.1.

As bordas encontradas por este módulo são enviadas para o módulo de representação em uma estrutura array (fig. 3.16), dita imagem gradiente.

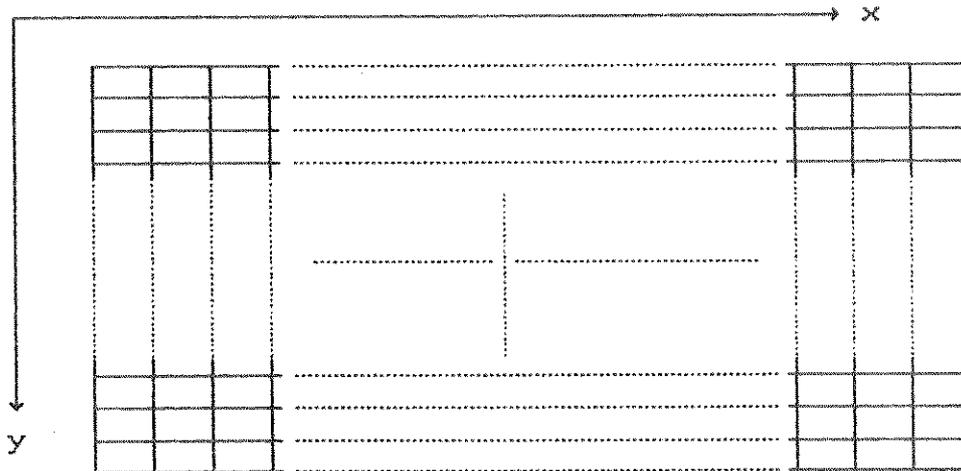
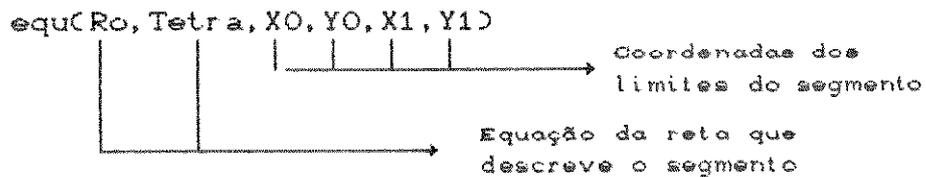


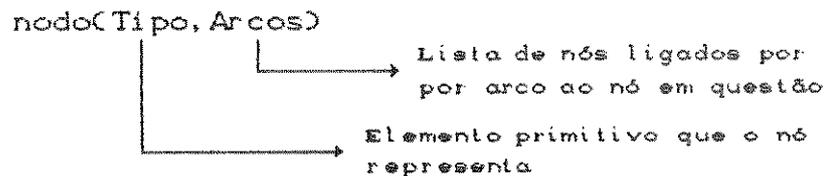
Figura 3.16 - Estrutura de array com a imagem gradiente.

### 3.2.2 - MÓDULO DE REPRESENTAÇÃO

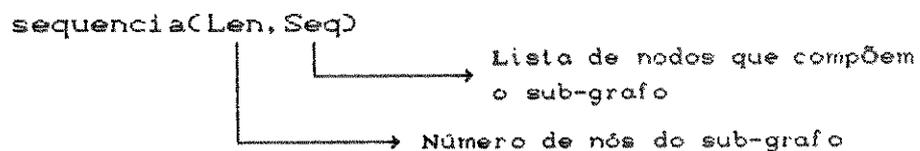
Inicialmente, as bordas, representadas no array "imagem gradiente" são submetidas a um processo de aproximação poligonal, produzindo uma lista de segmentos de reta da forma:



Então, a lista de segmentos de reta representando a cena é verificada e uma estrutura de grafo é gerada. Tal grafo é representado, internamente, por uma estrutura de adjacência cujas relações são da seguinte forma:



Os sub-grafos do grafo imagem são determinados, sendo internamente representados da seguinte forma:



### 3.2.3 - MÓDULO DE AQUISIÇÃO DE MODELOS

O procedimento de aquisição de um modelo consiste em, inicialmente, determinar-se o contorno (uma sequência de segmentos) do objeto no sentido anti-horário. Para isto, verifica-se o sentido do ângulo descrito ao se percorrê-lo. Se ele é positivo (+360°) a sequência descreve o contorno no sentido anti-horário. Em caso contrário, ela descreve o contorno no sentido horário tendo, então, sua ordem invertida. Os segmentos, agora, são direcionados (fig. 3.17a), existindo oito possíveis situações para cada segmento (fig. 3.17b)

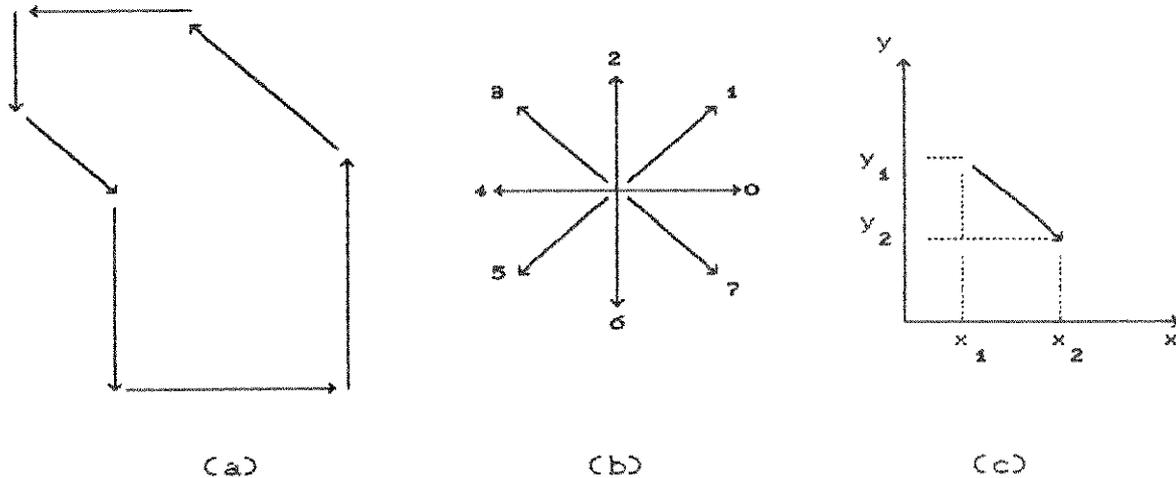


Figura 3.17 - Cálculo do centro de massa do objeto.

Cada um destes segmentos pode ser considerado como descrevendo um trapézio com relação aos eixos (fig. 3.17c). Sendo tais segmentos direcionados de forma a percorrer o contorno no sentido anti-horário, a área e o centro de massa do objeto podem ser calculados em função das áreas e centros de massa dos trapézios associados a cada segmento, sendo que alguns deles contribuem com valores de área negativa (com relação ao eixo x os segmentos 0, 1, e 7 e com relação ao eixo y os segmentos 6, 7 e 8). Assim, a área e o centro de massa do objeto podem ser calculados da seguinte forma:

$$\text{Área total} = \sum_{i=1}^{\text{nº de segtºs}} A_i \quad e$$

$$\bar{x} = \frac{\sum_{i=1}^{n^{\circ} \text{ de segtes}} x_i \cdot A_i}{\text{Área total}}$$

$$\bar{y} = \frac{\sum_{i=1}^{n^{\circ} \text{ de segtes}} y_i \cdot A_i}{\text{Área total}}$$

Sendo a origem do sistema de coordenadas do modelo coincidente com o centro de massa do objeto modelo, e de mesma orientação que o sistema de coordenadas da imagem, uma simples transformação de translação de  $(\bar{x}, \bar{y})$  equivale a descrever o objeto no sistema de coordenadas do modelo. A única modificação na descrição dos segmentos refere-se ao parâmetro  $\rho$  da equação da reta. Sendo, agora,

$$x' = x + x_0 \quad \text{e} \quad y' = y + y_0$$

um segmento de reta passa a ser descrito pela equação

$$(x' - x_0)\cos\theta + (y' - y_0)\sin\theta = \rho \quad \text{ou}$$

$$x'\cos\theta + y'\sin\theta = \rho + x_0\cos\theta + y_0\sin\theta$$

ou seja,

$$\rho' = \rho + x_0\cos\theta + y_0\sin\theta$$

A estrutura do grafo modelo armazenado é descrito da mesma forma que o grafo imagem, ou seja:

nodo(Tipo, Arcos)

### 3.24 - MÓDULO DE RECONHECIMENTO E LOCALIZAÇÃO

O isomorfismo é estabelecido através de um algoritmo de máximo clique (vide seção 2.3.3). Considera-se o conjunto de propriedades  $P = \langle \text{tipo de vértice (ângulo ou segmento)} \text{ e } \text{ângulo de abertura (caso o tipo seja ângulo)} \rangle$ . As relações de compatibilidade mútua são representadas pelas transformações de escalamento e rotação que relacionam o modelo à imagem. A transformação de translação é determinada da seguinte forma:

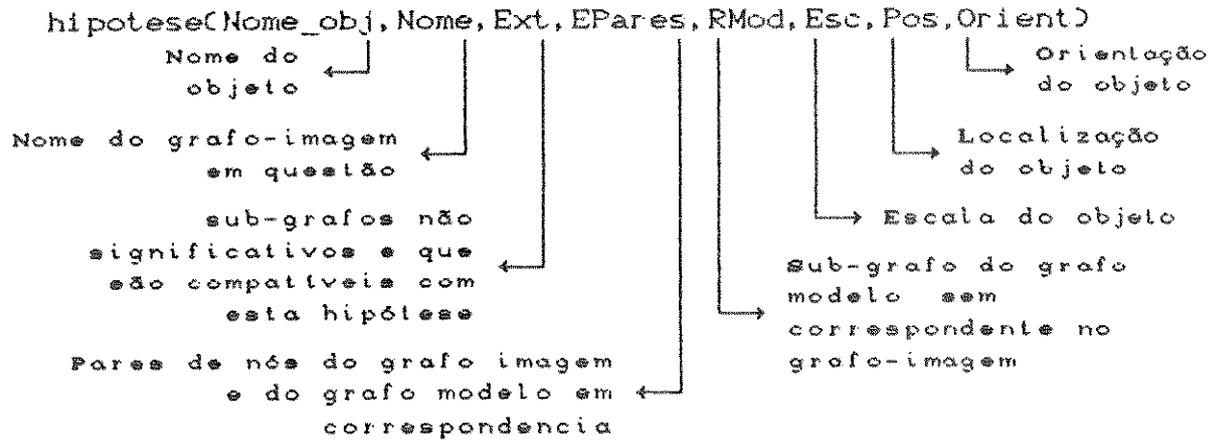
$$x_0 = u_m - (x_m \cos\theta - y_m \sin\theta) * Esc \quad \text{e}$$

$$y_0 = v_m - (x_m \sin\theta - y_m \cos\theta) * Esc$$

onde  $(u_m, v_m)$  representa um ponto no sistema de coordenadas do

objeto e  $(x_m, y_m)$  é o seu correspondente no sistema de coordenadas do modelo.

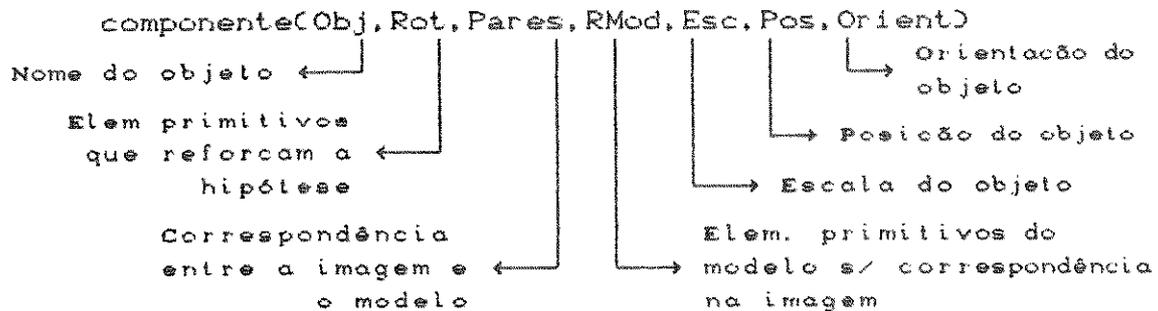
Cada hipótese é armazenada numa estrutura do tipo:



e cada possível interpretação da cena é representada numa estrutura do tipo:

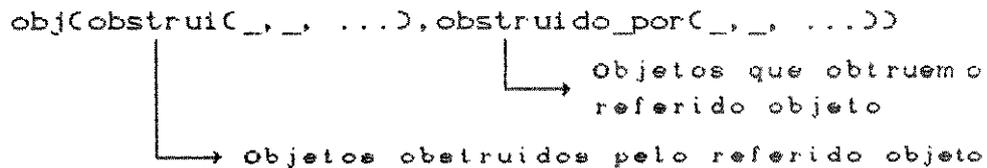


onde cada elemento  $Comp_i$  é uma estrutura do tipo:



### 3.2.5 - VERIFICAÇÃO DA CENA

A verificação das junções  $T$  acrescenta aos elementos componentes da cena informações sobre as relações de sobreposição entre os objetos. Tal procedimento consiste em alterar o Obj da estrutura componente para a seguinte forma:



### 3.3 - SUMÁRIO

Ao longo deste capítulo foram discutidas os vários aspectos envolvidos na implementação de um sistema de visão computacional capaz de reconhecer e localizar objetos parcialmente visíveis. Foram apresentados os vários módulos que o compõem e as restrições a ele incorporadas.

No capítulo que se segue, o desempenho do sistema é ilustrado através de vários exemplos de aplicação.

## CAPÍTULO 4

### EXEMPLOS DE APLICAÇÃO

#### 4.0 - GENERALIDADES

Neste capítulo, ilustra-se o desempenho do sistema através de experimentos, sendo expostos os resultados obtidos e as dificuldades por ele encontradas ao realizar a tarefa de interpretação das cenas propostas. As imagens utilizadas são sintetizadas com uma definição de 128x128. Os níveis de cinza variam dentro de uma faixa de 16 valores e as bordas são supostas compostas com uma linha (seção 2.1.2.1).

São considerados os seguintes modelos de objetos:

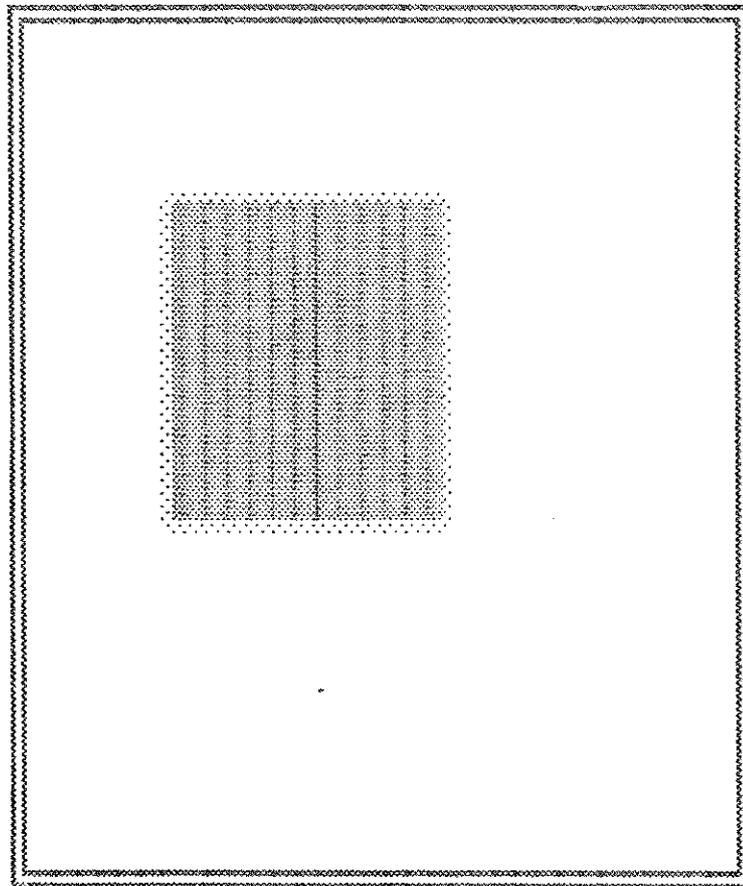


Figura 4.1 - Objeto 1: *quadrado*.

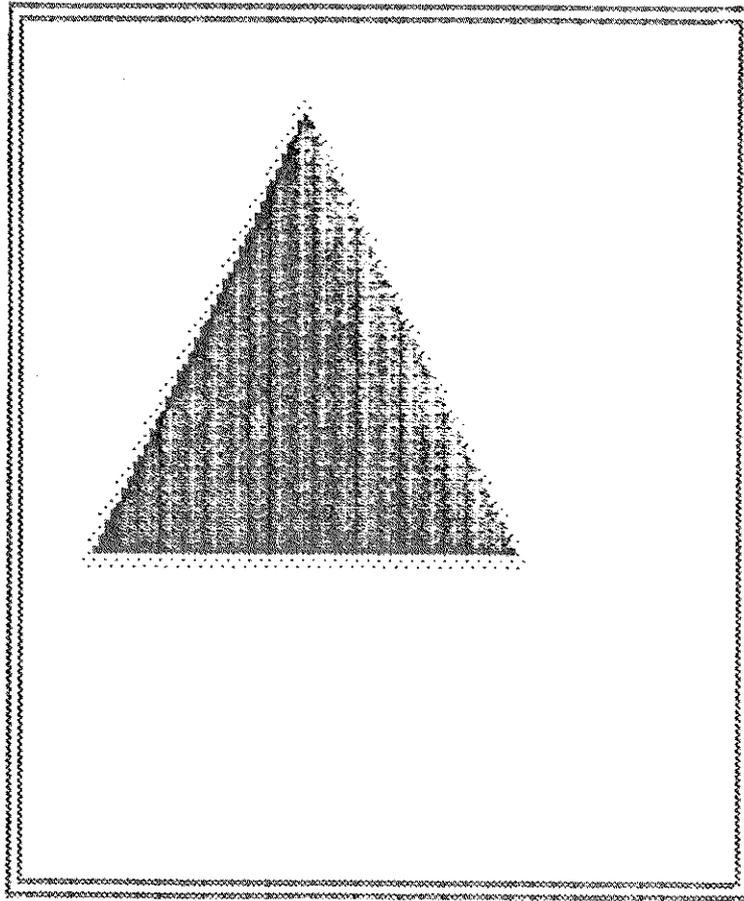


Figura 4.2 - Objeto 2 : *triângulo*.

Estes exemplares são tomados como referência, ou seja, são representantes das suas respectivas classes. A orientação e escala dos objetos reconhecidos são determinados com relação a estes modelos de referência.

#### 4.1 - RECONHECENDO E LOCALIZANDO UM OBJETO

A seguinte cena foi apresentada ao sistema:

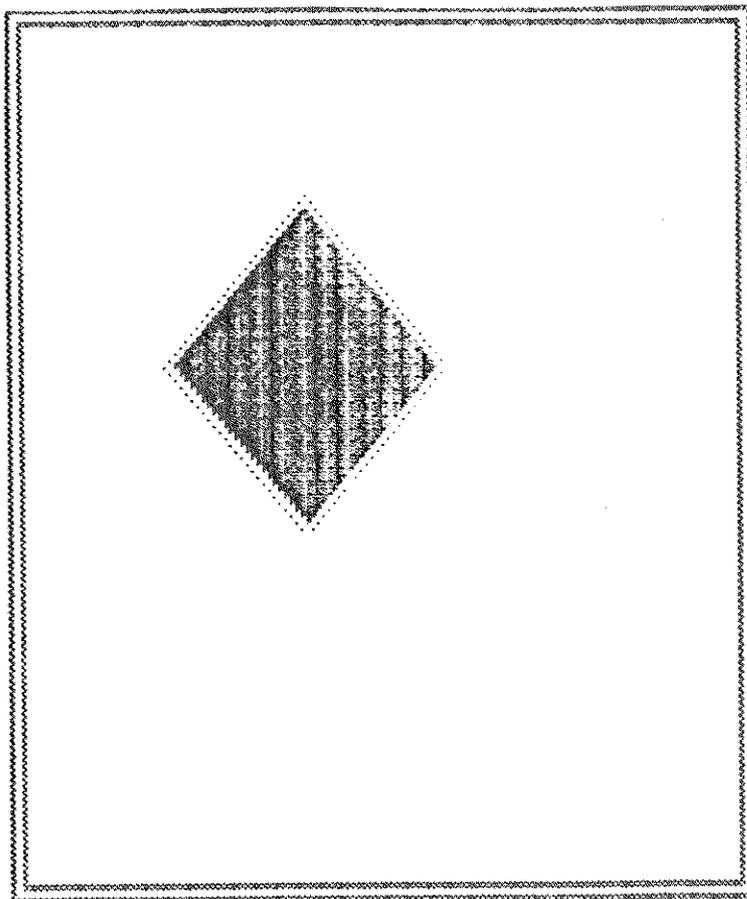


Figura 4.3 - Cena 1: Um objeto isolado.

O objeto presente na imagem é um "quadrado" e está inclinado de um ângulo de 45 graus com relação ao modelo de referência.

A resposta obtida do sistema é apresentada a seguir:

Objeto: quadrado

Escala: 0.66838199

Posição: 50.37117096 : 50.3722037

Orientação: 0.86095495 ( 49.3 graus )

Apesar das distorções causadas pelo fato de tal figura ter sido obtida de um monitor de vídeo de microcomputador, pode-se observar que a localização, a escala e a orientação do objeto, dentro de uma margem de tolerância estão corretos. Note-se que, sendo o objeto "quadrado" simétrico segundo dois eixos, por não ter havido qualquer tratamento para tais casos, a orientação obtida como resposta poderia ter sido  $\pm 45$  ou  $\pm 135$  graus.

#### 4.2 - INTERPRETANDO UMA CENA SEM OBJETOS SOBREPOSTOS

A seguinte cena foi apresentada ao sistema:

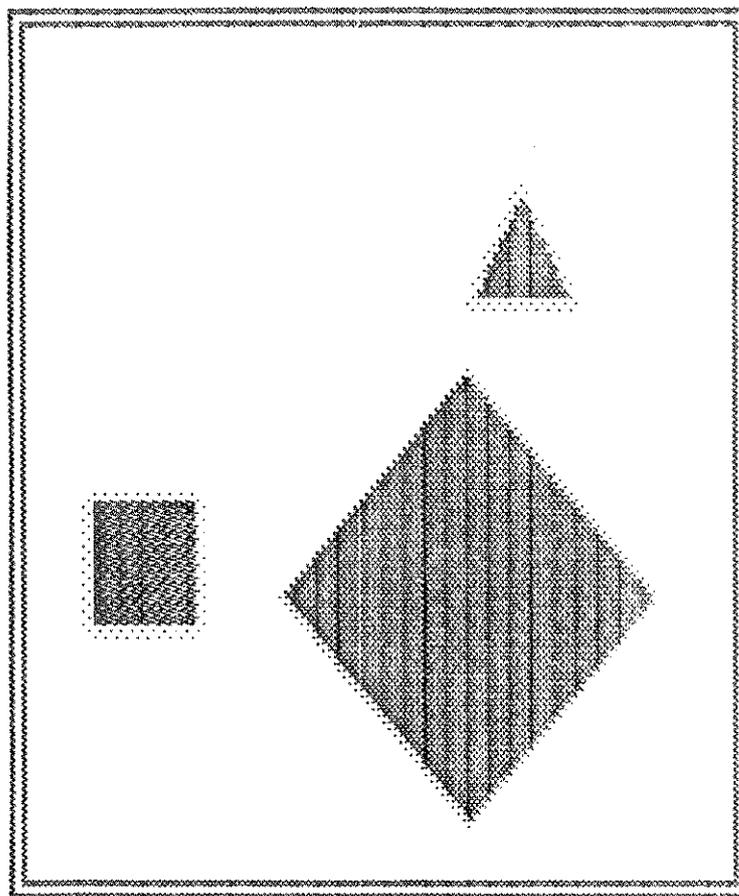


Figura 4.4 - Cena 2: Cena com objetos não sobrepostos.

A cena é composta por três objetos, dois "quadrados" e um "triângulo". A resposta obtida do sistema foi a seguinte:

Objeto: quadrado

Escala: 0.96068906

Posição: 80.9865433 : 85.48374668

Orientação: -0.72683202 ( - 41.6 graus )

Objeto: triângulo

Escala: 0.18278604

Posição: 90.5747005 : 39.77839778

Orientação: -2.08365956 ( - 119.4 graus )

Objeto: quadrado

Escala: 0.37827923

Posição: 21.26563616 : 80.32494074

Orientação: - 0.01559634 ( - 0.9 graus )

Novamente, observa-se que, considerada uma margem de tolerância, todos os objetos presentes na cena foram identificados e localizados corretamente. Entretanto, neste caso, nota-se que o "triângulo" e um dos "quadrados" foram localizados com a orientação de um de seus eixos simétricos.

#### 4.3 - RECONHECENDO OBJETOS PARCIALMENTE VISÍVEIS

A seguinte cena foi apresentada ao sistema:

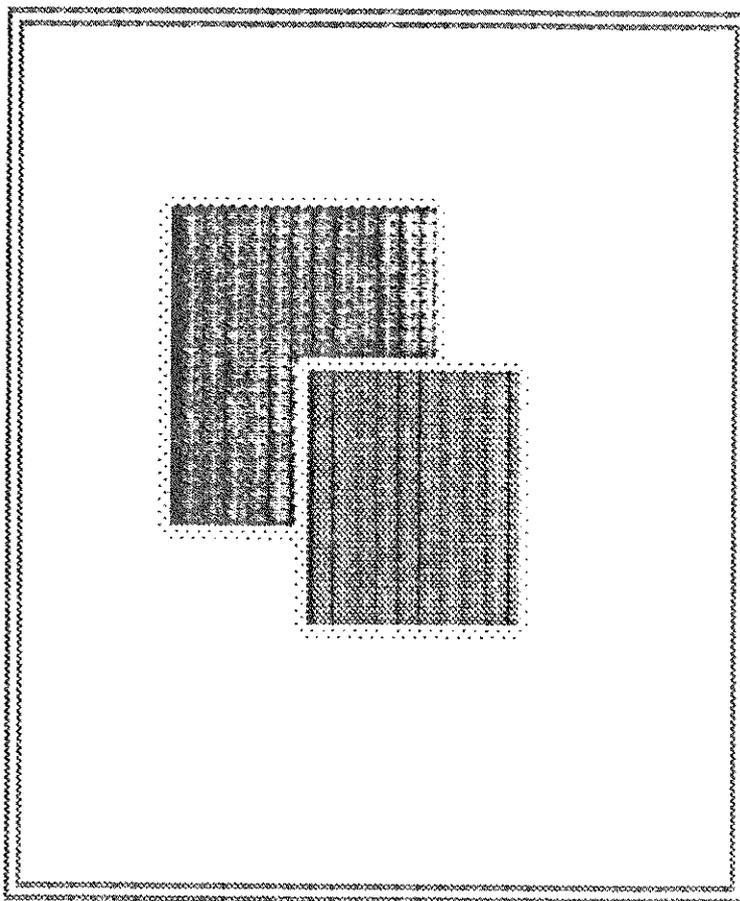


Figura 4.5 - Cena 3: Cena com objetos parcialmente visíveis.

A cena é composta por dois objetos "quadrados" sendo que um deles é apenas parcialmente visível.

A resposta obtida do sistema foi a seguinte:

Objeto: quadrado

Obstrui: quadrado

Escala: 0.79404383

Posição: 71.67015534 : 70.09872766

Orientação: - 1.57079633 ( - 90.4 graus )

Objeto: quadrado

Obstruido\_por: quadrado

Escala: 1.0

Posição: 51.28308677 : 50.65413368

Orientação: 0.0

Mais uma vez, considerada uma margem de tolerância, a resposta obtida pelo sistema está correta. Os objetos foram corretamente identificados, localizados e a relação de sobreposição existente foi determinada.

#### 4.4 - PROBLEMAS E DIFICULDADES ENCONTRADOS

Embora, para os exemplos mais simples, apresentados nas seções anteriores, o sistema tenha apresentado um desempenho satisfatório, este, ainda, está sujeito a imperfeições que dificultam sua aplicação para cenas mais complexas. Estas referem-se, essencialmente, ao problema de detecção de elementos primitivos (segmentos e ângulos) que determinam a forma dos objetos.

A figura 4.6 ilustra uma imagem segmentada. Note que o algoritmo de afinamento tende a causar uma distorção em pontos "críticos" para a análise da imagem, como o encontro de dois segmentos (junções T) e os cantos (ângulos). Este tende a buscar o eixo simétrico da borda (considerada como uma região alongada) e nem sempre realça ou mantém tais características intactas.

Outro fator causador de imperfeições é o próprio algoritmo de aproximação poligonal utilizado (seção 2.2.1.a.2). Este é um algoritmo de agrupamento e adota um limite de tolerância para a aproximação de cada segmento. Considerando, ainda, que ele atua sobre a imagem segmentada (com a borda afinada), uma vez que a definição é fixa (128 x 128 pixels), a saída de tal processo estará tanto mais sujeita a erros quanto maior os detalhes dos objetos na cena. A figura 4.7 ilustra os segmentos de reta encontrados pelo processo de aproximação poligonal. Note que os segmentos obtidos podem ser significativamente diferentes do que se esperaria em pontos estratégicos da imagem a ser analisada.

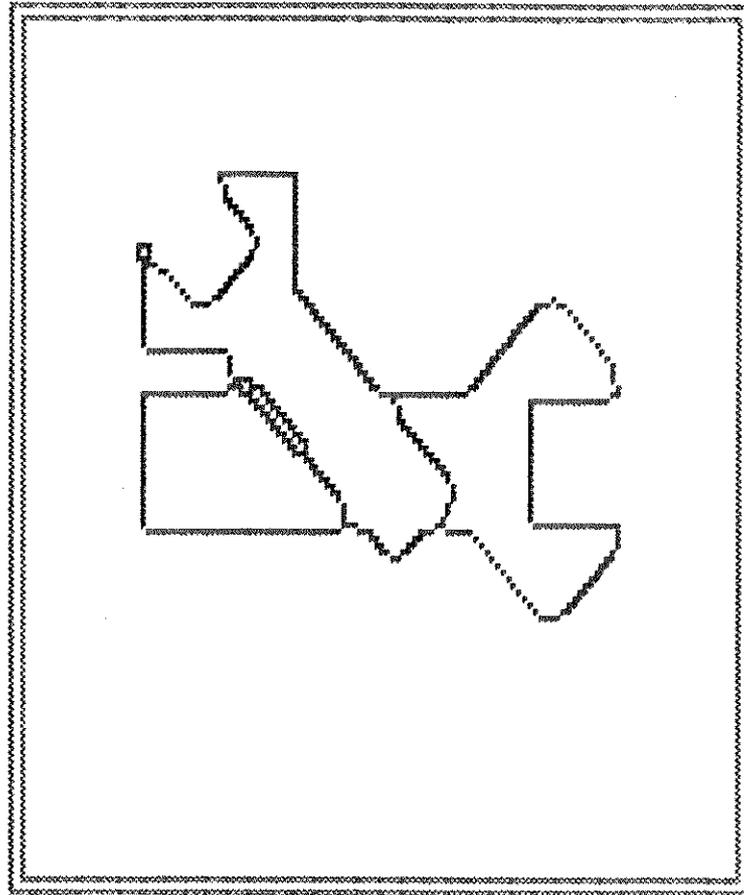


Figura 4.6 - *Distorção na imagem segmentada.*

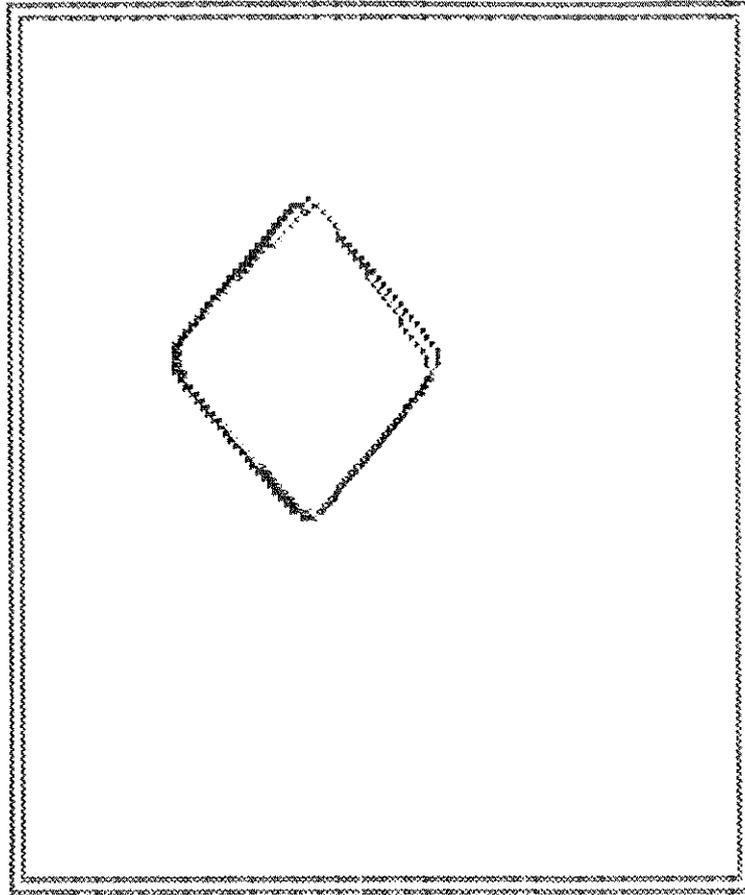


Figura 4. 7 - Erro na aproximação poligonal.

## CAPÍTULO 5

### DISCUSSÃO E CONCLUSÃO

#### 5.1 - ANÁLISE

No capítulo anterior, apresentou-se os resultados obtidos pelo sistema ao interpretar cenas em que os objetos podiam estar visíveis ou apenas parcialmente visíveis. Para cenas simples, o sistema foi capaz de reconhecer, determinar a escala e localizar objetos, ainda que estes estivessem apenas parcialmente visíveis. Assim, o sistema, como projetado e implementado, é capaz de desempenhar as funções a que se destina. Entretanto, algumas dificuldades, também apresentadas no capítulo anterior, ainda impedem o sistema de obter um melhor desempenho para cenas mais complexas.

Para que o sistema possa ter sua capacidade estendida, visando não só superar as dificuldades presentes, mas também aquelas que se espera encontrar ao lidar com cenas reais, os seguintes aspectos devem ser considerados:

- 1) As características locais e o processo utilizado para detectá-los.
- 2) A estratégia de reconhecimento e os algoritmos utilizados.
- 3) As restrições impostas ao domínio de aplicação e forma de aquisição da imagem.

Como abordado no capítulo três, devido ao fato dos objetos não serem inteiramente visíveis, a estratégia de reconhecimento baseia-se na utilização de características locais. Tal estratégia é encontrada em praticamente todos os sistemas que trabalham com este tipo de problema. Por exemplo, Bolles e Cain [BOLLES-83] propuseram o método de foco em características locais considerando os furos e os ângulos (canto) no contorno do objeto como características locais, Turney et al. [TURNNEY-85] introduziram o conceito de característica saliente, que é um sub-padrão do objeto (ex: uma parte de seu contorno) capaz de distingui-lo de outros objetos, Perkins [PERKINS-78] considerou as concurvas, um conjunto de curvas (segmentos de reta e arcos circulares) derivados do processo de detecção de bordas e Bhanu e Ming [BHANU-87] trabalharam com os ângulos formados por segmentos de reta consecutivos numa aproximação poligonal. No presente sistema, utilizam-se as únicas características locais disponíveis para os objetos do domínio de aplicação considerado, ou seja, os ângulos e segmentos de reta de seus contornos. As dificuldades expostas no capítulo quatro ilustram o tipo de problema a que está sujeito um sistema nesta fase. Assim, a detecção deve ser tal que

as características locais sejam extraídas de forma a refletir a estrutura do objeto o mais fielmente possível.

Para estabelecer a correspondência com o modelo (reconhecimento), utiliza-se um algoritmo grafo-teórico (maximal-clique), sendo que as restrições impostas por tal correspondência determinam a escala, a posição e a orientação dos objetos identificados. Embora, a complexidade do caso médio seja, em geral, de maior interesse, existindo algoritmos com tempo proporcional a  $N^3$  [ULLMAN-76], onde  $N$  é o número de nós no grafo, no pior caso, tal algoritmo depara com o problema da "explosão combinatorial", sendo o tempo necessário para resolver o problema proporcional a uma função exponencial do comprimento da entrada. Ele é classificado como um problema NP-completo [BALLARD-82]. Para amenizar tal problema, vários tipos de "heurísticas" podem ser utilizadas [ROSENFELD-82], como por exemplo, considerar apenas a correspondência entre pares de nós que sejam do mesmo tipo (ângulo ou segmento). A consideração das junções  $T$  também contribui para amenizar este problema de complexidade computacional, uma vez que ele divide o grafo-imagem em sub-grafos, reduzindo, assim, o tamanho do grafo de entrada do algoritmo. Contudo, tal algoritmo é um tanto quanto ingênuo e apesar da utilização de "heurísticas" como acima citado, por vezes, o esforço computacional dispendido pode tornar-se excessivo, o que é um fator limitante quando se pensa em aplicações reais.

Quanto às restrições impostas ao domínio de aplicação e forma de aquisição da imagem, obviamente, implicam em facilidades no projeto e implementação de um sistema. Por exemplo, a consideração das junções  $T$  só é possível aqui, devido ao fato de os objetos possuírem formas poligonais e serem caracterizados exclusivamente pelo seu contorno. Por outro lado, tais restrições também implicam numa especialização do sistema, tornando-o efetivo apenas para o domínio considerado e sob as condições de aquisição de imagens oferecidas.

## 5.2 - SUGESTÕES PARA POSTERIOR DESENVOLVIMENTO

Considerando os aspectos do sistema, acima discutidos, o caminho natural na continuação do presente trabalho está em procurar, para cada um dos problemas levantados, soluções capazes de suplantar as dificuldades atualmente encontradas.

Dois caminhos de investigação são sugeridos:

- 1) Utilização de imagens binárias e objetos representados por aproximação poligonal.
- 2) Utilização de imagens graduadas em níveis de cinza com outras características além de contorno de forma poligonal.

Ao considerar imagens binárias, cenas contendo objetos sobrepostos originam o que se pode chamar um "objeto aparente" (fig. 5.1). Neste caso, não é possível determinar as relações de sobreposição. Considerando, então, objetos com contornos descritos por poligonais (apesar de se poder aproximar objetos com qualquer tipo de contorno por polígonos, é desejável que o domínio de aplicação considerado seja constituído por objetos com contornos de forma poligonal), pode-se utilizar outros algoritmos de aproximação poligonal, além do utilizado neste trabalho. Por exemplo, algoritmos baseados na estratégia de explicitação e agrupamento poderiam fornecer resultados mais satisfatórios (vide seção 2.2.1.a.2).



Figura 5.1 - Objeto aparente de dois retângulos sobrepostos.

Dentro desta linha, ainda, com o objetivo de suplantar as dificuldades de esforço computacional encontradas com o algoritmo "ingênuo" de casamento de grafos, propõe-se a utilização de uma estratégia baseada em processos cooperativos ou de relaxação [BALLARD-82]. Este é um mecanismo computacional que permite a um conjunto de processos locais, cada um deles associado a uma parte do problema, interagir entre si, buscando uma solução globalmente consistente.

Considerando, agora, o segundo caminho de investigação, acima sugerido, propõe-se a utilização da transformação de Hough generalizada [BALLARD-82, DAVIS-82 e CASASANT-87] como detector de características locais. Este tipo de transformação é uma modificação da transformação de Hough exposta na seção 2.1.2.2.b e detecta formas arbitrárias, mesmo aquelas não descritas por forma paramétrica. Como no caso anterior, seria interessante a utilização de uma estratégia de reconhecimento baseada em processos de relaxação. Tal combinação poderia levar a um sistema com características interessantes (fig. 5.2). Uma delas

seria a natureza paralela que o sistema adquiriria, uma vez que ambos os algoritmos envolvidos são favoráveis para tal. A outra seria a possibilidade de lidar com medidas de incerteza, tornando o sistema mais flexível ao lidar com imprecisões.

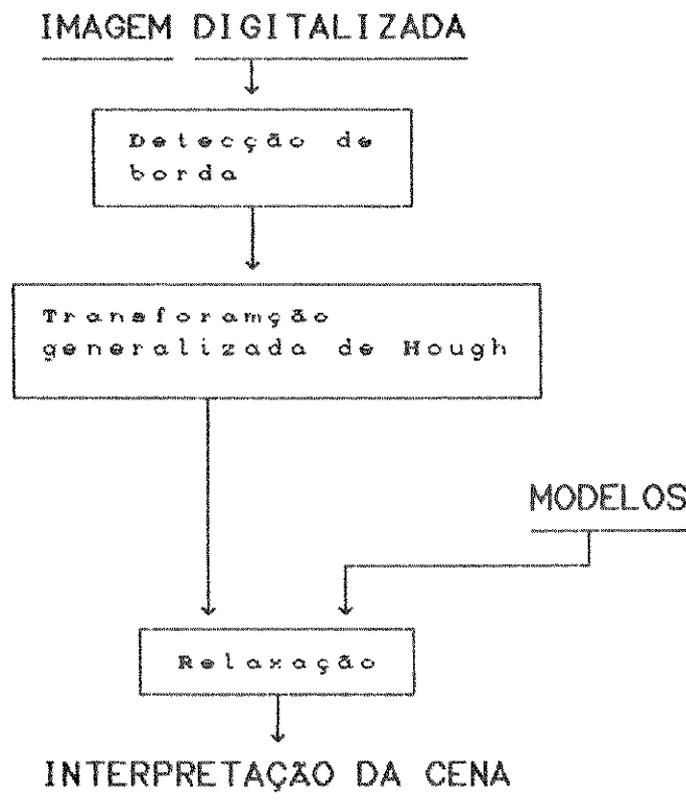


Figura 5.2 - Sistema utilizando a transformação generalizada de Hough e processos de relaxação.

### 5.3 - CONSIDERAÇÕES FINAIS

O presente trabalho visou o estudo do problema de "Bin-picking". Para isso, a partir de um sistema de visão computacional, de uma forma geral, considerou-se as particularidades de tal problema e implementou-se um sistema capaz de reconhecer, localizar e determinar as relações de sobreposição entre os objetos de um mundo restrito em cenas onde eles se dispunham de uma forma aleatória, podendo uns se sobreporem a outros. Este foi implementado utilizando-se as linguagens C e Prolog e encontra-se disponível no laboratório de Computação e

O sistema foi capaz de desempenhar as funções para as quais foi projetado e os resultados obtidos apresentados no capítulo quatro. Também foram apresentadas as dificuldades encontradas no desenvolvimento do trabalho e que, ainda, impõe limitações à atual implementação do sistema. Neste capítulo, discutiu-se o sistema como implementado e identificou-se os principais problemas encontrados ao lidar com objetos parcialmente visíveis. Também foram sugeridas algumas possibilidades de investigação, visando dar ao sistema uma maior flexibilidade e capacidade de lidar com imagens mais complexas e obtidas de uma cena real. Assim, uma vez que o objetivo inicial deste trabalho era o de identificar as dificuldades envolvidas ao abordar tal problema, considera-se que os objetivos foram alcançados.

Finalizando, não se pode deixar de colocar que o problema de visão computacional, na sua essência, é um problema tri-dimensional (3-D). Embora, este trabalho tenha se restringido a um mundo bi-dimensional (2-D), note-se que para um mundo 3-D, apesar de uma quantidade muito maior de informações terem de ser utilizadas, uma vez que, a partir de um ponto de observação, o que se vê é apenas parte da superfície do objeto, o reconhecimento de objetos no mundo 3-D deve se dar a partir de informações parciais, pois este é apenas parcialmente visível.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [AMBLER-75] - Ambler, A. P., H. G. Barrow, C. M. Brown, R. M. Burstall e R. J. Popplestone - "A versatile system for computer-controlled assembly" - Artificial Intelligence Vol. 6 No 2- 1975 - pp. 129-156.
- [BALLARD-81] - Ballard, D. H. - "Generalizing the Hough Transform to detect arbitrary shapes" - Pattern Recognition Vol. 13 No. 2 - 1981 - pp. 111-122.
- [BALLARD-82] - Ballard, D. H. e C. M. Brown - "Computer Vision" - Prentice Hall - 1982.
- [BERZTISS-73] - Berztiss, A. T. - "A backtrack procedure for isomorphism of directed graphs" - Journal of the ACM Vol 20 No 3 - 1973 -pp 365-277.
- [BHANU-87] - Bahnu, B. e J. C. Ming - "Recognition of occluded objects: a cluster structure algorithm" - Pattern Recognition Vol. 20 No. 2 - 1987 - pp. 199-211.
- [BOLLES-83] - Bolles, R. C. e R. A. Cain - "Recognising and locating partially visible objects: the local-feature-focus method" - em Robot Vision - editado por Alan Pugh - Springer Verlag - 1983.
- [BRON-73] - Bron, C. e J. Kerbosch - "Algorithm 457 - Finding all cliques of an undirected graph [H]" - Communications of the ACM Vol. 16 No 9 - 1973 - pp. 575-577.
- [CASASENT-87] - Casasent, D. e R. Krishnapuram - "Curved object location by Hough Transformations and inversions" - Pattern Recognition Vol. 20 No. 2 - 1987 - PP. 181-188.
- [CHIN-86] - Chin, R. T. e C. R. Dyer - "Model-Based Recognition in Robot Vision" - Computing Surveys vol. 18 no 1 - 1986 - pp. 67-108.
- [DAVIS-75] - Davis, L. S. - "A survey of edge detection techniques" - Computer Vision Graphics and Image Processing Vol. 4 - 1975 - pp. 248-270.
- [DAVIS-82] - Davis, L. S. - "Hierarchical Generalized Hough Transforms and Line-Segment Based Generalized Hough Transforms" - Pattern Recognition Vol. 15 No. 4 - 1982 - pp. 277-285.

- [DUDA-72] - Duda, R. O. e P. E. Hart - "Use of the Hough Transformation To Detect Lines and Curves in Pictures" - Communications of the ACM Vol. 15 No. 1 - 1972 - pp. 11-15.
- [DUDA-73] - Duda, R. e P. E. Hart - "Pattern Classification and Scene Analysis" - Willey and Sons - 1973.
- [FU-81] - Fu, K. S. e J. K. Mui - "A survey on image segmentation" - Pattern Recognition vol. 13 - 1981 - pp. 3-16.
- [GONZALEZ-78] - Gonzalez, R. C. e M. Thomason - "Suntatic Pattern Recognition" - Addison Wesley - 1978.
- [GONZALEZ-82] - Gonzalez, R. C. e R. Safabakhsh - "Computer vision techniques for Industrial Applications and Robot Vision" - Computer - Vol. 15 No 12 - 1982 - pp. 17-32.
- [GONZALEZ-87] - Gonzalez, R. C. e P. Wintz - "Digital Image Processing" - Second Edition - Addison Wesley - 1987.
- [GOVINDAN-87] - Govindan, V. K. e A. P. Shivaprasad - "A pattern adaptative thinning algorithm" - Pattern Recognition Vol. 20 No. 6 - 1987 - pp. 623-637.
- [HOROWITZ-76] - Horowitz, S. L. e T. Plavidis - "Picture segmentation by a tree traversal algorithm" - Journal of de ACM Vol. 23 No. 2 - 1976 - pp. 368-388.
- [HUECKEL-71] - Hueckel, M. H. - "An operator which locates edges in digitized pictures" - Journal of the ACM Vol. 18 No. 1 - 1971 - pp.113-125.
- [HUECKEL-73] - Hueckel, M. H. - "A local visual operator which recognizes edges and lines" - Journal of the ACM Vol. 20 No. 4 - 1973 - pp. 634-647.
- [ILLINGWORTH-88] - Illingworth, J. e J. Kittler - "A survey of the Hough Transform" - Computer Vision Graphics and Image Processing No 9 - 1988 - pp. 87-116.
- [KIMME-75] - Kimme, C., D. Ballard e J. Sklansky - "Finding circles by an Array Accumulators" - Communications of ACM Vol. 18 Nro. 18 - 1975 - pp. 120-122.
- [MARTELLI-72] - Martelli, A. - "Edge detection using heuristic search methods" - Computer Graphics and Image Processing Vol. 1 - 1972 - pp.169-182.
- [MARTELLI-76] - Martelli, A. - "An application of heuristic search methods to edge and contour detection" - Communications of the ACM Vol. 19 No. 2 - 1976 - pp. 73-83.

- [NACCACHE-84] - Naccache, N. J. e R. Shinghai - "SPTA: A proposed algorithm for thinning binary patterns" - IEEE Transactions on systems, man and cybernetics Vol. SMC-14 No 3 - 1984 - pp. 409-418.
- [PERKINS-85] - Turney, J. L., T. N. Mudge e R. A. Volz - "Recognizing partially occluded parts" - IEEE transactions on Pattern Analysis and Machine Intelligence Vol. PAMI-7 No. 4 - 1985 - pp.410-421.
- [PLAVIDIS-77] - Plavidis, T. - "Structural pattern recognition" Springer New York - 1977.
- [PUGH-83] - Pugh, A. - "Second generation Robotics" - em Robot Vision - editado por Alan Pugh - Springer Verlag - 1983.
- [REINGOLD-77] - Reingold, E. D., J. Nievergelt e N. Deo - "Combinatorial Algorithms - Theory and Practice" - Prentice Hall - 1977.
- [ROSEN-79] - Rosen, C. A. - "Machine Vision and Robotics" - em Computer Vision and Sensor-Based Robots - editado por Dodd, G. G. e Rossol, L. - Plenum Press - 1979.
- [ROSENFELD-82] - Rosenfeld, A. e A. C. Kak - "Digital picture processing" - second edition vols 1 e 2 - Academic press inc. - 1982.
- [ROBINSON-77] - Robinson, G. S. - "Edge detection by compass gradient masks" - Computer graphics and image processing Vol. 6 - 1977 - pp. 492-501.
- [STOCKMAN-77] - Stockman, G. C. e A. K. Agrawala - "Equivalence of Hough Curve detection to template matching" - Communications of the ACM Vol. 20 No. 11 - 1977 - pp. 820-822.
- [SUZUKI-87] - Suzuki, S. - "Binary picture thinning by an iterative parallel two-subcycle operation" - Pattern Recognition Vol. 20 No. 3 - 1987 - pp.297-307.
- [ULLMAN-76] - Ullman, J. R. - "An algorithm for subgraph isomorphism" - Journal of the ACM Vol. 23 No 1 - 1976 - pp. 31-42.
- [WESZKA-74] - Weszka, J. S., R. N. Nagel e A. Rosenfeld - "A threshold selection technique" - IEEE transactions on computers - Vol. C-23 - 1974 - pp. 1322-1326.
- [WESZKA-78] - Weszka, J. S. - "A survey of threshold selection techniques" - Computer graphics and image processing Vol. 7 - 1978 - pp. 259-265.

- [WINSTON-75] - Winston, P. H. - "Learning structural descriptions from examples" - em Psychology of Computer Vision - editado por Patrick H. Winston - McGraw-Hill - 1975.
- [YASNOFF-77] - Yasnoff, W. A., J. K. Mui e J. W. Bacus - "Error measures for scene segmentation" - Pattern Recognition vol. 9 - 1977 - pp. 217-231.
- [ZHANG-84] - Zhang, T. Y. e C. Y. Suen - "A fast parallel algorithm for thinning digital patterns" - Communications of the ACM Vol. 27 No. 3 - 1984 - pp. 236-239.
- [ZUCKER-76] - Zucker, S. W. - "Region growing: childhood and adolescence" - Computer graphics and image processing Vol. 5 - 1976 - pp. 382-399.