



UNIVERSIDADE FEDERAL DA PARAÍBA - UFPB
Centro de Ciências e Tecnologia - CCT
Departamento de Sistemas e Computação - DSC
Coordenação de Pós-Graduação em Informática - COPIN

**IDENTIFICAÇÃO DE VÉRTICES EM IMAGENS
MULTI-VISTA DE FORMAS 3D.**

Vânia Cordeiro da Silva

Campina Grande, julho de 1999

Vânia Cordeiro da Silva

IDENTIFICAÇÃO DE VÉRTICES EM IMAGENS

MULTI-VISTA DE FORMAS 3D.

Dissertação submetida ao corpo docente da Coordenação do Curso de Mestrado em Informática da Universidade Federal da Paraíba – Campus II como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Orientador: João Marques de Carvalho, Ph. D.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Matemática Computacional e Computação Gráfica

Campina Grande, julho de 1999



5586i Silva, Vania Cordeiro da
Identificacao de vertices em imagens multi-vista de
formas 3D / Vania Cordeiro da Silva. - Campina Grande,
1999.
71 f.

Dissertacao (Mestrado em Informatica) - Universidade
Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Processamento Digital de Imagens 2. Modelagem
Poliedral 3. Deteccao de Vertices 4. Reconhecimento de
Padroes 5. Visao Computacional I. Carvalho, Joao Marques de
II. Universidade Federal da Paraiba - Campina Grande (PB)

CDU 004.383.5(043)

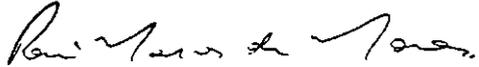
**IDENTIFICAÇÃO DE VÉRTICES EM IMAGENS MULTI-VISTA DE
FORMAS 3D**

VÂNIA CORDEIRO DA SILVA

DISSERTAÇÃO APROVADA EM 09.07.1999


PROF. JOÃO MARQUES DE CARVALHO, Ph.D
Orientador

M. Fátima S. Farias
PROFª MARIA DE FÁTIMA SANTOS FARIAS, D.Sc
Examinadora


PROF. RONEI MARCOS DE MORAIS, Dr.
Examinador


PROF. ELMAR UWE KURT MELCHER, Dr.
Examinador

CAMPINA GRANDE – PB

À minha fonte de inspiração, meu esposo Álvaro Vinícius.

Agradecimentos

Agradeço, primeiramente à Deus, por poder concluir mais uma etapa da minha caminhada.

Agradeço também as seguintes pessoas sem as quais este trabalho não seria possível: ao meu esposo, Álvaro, por suas sugestões e apoio, por entender e suportar minha ausência, e a minha família, pela ajuda inestimável. Ao professor João Marques pela valiosa orientação, dedicação e profissionalismo, que são suas características.

Ao professor Elmar pelas contribuições.

Meus sinceros agradecimentos aos colegas do curso de Computação, aos colegas do grupo de Processamento de Imagens coordenado pelo professor João Marques, em particular a Luciana e Josemar.

Agradeço a todos aqueles que contribuíram direta ou indiretamente para a elaboração desta dissertação, cujos nomes não constam desta minúscula lista.

Finalmente agradeço a todos os amigos integrantes da lista mundica-l pelo companheirismo mesmo a distância, e pela ajuda em todos os momentos em que fora necessário aliviar o stress.

Resumo

O objetivo desta dissertação é o desenvolvimento de um método para identificação de vértices tridimensionais em imagens multi-vista de cenas compostas de um ou mais objetos. Esta identificação permite a definição de modelos poliedrais dos objetos, que podem ser utilizados para reconhecimento 3D. Os algoritmos desenvolvidos operam por fluxo de dados, de modo a permitir que todo o processo de modelagem possa ser executado em tempo real. O método trabalha com três imagens reais de uma cena, obtidas de três pontos de vista diferentes, também conhecidas como imagens multi-vistas: superior, frontal e lateral. O método reconhece não só o contorno externo do objeto, como também suas arestas internas.

Abstract

The objective of this dissertation is the development of a method for identification of 3D vertices for multi-view images of scene composed by one or more objects. This identification allows the definition of polyhedral object models, which can be used for 3D recognition. The developed algorithms operate by data-flow, thus enabling the whole modelling process to be performed in real-time. The method works with three real images of a scene, taken from three different points of view, also known as multi-view images: top, side and front views. The method recognizes not only the external contour of the object, but also its internal edges.

Sumário

1.	Introdução Geral	1
1.1.	Sistemas de Visão	3
1.2.	Caracterização do Problema	5
1.3.	Objetivo e Organização da Dissertação	9
2.	Reconhecimento Através de Modelos.....	12
2.1.	Características dos Sistemas de Visão Baseados no Casamento de Modelos.....	13
2.2.	Construção dos Modelos.....	14
2.2.1.	Construção de Modelos a Partir do Próprio Objeto.....	15
2.2.2.	Construção de Modelos por Sistemas CAD	17
2.3.	Extração das Características	20
2.3.1.	Aquisição das Imagens	20
2.3.2.	Descrição das Imagens	22
2.4.	Casamento de Características	24
3.	Pré-Processamento	27
3.1.	Número de Conectividade	28
3.2.	Gradiente Spline	29
3.3.	Binarização	30
3.4.	Esqueletização	31
3.4.1.	Algoritmo de Pré-processamento de Stentiford.....	32
3.4.2.	O Algoritmo de Zhang-Suen	35
3.4.3.	Pós-Processamento de Holt	36

4.	Geração e Extração de Vértices.....	38
4.1.	Geração dos Vértices Formados por Três Arestas.....	39
4.2.	Geração dos Vértices Formados por Duas Arestas.....	43
4.3.	Extração dos Vértices	47
4.4.	Correspondência dos Vértices nas Multi-Vistas.....	49
5.	Testes Efetuados e Resultados Obtidos.....	52
5.1.	As Imagens	52
5.2.	Os Resultados	55
6.	Conclusões, Contribuições e Trabalho Futuro	70

Lista de Figuras

1.1	Sistema de Visão para Inspeção Industrial.....	4
1.2	Diagrama do sistema de visão baseado no casamento de modelos.....	5
1.3.	Diagrama em blocos de sistema de reconhecimento de objetos	6
1.4.	Estrutura de aquisição das imagens multi-vistas para sistemas de visão 3D.....	8
1.5.	Diagrama em blocos do método desenvolvido	9
2.1.	Estrutura para a metodologia da triangulação.....	21
3.1.	Um exemplo de números de conectividade: a) conectividade = 1, b) conectividade = 2, c) conectividade = 3, d) conectividade = 4 e e) conectividade = 0.....	28
3.2.	Máscaras utilizadas para o cálculo do gradiente Spline	30
3.3.	Máscaras para afinamento.....	32
3.4.	Máscaras utilizadas na etapa de pré-processamento: ênfase do ângulo agudo.....	34
3.5.	Máscaras utilizadas por Holt para deletar pixels formadores de escadas.....	37
4.1.	Exemplos da perda do pixel de encontro das arestas.....	38
4.2.	Exemplos de vértices 3x3 formados por três arestas.....	39
4.3.	a) Imagem original, b) imagem esquelizada e c) imagem extraída os vértices 3x3, vértices falsos.....	40
4.4.	Exemplos das possíveis continuações das arestas.....	41
4.5.	Vértices de 3 arestas extraídos a) levando em conta o pixel central, b) sem levar em conta o pixel central	42

4.6. Padrões de vértices com pesos, a) peso 4: todas as arestas retas, b) peso 1: todas as arestas com efeito escadaria e c) peso 3: arestas mistas	43
4.7. A mesma imagem com a) vértices extraídos com seus respectivos pesos e b) depois do filtro local.....	44
4.8. Problema decorrente do aumento dos vértices.....	45
4.9. Vértices extraídos a) de tamanho 5x5 e b) de tamanho 7x7	46
4.10. Exemplos das direções permitidas para os vértices 7x7	47
4.11. Cálculo do ângulo entre as arestas.....	48
4.12. Correspondência de vértices nas multi-vistas	50
5.1 Cena 5 – vista superior: etapas do pré-processamento a) imagem original, b) imagem gradiente, c) imagem binarizada, d) imagem esqueletizada, e) vértices extraídos com seus pesos, f) vértices após filtro de máximo local 5x5	56
5.2. Cena 5 – vista lateral: etapas do pré-processamento a) imagem original, b) imagem gradiente, c) imagem binarizada, d) imagem esqueletizada, e) vértices extraídos com seus pesos, f) vértices após filtro de máximo local 5x5	59
5.3. Cena 5 – vista frontal: etapas do pré-processamento a) imagem original, b) imagem gradiente, c) imagem binarizada, d) imagem esqueletizada, e) vértices extraídos com seus pesos, f) vértices após filtro de máximo local 5x5	61
5.4. Cena 2 – vista superior: etapas do pré-processamento a) imagem original, b) imagem gradiente, c) imagem binarizada, d) imagem esqueletizada, e) vértices extraídos com seus pesos, f) vértices após filtro de máximo local 5x5	62
5.5. Cena 2 – vista lateral: etapas do pré-processamento a) imagem original, b) imagem gradiente, c) imagem binarizada, d) imagem esqueletizada,	

e) vértices extraídos com seus pesos, f) vértices após filtro de máximo local 5x5	64
5.6. Cena 2 – vista frontal: etapas do pré-processamento a) imagem original, b) imagem gradiente, c) imagem binarizada, d) imagem esqueletizada, e) vértices extraídos com seus pesos, f) vértices após filtro de máximo local 5x5	65
5.7. Vértices resultantes da correspondência para a cena 5: a) vista frontal, b) vista lateral, c) vista frontal.....	66
5.8. Vértices resultantes da correspondência para a cena 2: a) vista frontal, b) vista lateral, c) vista frontal.....	67
5.9 a) Vista lateral da Cena 5, b) Ampliação da região marcada por uma elipse.....	68
5.10 a) Vista lateral da Cena 2, b) Ampliação da região marcada por uma elipse.....	69

Lista de Tabelas

5.1. Imagens testadas com suas respectivas dimensões	53
5.2. Resumo dos resultados	54

Capítulo 1

Introdução Geral

É excitante a grande variedade de formas, cores e movimentos dos elementos físicos do nosso mundo, com os quais nós podemos interagir ou simplesmente contemplar. Ver uma rosa desabrochando, o sorriso de uma criança, o condor voando e outros espetáculos da natureza é, com certeza, um bálsamo para a alma humana. Contudo, além de um prazer, ver é uma necessidade pois “ouvir” o mundo, “tocar” o mundo, “cheirar” o mundo não nos transmite tanta informação e de forma tão rápida, quanto poder “ver” o mundo. A importância da visão foi sempre reconhecida pela humanidade, de forma mais ou menos explícita, desde tempos remotos. Napoleão dizia que “Uma imagem vale mais do que mil palavras”, já São Tomé dizia que tinha que se “Ver para crer”. Pode-se citar ainda os ditos populares [5]: “O que os olhos não vêem, o coração não sente”, ou, “Longe dos olhos, longe do coração”.

A visão humana é um sistema biológico complexo, por isso mesmo ainda não inteiramente compreendido pela ciência, que permite a captação da radiação eletromagnética em um espectro visível, convertendo-a em sinais neuronais e realizando operações de processamento de baixo, médio e até mesmo de alto nível, que ocorrem em partes diferentes do cérebro.

Uma fonte luminosa emite raios luminosos sobre superfícies de objetos, que são refletidos e percebidos pelo sistema visual. As células nervosas da retina transformam as informações como intensidade e comprimento de onda de luz em sinais nervosos remetidos ao cérebro pelo *nervo óptico*. Essa informação é armazenada e, uma vez processada, pode ser gerado algum tipo de resposta nervosa que será provavelmente transmitida para alguma parte ou órgão do corpo humano, como glândulas, pulmões, braços ou pernas por exemplo.

O sistema visual humano consegue assimilar um grande número de informações que se apresentam em uma infinidade de padrões diferentes a curtos intervalos de tempo. Seu processamento é muito eficiente, além de extremamente versátil. O olho humano consegue adaptar-se à uma grande diversidade de situações novas e até adversas, quanto a níveis de

intensidade luminosa.

A visão humana é, sem dúvida, a nossa principal fonte de obtenção de informação, por natureza, a mais rápida e completa [5].

A necessidade crescente de automatização de atividades quotidianas, sejam elas prejudiciais, repetitivas ou que exijam uma maior precisão, incentiva pesquisas e estudos para expandir ou criar novos campos. Com a nossa visão não poderia ser diferente: começamos com os óculos, depois binóculos, lunetas, telescópios, microscópios para, hoje, falarmos em visão por computador. A área de visão computacional ou visão artificial, apesar de não ser objeto recente de pesquisa, está em contínuo desenvolvimento.

A construção de máquinas que possam “ver” tem sido tema de pesquisa de grande interesse nas últimas três décadas. Apesar de ainda não se ter conseguido dotar uma máquina com a capacidade total do nosso complexo sistema de visão, muitos progressos já foram e estão sendo alcançados ao se construir máquinas capazes de realizar tarefas visuais específicas.

Podem ser citados vários exemplos da utilidade (aplicabilidade) da visão computacional. Em robótica podemos controlar um robô que precise escolher uma determinada peça entre muitas, encaixá-las ou simplesmente acertar um alvo. O efetivo controle de qualidade de um produto sem contato direto com este, tal como se faz com circuitos impressos, é um outro exemplo de tarefa de automação industrial que pode ser executada através da inspeção visual automatizada. Todavia não é apenas em tarefas industriais que se pode encontrar aplicação para visão computacional. Em equipamentos médicos, por exemplo, tem-se a análise de tomografias de ressonância magnética, ou ainda imagens microscópicas [30].

Como exemplos do uso de máquinas dotadas de visão computacional para o reconhecimento de padrões pode-se citar o reconhecimento de impressões digitais, assinaturas, voz, leitura de caracteres manuscritos ou não, palavras ou até mesmo textos e, mais recentemente, identificação pessoal através da leitura da íris humana. Uma máquina com capacidade de visão inteligente pode ser aplicada em diversas outras áreas tais como atividades militares e pesquisas espaciais para, por exemplo, teleguiar um míssil ou localizar alvos em imagens de satélites, bem como o uso de radares e imagens infra-vermelho [31].

1.1. Sistemas de Visão

Sistemas de visão necessitam de uma maneira de adquirir, processar e visualizar imagens. Um sistema genérico mínimo para o processamento de imagens é constituído de captadores (sensores) de visão, de um hardware de digitalização de imagens, uma memória para armazená-las, e um computador que execute os algoritmos de processamento de imagens.

Os objetos presentes nas imagens são captados através da irradiação de energia, seja por reflexão, transmissão ou emissão; todas inerentemente contínuas. A digitalização da imagem se faz necessária para que a imagem analógica possa ser processada por um computador digital.

A imagem digital nada mais é do que uma matriz de pontos, onde o valor de cada elemento desta matriz está associado ao valor do nível de cinza ou nível de brilho (em uma imagem preto e branco), ou ainda cor do ponto correspondente na grade discreta de amostragem da imagem. A amostragem, em geral, é seguida por uma quantização, para que os valores obtidos tenham uma precisão finita dependente tanto da aplicação quanto do equipamento utilizado. Os índices dos elementos da matriz correspondem às coordenadas espaciais dos pontos de amostragem. Estes elementos são usualmente conhecidos como pixels (*"picture elements"*). Em muitas aplicações, que trabalham com imagens monocromáticas, um único valor é medido em cada ponto amostrado, representando o valor do brilho daquele ponto na imagem em uma dada banda espectral. Outras aplicações requerem imagens multi-espectrais e, portanto, vários valores são medidos para cada ponto amostrado, representando a intensidade do brilho do ponto em um conjunto de bandas espectrais [33].

Deste modo, cada pixel será representado por uma k -upla desses valores. Em imagens tridimensionais o valor da matriz pode representar a profundidade da imagem para aquele índice, ou seja, representa a distancia entre o plano em que se encontra o sensor óptico utilizado para a aquisição da imagem e o objeto nas coordenadas correspondentes (índice da matriz).

As imagens utilizadas em sistemas de visão computacional são, em geral, obtidas dentro dos limites da faixa de energia visível do espectro eletromagnético. Assume-se que a

cena é iluminada por uma fonte de luz natural ou artificial, e que o sensor óptico incorporado ao sistema de visão computacional transforma qualquer imagem da cena em uma imagem plana, perpendicular ao campo plano de visão do sensor óptico. Não são detectados quaisquer objetos transparentes.

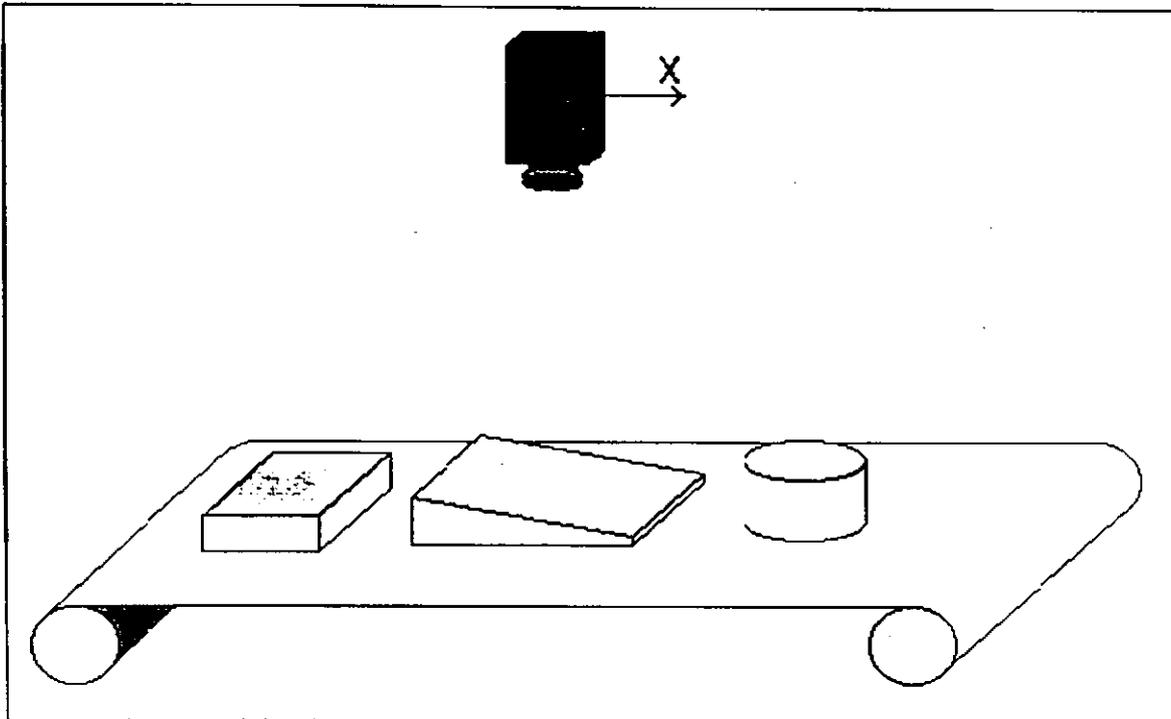


Figura 1.1. - Sistema de visão para Inspeção Industrial

O tratamento aplicado a uma imagem digital depende da aplicação a que ela se destina, variando de acordo com a natureza das imagens, natureza da aplicação, qualidade das imagens e o conhecimento prévio do meio em estudo. De uma maneira geral, um sistema de visão computacional tem que ser capaz de extrair de uma cena as informações necessárias à aplicação, através da análise da mesma. Também tem que ser capaz de identificar os objetos contidos na cena bem como as suas formas.

Com relação à área de automação industrial, que é uma área com grande potencial de aplicação para este tipo de sistemas, a visão computacional pode ser utilizada para o reconhecimento, quer seja de um objeto ou de partes dele. Este é o caso das tarefas de inspeção visual, como exemplificado na Figura 1.1, a qual mostra uma esteira rolante onde os objetos saídos de uma linha de produção tem suas formas inspecionadas por um sistema de visão computacional [3].

1.2. Caracterização do Problema

Diversos problemas de visão computacional podem ser resolvidos com a utilização de sistemas de visão que operem em duas dimensões apenas (2D). Outros exigem um sistema de visão mais sofisticado que trabalhe em três dimensões (3D). Uma grande parte de ambos os tipos de sistemas (2D e 3D), funciona tentando casar uma ou mais características da cena (imagem de entrada) com as características pertencentes a um conjunto de modelos pré-estabelecidos, criados durante a fase de treinamento do sistema. Estes sistemas de visão são conhecidos como *Sistemas de Reconhecimento Baseados no Casamento de Modelos (model based)*, e sua idéia encontra-se resumida em um diagrama de blocos apresentado na Figura 1.2. A principal vantagem desta classe de sistemas é a grande compressão de dados obtida através da representação dos objetos por modelos ou por um pequeno número de atributos (características). Este assunto será abordado com um nível maior de detalhamento no Capítulo 2.

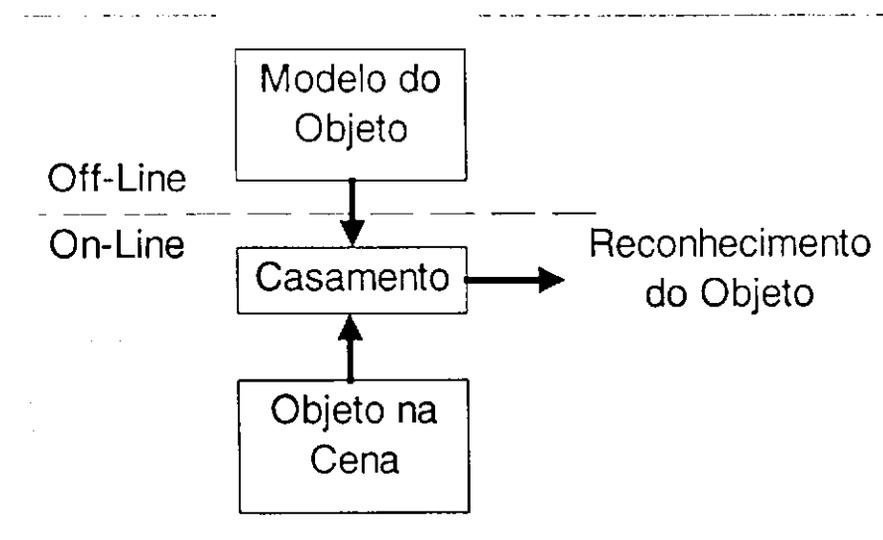


Figura 1.2. - Diagrama do Sistema de Visão baseado no Casamento de Modelos.

Uma das abordagens mais utilizadas em sistemas de visão para o reconhecimento de objetos em duas dimensões é baseada no casamento de modelos poligonais, onde as formas do objeto são representadas por polígonos. Uma maneira de se realizar uma modelagem

poligonal de um objeto bidimensional, ou de parte dele, é identificando o contorno da sua forma e neste, os pontos que serão os vértices de uma aproximação poligonal fechada para este objeto [2]. O contorno bidimensional de qualquer objeto contém grande quantidade de informação sobre a forma deste objeto, podendo portanto ser utilizado para o reconhecimento do mesmo. Conseqüentemente, modelos de contornos bidimensionais, desde que suficientemente precisos, também podem ser aplicados para a mesma finalidade, ou seja, em processos de identificação de formas ou objetos.

No caso do reconhecimento de objetos em 3D (três dimensões), duas ou mais imagens de diferentes pontos de vista do objeto podem ser obtidas, a partir das quais uma aproximação (modelo) polidral do mesmo será gerada.

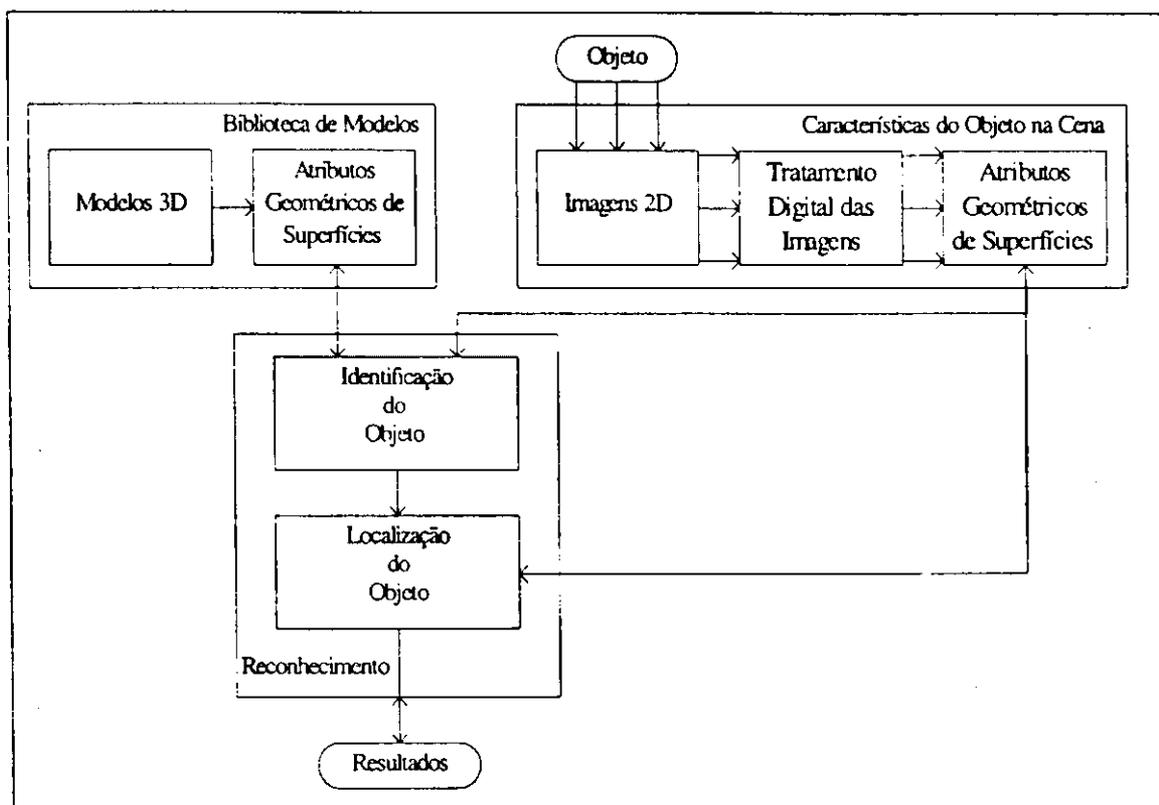


Figura 1.3. - Diagrama em Blocos de Sistema de Reconhecimento de Objetos 3D

As características geométricas extraídas deste modelo do objeto presente na cena podem então ser calculadas e comparadas com aquelas armazenadas em uma biblioteca de modelos previamente construída, de modo a permitir a identificação do mesmo. Um ponto de importância fundamental do processo de modelagem é a identificação correta dos vértices tridimensionais que definirão as superfícies das quais é formado o poliedro.

Diversos métodos de reconhecimento de objetos em cenas tridimensionais baseados em modelos foram e estão sendo desenvolvidos, sobretudo na década passada e na presente. Alguns destes trabalhos são descritos no Capítulo 2. Uma revisão bibliográfica mais completa desta área pode ser encontrada na tese de Farias [1].

O método de reconhecimento de objetos poliédricos para cenas tridimensionais que está apresentado no diagrama em blocos da Figura 1.3, é descrito por Farias [1]. Neste, atributos geométricos extraídos de modelos construídos a partir de três imagens bidimensionais de diferentes pontos de vista dos objetos (imagens multi-vistas) são casados com atributos geométricos encontrados em modelos 3D previamente construídos e armazenados. O método consiste de três etapas:

1. Construção da Biblioteca de Modelos;
2. Extração de Características do Objeto na Cena;
3. Casamento das Características (identificação e localização).

A biblioteca de modelos consiste de modelos poliédricos tridimensionais e de um respectivo conjunto de atributos geométricos de superfícies, correspondentes a cada um dos modelos dos objetos passíveis de reconhecimento. Na etapa de extração de características do objeto presente na cena, são captadas três imagens do objeto; uma frontal, outra lateral e uma última superior, correspondendo respectivamente aos planos YZ, XZ e XY, conforme estrutura exposta na Figura 1.4. Estas imagens passam por um processo de digitalização e extração de algumas características bidimensionais. Tais características são reunidas e um conjunto de atributos geométricos 3D correspondentes às superfícies visíveis do objeto é então elaborado [1].

Os modelos no trabalho de Farias [1], foram elaborados no ambiente tridimensional do sistema CAD – Auto CAD R32, como se estivessem em repouso sobre uma mesa. Mais informações sobre este tipo de modelagem podem ser encontradas no Capítulo 2.

Com as informações provenientes do conjunto de atributos da cena, o bloco de identificação faz uma busca na biblioteca de modelos, determinando qual modelo contém aqueles atributos, através do casamento de características. Nesta etapa o sistema identifica não apenas o modelo do objeto encontrado na cena, como também as superfícies do mesmo correspondentes às respectivas superfícies visíveis. Com as informações do modelo

identificado e da própria cena, o sistema determina a localização espacial do modelo no ambiente da cena [1].

O casamento de características, que é o reconhecimento propriamente dito, ocorre em duas fases: a geração de hipóteses e a verificação das hipóteses geradas. As hipóteses são geradas a partir de atributos geométricos locais e globais das superfícies e são verificadas (confirmadas ou não), a partir de atributos relacionais entre as superfícies.

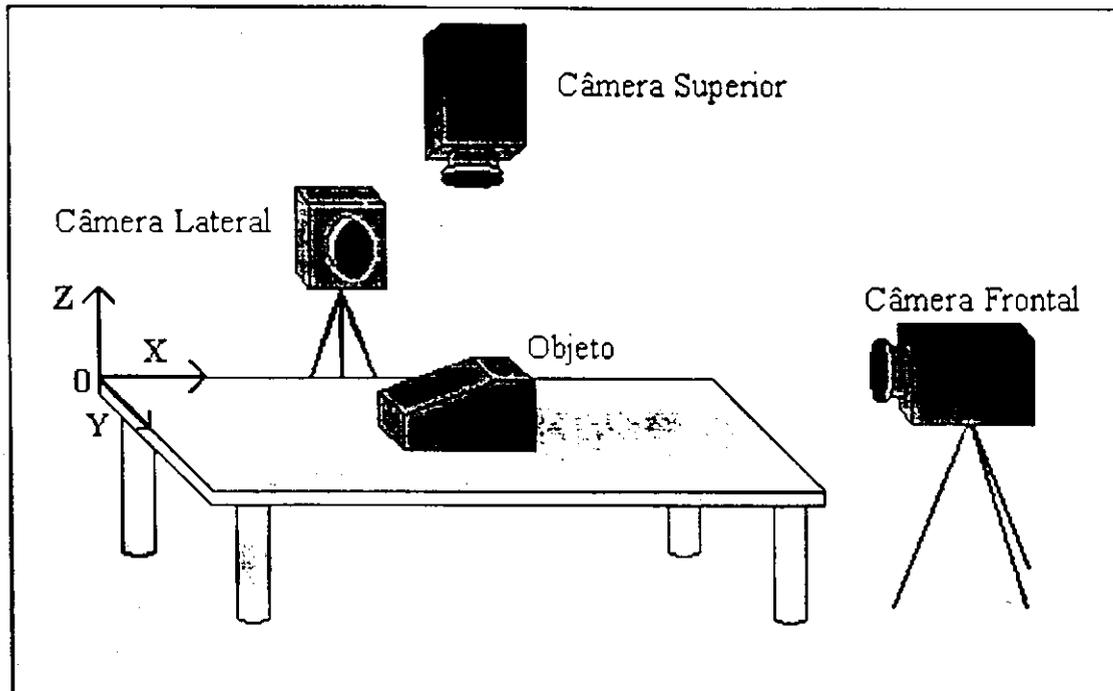


Figura 1.4. - Estrutura de Aquisição das Imagens Multi-vistas

Como exemplo das características extraídas das cenas, pode-se citar os pontos que compõem a lista dos vértices tridimensionais do objeto, que são necessários para se obter as outras características relevantes para o reconhecimento, como quantidade de vértices, área, perímetro, comprimento das arestas, centróide e normal unitária por exemplo. Conseqüentemente o processo de extração de vértices é de fundamental importância para a confiabilidade do sistema de visão.

1.3. Objetivos e Descrição do Sistema

Os algoritmos de extração de vértices utilizados em sistemas de visão bidimensionais geralmente encontrados na literatura, como o descrito em Melcher et alii. [8], não são eficazes quando aplicados para o caso tridimensional. Isto ocorre porque estes algoritmos não são capazes de detectar a existência de vértices definidos por arestas internas ao contorno do objeto, tornando necessária a extração de mais informações.

Para o sistema de visão tridimensional de Farias [1] que foi descrito de forma resumida na seção anterior, os algoritmos responsáveis pela obtenção das características dos objetos presentes na cena não se encontram em uma forma que permita a extração automática dos vértices dos contornos ainda durante a fase de aquisição das imagens, ou seja, em tempo real. Esta extração tem por objetivo possibilitar a implementação do sistema como um todo por uma arquitetura paralela de alto desempenho.

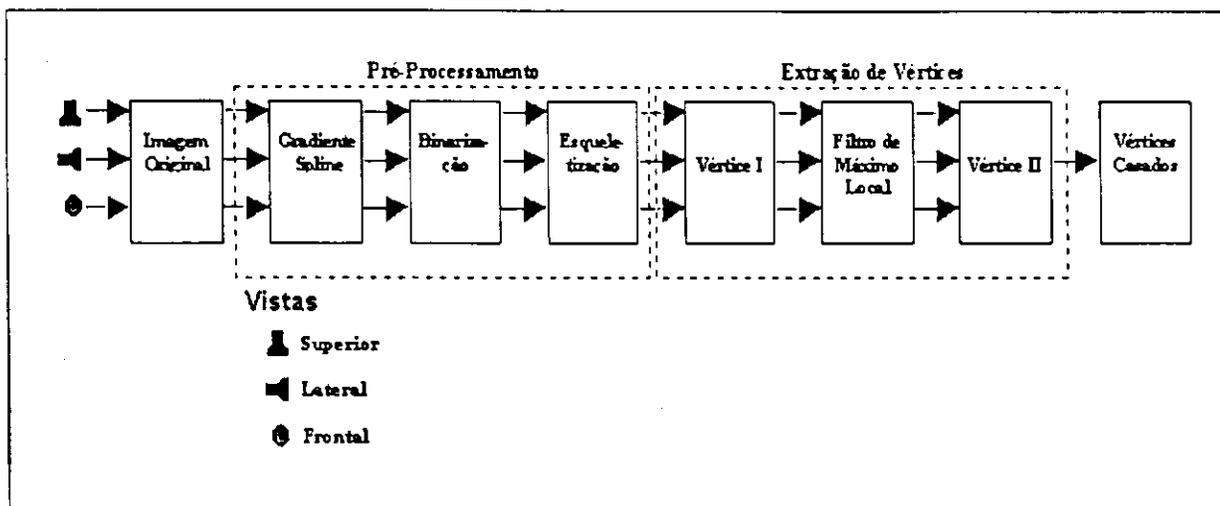


Figura 1.5: Diagrama em Blocos do Método Desenvolvido

Este problema constitui, portanto, o objetivo do presente trabalho: desenvolvimento de um método para identificação de vértices tridimensionais em imagens multi-vista. A identificação destes vértices, permite a definição de um modelo poliedral do objeto, o qual pode ser utilizado em um processo de reconhecimento de formas. Os algoritmos que foram aqui desenvolvidos operam por fluxo de dados, de modo a permitir que todo o processo de

modelagem do objeto possa ser realizado durante a etapa de aquisição da imagem, ou seja, em tempo real. Um algoritmo assim caracterizado pode ser implementado por uma arquitetura paralela, a qual pode ser realizada através de hardware dedicado, por um ASIC (Application Specific Integrated Circuit) ou por um FPGA's (Field Programmable Gate Array).

Desta maneira, o trabalho desta dissertação de mestrado viabiliza a construção de um sistema de visão tridimensional de alto desempenho totalmente automatizado, onde as etapas necessárias ao processamento das imagens e à modelagem do objeto podem ser implementadas por hardware, enquanto a tarefa de reconhecimento propriamente dita é implementada via software.

Na Figura 1.5 o sistema de extração de vértices desenvolvido neste trabalho é apresentado na forma de um diagrama de blocos. As operações indicadas são realizadas sobre cada uma das vistas da cena. Uma descrição sucinta destas operações é dada abaixo. Maiores detalhes serão descritos nos Capítulos 3 e 4.

- Gradiente Spline: Apaga regiões homogêneas e realça as bordas, deixando apenas os contornos.
- Binarização: Transforma imagens em níveis de cinza em imagens preto e branco, sendo os contornos marcados com 1 e o fundo com 0.
- Esqueletização: Extrai a linha essencial da estrutura da imagem, com 1 pixel de largura.
- Vértice I: Identifica os vértices e os marca com seus respectivos pesos.
- Filtro de máximo local: Em uma janela 5x5 descobre o vértice de maior peso e apaga os outros.
- Vértice II: Identifica os vértices que têm correspondência (vértices casados) em outras vistas e apaga os outros.

1.4. Organização da Dissertação

O texto que segue está organizado da seguinte maneira: o Capítulo 2 apresenta, como já foi dito, uma revisão bibliográfica sobre os sistemas de visão para reconhecimento de objetos baseados no casamento de modelos. No Capítulo 3 podem ser encontradas todas as etapas de pré-processamento aplicado às imagens, necessário para a extração dos vértices. O Capítulo 4 descreve o método de geração e extração dos vértices formados pelo encontro de duas ou mais arestas, proposto neste trabalho. O Capítulo 5 analisa as imagens resultantes dos testes efetuados para os algoritmos descritos em capítulos anteriores. O Capítulo 6 apresenta as conclusões obtidas para o trabalho desenvolvido e aqui exposto, bem como suas contribuições e sugestões para trabalhos futuros. Na seqüência estão citadas as referências bibliográficas utilizadas para o bom andamento deste trabalho.

Capítulo 2

Reconhecimento Através de Modelos

Quando um homem vê um determinado objeto pela primeira vez, ele é capaz de, sem nenhum esforço adicional, extrair e organizar informações sobre este novo elemento tais como textura, geometria, funcionalidade, etc. Uma vez estando estas características armazenadas em sua memória, o ser humano torna-se apto a reconhecer este objeto na próxima vez que o vir, mesmo que ele esteja sob um ponto de vista diferente do original, ou com um tamanho diferente ou ainda com uma parte escondida ou mesmo faltando. Portanto, o que ocorre com o homem de uma maneira geral, é tentar casar algumas características visíveis de um objeto com as mesmas características de objetos já conhecidos para poder identificá-lo.

Analogamente, o reconhecimento de objetos pelo computador exige deste último um conhecimento a priori sobre o objeto, o que normalmente é obtido através de um modelo armazenado na memória do computador. Então, o processo do reconhecimento é essencialmente o problema de tentar encontrar dentre aqueles modelos de objetos armazenados num banco de dados, o que melhor combine com os dados extraídos da imagem que contém o objeto observado.

Conseqüentemente, o desempenho destes sistemas de visão para o reconhecimento de objetos está diretamente associado ao método escolhido para descrever o modelo do objeto, seja em duas ou três dimensões. Quanto mais fiel ao objeto real for esta descrição (modelo), mais confiável e rápido será o resultado final do casamento.

Um sistema de visão que faz uso de modelos geométricos de objetos é conhecido como *sistema de visão para reconhecimento baseado no casamento de modelos geométricos*. Estes sistemas funcionam basicamente tentando encontrar a melhor correspondência entre algum conjunto pré-definido de características geométricas do modelo de um objeto observado numa imagem, e os mesmos tipos de características para modelos de objetos já conhecidos. Quase todas as aplicações que necessitam reconhecer seja o objeto como um todo ou apenas alguma característica específica deste, precisam também determinar efetivamente sua

localização e orientação exatas na cena.

Dependendo da natureza do problema para o qual o sistema de visão se destina, ele pode trabalhar em duas ou três dimensões. Todavia, alguns dos sistemas de visão que trabalham em três dimensões utilizam modelos e estratégias de reconhecimento também empregadas pelos sistemas que operam em apenas duas [16].

Este capítulo faz uma breve revisão bibliográfica a respeito dos sistemas de visão que fazem o reconhecimento de objetos através do casamento de modelos, dando ênfase aos sistemas que reconhecem objetos tridimensionais, pois este é o objetivo final do presente trabalho.

Na próxima seção serão examinadas as características dos sistemas de visão baseados no casamento de modelos geométricos, seguida da seção 2.2. que analisa técnicas de construção de modelos para estes sistemas. A seção 2.3 contém mais informação sobre a extração das características da imagem, e finalmente, mais detalhes sobre o casamento das características do objeto real com as do modelo podem ser encontrados na seção 2.4.

2.1. Características dos Sistemas de Visão Baseados no Casamento de Modelos

Este tipo de sistema de reconhecimento trabalha em dois momentos distintos: a fase de treinamento do sistema e a fase de classificação, ou reconhecimento propriamente dito. A primeira fase acontece *off-line*, ou seja, em um momento anterior ao tempo de execução do sistema. É nesta fase em que os objetos que serão passíveis de reconhecimento pelo sistema são escolhidos, bem como através de quais características eles serão representados e como serão construídos seus modelos. Na segunda fase, a imagem de um objeto real é adquirida, digitalizada, modelada, suas características relevantes ao processamento são extraídas e, a procura por uma correspondência entre as características do modelo do objeto presente na imagem real e as dos modelos armazenados é realizada para que o objeto seja efetivamente reconhecido.

As três principais etapas de operação de um sistema de visão para reconhecimento baseado no casamento de modelos, segundo Chin e Dyer [16] são:

1. Construção dos modelos dos objetos;
2. Extração das características do objeto da cena;
3. Casamento das características do modelo com o objeto.

Vários trabalhos sobre sistemas de reconhecimento tem sido publicados defendendo relacionamentos diferentes entre estas três etapas. Estas diferenças somadas às estratégias empregadas no algoritmo de reconhecimento personalizam cada sistema, de modo que se tornam quase específicos em sua atuação [1]. Cada uma destas etapas serão examinadas separadamente nas próximas seções.

2.2. Construção dos Modelos

Como visto anteriormente, a modelagem do objeto é um fator crítico para o desempenho final de um sistema de visão baseado no casamento de modelos. Os modelos tridimensionais permitem uma descrição do objeto mais geral e ao mesmo tempo mais completa, independente do ponto de vista em que se encontra o observador.

Na literatura podem ser encontradas duas abordagens distintas para a construção dos modelos em três dimensões como em Chin e Dyer [16], Besl e Jain [17], ou ainda Arman e Aggarwal [18]. Uma abordagem utiliza imagens reais do próprio objeto a ser reconhecido para gerar os modelos, e a outra utiliza imagens do objeto desenhadas pelo computador, através dos recursos de sistemas CAD (Computer-Aided Design/Drafting) para elaborar os modelos.

De uma forma resumida podemos afirmar que na primeira abordagem, são extraídas com antecedência características de imagens reais, que podem ser obtidas de vários pontos de vista do objeto, integrando-as de forma coerente para que possam apresentar informações do objeto sob todos os ângulos de vista. Na segunda, os sistemas CAD permitem que um usuário desenvolva os modelos dos objetos no computador. Além da geração e visualização em três dimensões no computador, os sistemas CAD tem facilidades para que sejam extraídas dos modelos as características geométricas mais freqüentemente utilizadas [1].

Ambas as abordagens para a construção dos modelos encontram-se mais explicadas nas

subseções que se seguem.

2.2.1. Construção de Modelos a Partir do Próprio Objeto

Os pesquisadores têm investigado dois tipos principais de representação em três dimensões de objetos: 1) representação exata e 2) representação multi-vista. Na primeira representação um modelo genérico simples é construído a partir das propriedades do objeto no espaço físico que ele ocupa. Na segunda, um conjunto de características bidimensionais são extraídas de imagens distintas do objeto (ditas “multi-vistas”), sendo estas obtidas a partir de diferentes pontos de vista do mesmo [16].

A representação exata inclui a classe dos métodos baseados nas especificações geométricas e topológicas de um objeto em três dimensões. As principais técnicas de representações que podem ser encontradas na literatura são: *representação por superfícies*, *representação por volume*, *representação por esqueletização* e, por último, a *representação por wire-frames* [1].

A *representação por superfícies* de um objeto é especificada em termos de suas bordas em três dimensões, as quais delimitam suas faces [1, 4, 18]. Isto é feito através da obtenção, primeiro, dos pontos que representam os vértices do objeto, para em seguida obter-se as arestas a fim de, então, poder-se descobrir as faces ou superfícies que formam o objeto, ou seja, que compõem uma descrição poliedral do objeto.

A *representação por volume* descreve um objeto tridimensional em termos de sólidos geométricos [1, 4, 18]. Os sólidos mais utilizados para isto são os cilindros, os cones, as esferas e os poliedros em geral. A principal vantagem desta classe de representação é que ela fornece uma descrição exata do objeto. A principal desvantagem é que é complexa sua utilização em sistemas de reconhecimento de objetos em tempo real, uma vez que em seu processamento é necessário executar projeções de 2D para 3D ou ainda de 3D para 2D para que se possa casar características de imagens em duas dimensões observadas com os modelos tridimensionais.

A *representação por esqueleto* de um objeto tridimensional se refere a uma figura composta por uma “armação de fios” (*wire-frames*) a qual corresponde à estrutura principal

do objeto, ou seja, ao seu esqueleto [18].

A *representação por wire-frames* constrói o modelo em três dimensões do objeto a partir apenas de suas arestas, fazendo uso de uma lista dos pontos de vértices e de uma lista dos pares destes que formam cada aresta do objeto [1, 17]. Os modelos gerados por esta representação não são suficientes por si só para definir onde as faces do objeto estão ou algumas quantidades como a área de cada superfície ou o volume do objeto.

O segundo tipo de representação de objetos tridimensionais, a representação multi-vistas, é determinada pela descrição em duas dimensões de características dos objetos, obtidas em cada uma das vistas, também bidimensionais, aos quais são a posteriori, combinadas entre si para formar o modelo em três dimensões [1, 4, 16, 17, 18].

A vantagem desta representação é que inúmeras técnicas de análise para imagens em duas dimensões estão disponíveis, existindo uma vasta literatura a respeito como apresentado em Zhang e Faugeras [27].

Uma classe desta representação é baseada nas descrições das vistas características extraídas do objeto, a qual requer todas as vistas topologicamente distintas. Baseado nesta equivalência topológica de pontos de vista vizinhos, pode ser definido um “grafo de aspecto” de pontos de vista de características distintas [16].

Outro tipo de representação multi-vista é a representação de nome “vista-esfera discreta”. Nesta representação, o objeto se transforma em uma espécie de “esfera visual” giratória, que tem todos os seus pontos externos visíveis, vistos a uma distância fixa do sensor óptico ao ponto do objeto, armazenando os resultados para cada ponto visualizado.

A representação de objetos a partir de características extraídas do próprio objeto, ou de imagens destes, tem a desvantagem de que os erros introduzidos na etapa de aquisição dos dados (imagem), devidos por exemplo, a erros de calibração de câmera, ruído, distorções no sensor, erro no cálculo das distâncias, etc., podem afetar a etapa de modelagem, o que pode implicar em falhas do processo na etapa de casamento. Além disto, algumas características da imagem podem se perder por causa do efeito “sombras”, onde parte da cena ou mesmo do objeto permanece invisível à fonte de luz ou ao sensor. Acontecendo isto, não é possível obter um modelo completo do objeto.

2.2.2. Construção de Modelos por Sistemas CAD

A construção de modelos para sistemas de reconhecimento de objetos necessita de técnicas eficientes que descrevam completamente um objeto em três dimensões. Os sistemas CAD (“Computer-Aided Design / Drafting”) podem ser utilizados para este fim, seja para projetar a geometria do objeto (síntese de imagem), ou para derivar as características (extrair os atributos) geométricas dos objetos projetados, essenciais aos procedimentos de reconhecimento.

Os sistemas CAD representam um sistema muito utilizado em diversas áreas de pesquisa científica, principalmente nas engenharias mecânica, elétrica e civil, para o desenvolvimento de projetos auxiliados por computador, com importantes recursos tanto em duas quanto em três dimensões.

Os sistemas CAD são ferramentas que oferecem facilidades gráficas para a construção dos modelos, com escalas apropriadas para plotagem e visualização. O CAD permite manipular os modelos dos objetos construídos de várias formas, tais como rotacionar ou transladar para se conseguir acesso às vistas diferentes (projeções) dos objetos, transformar formas bidimensionais em sólidos por meio de extrusão e revolução, estender curvar, combinar características simples em sólidos por meio complexos com as operações booleanas de união, subtração e interseção e adicionar características a sólidos através de chanfros e arredondamentos [1].

Operando com grande precisão, os sistemas CAD também são capazes de analisar o projeto de forma visual e material, acrescentar superfícies em imagens em três dimensões, definir fontes de iluminação, bem como adicionar ou retirar cores e textura para deixar os desenhos com aparência a mais realista possível. Além disso, o CAD produz desenhos apropriados para criação de animações através de programas adicionais, tipo o ‘3D-Studio’ e ‘Animator’.

A construção de modelos por CAD possui algumas vantagens sobre os métodos que constróem modelos a partir de imagens reais do objeto, quais sejam :

Os modelos CAD são exatos e livres dos erros que eventualmente ocorrem quando da

aquisição da imagem real, podendo gerar as partes não visíveis de um determinado ponto de vista. Provêem facilidade para manusear uma ampla classe de objetos, bem como para gerar bases de dados geométricas e bibliotecas padronizadas de modelos, adequadas para diversas áreas da engenharia [18]. Além disso, o CAD pode simular situações (cenas) com os modelos antes mesmo da elaboração da estrutura física de aquisição das imagens dos objetos, para testes e estudos do sistema.

Tradicionalmente, os sistemas CAD podem ser divididos em três grupos [1]. No primeiro grupo, destacam-se os sistemas que são utilizados nos *mainframes*. Estes são computacionalmente poderosos e com muitas opções de manipulação de modelos. É o caso do "CATIA", do "I-DEAS", do "CADAM" e outros. No segundo grupo estão aqueles utilizados nos PC's, com capacidade menor que os do primeiro grupo, mas munidos das tarefas essenciais ao projeto de modelos. É o caso do "AutoCAD", do "CAD KEY", do "Personal Designer" e outros. No terceiro e último grupo estão aqueles que operam especificamente em estações de trabalho RISC como o "ANVIL 500", o "CADRA-III", o "CIM CAD 3D" e outros, com capacidade intermediária de processamento. Entretanto, com a constante evolução dos processadores e dos softwares é de se esperar que esta distinção tenda a desaparecer, com o surgimento dos sistemas CAD multi-plataforma de alto desempenho.

Os diversos sistemas CAD diferem entre si pelo poder de manipular as características geométricas dos modelos, e de mostrá-las sob pontos de vista a partir de vários ângulos de iluminação propostos, por operadores de edição, velocidade de operação, custo e disponibilidade de programas auxiliares. Alguns CAD também utilizam formatos padronizados de armazenamento de dados para permitir a troca de modelos entre sistemas diferentes [32].

Os primeiros modelos de sólidos tridimensionais construídos por sistemas CAD, provavelmente, utilizaram a representação de armação de linhas (*wire-frames*), onde o objeto é representado por um conjunto de arestas (linhas) interligadas. Tal representação, por si só, não era suficiente para definir as faces do objeto e portanto poderia confundir o usuário [1].

Na literatura encontram-se várias formas de representação de sólidos em três dimensões para sistemas CAD [1, 4, 16, 17, 18], das quais as mais importantes se enquadram nas seguintes classes: 1) representação por bordas de superfície, 2) representação por volume, 3) representação por imagem Gaussiana estendida, e 4) representação por varredura.

A primeira representação, por bordas de superfície, é a mais utilizada nos sistemas CAD e em Computação gráfica. Nesta representação, um sólido pode ser descrito de três formas: por um conjunto de superfícies, um conjunto de curvas espaciais e/ou grafo relacional descrevendo as conectividades entre as superfícies

A representação por volume descreve um sólido baseado no espaço volumétrico que ele ocupa. Segundo Farias [1], um meio de representar um objeto volumetricamente seria colocá-lo em algum sistema de coordenadas tridimensionais e dividir o volume que ele ocupa em elementos menores de volume, tal que estes elementos menores englobem partes do objeto.

A maioria das descrições volumétricas empregam cubos ou paralelepípedos conhecidos como *voxels*. A dimensão do voxel é relativa, um único voxel pode ocupar um grande espaço ao representar um objeto ou ainda um grande número de voxels pode ser requerido para representar um determinado objeto. O critério de erro para definir a dimensão do voxel vai depender da aproximação desejada no projeto.

A representação por imagem Gaussiana estendida utiliza o método da esfera Gaussiana para representar objetos sólidos através da orientação das superfícies do objeto. Se um sólido possui um conjunto de vetores normais unitários associado com cada uma de suas faces tal que cada vetor normal possa ser transladado para uma origem comum, retendo a sua direção original, então as extremidades destes vetores definem uma esfera Gaussiana e o lugar destas extremidades chama-se Imagem Gaussiana do sólido.

A maior dificuldade em utilizar a imagem Gaussiana é quando os sólidos têm superfícies não convexas, porque dois sólidos não convexas poderão ter a mesma imagem Gaussiana. E além disto, algumas informações da superfície do sólido são perdidas, uma vez que apenas a informação sobre a direção da normal é preservada.

A última representação, por varredura de um objeto, é descrita como sendo o volume do objeto varrido por um conjunto de seções transversais ao longo de um eixo, obedecendo uma regra de varredura. Em outras palavras, um espaço tridimensional é descrito por uma função bidimensional que pode ser definida para todos os pontos pertencentes a um eixo do objeto.

Os trabalhos desenvolvidos que fazem o uso de ferramentas CAD para a construção de modelos têm apresentado melhores resultados não somente com relação a precisão mas

também com relação a velocidade de processamento, por estes motivos vem tendo maior aceitação nos trabalhos desenvolvidos nos últimos anos [19].

2.3. Extração das características

Nesta seção podem ser observados dois problemas básicos para sistemas de reconhecimento: a aquisição das imagens do objeto e a escolha das características a serem extraídas das imagens. Estes problemas devem ser abordados antes de se trabalhar no algoritmo de reconhecimento propriamente dito, ou seja, ao se iniciar o projeto do sistema de reconhecimento de objetos.

A estrutura utilizada para a aquisição dos dados em três dimensões deve ser tal que o sistema consiga captar dos objetos os mesmos atributos geométricos que descrevem o modelo. As imagens capturadas são projeções das partes visíveis do objeto sob um determinado ponto de vista, o ângulo de visão da câmera.

O problema de selecionar quais características geométricas do objeto serão componentes de sua descrição, está integralmente relacionado ao problema de definição do modelo, uma vez que as características nos dois casos serão as mesmas.

Uma breve apresentação sobre os dois problemas aqui abordados, como pode ser feita a aquisição das imagens reais dos objetos e como descrever a imagem adquirida, é fornecida nas duas sub-seções a seguir.

2.3.1. Aquisição das Imagens

A aquisição das imagens reais dos objetos é um importante aspecto dos sistemas de reconhecimento. As técnicas de reconhecimento de objetos tridimensionais empregam métodos baseados na estrutura geométrica formada pela câmera, fonte luz (se houver) e o objeto, para compor a descrição da cena em três dimensões.

Uma grande parte dos sistemas de visão que fazem o reconhecimento de objetos

tridimensionais baseados no casamento de modelos e mencionados na literatura de visão por computador, apresentam a *estereoscopia* como modo de obtenção das imagens dos objetos. A estereoscopia é um sistema de montagem que extrai várias imagens do objeto, obtidas de vários pontos de vista diferentes, para capturar o volume (informação tridimensional) do objeto [18].

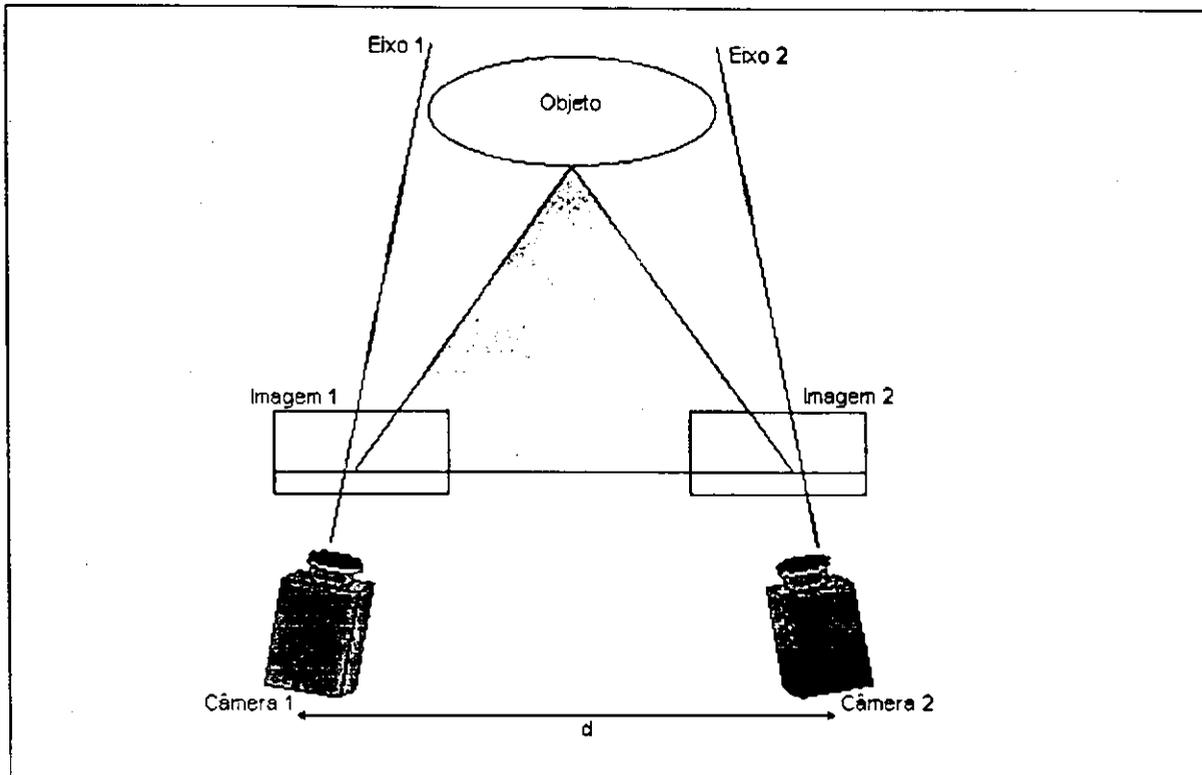


Figura 2.1.: Estrutura para a metodologia da triangulação.

A estereoscopia está dividida em duas abordagens diferentes: a estereoscopia ativa e a estereoscopia passiva [4 e 18]. Na abordagem passiva a informação tridimensional é obtida da cena utilizando-se apenas a energia existente no ambiente, como luz natural refletida. Na abordagem da estereoscopia ativa a informação tridimensional é derivada de projeções de ondas de energia externa, como por exemplo as ondas de sonar e luz de laser.

As técnicas mais comumente utilizadas na estereoscopia ativa são duas: os métodos baseados em triangulação (cuja estrutura encontra-se apresentado na Figura 2.1.) e os métodos baseados no retorno de feixe (*time-of-flight*) [1, 4, 18].

A triangulação utiliza uma fonte de laser que emite um feixe de luz sobre o objeto cujos reflexos são recebidos por uma câmera estrategicamente posicionada onde os reflexos podem

ser alcançados. A varredura de um feixe de luz pode ser feito de dois modos, com um refletor que pode rotacionar e é utilizado para projetar o feixe de luz no objeto ou com o objeto colocado em um determinado local que pode ser uma mesa giratória, para que o mesmo possa deslizar ou rodar na frente do feixe de luz. Então aplicando-se os conhecimentos de geometria na montagem do sistema, a distância de cada ponto iluminado até o sensor é calculada, permitindo assim a definição do objeto em três dimensões [1, 4]

A vantagem desta montagem está no fato dela conseguir fornecer muitas amostras uniformes e espaciais dos objetos na cena. Como desvantagens podemos citar a limitação do número e tamanho dos objetos contidos na cena, mas a principal desvantagem é o chamado efeito "sombra" resultante da triangulação, onde uma região do objeto pode ficar invisível por não ter sido iluminada pelo laser.

Os métodos baseados no retorno de feixe consistem de um transmissor de sinal direcionado ao objeto e de um receptor (sensor) eletrônico que capta o sinal refletido pelo objeto presente na cena e calcula o tempo gasto pelo retorno do sinal (reflexão) e sua intensidade. Este tempo é proporcional à distância e às dimensões do objeto. Na prática são utilizados transmissores de ultra-som ou de luz de laser. Esta metodologia apesar de ser mais sofisticada, tem resolução pobre devido às dificuldades em gerar feixes acústicos estreitos.

Após a obtenção da imagem do objeto (descrição de baixo nível), estas passam por processos de tratamento digital de sinais, visando extrair as informações úteis. (realce, redução de ruídos, compressão de imagens, quantização, etc.) e segmentação para em seguida serem retiradas as características úteis ao reconhecimento.

2.3.2. Descrição das Imagens

A decisão da escolha dos atributos ou características que serão extraídas é subjetiva e dependente da aplicação. As características mais importantes na análise da imagem digital se enquadram em uma das três classificações abaixo [1, 3 e 16]:

1. Características Globais,
2. Características Locais,

3. Características Relacionais.

As características globais estão relacionadas com as propriedades geométricas, como perímetro, área, momentos de inércia, centróide, níveis de cinza dos objetos, normal, cor e contorno. Estes atributos são fáceis e rápidos de serem obtidos, podendo ser computados em tempo real. Além disso, são invariantes às transformações de translação e rotação, o que significa que os objetos podem se encontrar em qualquer posição e orientação. Entretanto, sistemas de reconhecimento que fazem o uso unicamente desta classe de características dificilmente conseguem reconhecer corretamente um objeto que esteja com uma ou mais partes oclusas, ou mesmo faltando, ou ainda em contato com outros objetos.

Dentre os exemplos de características locais, que estão relacionadas com as bordas do objeto ou modelo, podem ser citados os segmentos de linha, comprimento de arcos e vértices. Estes são atributos que exigem sistemas mais sofisticados e complexos para serem extraídos das imagens. Eles não são invariantes as transformações de translações e às rotações, o que obviamente significa uma desvantagem.

As características relacionais, que descrevem as relações (dentre estas, as relações geométricas têm um interesse particular) que existem entre as características locais, incluem: o ângulo entre duas arestas ou duas normais, arestas entre as superfícies, distâncias entre centróides de superfícies, dentre outras. Estes atributos são mais adequados para as cenas em que os objetos se encontrem em contato ou parcialmente oclusos, bem como a classe das características locais.

De uma maneira geral, extraem-se ao menos dois tipos de características do mesmo objeto, como uma característica local e uma global, ou uma local e a outra relacional, ou ainda uma característica global e a outra relacional, de modo que uma delas tenha um nível maior de detalhamento do que a outra. Este detalhe fará a distinção entre um modelo do objeto dentre os demais. Por exemplo, numa determinada superfície as características locais podem ser a aresta e o vértice, enquanto as características globais podem ser o comprimento da aresta, sua área e o número de vértices. Mesmo que dois modelos coincidam com as características locais, eles se diferenciarão nas características globais [1].

Segundo Chin e Dyer [16], as representações das características mais utilizadas são três. A primeira é o *mapa de profundidade* onde temos uma tabela tridimensional com os índices x, y e z representando respectivamente os valores da linha, da coluna e da intensidade

luminosa. A segunda representação é a *estrutura de árvore hierárquica*, que é composta de níveis e sub-níveis representando as partes e sub-partes do objeto. A terceira e última representação é a *estrutura de grafos relacionais* que é composta de *links* e nós representando partes e ligações.

2.4. Casamento de Características

O último dos componentes principais de um sistema de visão para o reconhecimento é o casamento de características, onde acontece o reconhecimento (ou identificação) propriamente dito. Ele é também considerado o mais difícil e complexo, embora seu desempenho dependa em grande parte do processo de modelagem, como dito anteriormente [16].

Neste, o sistema procura encontrar a correspondência entre os dados observados na cena com os mesmos dados já armazenados em uma biblioteca de modelos dos objetos ou classes de objetos, obtidos durante a fase de treinamento de sistema. Este modelos representam os objetos que se espera 'ver' nas imagens, obedecendo critérios de erro determinados a priori.

Nas últimas décadas muitas estratégias de casamento baseadas em modelos têm sido desenvolvidas. Estas estavam primeiramente preocupados somente com o reconhecimento do objeto, contudo, uma tendência verificada nos últimos anos foi a produção de informações extras, como a posição e orientação dos objetos identificados na cena, ou ainda a localização de uma superfície onde deverá ser feita alguma operação específica [1].

Isto pode ser feito através de uma matriz de transformação que faz o mapeamento entre o sistema de coordenadas do modelo e o sistema de coordenadas da imagem observada. Este procedimento requer o conhecimento de transformações geométricas, no entanto, esta correspondência não é trivial de ser obtida, uma vez que não existe qualquer referência direta para se obter a transformação apropriada [26].

O casamento está relacionado com o tipo de representação de dados utilizados nas características extraídas do objeto e também foram utilizados na fase de modelagem.

Na maioria dos trabalhos encontrados na literatura, a etapa do reconhecimento é realizada em dois passos, são eles:

1. A Geração de Hipóteses,
2. A Verificação das Hipóteses.

Como mencionado anteriormente, reconhecer um objeto é um problema de casamento entre a cena descrita com um modelo armazenado. O processo de reconhecimento se inicia pela geração de hipóteses. Em algum estágio do processamento deveremos ter um conjunto de pares de características correspondendo, respectivamente, a um atributo específico do objeto da cena e ao mesmo atributo de um modelo que casem.

Considerando M características extraídas do objeto presente na cena, e uma biblioteca composta por N modelos, temos o seguinte exemplo:

(característica 1 / objeto, característica 1 / modelo 1)

(característica 1 / objeto, característica 1 / modelo 2)

...

(característica 1 / objeto, característica 1 / modelo N)

(característica 2 / objeto, característica 2 / modelo 1)

...

(característica M / objeto , característica M / modelo N)

Para este casamento as características encontradas no objeto na cena devem ser arbitrariamente ordenadas e a primeira característica da cena deve ser testada com a mesma característica de cada um dos N modelos presentes na biblioteca. O processo se repete para a segunda característica, e assim por diante, para as M características.

Isto introduz um dos maiores problemas do casamento baseado em modelos: existirão muitos possíveis pares casando para a mesma característica, tornando lento o processo de reconhecimento. Assim, o problema consiste em encontrar uma metodologia de procura e casamento eficientes [4], para controlar o número de hipóteses geradas.

As hipóteses são geradas independentemente umas das outras. O algoritmo deve ser capaz de alcançar este passo utilizando uma descrição parcial do objeto e uma descrição

completa do modelo, para garantir seu funcionamento caso aconteçam possíveis pontos de perdas de informações devido a prováveis oclusões parciais, ou um único ponto de vista, ou um único campo de vista, etc. [18].

A verificação de hipóteses, é estabelecida através de um novo casamento, agora apenas entre os modelos selecionados no primeiro passo com um segundo conjunto de descrições mais detalhadas do objeto. Neste casamento, apenas um modelo deve ser selecionado dentre aqueles candidatos, ou nenhum, significando que o sistema não é capaz de reconhecer aquele objeto em particular. O modelo que foi selecionado é o modelo procurado e o reconhecimento do objeto é concluído com a descrição completa do modelo encontrado.

Determinar a localização do modelo identificado na cena é o mesmo que reconstruir o modelo no sistema de referência da cena. Este processo de reconstrução ocorre, mais uma vez, através de transformações geométricas que transportam o modelo elaborado em seu sistema de referência para dentro do sistema de coordenadas da cena, o mundo real. Tais transformações envolvem operações de translações e rotações de superfícies no espaço tridimensional [4].

Neste capítulo pôde ser visto um resumo da metodologia de reconhecimento de objetos tridimensionais através do casamento de modelos, algumas de suas características e suas três principais etapas. No próximo capítulo pode ser encontrado explicações a cerca do pré-processamento aplicado às imagens que se faz necessário a este presente trabalho, bem como as etapas em que ele é executado.

Capítulo 3

Pré-Processamento

Algoritmos para modelagem poligonal ou poliedral requerem imagens com formas bem definidas e contínuas. Produzir estas formas: é este o objetivo da fase de pré-processamento, descrita neste capítulo, à qual as imagens originais são submetidas no momento inicial do processamento. No final desta fase as imagens deverão estar reduzidas a formas compostas unicamente por linhas com apenas um *pixel* de largura, ditas *imagens esqueletizadas*, representando os contornos tanto externos quanto internos dos objetos presentes na cena.

O pré-processamento está dividido em três etapas distintas:

- Geração das imagens gradientes;
- Binarização;
- Esqueletização;

Para a execução em tempo real dos algoritmos de processamento de imagem, é necessário que estes sejam do tipo *data-flow* (fluxo de dados), onde o processamento se dá em função dos dados, a medida que chegam os dados fluem pelas operações. Isto possibilita que o processamento ocorra ainda durante a fase de aquisição das imagens. Nestes algoritmos os pixels das imagens são capturados, enviados e processados serialmente, pixel por pixel, linha após linha. Para cada pixel que chega, o algoritmo do tipo *data-flow* produz um outro de saída que é determinado em função do pixel de posição correspondente da entrada e de um conjunto pré-definido de seus vizinhos de entrada.

Em geral, estes algoritmos necessitam aguardar até que cheguem pixels suficiente para formar o conjunto pré-definido de vizinhos, que pode, por exemplo, assumir a forma de uma janela de tamanho $N \times N$ pixels (N varia de acordo com as necessidades dos algoritmos). Os pixels transformados de saída serão os componentes de entrada da etapa ou algoritmo subsequente do processamento *data-flow*, e assim por diante.

Antes que sejam descritas as etapas acima mencionadas do pré-processamento, é

preciso que se entenda o conceito de número de conectividade de um pixel, descrito a seguir.

3.1. Número de Conectividade

O conceito de número de conectividade de um pixel deve ser claramente explicado para evitar interpretações erradas, pois se está utilizando somente uma pequena parte (janela) da imagem original de cada vez, onde as regras de segmentação não são muito claras. Às vezes, um único *pixel* pode conectar duas seções maiores do objeto e é intuitivamente óbvio que este pixel não pode ser removido. Caso contrário serão criados dois objetos diferentes onde originalmente existia apenas um [11].

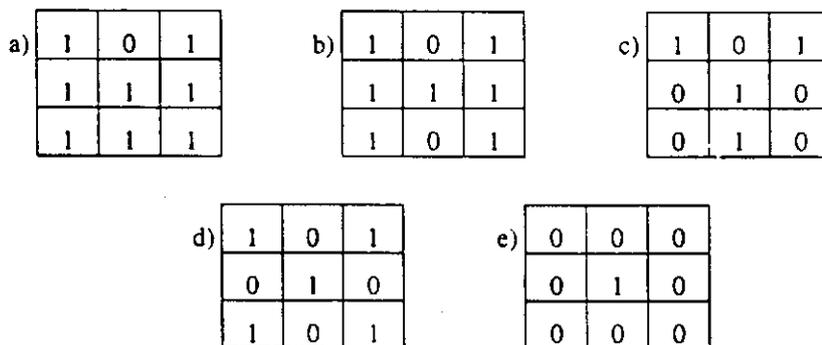


Figura 3.1: Exemplos de números de conectividade: a) conectividade = 1, b) conectividade = 2, c) conectividade = 3, d) conectividade = 4 e e) conectividade = 0.

Um exemplo de medida de conectividade, que pode ser encontrado em Yokoi et alii. [13], está apresentado na Figura 3.1. No exemplo, a) o pixel central da janela 3x3 não conecta nenhuma região, e por isso mesmo pode ser removido durante um processo de afinamento: sua conectividade é 1. Em b) se o pixel central for excluído o lado direito e o esquerdo serão desconectados, e então perderemos a informação de se tratar de um objeto só: sua é conectividade igual a 2. Em c) o pixel central conecta 3 partes de um mesmo objeto, sua conectividade é então igual a 3. Em d) tem-se o maior número de conectividade possível, 4. Finalmente, em e) aparece o pixel central isolado, sua conectividade é portanto, zero.

Esta medida, a de conectividade, pode ser expressa por:

$$C_s = \sum_{k \in S} N_k - |N_k \oplus N_{k+1} \oplus N_{k+2}|$$

Onde: N_k , $k = 1, 2, \dots, 8$ é o valor do nível de cinza de cada um dos oito vizinhos imediatos do pixel central numa janela 3×3 , e $S = \{1, 3, 5, 7\}$. N_1 é o valor de cinza do pixel exatamente à direita do pixel central e os outros são numerados no sentido anti-horário ao redor do pixel central. O valor de N_k será um se o nível de cinza do pixel for branco (fundo) e zero se for preto (objeto). O pixel central é N_0 .

Outra maneira de se medir a conectividade de um pixel computacionalmente é analisando a vizinhança dos seus oito pixels vizinhos mais próximos, na ordem sugerida acima: $N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8$ e N_1 . A quantidade de mudanças de cor (preto-branco ou branco-preto) encontrada, será o valor do número de conectividade do pixel central [11].

3.2. Gradiente Spline

O objetivo do cálculo da imagem gradiente é o de possibilitar a extração das bordas dos objetos presentes na cena.

O operador gradiente Spline é baseado em *splines cúbicas* como descrito por Melcher et alli [7]. O valor do gradiente em um dado ponto da imagem de entrada é obtido fazendo-se a convolução desta imagem com as quatro máscaras mostradas na Figura 3.2, de tamanho 5×5 . Portanto, o gradiente Spline bidimensional é resultante dos gradientes horizontal, vertical e diagonais, g_{px} , g_{py} , g_{p1} , g_{p2} , respectivamente, e pode ser expresso matematicamente por:

$$g_p = 3(|g_{px}| + |g_{py}|) + 2(|g_{p1}| + |g_{p2}|)$$

O peso dos valores dos gradientes g_{px} e g_{py} é maior do que o peso dos gradientes g_{p1} e g_{p2} por um fator de $3/2$. Isto é uma aproximação do fator de $\sqrt{2}$ (raiz de dois) que corresponde à taxa das distâncias entre o pixel central e dois pontos vizinhos, nas direções vertical e diagonal, respectivamente.

Uma das principais vantagens deste operador gradiente é que ele é menos sensível a

ruídos do que outros operadores, como mostra comparação feita em Melcher et al. [7]. Isto se deve ao fato de que o gradiente Spline faz uso de janelas de tamanho 5x5 nas máscaras, maior que o mais comumente utilizado por outros operadores gradientes, e também por determinar seus resultados a partir do valor médio das derivadas nas direções horizontal, vertical e diagonais.

g_{px}	0	0	0	0	0
	0	0	0	0	0
	-1	8	0	-8	1
	0	0	0	0	0
	0	0	0	0	0

g_{py}	0	0	-1	0	0
	0	0	8	0	0
	0	0	0	0	0
	0	0	-8	0	0
	0	0	1	0	0

g_{dx}	-1	0	0	0	0
	0	8	0	0	0
	0	0	0	0	0
	0	0	0	-8	0
	0	0	0	0	1

g_{dy}	0	0	0	0	1
	0	0	0	-8	0
	0	0	0	0	0
	0	8	0	0	0
	-1	0	0	0	0

Figura 3.2: Máscaras utilizadas para o cálculo do gradiente Spline

Além disso, as bordas produzidas pelo gradiente Spline são mais precisas, devido ao fato de que os valores do gradiente dos pixels de contorno são significativamente maiores que os valores de gradiente dos pixels de suas vizinhanças. Isto acontece porque a interpolação através de splines cúbicas produz uma aproximação ótima da função original [9], resultando também em uma aproximação ótima do gradiente desta função.

3.3. Binarização

As imagens utilizadas no presente trabalho são formadas por 32 níveis de cinza (maiores informações a respeito das imagens podem ser encontradas no Capítulo 5), conseqüentemente as imagens gradientes também o são. Nesta etapa, as imagens gradiente são binarizadas, ou seja o conjunto de valores que cada pixel pode assumir é reduzido para duas possibilidades: zero ou um. O valor zero representa o fundo da imagem, ou seja, a ausência de objetos na cena, enquanto o valor um fica reservado para os objetos presentes na cena.

Este é o único algoritmo utilizado por esta dissertação que não trabalha com janelas deslizantes sobre a imagem. Ele processa um pixel por vez na mesma seqüência em que estes aparecem na imagem (da esquerda para a direita, de cima para baixo). O valor do pixel de saída é determinado em função do valor do pixel de entrada de posição correspondente, que é comparado com um limiar de decisão. Neste trabalho o limiar escolhido foi 16 por ser este o valor médio da gama de valores possíveis de níveis de cinza das imagens de entrada utilizadas nos testes. O limiar 16 foi escolhido empiricamente. Testes indicaram que o valor do limiar não influe significativamente no resultado da esqueletização. O valor do gradiente do pixel lido é comparado com o valor do limiar de decisão, 16; se for maior que este, o valor do pixel de saída na posição correspondente ao pixel de entrada será um, caso contrário o valor de saída será zero.

A binarização das imagens gradientes é feita para facilitar a execução dos algoritmos pertencentes às próximas etapas do pré-processamento, deixando as imagens com o formato o mais simples possível. Esta simplificação é possível, devido ao fato da modelagem dos objetos ser feita apenas em função de seus contornos, não necessitando dos valores de cinza dos pixels internos.

3.4. Esqueletização

A necessidade desta última fase deve-se ao fato das bordas das imagens gradientes geralmente serem muito "grossas", ou seja, possuírem mais do que um pixel de largura.

Neste trabalho, a etapa de segmentação de imagens não poderia ser implementada por um método do tipo "*boundary tracking*", como o apresentado em Cortez [3], comumente utilizado para este fim, no qual o contorno do objeto é obtido através de rastreamento pixel a pixel da imagem de entrada, de forma contínua e sequenciada a partir de um pixel pertencente ao contorno. Para isso seria necessário ter-se a imagem completa antes do início do processamento do algoritmo, o que não é possível em sistemas de execução em tempo-real, conforme explicado anteriormente neste capítulo.

Então a solução foi pesquisar, desenvolver e implementar algoritmos que funcionem através de janelas deslizantes sobre a imagem, como por exemplo, os algoritmos que

trabalham com morfologia matemática.

Para o bom funcionamento destes algoritmos, é importante que a imagem de entrada tenha sua espessura reduzida, sem alterar o seu tipo e número de conectividade (Seção 3.1), fazendo com que a imagem de saída tenha o mesmo número de partes componentes. Uma transformação deste tipo chama-se de *afinamento* [10]. Afinando-se a mesma imagem sucessivamente, até que sua estrutura seja representada por uma linha com apenas um pixel de largura, estamos fazendo a *esqueletização* desta imagem. Esqueletização de imagens pode ainda ser definida como o ato de identificar os pixels pertencentes a um objeto que são essenciais para a comunicação da estrutura deste objeto, ou seja, seu esqueleto.

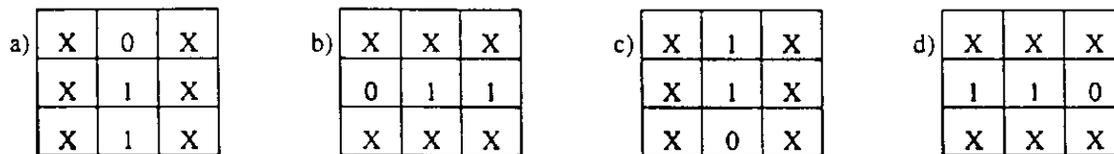


Figura 3.3: Máscaras para Afinamento.

A maioria dos algoritmos de afinamento são baseados em retiradas sucessivas de camadas de pixels mais externos da imagem até que nenhuma camada mais possa ser retirada. Existem um conjunto de regras definindo quais pixels podem ser removidos, e frequentemente algum tipo de esquema de casamento de padrão é usado para implementar estas regras [11].

Bons resultados de esqueletização foram conseguidos para este trabalho com a combinação de três algoritmos diferentes de afinamento, conforme sugestão estudada em Parker [11]: os algoritmos de pré-processamento das imagens para afinamento de Stentiford [12], combinados com os algoritmo de afinamento propriamente dito de Zhang-Suen [14] e por último o algoritmo de pós-processamento de Holt [15], todos descritos resumidamente abaixo.

3.4.1. Algoritmo de Pré-processamento de Stentiford

Um típico exemplo de um algoritmo de afinamento do gênero exposto é o descrito por Stentiford em 1983 [12], que utiliza máscaras de tamanho 3x3. O casamento destas máscaras com uma janela da imagem, significa marcar para futura exclusão o pixel central da janela.

Ao termino da execução do algoritmo de casamento, um outro processo é iniciado, que muda para branco (zero) os pixels marcados. As máscaras utilizadas por Stentiford são mostradas na Figura 3.3. Os padrões de pixels pretos (uns) e brancos (zeros) nas máscaras devem ter seus correspondentes nas janelas da imagem de mesma cor para que aconteça o casamento. As posições dos pixels identificados com Xs indicam que nas janelas da imagem sua cor é irrelevante para que haja o casamento.

A operação do algoritmo é [11]:

1. Varrer a imagem até encontrar uma janela que case com a máscara da Figura 3.3. a;
2. Se o pixel central não for um ponto final de uma reta, ou seja, se na janela de tamanho 3x3 da imagem não houver apenas mais um pixel preto além do pixel central, e possuir número de conectividade (explicado na Seção 3.1.) igual a um, então marcar o pixel central da janela na imagem para futura exclusão;
3. Repetir os passos 1 e 2 até completar a varredura da imagem;
4. Repetir os passos do 1 ao 3 para as outras máscaras da Figura 3.3, uma de cada vez e na ordem em que são mostradas;
5. Excluir da imagem os pixels marcados para exclusão, mudando seu valor para zero (branco);
6. Repetir todos os passos até que nenhum pixel seja marcado para exclusão, que é a condição de parada.

Existe um grupo de problemas clássicos com os algoritmos de afinamento que podem ser percebidos no final da esqueletização. Eles são clássicos por que tendem a aparecer numa grande variedade dos algoritmos deste tipo, fazendo com que pesquisadores da área tentem se antecipar a eles. O primeiro deste problemas é chamado de *gargalo*, e aparece quando se afina a interseção de dois segmentos de linhas grossas (linhas com vários pixels de largura). Neste caso, o ponto de interseção é transformado em um pequeno segmento de reta entre os esqueletos das linhas originais.

Um outro problema que pode ser criado é chamado de *cauda*, e aparece, como o anterior, quando do encontro de dois segmentos de linhas grossas. Neste caso, o ponto de

interseção se dá no final dos segmentos, com um ângulo agudo, e a esqueletização causa o aparecimento de um pequeno segmento de reta antes da bifurcação, onde começam as duas linhas. Finalmente, e talvez o mais comum de todos os problemas, a criação de linhas extras com uma das extremidades unida ao esqueleto real do objeto, oriundas de imperfeições ou mesmo ruídos ainda presentes nas imagens.

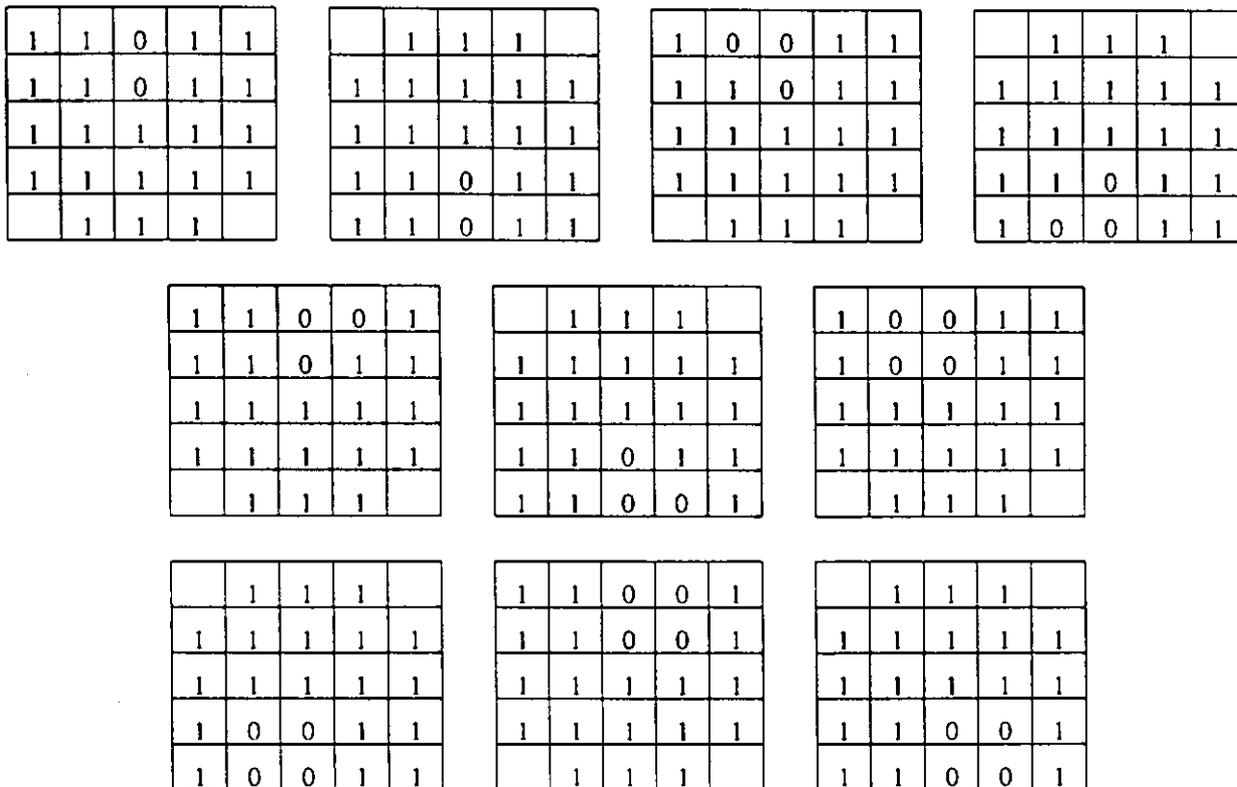


Figura 3.4: Máscaras usadas na etapa de pré-processamento: ênfase do ângulo agudo.

Stentiford sugere um estágio de pré-processamento para minimizar estes problemas com o afinamento [12], utilizando algoritmos de suavização de imagens para corrigir as imperfeições e/ou ruídos. Para o problema do “gargalo”, acima citado, ele sugere um procedimento chamado “ênfase ao ângulo agudo”, pelo qual os pixels perto da interseção das linhas têm seus valores mudados para zero (branco) se o ângulo formado for um ângulo agudo. Isto é decidido usando-se as máscaras de tamanho 5x5 mostradas na Figura 3.4. O casamento de uma destas máscaras com uma janela da imagem faz com que o pixel de posição correspondente na saída (central) seja marcado para futura exclusão. Neste caso, outra iteração ocorre, a qual faz uso para o casamento somente das seis primeiras máscaras. Se algum pixel ainda for marcado para futura exclusão na segunda iteração, um último passo utilizando somente as duas primeiras máscaras é realizado.

Apesar de resultados regulares, o método proposto por Stentiford [12] continua falho. A utilização de apenas três estágios no procedimento “ênfase do ângulo agudo” não é suficiente para alguns casos de bordas muito grossas. As máscaras da Figura 3.4. não prevêm todas as situações nas quais podem acontecer os fenômenos de *gargalo e cauda*. Como resultado, o método não é satisfatoriamente confiável para ser utilizado em um processo maior de reconhecimento de padrão.

3.4.2. O Algoritmo de Zhang-Suen

Um outro algoritmo de afinamento de grande aceitação e uso é o de Zhang de 1984, que tem sido muito utilizado como a base de comparação para métodos de afinamento e é simples e rápido para se implementar [11].

O algoritmo é dividido em duas etapas distintas. Na primeira, um pixel de posição (x,y) da imagem de entrada é marcado para deleção se todas as quatro condições abaixo foram atendidas simultaneamente:

1. O número de conectividade do pixel testado for igual a um;
2. O pixel tiver um mínimo de dois e um máximo de seis vizinhos com valor 1 (preto), em sua vizinhança de oito;
3. Pelo menos um dentre os pixels das posições $(x,y+1)$, $(x-1,y)$ e $(x,y-1)$ possuírem valor zero (branco);
4. Pelo menos um dentre os pixels das posições $(x-1,y)$, $(x+1,y)$ e $(x,y-1)$ possuírem valor zero (branco);

No final desta etapa os pixels marcados para deleção serão excluídos. A imagem resultante é submetida, então, à última etapa, na qual os seguintes testes são aplicados para cada pixel em (x,y) :

1. O número de conectividade do pixel testado é igual a 1?
2. O pixel possui um mínimo de dois e um máximo de seis vizinhos com valor 1 (preto) em sua vizinhança de oito;

3. Pelo menos um dentre os pixels das posições $(x-1,y)$, $(x,y+1)$ e $(x+1,y)$ possui valor zero (branco);
4. Pelo menos um dentre os pixels das posições $(x,y+1)$, $(x+1,y)$ e $(x,y-1)$ possui valor zero (branco);

Novamente, ao final da segunda etapa os pixels que satisfizeram simultaneamente os testes acima são excluídos. O algoritmo só é interrompido quando, no final da segunda etapa, não existirem mais pixels marcados para deleção, indicando que a esqueletização está completa.

Nas imagens esqueletizadas por este algoritmo, o problema de aparecimento de “cauda” não foi detectado. Entretanto o fenômeno chamado de “gargalo” ainda está presente, bem como os efeitos decorrentes de ruídos e imperfeições. Contudo estes problemas podem ser minimizados seguindo-se as sugestões de Stentiford, de utilizar um algoritmo suavizador, e fazer uso também do procedimento intitulado de “ênfase do ângulo agudo”, anteriormente descrito. Os resultados obtidos são significativamente melhores e mais confiáveis do que os do algoritmo de Stentiford.

3.4.3. Pós-processamento de Holt

Podemos citar ainda um último algoritmo de afinamento, apesar de seus resultados serem bons mas não tanto como a combinação do pré-processamento de Stentiford com o algoritmo de afinamento de Zhang-Suen.

Este algoritmo, proposto por Holt em 1987 [15], é rápido e não envolve sub-iterações. Para tanto, são utilizadas expressões lógicas, operadas sobre janelas de tamanho 3x3 da imagem de entrada, para pesquisar se o pixel central desta pode ser deletado ou não:

$$v(C) \wedge (\sim \text{borda}(C) \vee (\text{borda}(L) \wedge v(N) \wedge v(S)) \vee (\text{borda}(S) \wedge v(O) \wedge v(L)) \vee (\text{borda}(L) \wedge \text{borda}(SL) \wedge \text{borda}(S)))$$

Onde: C é o pixel central da janela analisada, a função v fornece o valor do pixel de parâmetro (1 = VERDADEIRO se for um pixel pertencente ao objeto, e 0 = FALSO caso

contrário). A função *borda* retorna VERDADEIRO se C for um pixel de borda do objeto – isto corresponde a ter entre dois e seis vizinhos mais próximos com valor 1 e o número de conectividade igual a 1. As letras N, S, L e O correspondem aos quatro pontos cardiais, norte, sul, leste e oeste respectivamente, indicando a direção do pixel testado a partir do pixel central. O pixel designado por L equivale ao da posição $(x,y+1)$ do algoritmo de Zhang-Suen.

0	1	X	X	1	0	0	X	X	X	X	0
1	1	X	X	1	1	X	1	1	1	1	X
X	X	0	0	X	X	X	1	0	0	1	X

Figura 3.5: Máscaras usadas por Holt para deletar pixel formadores de escadas.

Freqüentemente quando o algoritmo de afinamento termina sua operação, ainda existem pixels que poderiam ser deletados. Dentre estes, se destacam aqueles pixels que desenharam no esqueleto do objeto uma estrutura parecida com degraus de uma escada. Claramente, metade destes pixels formadores do efeito escada poderiam ser removidos sem afetar a forma do objeto como um todo. Isto pode ser feito, basicamente, utilizando-se as máscaras de tamanho 3x3 apresentadas na Figura 3.5., nas quais o pixel da imagem de saída de posição correspondente à do pixel central das máscaras, é excluído quando acontece casamento.

Terminada a fase do pré-processamento obrigatório das imagens, estas se encontram compostas apenas por linhas com um pixel de largura que correspondem ao formato do(s) objeto(s) presente(s) na imagem original e binarizadas. Estas são as condições mínimas necessárias para o processamento propriamente dito que é a extração dos seus vértices, assunto do próximo capítulo.

Capítulo 4

Geração e Extração dos vértices

Uma vez estando a imagem devidamente esquelizada e binarizada, o próximo passo é a extração dos vértices da mesma. Para tanto a solução encontrada foi procurar em janelas da imagem, estruturas que casem com algum dos padrões de vértices aceitáveis, esta solução foi baseada no trabalho de Villar [2]. Os padrões de vértices foram previamente gerados e armazenados em dois arquivos distintos: o primeiro contendo os vértices formados pelo encontro de três arestas e o outro contendo os vértices formados pelo encontro de duas arestas.

As janelas dos padrões são de tamanhos diferentes, mas nos dois casos os vértices gerados são de tamanho $N \times N$, sendo N um número ímpar, e o *pixel* correspondente ao vértice propriamente dito é sempre o *pixel* central da janela $N \times N$. O algoritmo criado e implementado que faz a procura destes padrões é do tipo *data-flow*, o que garante uma execução em tempo real.

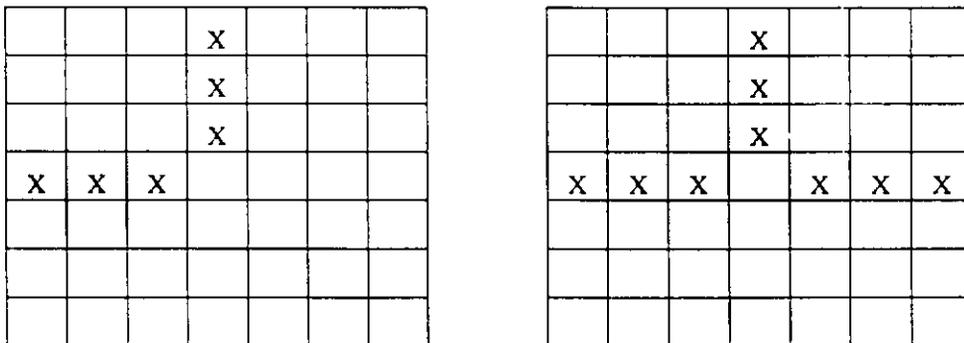


Figura 4.1.: Exemplos da perda do pixel de encontro das arestas

Como o objetivo da esquelização é deixar o contorno da imagem com no máximo um pixel de largura, observou-se nas imagens resultantes deste processo que alguns pixels que representavam os pontos de encontro das arestas (vértices propriamente ditos) tinham sido apagados, conforme mostrado na Figura 4.1. Algumas modificações no algoritmo de esquelização foram feitas com a intenção de preencher estas discontinuidades. Este objetivo foi inicialmente atingido porém com o prejuízo da perda de informação necessária

ao processo de extração dos vértices, ou seja, em determinados pontos da imagem não se têm um único pixel de largura. Uma outra solução testada foi a utilização de algoritmos que preencham as descontinuidades automaticamente, o que também não foi satisfatório por motivo semelhante.

Uma outra alternativa pensada e que foi finalmente adotada, foi modificar os algoritmos que fazem a procura de ambos os padrões de vértices nas janelas da imagem esqueletizada. Nesta forma, estes algoritmos, ao procederem o casamento, não exigem que o pixel central da janela da imagem contenha valor um (preto), desde que os demais pixels coincidam, para marcar a saída como vértice. Com isso, janelas como as apresentadas na Figura 4.1 são reconhecidas como contendo vértices. A solução tentada tem custos de execução e armazenamento aceitáveis, uma vez que não será preciso aguardar o tempo de se executar nenhum algoritmo adicional, conseqüentemente não implementa-lo em hardware, apenas modificar um já existente.

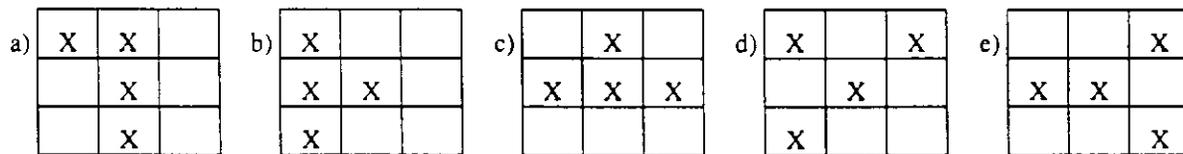


Figura 4.2.: exemplos de vértices 3x3, formados por 3 arestas

Os algoritmos implementados varrem os demais *pixels* do padrão de vértice procurando as posições onde eles sejam diferentes de zero, e para cada um destes verificam se na mesma posição da janela da imagem o *pixel* de posição correspondente também o é (menos o *pixel* central, conforme explicado). Se em todos os casos a condição for satisfeita, na saída o *pixel* central da janela será marcado como um candidato a vértice. A maneira exata como é feita esta marcação (pesos) será explanada posteriormente. Para os demais *pixels* (iguais a zero) nenhum teste é feito.

4.1. Geração dos Vértices Formados por Três Arestas

Para a geração do primeiro arquivo, foi criado e implementado um algoritmo que gera todas as possibilidades diferentes de se distribuir dentro de uma janela 3x3 quatro pixels, representando o encontro de três arestas no ponto central da janela, como nos exemplos

mostrados na Figura 4.2. Procurando-se vértices formados pelo encontro de apenas três arestas, garante-se encontrar também os que são formados pelo encontro de mais do que três arestas. No total são gerados 56 padrões de vértices distintos através deste procedimento.

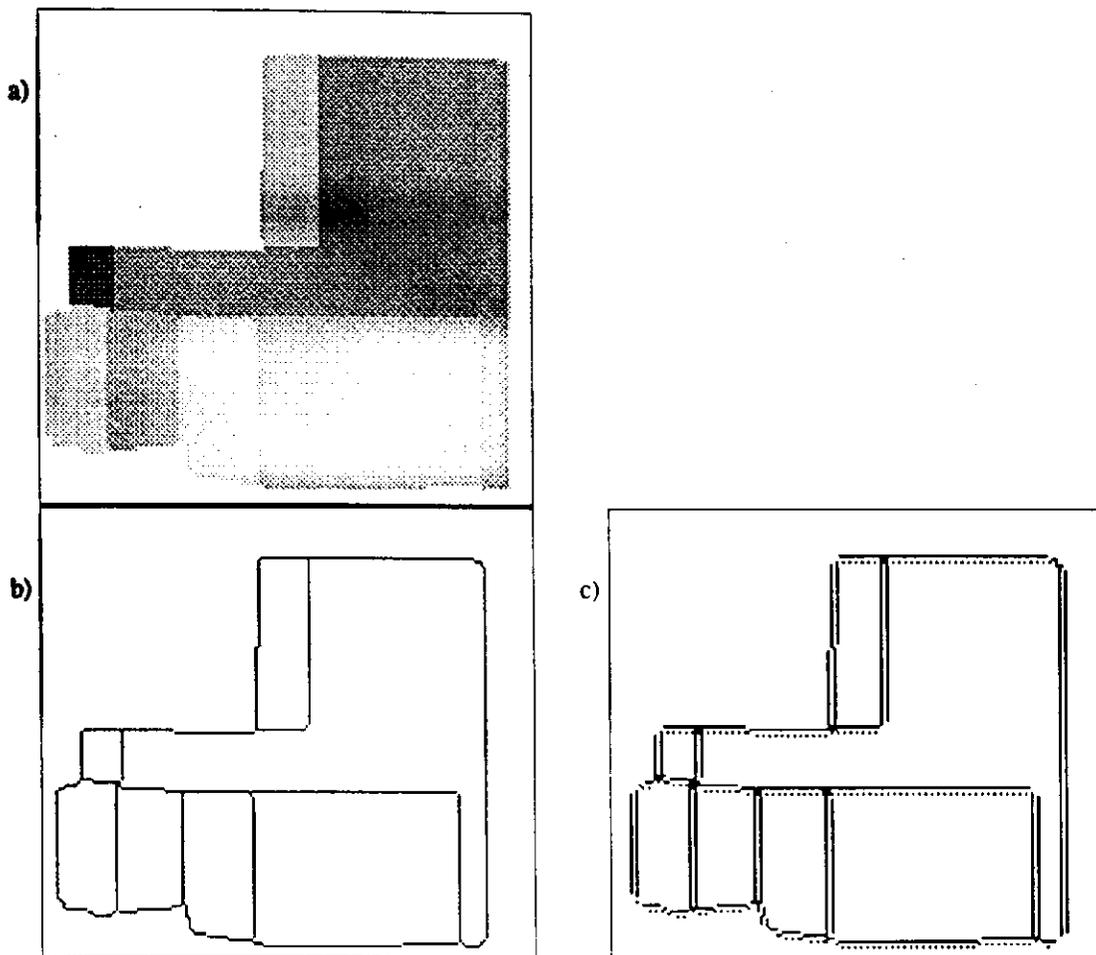


Figura 4.3: a) Imagem original, b) Imagem esqueletizada, c) imagem formada pelos vértices 3x3, com vértice falsos.

Ao se processar as imagens com os vértices gerados os possíveis ruídos e falhas decorrentes da aquisição e digitalização ocasionam a detecção de falsos candidatos a vértices. Entretanto o problema maior vem do fato de que não é exigido o *pixel* central da janela diferente de zero para que ali esteja um vértice, padrões de vértices similares ao exemplo da Figura 4.2-b geram uma quantidade inaceitável de falsos pontos de vértices, que chegam a formar uma nova aresta inteira, como ilustrado na Figura 4.3, para a vista lateral da cena 5.

Portanto, a utilização de padrões de vértices gerados em janelas de tamanho 3x3 que permitem que sejam reconhecidos os que possuem arestas de até um *pixel* de largura, não revelou-se uma solução adequada para o problema.

O próximo passo tentado foi aumentar o tamanho dos padrões de vértices de 3x3 para 5x5. Aproveitando os padrões já existentes, criou-se e implementou-se um algoritmo que os ampliasse, seguindo os caminhos ou direções possíveis de continuação das três arestas, a partir dos três *pixels* de suas extremidades (excluindo-se aqui o *pixel* central). Para tanto foi considerado um comportamento legítimo das arestas que deveriam ser aumentadas, tendendo a ficarem retas ou com pequenas quebras parecidas com degraus, o que é, por isso mesmo, conhecido como efeito “escadaria”. Estes caminhos possíveis são três para cada posição de *pixel* na janela 3x3 dos vértices e estão expostos na Figura 4.4. O algoritmo possui três laços aninhados, um para cada *pixel* de extremidade. Os dois *pixels* mais externos são fixados enquanto se varre todas as possibilidades de ocorrência do terceiro. Como resultado desta operação obtivemos um total de 1512 padrões de vértices de tamanho 5x5, formados pelo encontro de três arestas.

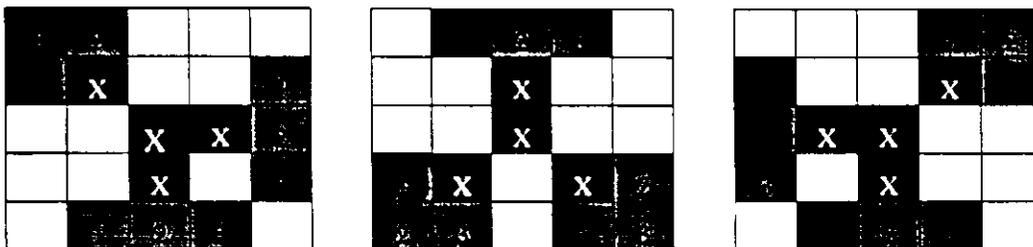


Figura 4.4.: Exemplos das possíveis continuações das arestas.

Os resultados obtidos ao processarmos estes novos vértices na imagem esquelizada da Figura 4.3-a, são mostrados na Figura 4.5, de duas maneiras: a) exigindo-se o *pixel* central da janela da imagem com valor 1, e b) sem exigí-lo.

Pode ser observado a grande quantidade de vértices falsos detectados em ambas as Figuras (pontos pretos), mas a quantidade maior é sem dúvidas na imagem 4.5.b. Pode ser observado também que estes vértices falsos estão aglomerados em torno de posições que realmente contem vértices. Portanto, o algoritmo precisa estar apto a escolher dentre tantos vértices quais são os verdadeiros.

A solução encontrada foi a de atribuir pesos aos 1512 padrões de vértices gerados. Deste modo, o uso de um filtro de máximo local permite que sejam selecionados aqueles vértices com maior valor de peso, dentro de uma janela. O algoritmo de filtro de máximo local está explicado na final de seção 4.3.

Uma análise da estrutura dos vértices falsos, revelou que, em sua maioria, estes são compostos por arestas que apresentam o chamado efeito “escadaria”. Com isso, a condição escolhida para a atribuição dos pesos aos vértices é: quanto maior for a ocorrência do efeito escadaria em um padrão de vértice, menor o peso correspondente. Como os padrões são formados por 3 arestas, foram definidos quatro pesos, listados abaixo em ordem decrescente:

Peso 4- nenhuma aresta com o efeito escadaria,

Peso 3- uma aresta com o efeito escadaria,

Peso 2- duas arestas com o efeito escadaria,

Peso 1- todas as três arestas com o efeito escadaria.

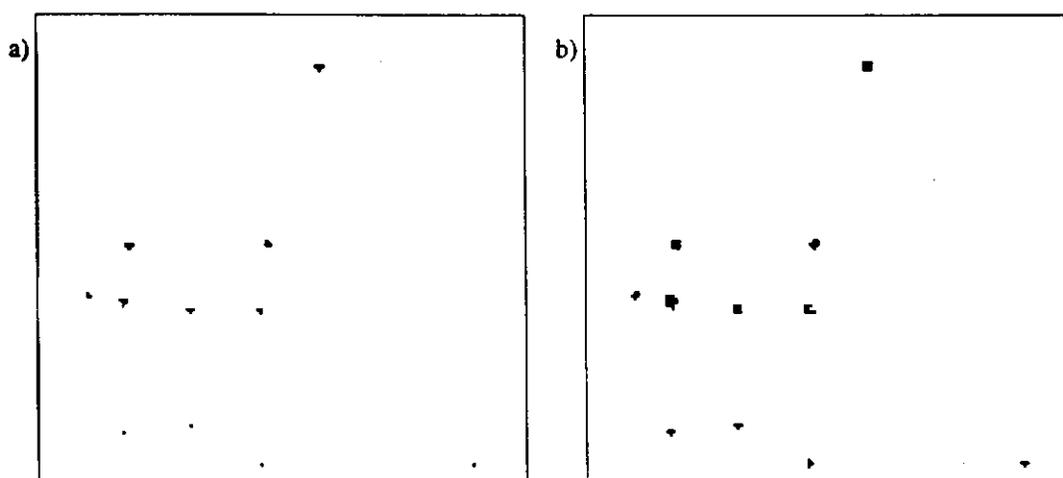


Figura 4.5.: Vértices de três arestas extraídos a) levando em conta o pixel central, b) sem levar em conta.

Exemplos de alguns vértices com seus respectivos pesos são mostrados na Figura 4.6.

Para a implementação do algoritmo filtro de máximo local, a princípio pensou-se em usar uma janela de tamanho 3x3, o que chegou até a ser implementado. Entretanto, observando as imagens com seus vértices extraídos, percebemos que apareceram vértices falsos com distância de até dois pixels de um vértice verdadeiro. Conseqüentemente aumentou-se o tamanho da janela do filtro para 5x5 a fim de garantir melhores resultados. Na Figura 4.7-a aparece um exemplo de uma imagem com os vértices extraídos desta vez com seus pesos (quanto mais escuro o ponto, maior o peso), e na 4.7-b a mesma imagem depois de se aplicar o filtro de máximo local com janela de tamanho 5x5.

Estando os resultados obtidos para vértices formados pelo encontro de três ou mais arestas satisfatórios, como consequência do trabalho aqui exposto, passa-se agora a apresentar o caso de vértices formados pelo encontro de apenas duas arestas.

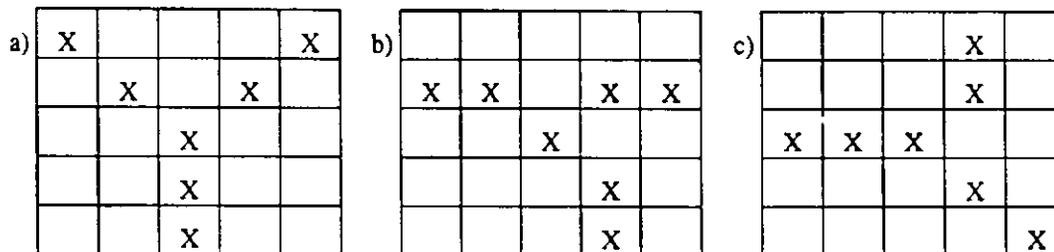


Figura 4.6.: Padrões de vértices com pesos, a) peso 4: todas as arestas retas, b) peso 1: todas as arestas com efeito escadaria, c) peso 3: arestas mistas.

4.2. Geração dos Vértices Formados por Duas Arestas

Usou-se a mesma idéia dos vértices formados por três arestas para criação dos vértices de duas; desenvolveu-se e implementou-se um algoritmo que gera dentro de uma janela 3x3 todas as combinações possíveis de três *pixels* sendo um fixo no centro, o que representa o encontro de duas arestas com um *pixel* de largura, excluindo-se obviamente aquelas combinações em que os três *pixels* pertençam à mesma reta. Foram gerados 24 padrões de vértices, ocasionando inicialmente os mesmos problemas já discutidos para o caso de vértices formados por três arestas, ou seja, a detecção de aglomerados de vértices falsos em torno de vértices verdadeiros.

A solução adotada para o problema acima descrito foi a mesma: aumentar os padrões de vértices de 3x3 para 5x5, através de um algoritmo que detecta os dois *pixels* de extremidade e calcula as três possíveis direções em que cada aresta pode crescer dependendo de suas posições dentro da janela (as mesmas direções da Figura 4.4), verificando se os *pixels* não pertencem a uma mesma reta. Isso é feito através de dois laços aninhados, um para cada pixel de extremidade. Desta maneira foram gerados 216 padrões de vértices formados pelo encontro de duas arestas em uma janela 5x5.

Neste ponto deparou-se com um fato que aconteceu na geração de ambos os padrões de vértices de tamanho 3x3, com duas e três arestas. Analisando os caminhos possíveis de se estender uma aresta, que são iguais em ambos os casos, percebeu-se que eles podem ser

coincidentes dependendo das posições dos seus pixels de extremidades, uma vez que os algoritmos que fazem esta ampliação o fazem de maneira exaustiva. Como mostra a Figura 4.8, o *pixel* sombreado com preto é um caminho possível de se estender as duas atestas sombreadas com *pixels* cinza.

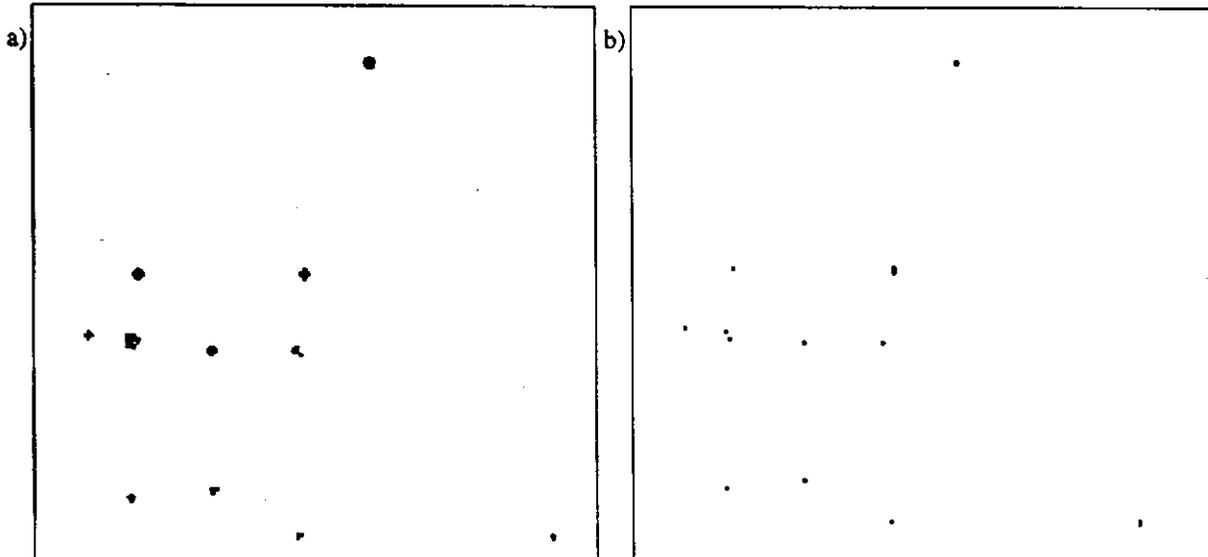


Figura 4.7.: A mesma imagem com a) vértices de três arestas extraídos com seus respectivos pesos, b) vértices depois do filtro de máximo local

Um algoritmo que detecta e elimina estes vértices, cujas arestas estendidas se cruzam, testando se a posição do *pixel* de extremidade de cada uma é o mesma, foi criado e implementado, e no final acarretou em uma diminuição do número de vértices de tamanho 5x5 de 1512 para 1335 para os formados pelo encontro de três arestas. Para o caso dos vértices formados por duas arestas a diminuição é de 216 para 208, o que não altera o resultado da extração dos vértices. A mesma imagem da Figura 4.3.a está apresentada na Figura 4.9 com seus vértices extraídos. Podemos observar que neste caso apareceram muitos vértices falsos, como no caso anterior de três arestas, contudo desta vez os vértices falsos não só estão aglomerados em torno de locais onde realmente temos vértices, eles também estão posicionados ao longo das arestas do contorno da imagem. Portanto, a solução aqui não pode ser a atribuição de pesos aos vértices.

Devido ao fato do tamanho da janela dos vértices ser 5x5, tem-se que os padrões de vértices formados por duas arestas só pesquisam o comportamento dos dois pixels mais próximos do ponto onde se localiza o vértice. Com isso podem ser reconhecidos como contendo vértices regiões em que não temos o encontro de duas arestas diferentes, e sim uma

única aresta em que um pixel esteja desalinhado dos demais. Esta situação pode ser natural da imagem ou decorrentes de falhas na aquisição e/ou digitalização das imagens. Para evitar este tipo de problema, deve-se examinar uma maior região da imagem analisada, conseqüentemente vértices de tamanho 5x5 tornam-se insuficientes.

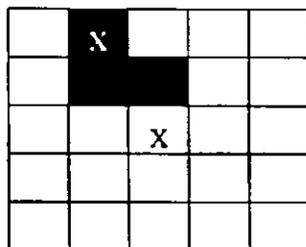


Figura 4.8.: Problema decorrente do aumento dos vértices.

Neste trabalho, criou-se e implementou-se um algoritmo para aumentar o tamanho dos padrões de vértice formados pelo encontro de duas arestas, de 5x5 para 7x7. O princípio utilizado é o mesmo do algoritmo que ampliou os padrões de 3x3 para 5x5, ou seja, estender as arestas a partir de suas extremidades. Entretanto, as direções que os prolongamentos das arestas podem tomar sofrem aqui uma alteração, para garantir um comportamento legítimo às mesmas, deixando-as retas ou com pequenos 'degraus' sempre no sentido o mais próximo possível do original da aresta. Em determinadas posições temos somente dois caminhos possíveis para continuar a aresta, em outros temos três direções a escolher, como vinha sendo feito até então. Exemplos das direções permitidas estão na Figura 4.10.

Com esse procedimento gerou-se 1103 padrões de vértices de tamanho 7x7 formados pelo encontro de duas arestas. Foi, então, executado o algoritmo já descrito anteriormente que elimina os vértices que tenham arestas que se interceptam, para ambos os tipos de vértices de tamanho 5x5 (formados por duas e por três arestas). Neste caso, para um limiar de distância de 3 pixels a partir do pixel central da janela, o número total de vértices foi reduzido de 1103 para 1089.

Após processamento das imagens com estes padrões de vértices, ainda pode-se verificar uma marcação insatisfatória de vértices falsos, apesar de em menor quantidade do que para o processamento com vértices 5x5, como mostrado na Figura 4.9. Contudo, alguns dos vértices falsos que não se encontram aglomerados ao redor de um verdadeiro, indicam locais onde as imagens sofrem uma pequena mudança de direção, o que não implica obrigatoriamente na

presença de um vértice. Estes vértices falsos apresentam uma característica comum: os ângulos entre suas arestas estão sempre compreendidos entre 120° e 180° .

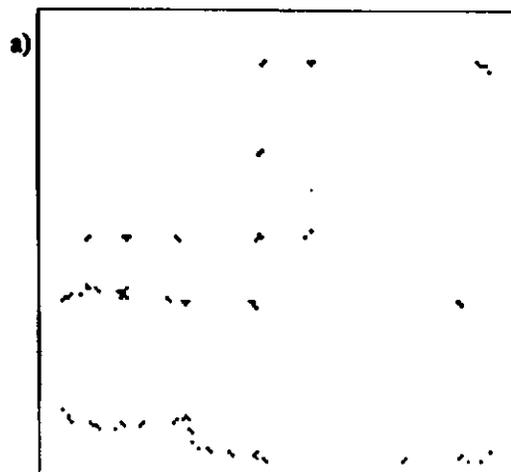


Figura 4.9.: Vértices extraídos de tamanho 5x5.

Não considerando relevantes vértices cujas arestas tenham ângulos entre si iguais ou maiores que 120° para efeito de descrição de modelos poliédricos de objetos, decidiu-se procurar por estes e eliminá-los do arquivo de vértices formados por duas arestas .

Considerando a futura implementação em hardware, cálculos que envolvem potenciação, radiciação, módulos, funções trigonométricas, e outras operações do gênero, para determinar o ângulo entre as arestas devem ser evitados. Uma solução alternativa é apresentada na Figura 4.11, onde percebe-se que a menor distância entre os últimos pixels de cada aresta (sombreados com preto), calculada em número de *pixels* do perímetro da janela 7x7 (sombreados com cinza), determina exatamente ao ângulo de 120° entre as mesmas. Consequentemente, calcula-se as duas possíveis distâncias entre os *pixels* de extremidade das arestas, percorrendo-se o contorno da máscara, uma no sentido horário e a outra no sentido anti-horário. Aqueles vértices cujas menores distâncias entre os últimos pixels de cada aresta forem maiores que 9 pixels serão descartados do arquivo final de vértices.

Após implementação e execução do algoritmo descrito acima, a quantidade de vértices de tamanho 7x7 formados pelo encontro de duas arestas foi reduzida de 1089 para 929. Para a atribuição de pesos a estes últimos vértices, a estratégia adotada foi a mesma usada para os vértices de três arestas, baseada na quantidade de arestas retas e arestas com o efeito “escadaria”:

- Peso 4: as duas arestas retas,

- **Peso 3:** uma aresta reta e a outra em degraus,
- **Peso 2:** as duas arestas em degraus.

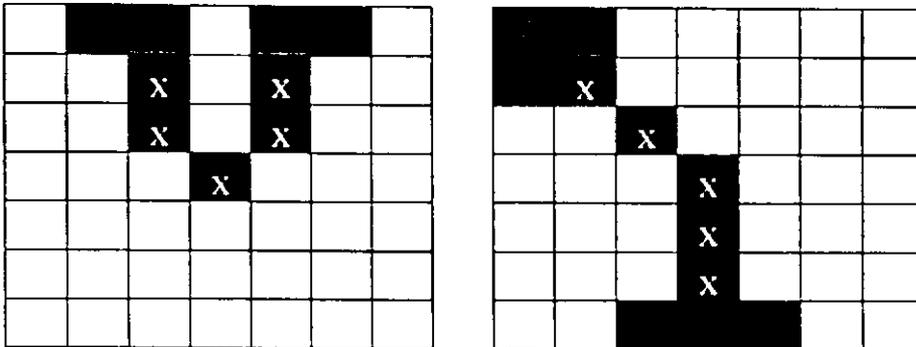


Figura 4.10.: exemplos das direções permitidas para os vértices 7x7.

Para os vértices formados pelo encontro de duas arestas não existe o peso 1. Os pesos começam do 4 para que a resposta do algoritmo de filtro de máximo local seja a mesma para ambos os tipos de vértices, aqueles formados pelo encontro de duas e de três arestas. Com isso é suficiente que este algoritmo seja executado uma única vez.

4.3. Extração dos Vértices

O algoritmo criado e implementado que procura fazer o casamento dos padrões de vértices armazenados, com janelas da imagem, trabalha com os dois tamanhos de janelas ao mesmo tempo, 5x5 e 7x7. Além disso, este algoritmo gerencia o deslizamento de ambas as janelas pela imagem de maneira independente mas simultânea, garantindo assim que a posição do *pixel* central de ambas as janelas na imagem original será sempre a mesma, durante todo o processamento.

O algoritmo primeiro tenta casar a janela 5x5 da imagem com algum padrão de vértice do arquivo dos vértices 5x5. Ao encontrar a primeira estrutura semelhante, marca o *pixel* do arquivo de saída, na posição correspondente ao do *pixel* central da janela da imagem, com o peso do vértice casado. Em seguida, desliza as duas janelas sobre a imagem, atualizando os *pixels* e recomeçando o procedimento de casamento. Caso não encontre nenhum padrão de vértice 5x5 que case com a estrutura da janela, ele então tenta casar a janela com os padrões de vértices 7x7. Ao encontrar a primeira estrutura semelhante, marca o *pixel* do arquivo de

saída na posição correspondente com o peso do vértice casado e desliza as janelas sobre a imagem, atualizando os *pixels*. Contudo não encontrando nenhum padrão de vértice que corresponda à janela da imagem, a saída será marcada com zero na respectiva posição do *pixel* central de ambas as janelas na imagem original.

X						
	X					
		X				
			X	X	X	X

Figura 4.11.: Cálculo do ângulo entre as arestas.

Visando diminuir o tempo de execução do algoritmo, poupando acessos a disco e comparações inúteis, antes de começar a procura de estruturas semelhantes entre os padrões de vértices e as janelas, é computada a quantidade de *pixels* diferentes de zero dentro de cada uma das janelas separadamente. Se em ambas as janelas da imagem, a 5x5 e a 7x7, não existir um mínimo de seis *pixels* diferentes de zero sem considerar o valor do *pixel* central, o procedimento de procura é interrompido para aquelas janelas, acontecendo então o deslizamento das mesmas sobre a imagem. O número mínimo de seis *pixels* dentro da janela da imagem foi escolhido por ser este o número de *pixels* que compõem os padrões de vértices de ambos os casos, de 2 e 3 arestas, sem ser considerado o *pixel* central do mesmo. Novas contagens serão feitas para as novas janelas, até que a condição seja satisfeita, e as tentativas de casamento propriamente ditas possam ser iniciadas.

Para que todos os *pixels* das imagens sejam avaliados, para decidirmos se são vértices ou não, todos devem ocupar a posição central das duas janelas da imagem, 5x5 e 7x7, uma e somente uma vez. Isto inclui os *pixels* das posições 0,0 e NxN, por exemplo, e todos os *pixels* localizados nas linhas e colunas do perímetro das imagens. Para tanto, as imagens são temporariamente envolvidas por “molduras” de *pixels*, que não tem efeito sobre o resultado final do processamento de extração de vértices, servindo apenas para preenchimento de todas as posições das janelas, quando estas estão deslizando pelos contornos das imagens. Estas “molduras” possuem três *pixels* de largura, todos preenchidos com valor zero.

O algoritmo de filtro de máximo local procura dentro de uma janela 5×5 , se há um outro *pixel* marcado com o peso maior que o do *pixel* central. Caso isto ocorra, o *pixel* central é marcado com zero, caso contrário, independente do seu peso, será marcado como o maior peso no arquivo de saída, peso 4, que corresponde a cor preta.

Os dois arquivos finais de padrões de vértices, o dos formados pelo encontro de duas arestas em uma janela de tamanho 7×7 , e o dos formados pelo encontro de três arestas em uma janela de tamanho 5×5 , estão ordenados pelo peso dos vértices, do maior (peso 4) para o menor. Assim fica garantido que o primeiro vértice casado com a janela da imagem será sempre o de maior peso.

4.4. Correspondência de Vértices nas Multi-Vistas

Como os vértices encontrados no final das duas primeiras seções estão aglomerados em torno da posição onde se localiza um vértice verdadeiro, a última etapa do processamento é então a seleção, dentre estes vértices, do que melhor representa o verdadeiro. Isto é feito através da utilização do algoritmo que faz a correspondência dos vértices nas vistas. Este baseia-se no algoritmo de propósito semelhante descrito e utilizado no trabalho de Farias [1].

Considere a situação apresentada na Figura 4.12, onde pode ser encontrado um vértice visível nas três vistas. Na vista superior tem-se um vértice V_1 com coordenadas (y_1, x_1) e espera-se encontrar a coordenada z_1 nos vértice V_2 e V_3 correspondentes nas demais vistas. Para isto, faz-se uma pesquisa nas vistas lateral e frontal, nesta ordem, tentando casar x_1 com x_2 e y_1 com y_2 . Portanto, espera-se encontrar um vértice que esteja visível na vista superior e no mínimo em uma das duas outras vistas, respectivamente: a lateral ou a frontal.

Uma janela de tamanho 10×10 varre a imagem até que um ponto marcado como vértice ocupe a posição central. Caso existam outros pixels diferentes de zero dentro da janela, um único par de coordenadas $V_m(y_m, x_m)$ representando a média das coordenadas destes pontos será utilizado na pesquisa nas vistas. Todos os vértices originais são apagados da imagem, ficando somente o *pixel* V_m equivalente às suas médias. Desliza-se a janela sobre a imagem até que outro vértice ocupe seu centro. Caso o *pixel* V_m de coordenadas (y_m, x_m) se encontre dentro dos novos limites da janela 10×10 também será empregado no cálculo de um novo

vértice médio V_m e será, como os outros, também apagado, após este cálculo.

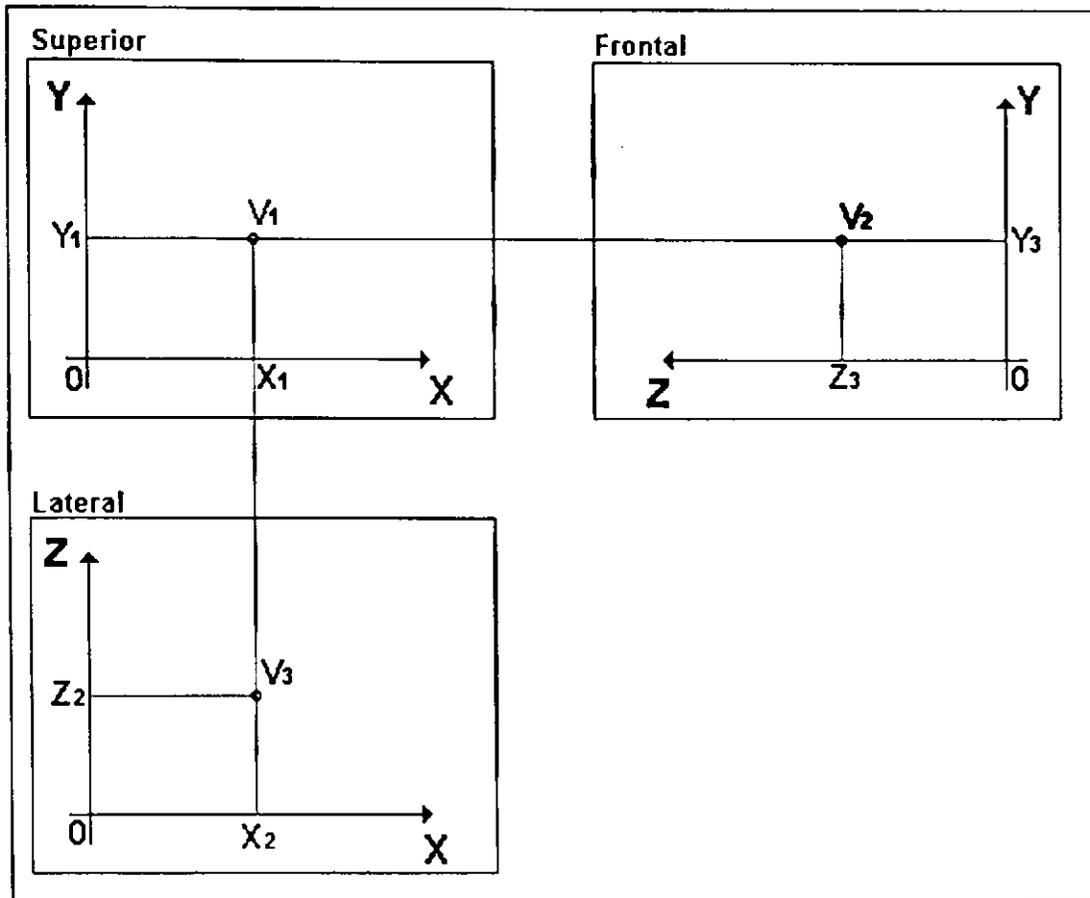


Figura 4.12.: Correspondência de Vértices nas Multi-Vistas.

Somente os vértices resultantes do processo acima descrito são utilizados no procedimento de casamento, que começa pela vista superior. O casamento entre as coordenadas nas pesquisas às vistas lateral e frontal ocorre, respectivamente, quando são satisfeitos os seguintes critérios de erro:

$$|x_1 - x_2| \leq \text{Erro}$$

$$|y_1 - y_2| \leq \text{Erro}$$

Sendo encontrado pelo menos um valor de x_2 ou de y_2 que satisfaça estas condições, o vértice é validado. Observe que se a pesquisa na vista lateral não tiver sucesso, existe ainda a possibilidade de sucesso na pesquisa da vista frontal. Caso as duas pesquisas não tenham sucesso o vértice testado é considerado indeterminado e conseqüentemente é rejeitado.

Na análise das vistas lateral e frontal, faz-se apenas a pesquisa na vista superior

considerando os critérios de erro abaixo:

$$|x_2 - x_1| \leq \text{Erro}$$

$$|y_3 - y_1| \leq \text{Erro}$$

Como no caso da análise da vista superior, se for encontrado ao menos um vértice que satisfaça a condição o vértice é validado, caso contrário é rejeitado.

Estes procedimentos são repetidos até que todos os vértices de cada vista sejam investigados.

O limiar de erro *Erro* utilizado nas condições acima corresponde a um erro fixo medido em dez unidades de pixels. Este valor é empírico e depende das condições em que foram adquiridas as imagens, quanto melhor as imagens o erro será menor, servindo para deixar uma margem de segurança aos erros que podem ocorrer nas dimensões das imagens (capítulo 5) e extração dos vértices.

As imagens utilizadas nos testes efetuados para avaliar o presente trabalho, bem como os resultados obtidos podem ser encontrados no próximo capítulo.

Capítulo 5

Testes Efetuados e Resultados Obtidos.

Neste Capítulo, a metodologia proposta para o reconhecimento de vértices tridimensionais em imagens multi-vista é avaliada através dos resultados obtidos experimentalmente.

Os algoritmos utilizados para o processamento das imagens foram todos implementados em linguagem C seguindo o padrão ANSI, garantindo assim a portabilidade dos mesmos. Os ambientes onde os algoritmos foram desenvolvidos foram três, de acordo com as disponibilidades de utilização de equipamentos nos laboratórios LABCOM (Laboratório de Computação), do Departamento de Ciências e Computação e LAPS (Laboratório de Automação e Processamento de Sinais), do Departamento de Engenharia Elétrica: máquinas PENTIUM com WindowsTM 9x e NT, e estações de trabalho IBM com Unix AIXTM.

Nas seções seguintes há uma breve apresentação das imagens utilizadas nos testes, seguida de uma discussão sobre os mesmos e da apresentação e análise dos resultados obtidos.

5.1. As Imagens

Serão descritas nesta seção seis imagens obtidas de duas cenas diferentes que contém objetos reais, originalmente utilizadas no trabalho apresentado em [1]. As três imagens da primeira cena foram obtidas com câmera fotográfica comum, seguida de posterior digitalização das mesmas através de scanner óptico. As outras três (da segunda cena) foram obtidas com câmera digital. Em cada caso, foi utilizada uma única câmera fotográfica para se obter as três diferentes vistas de cada uma das cenas, como mostrado na Figura 1.4.

Figuras	Vistas	Dimensões
Cena 1	Superior	410 x 682
	Lateral	232 x 677
	Frontal	232 x 408
Cena 2	Superior	337 x 374
	Lateral	199 x 374
	Frontal	202 x 339
Cena 3	Superior	718 x 574
	Lateral	460 x 583
	Frontal	464 x 719
Cena 4	Superior	438 x 295
	Lateral	190 x 295
	Frontal	188 x 435
Cena 5	Superior	273 x 390
	Lateral	175 x 392
	Frontal	174 x 272
Cena 6	Superior	409 x 279
	Lateral	189 x 284
	Frontal	187 x 407
Cena 7	Superior	433 x 286
	Lateral	178 x 290
	Frontal	177 x 426
Cena 8	Superior	174 x 180
	Lateral	128 x 181
	Frontal	128 x 170

Tabela 5.1.: Imagens testadas com suas respectivas dimensões.

As imagens foram digitalizadas em 256 níveis de cinza com taxa de amostragem de 75 pixels por polegada. A distância média entre os objetos e a câmera foi de 60 cm, o comprimento focal das lentes foi de 100 mm e o fator de ampliação foi de 4.16 para as imagens obtidas com câmera comum e 1 para as imagens obtidas com câmera digital. Após a aquisição, as imagens foram convertidas para arquivos de pixels em ASCII, com extensão .PIX, com seus níveis de cinza reduzidos para apenas 32. Esta conversão visa facilitar as etapas de seu processamento.

Figuras	Vistas	Vértices Verdadeiros	Vértices Encontrados	Vértices Verdadeiros Casados	Vértices Falsos Casados	Erro I (%)	Erro II (%)
Cena 1	Superior	16	44	16	1	0.00	6,25
	Lateral	21	66	20	0	4.76	0.00
	Frontal	13	20	13	0	0.00	0,00
Cena 2	Superior	12	19	11	1	8.33	8.33
	Lateral	16	39	12	0	25.00	0.00
	Frontal	16	35	12	0	25.00	0.00
Cena 3	Superior	12	34	12	0	0.00	0,00
	Lateral	27	52	27	2	0.00	7,41
	Frontal	18	33	18	1	0.00	5,55
Cena 4	Superior	16	36	15	1	6.25	6.25
	Lateral	15	42	12	1	20.00	6.66
	Frontal	22	47	20	0	10.00	0,00
Cena 5	Superior	14	36	14	1	0.00	7,14
	Lateral	22	42	22	0	0.00	0,00
	Frontal	20	41	17	0	15.00	0,00
Cena 6	Superior	8	25	8	1	0.00	12,50
	Lateral	8	20	8	0	0.00	0,00
	Frontal	7	11	7	0	0.00	0,00
Cena 7	Superior	10	14	10	0	0.00	0,00
	Lateral	17	58	15	4	11.76	23.53
	Frontal	14	29	14	0	0.00	0,00
Cena 8	Superior	8	20	8	1	0.00	12,50
	Lateral	13	47	12	0	7.69	0.00
	Frontal	6	13	6	0	0.00	0,00

Tabela 5.2.: Resumo de Resultados.

Cada imagem tem suas dimensões particulares, e estas encontram-se apresentadas na Tabela 5.1, no formato cartesiano (Linha X Coluna). Para efeito de visualização neste trabalho, diminuiu-se o fundo deixando apenas a figura do objeto em perspectiva. As vistas superiores sempre representam os planos Y-X (largura-comprimento) no sistema de coordenadas cartesianas, as vistas laterais representam os planos Z-X (largura-altura) e as

vistas frontais representam os planos Z-Y (comprimento-altura). A origem, ou seja, o par de coordenada (0,0) corresponde ao canto inferior esquerdo da mesa onde os objetos estão dispostos na imagem frontal.

Como pode ser observado na terceira coluna da tabela 5.5, nas vistas de uma mesma cena existem casos em que uma dada dimensão apresenta valores distintos. Esta variação, corresponde a um número bem pequeno de pixels, em média apenas dois. Um exemplo, é a cena 1, que na vista superior tem a dimensão de 390 pixels medidos ao longo do eixo X, enquanto na vista lateral esta mesma dimensão alcança 392 pixels. Esta discrepância é decorrente de falhas nos processos de aquisição e digitalização das imagens, que, conforme já foi explicado, é manual. Entretanto, este fato não é gerador de erros, devido à margem de tolerância existente no algoritmo que faz a procura por vértices correspondentes nas três vistas.

Na da cena 2 tem-se dois objetos superpostos que não podem ser vistos separadamente sob nenhum ponto de vista, enquanto que na da cena 5 temos a presença de dois objetos que, dependendo da posição do observador, podem ser vistos separadamente ou com oclusão parcial.

5.2. Os resultados

As imagens resultantes de cada etapa do processamento encontram-se nas Figuras 5.1 a 5.8, nas quais podem ser vistas a imagem original e as imagens de saída das três etapas do pré-processamento descrito no Capítulo 3: a imagem gradiente, a imagem gradiente binarizada e a imagem esqueletizada do objeto. Na seqüência tem-se as duas imagens do processamento para extração de contornos: os candidatos a vértices com seus respectivos pesos e os que restaram depois do filtro de máximo local (também descritos no Capítulo 3), e a imagem final dos vértices do(s) objeto(s) da cena de entrada, após o algoritmo que procura a correspondência dos vértices nas multi-vistas.

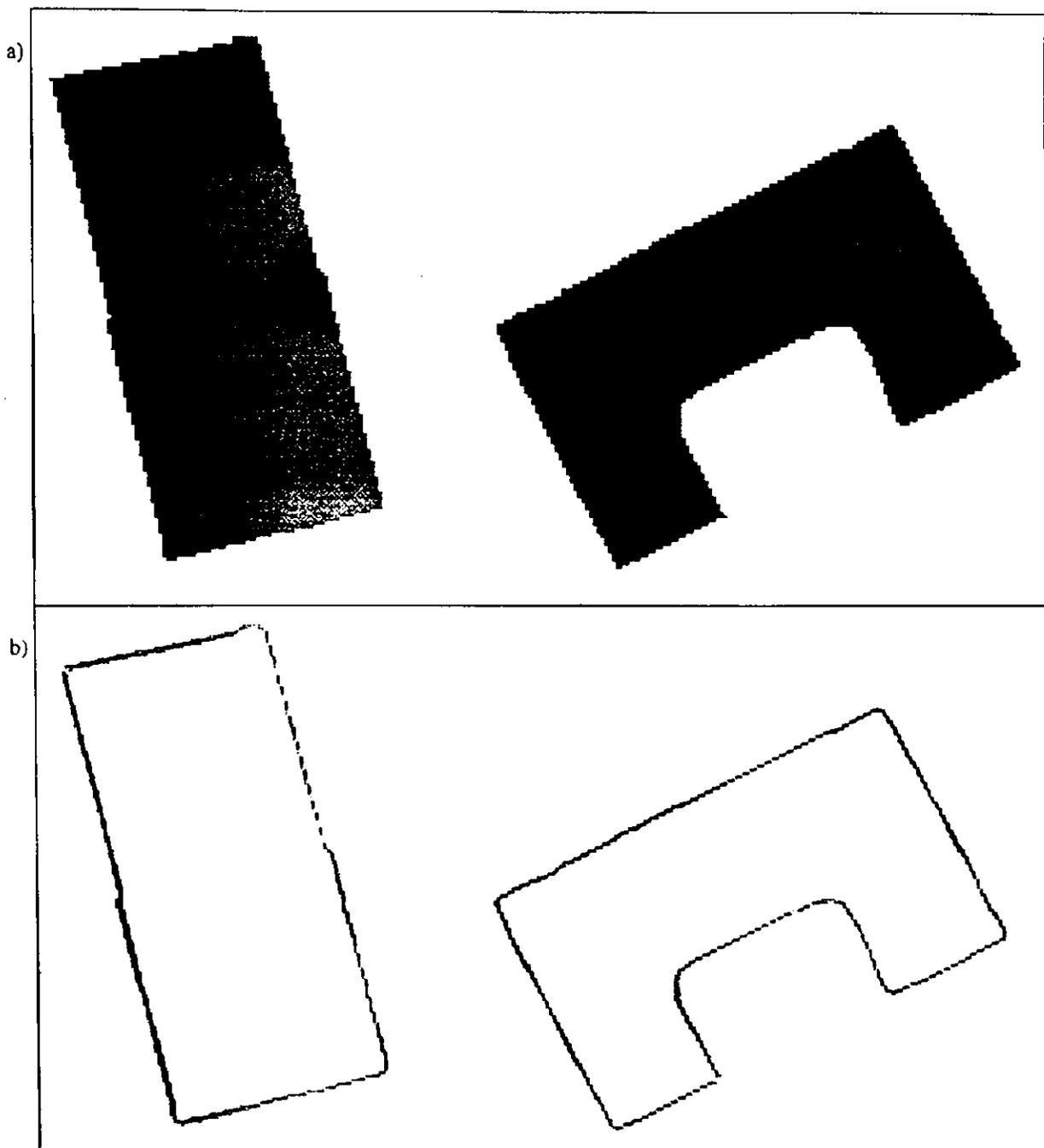


Figura 5.1.: Cena 5 – Vista Superior: Etapas do processamento

- a) imagem original, b) imagem gradiente,*
- c) imagem binarizada, d) imagem esqueletizada,*
- e) vértices extraídos com seus pesos,*
- f) vértices após filtro de máximo local 5x5,*

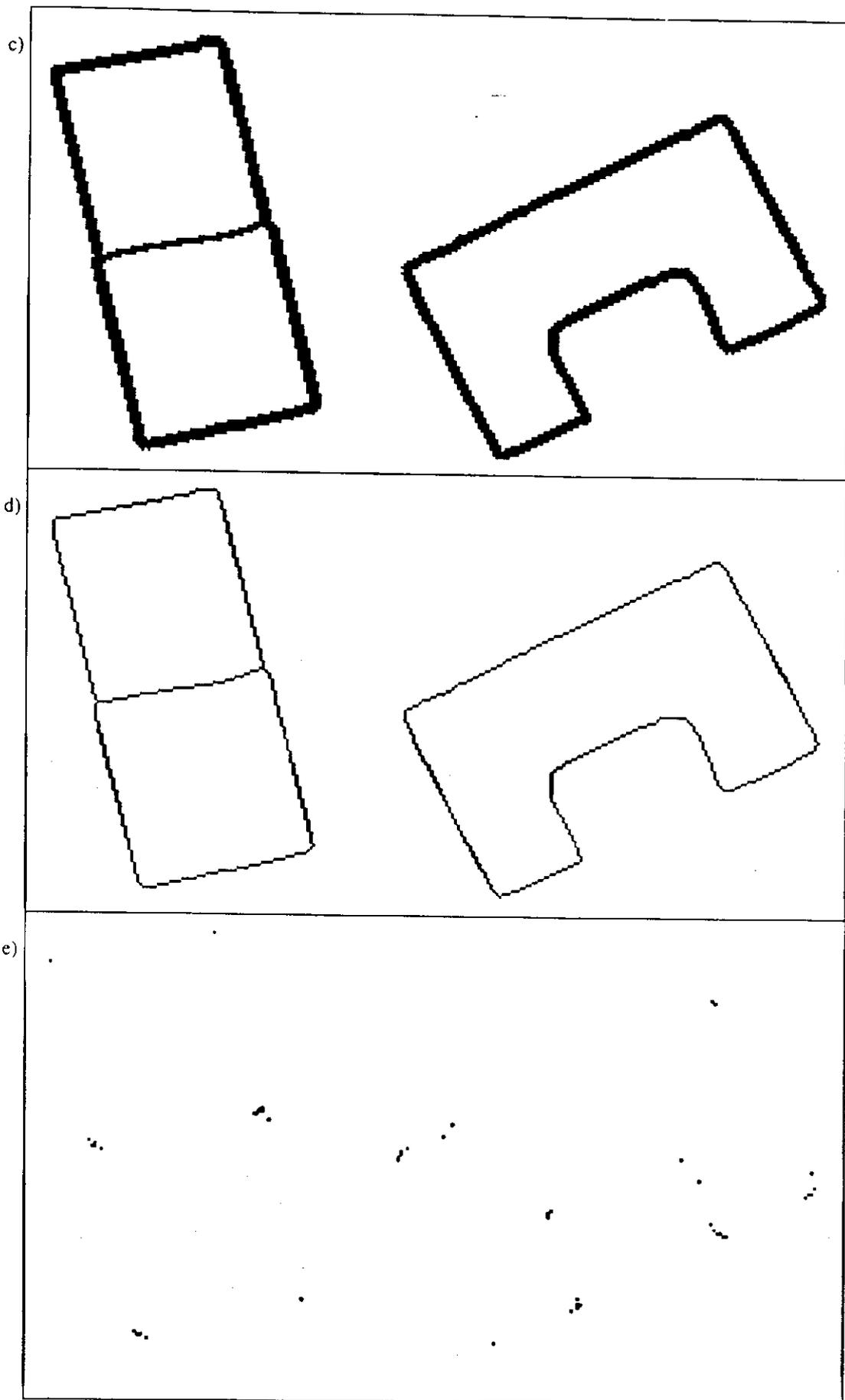


Figura 5.1.: Continuação.

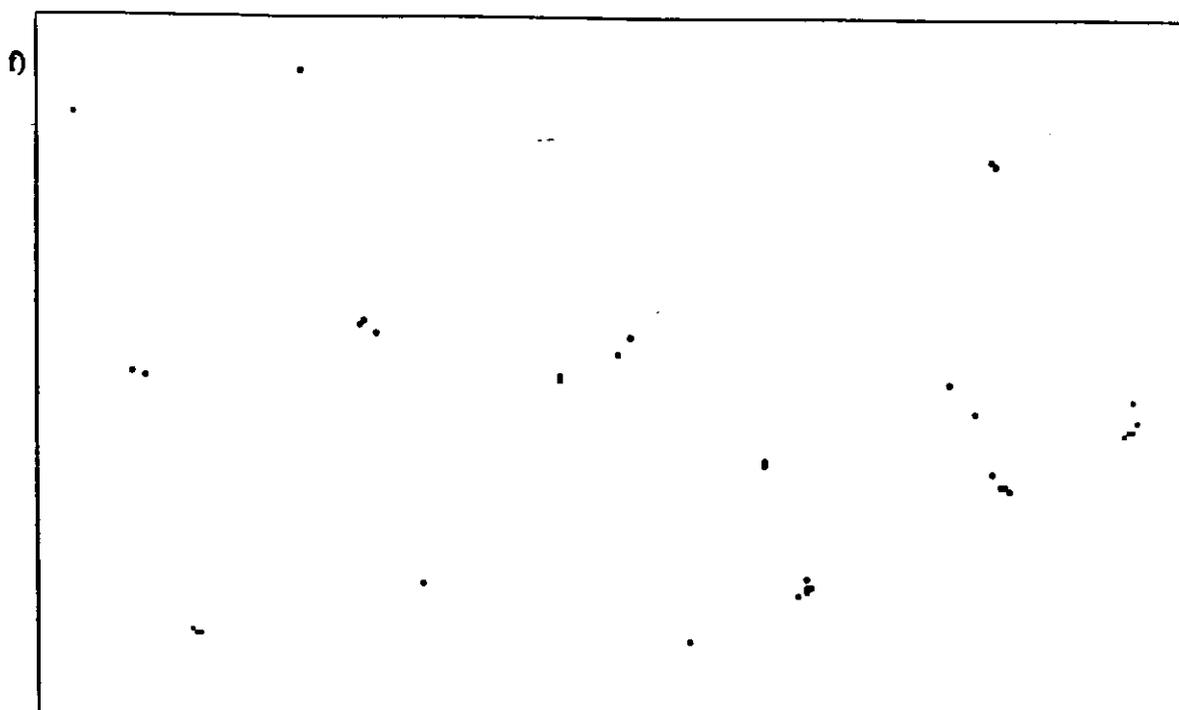


Figura 5.1.: Continuação.

A Tabela 5.2. apresenta um resumo dos resultados para as cenas utilizadas nos testes, que estão listadas na primeira e segunda colunas. A terceira coluna da tabela contabiliza o número total de vértices dos objetos presentes em cada imagem, segundo os modelos presentes na biblioteca, respeitando os pontos de vista e as oclusões parciais, quando estas existem. Esta coluna é utilizada como referencial de desempenho do método proposto. Na quarta coluna tem-se a quantidade de vértices encontrados no final do processamento das vistas das cenas individualmente. Na quinta coluna podem ser encontrados o números de vértices verdadeiros casados para cada imagem, resultado do processamento com o algoritmo que procura a correspondência dos vértices nas três vistas. Na sexta coluna tem-se o número de vértices falsos casados por vista, após o mesmo processamento.

As duas últimas colunas apresentam as taxas de erro, expressas em porcentagem, para cada vista individualmente. A taxa de erro *Erro I* que pode ser visto na sétima coluna representa a quantidade de vértices verdadeiros que não foram casados ao final do processamento. Este erro é calculado por:

$$\frac{|n^{\circ} \text{vértices verdadeiros} - n^{\circ} \text{vértices verdadeiros casados}| \times 100}{n^{\circ} \text{de vértices verdadeiros}}$$

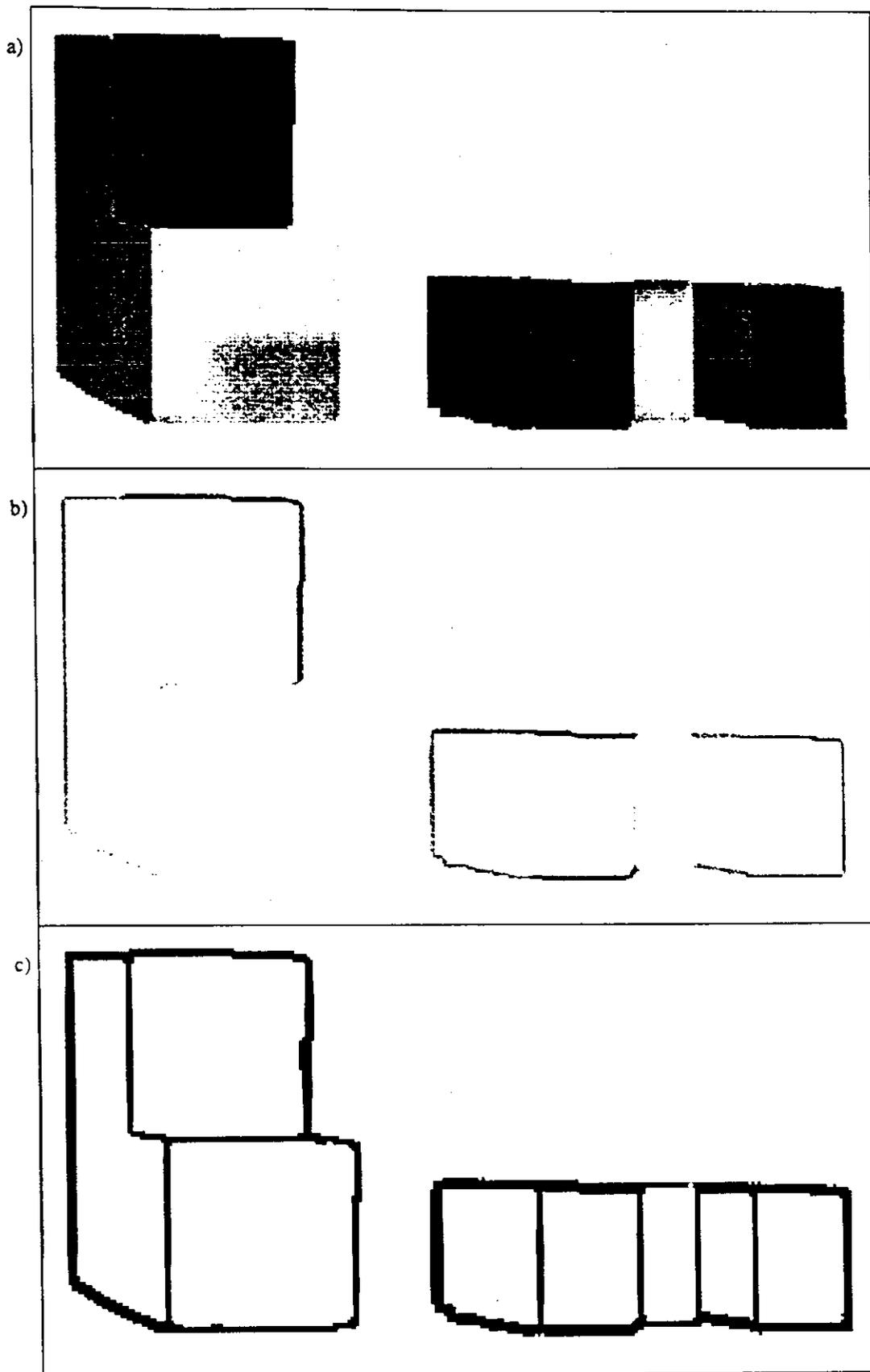
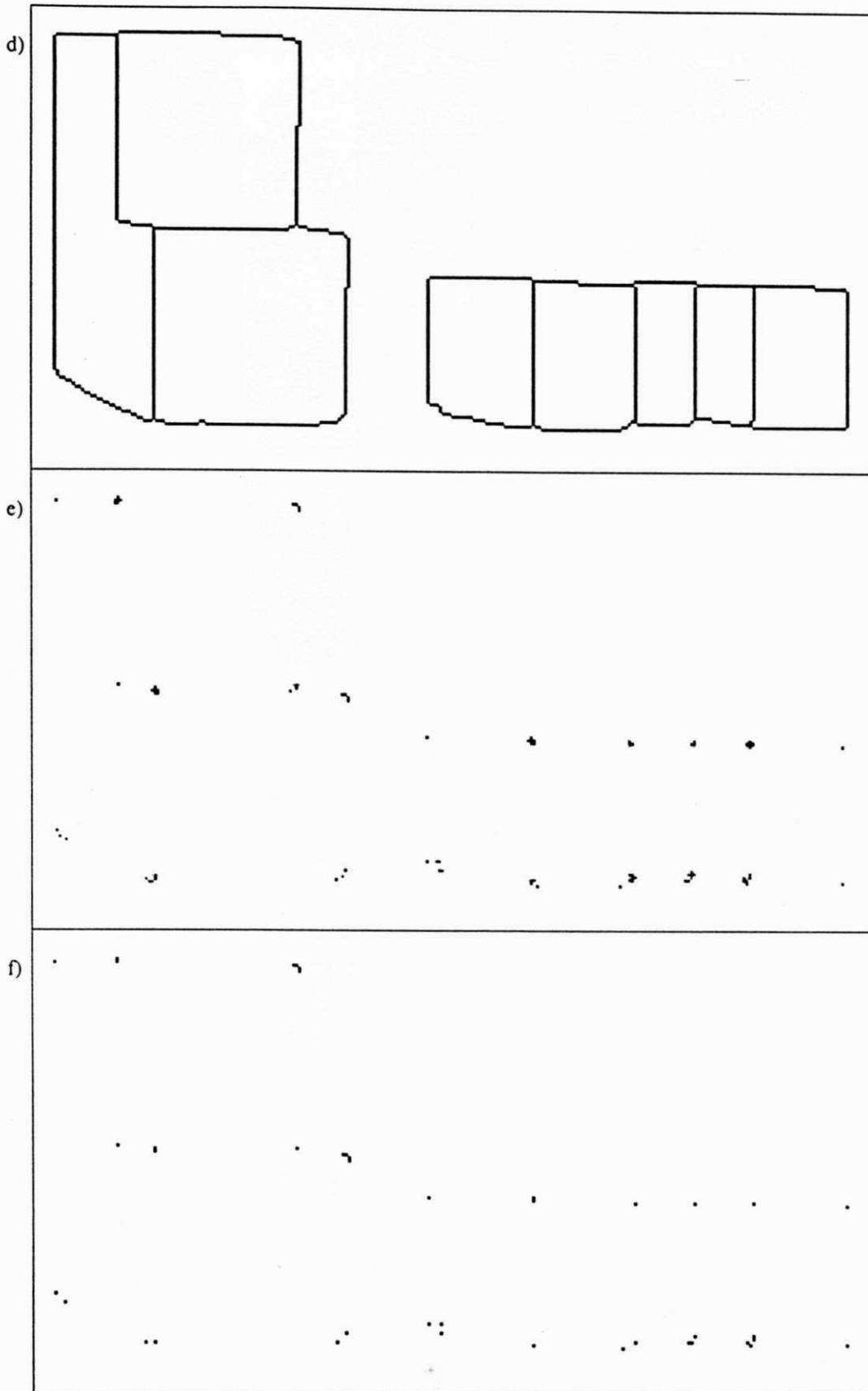


Figura 5.2.: Cena 5 – Vista Lateral: Etapas do processamento

a) imagem original, b) imagem gradiente,

c) imagem binarizada.



*Figura 5.2.: Continuação: d) imagem esqueletizada,
e) vértices extraídos com seus pesos,
f) vértices após filtro de máximo local 5x5.*

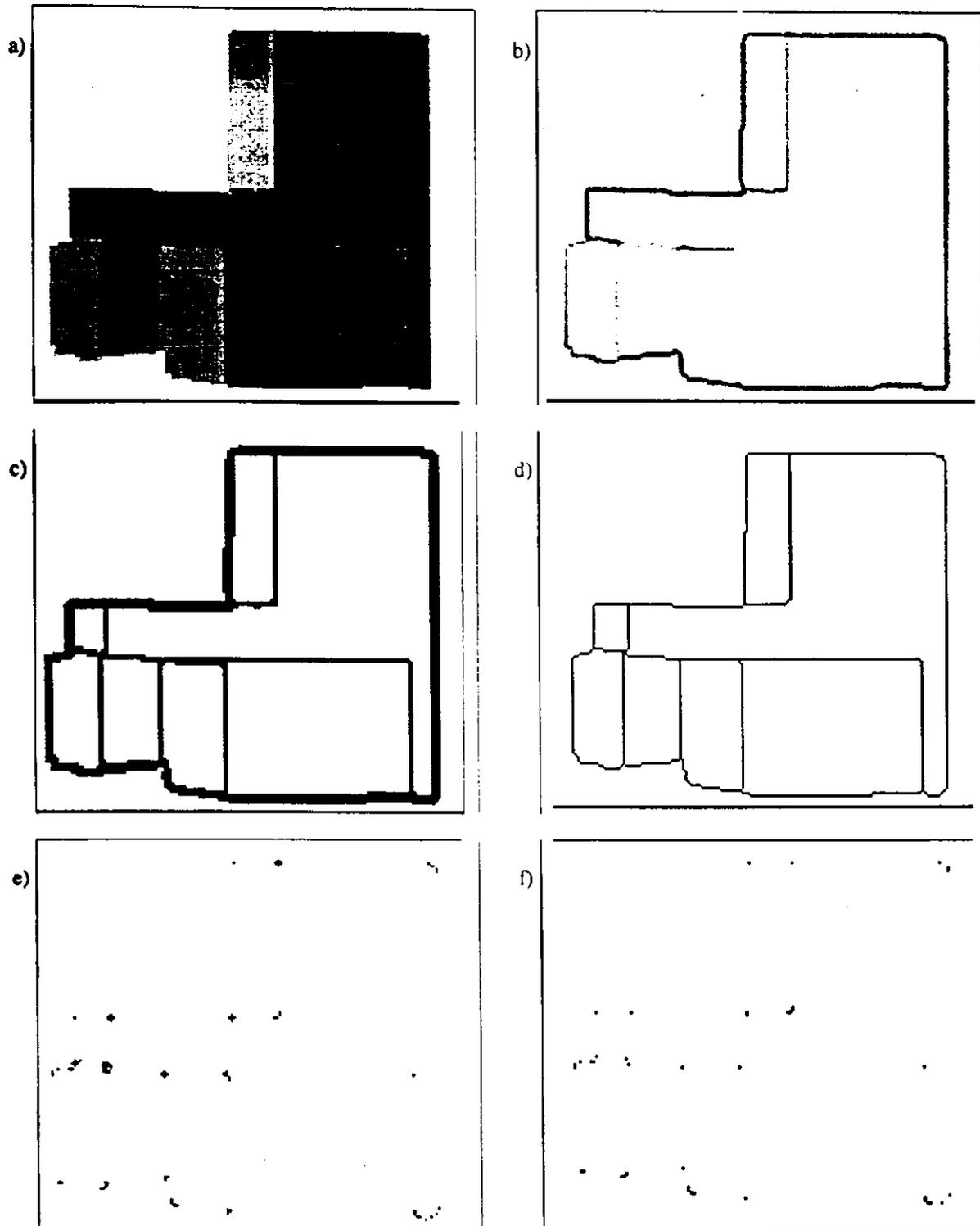


Figura 5.3.: Cena 5 – Vista Frontal: Etapas do processamento

- a) imagem original, b) imagem gradiente,*
- c) imagem binarizada, d) imagem esqueletizada,*
- e) vértices extraídos com seus pesos,*
- f) vértices após filtro de máximo local 5x5.*

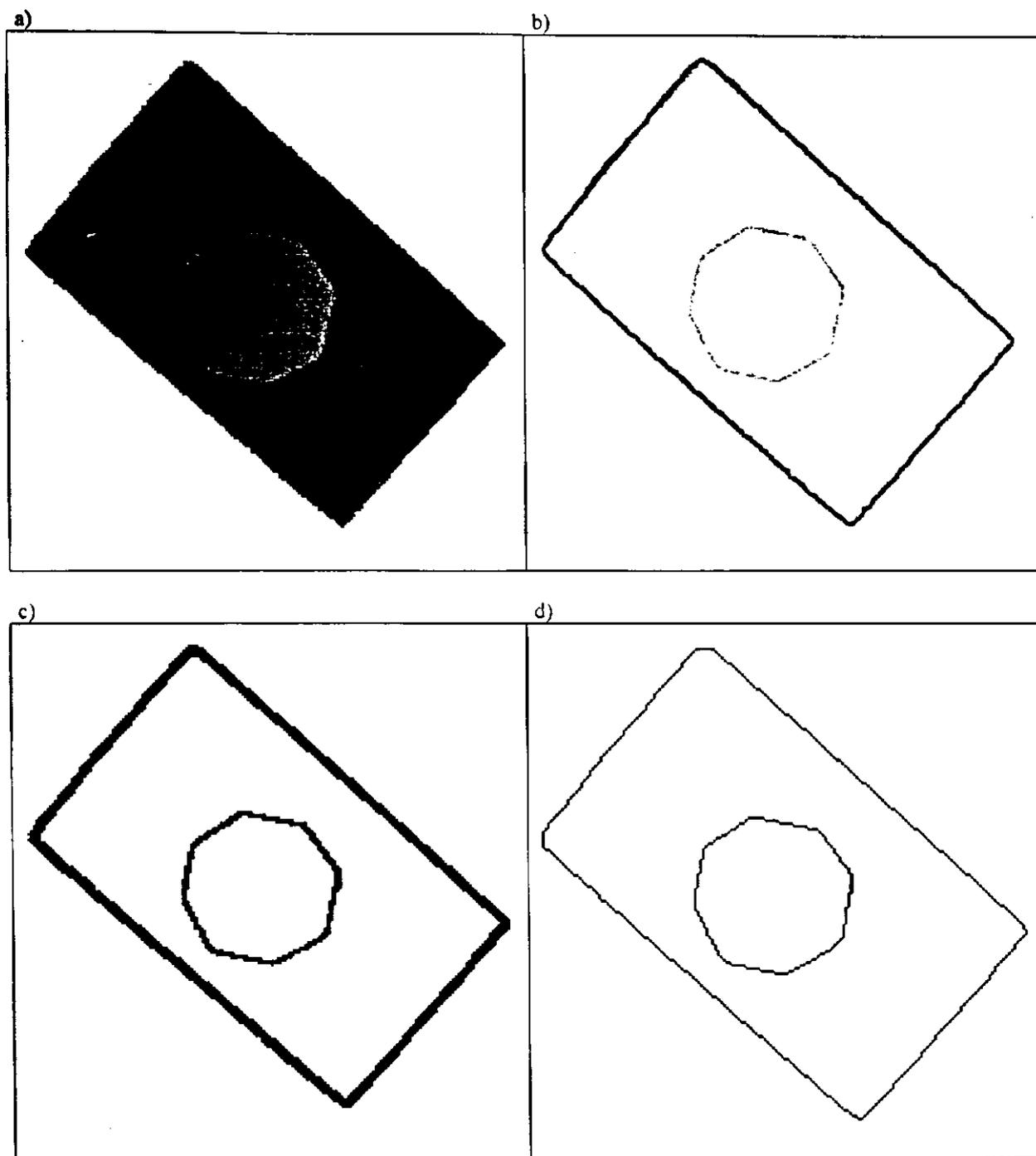


Figura 5.4.: Cena 2 – Vista Superior : Etapas do processamento

- a) imagem original, b) imagem gradiente,*
- c) imagem binarizada, d) imagem esqueletizada,*
- e) vértices extraídos com seus pesos,*
- f) vértices após filtro de máximo local 5x5.*

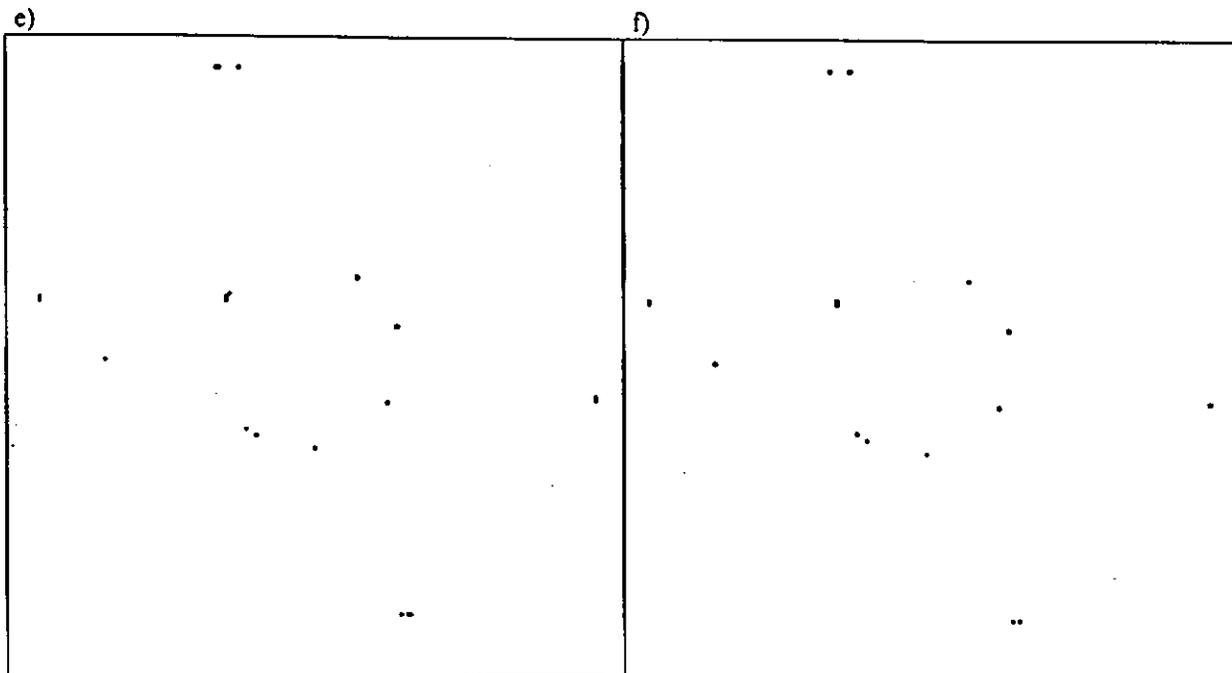


Figura 5.4.: Continuação.

A taxa de erro *Erro II* apresentado na última coluna contabiliza a quantidade de vértices falsos marcados como verdadeiros no final de todo o processamento, e é expressa por:

$$\frac{(\text{n}^\circ \text{vértices falsos casados}) \times 100}{\text{n}^\circ \text{vértices verdadeiros}}$$

Os vértices encontrados a mais representam locais nas arestas onde se tem mudanças de direção menores que 120°, decorrente de problemas na aquisição ou digitalização, que não existem no objeto real, como apresentado na Figura 5.9, onde pode-se ver a vista superior da cena 5, com uma região (marcada por uma elipse) ampliada embaixo.

Os vértices não encontrados, são justificados pelo algoritmo que faz a correspondência dos vértices nas multi-vistas, que exige uma distância mínima de dez pixels entre vértices. Esta situação é ilustrada na Figura 5.10, onde pode-se ver a vista lateral da cena 2 com uma região (marcada por uma elipse) ampliada embaixo. No final do processamento na região marcada só existirá um vértice.

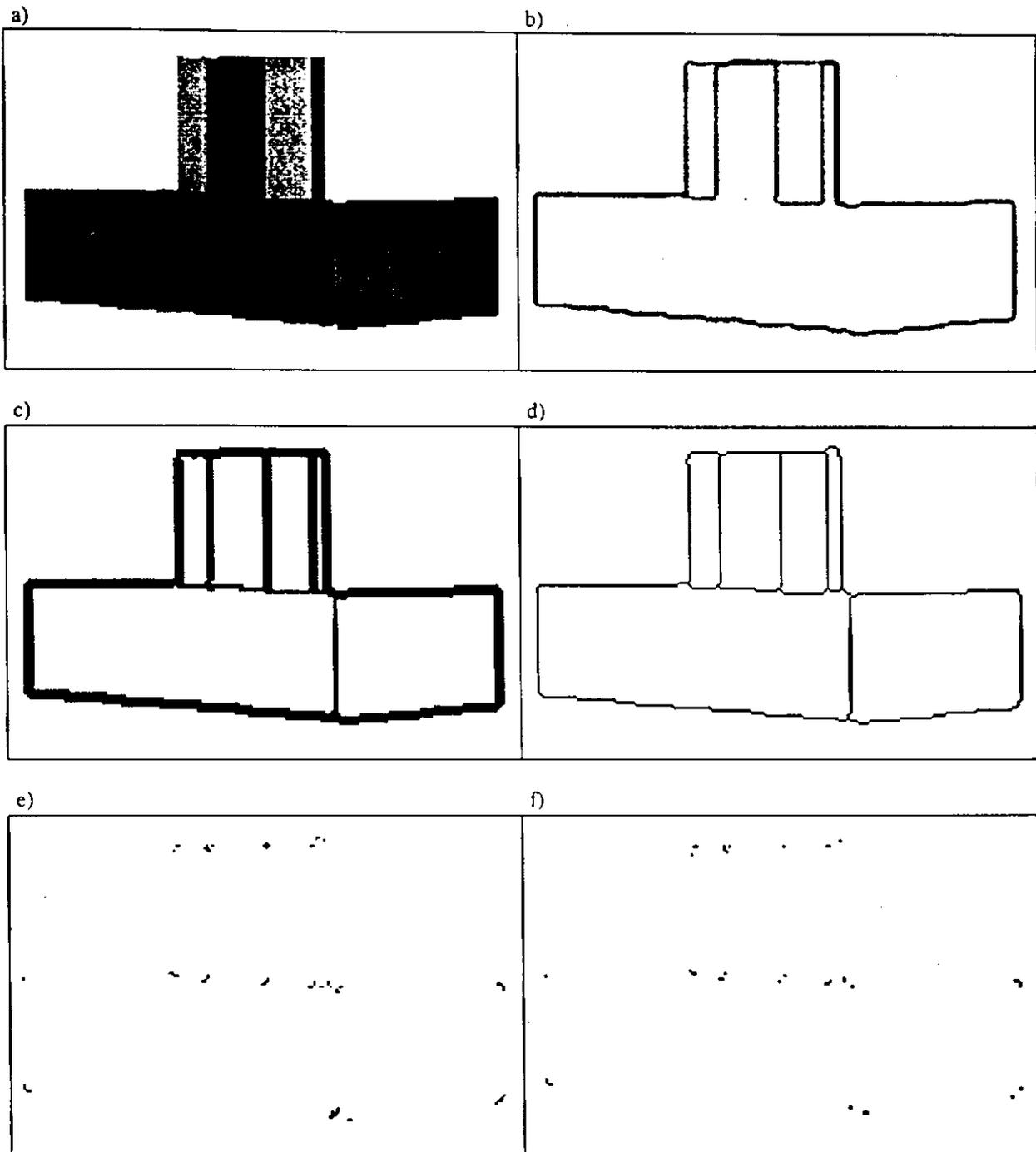


Figura 5.5.: Cena 2 – Vista Lateral : Etapas do processamento

- a) imagem original, b) imagem gradiente,*
- c) imagem binarizada, d) imagem esqueletizada,*
- e) vértices extraídos com seus pesos,*
- f) vértices após filtro de máximo local 5x5.*

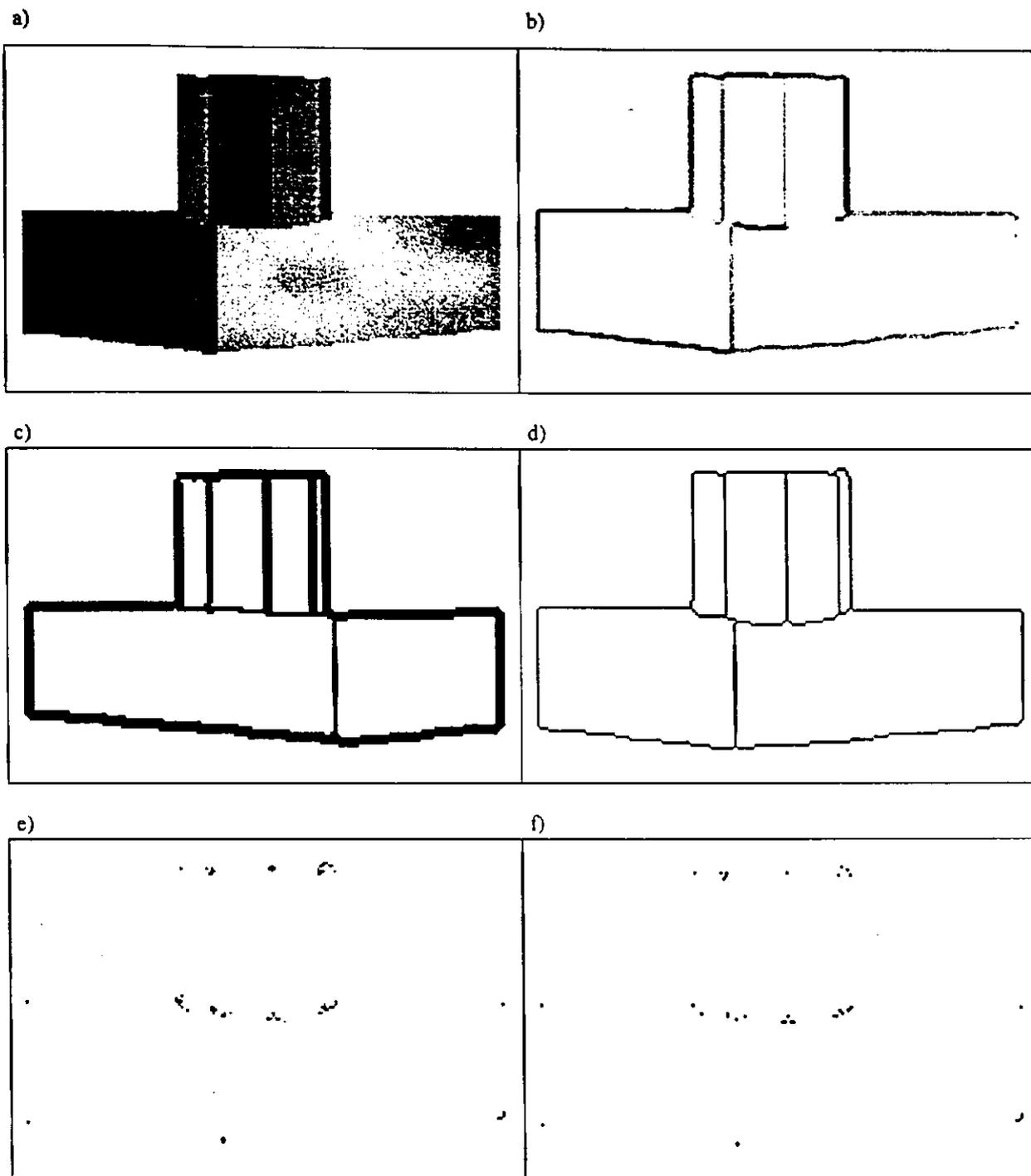


Figura 5.6.: Cena 2 – Vista Frontal : Etapas do processamento

- a) imagem original, b) imagem gradiente,
- c) imagem binarizada, d) imagem esqueletizada,
- e) vértices extraídos com seus pesos,
- f) vértices após filtro de máximo local 5x5.

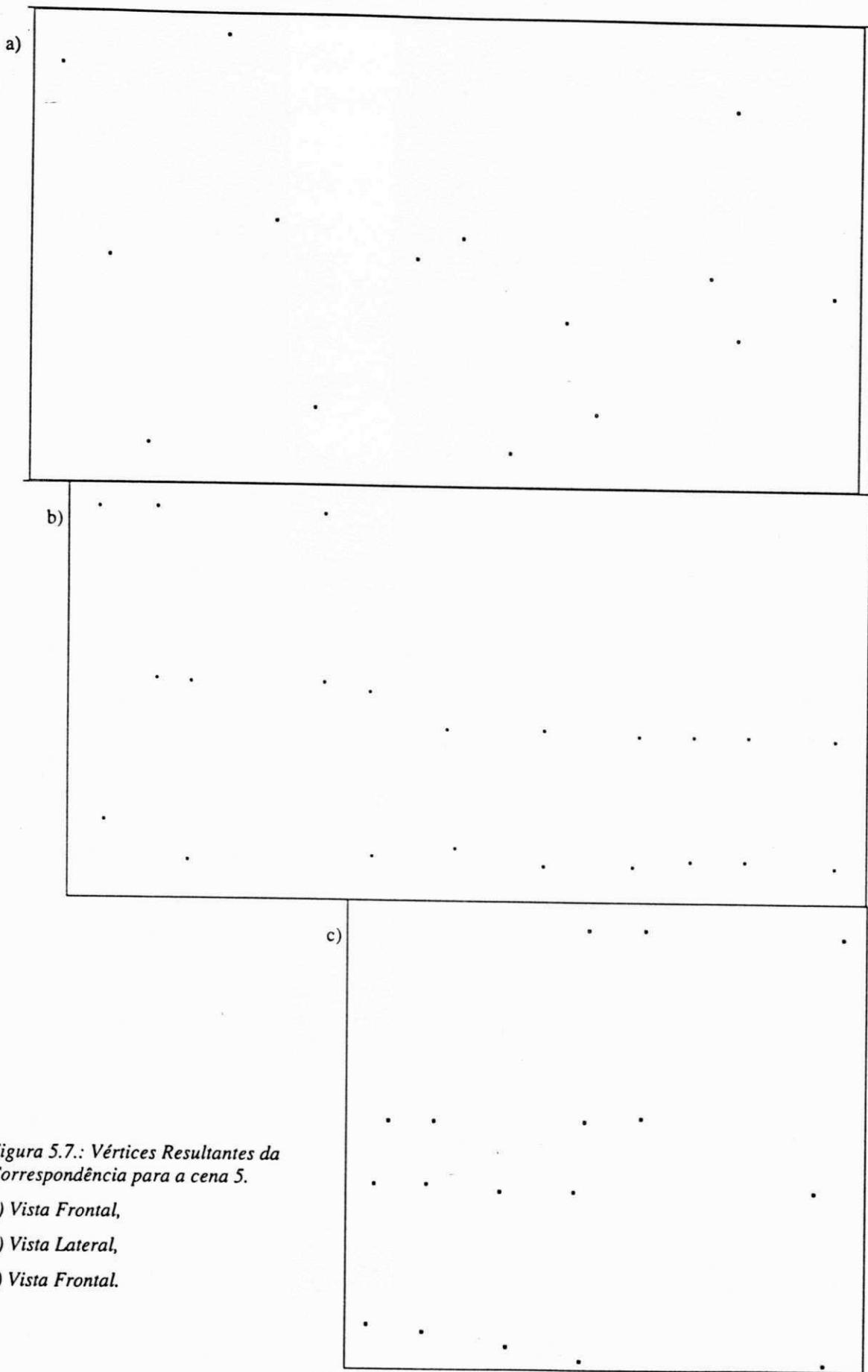
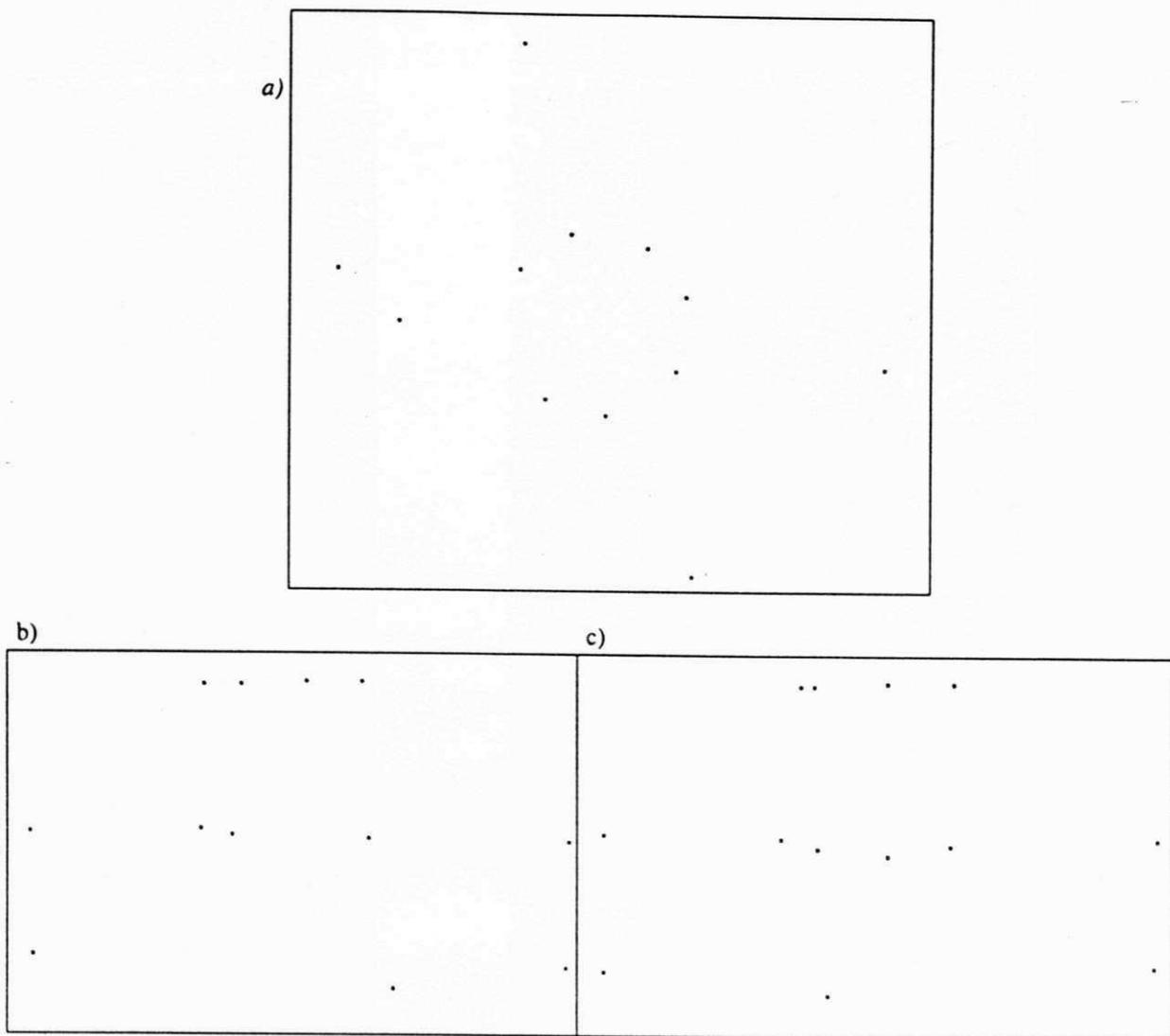


Figura 5.7.: Vértices Resultantes da Correspondência para a cena 5.

- a) Vista Frontal,*
- b) Vista Lateral,*
- c) Vista Frontal.*



*Figura 5.8.: Vértices Resultantes da Correspondência para a Cena 2:
 a) Vista Superior, b) Vista Lateral, c) Vista Frontal,*

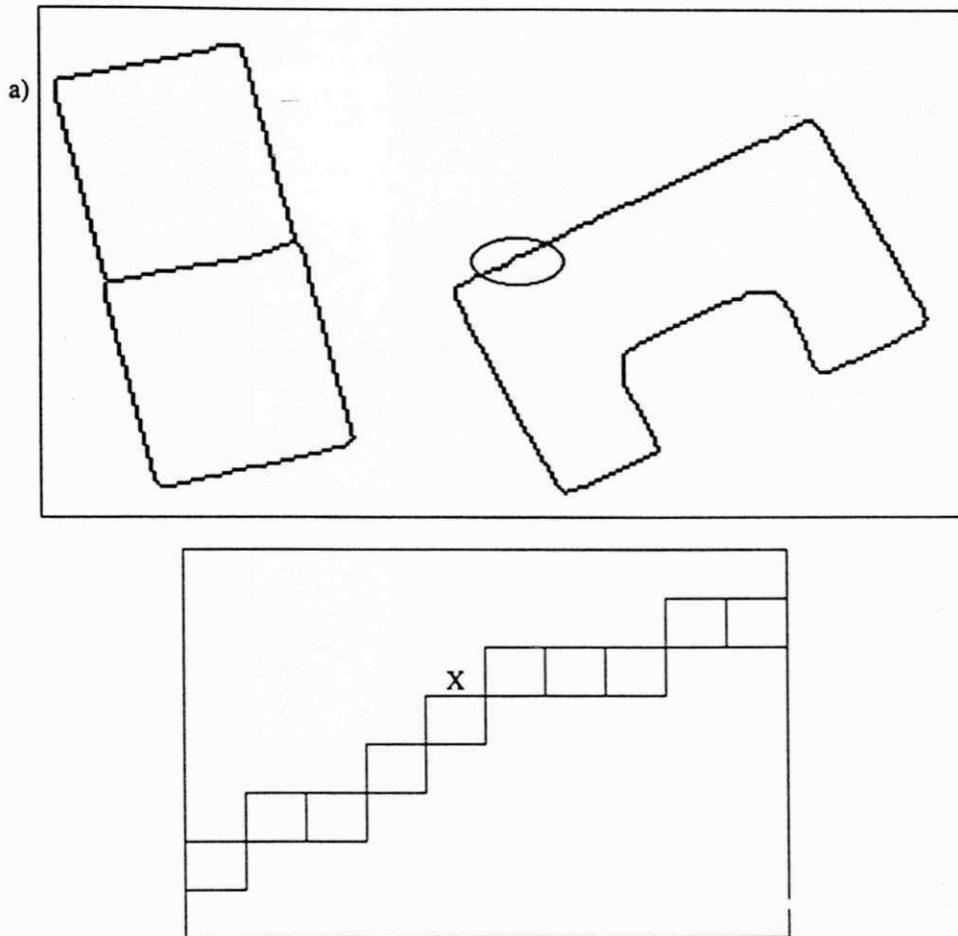


Figura 5.9: a) Vista Lateral da Cena 5, b) ampliação da região marcada por uma elipse.

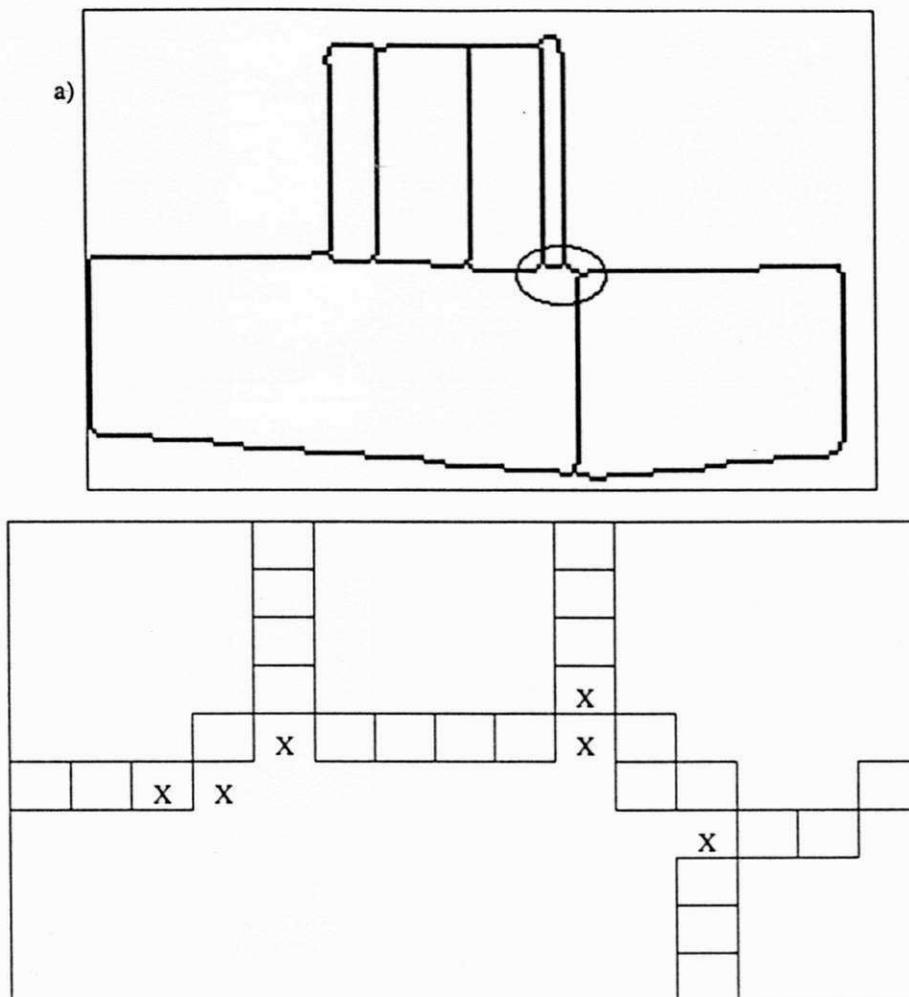


Figura 5.10: a) Vista Lateral da Cena 2, b) ampliação da região marcada por uma elipse.

Capítulo 6

Conclusões, Contribuições e Trabalhos Futuro

Este trabalho apresenta um método para identificação de vértices de objetos poliédricos, para fazer parte de um sistema de reconhecimento de imagens tridimensionais. O método trabalha com três imagens reais de um mesmo objeto de três pontos de vista diferentes, chamadas de imagens multi-vistas: uma vista superior, outra frontal e uma lateral. O método reconhece não só o contorno externo do objeto nas vistas, como também suas arestas internas.

O método trabalha em dois momentos diferentes, o primeiro, em *off-line*, que gera os padrões de vértices que serão utilizados para procurar os vértices nas imagens no segundo momento, em *on-line*.

Com o trabalho desenvolvido nesta dissertação de mestrado pode concluir-se ser viável a construção de sistemas de visão tridimensionais para reconhecimento do objetos poliédricos de alto desempenho totalmente automatizados, utilizando um conjunto de três vistas ortogonais da cena.

O método proposto funciona por *data-flow* o que possibilita sua utilização em sistemas de visão que executem em tempo real.

Foram realizadas pesquisas pela Internet na base de dados do ISI (Institute for Scientific Information) à procura de trabalhos similares e não foi encontrado nenhum método para identificação de vértices em tempo real de imagens multi-vistas. Tudo indica, portanto, que o método aqui proposto é o primeiro do gênero.

Pelos resultados obtidos nos testes apresentados no capítulo anterior, pode ser percebida a eficiência do método. Dependendo da qualidade das imagens pode ser alcançado uma taxa de erro de 0,0%.

Para a realização desde trabalho fez-se necessário modificar o algoritmo proposto por Farias [1] que faz a correspondência dos vértices nas imagens multi-vistas. Como aquele

algoritmo não faz a extração dos vértices automaticamente, não previa a ocorrência das situações encontradas neste trabalho, como a aglomeração de falsos vértices ao redor de um verdadeiro, ou as diferenças nas dimensões das imagens de uma mesma cena, ou ainda a necessidade de se adaptar as deformidades das figuras.

Como trabalhos que podem ser desenvolvidos futuramente podem ser citados:

- Projeto de uma arquitetura paralela para implementação dos algoritmos desenvolvidos, a fim de avaliar o comportamento do algoritmo em condições reais de operação. A utilização de FPGA's para esta implementação deve ser investigada.
- O desenvolvimento de um sistema de compressão de vídeo a partir dos modelos poliedrais produzidos.
- Um estudo aprofundado do sistema de aquisição e digitalização de imagens para corrigir os problemas enfrentados neste trabalho, devidos às deformidades dos objetos presentes nas imagens.
- Desenvolvimento de algoritmos que atenuem as mesmas deformidades.

Referencias

- [1] Farias, M. F. S. - *Reconhecimento de Objetos 3D Através de Atributos de Superfície de Modelos Poliedrais*. Tese de Doutorado em Engenharia Elétrica - COPELE/UFPb, dezembro de 1997.
- [2] Villar, Y. M. - *Projeto de um ASIC para Extração de Vértices de Imagens em Tempo Real*. Dissertação de Mestrado em Engenharia Elétrica - COPELE/UFPb, dezembro de 1997.
- [3] Cortez, P. C. - *Reconhecimento de Formas 2D Utilizando uma Técnica Seqüencial Integrada e Modelos Poligonais*. Tese de Doutorado em Engenharia Elétrica - COPELE/UFPb, 1996.
- [4] Marshall, A. D. e Martin, R. R. - *Computer Vision, Models and Inspection*. Editora World Scientific, 1992.
- [5] Da Silva, V. C. - *Normas de Compressão de Imagens*, Relatório Técnico Final da Disciplina Técnicas de Compressão de Imagens COPIN/UFPb, 1997.
- [6] Facon, J. - *Processamento E Análise De Imagens*, VI Escola Brasileño Argentina de Informática, 1993.
- [7] Melcher, E., Carvalho, J. M., Barros, M., Naviner, L. Naviner, J.F., et al. - A Novel Gradient Operator Suited for VLSI Implementation of 2D Shape Recogniton. Proceedings of IX SBCCI - *Simpósio Brasileiro de Concepção de Circuitos Integrados*, páginas 321-332, Março de 1996.
- [8] Melcher, E., Carvalho, J. M., Cortez, P. C., Naviner, L. Naviner, J.F. - A Vertex Detection Algorithm for VLSI Implementation. Proceeding of IX SIBGRAPI, páginas 367-368, outubro de 1996.

- [9] Melcher, E., Carvalho, J. M., Naviner, L. Naviner, J.F., Moreira, R. A. S., Villar, Y. M. e Morais, M. - VLSI Implementation of Contour Extraction from Real Sequences. VLSI'97 - *IFIP International Conference on Very Large Scale Integration*, Agosto de 1997.
- [10] Facon, F. - *Morfologia Matemática: Teoria e Exemplos*. Ed. Universitária Champagnat da Pontífica Universidade Católica do Paraná – Curitiba, 1996.
- [11] Parker, J. R.. - *Algorithms for Image Processing and Computer Vision*. Ed. John Wiley & Sons, Inc. United States of America, 1997.
- [12] Stentiford, F. W. M. and Mortimer, R. G. - Some New Heuristic for Thinning Binary Handprintes Characters for OCR. *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. SMC-13, No. 1, 1983 January/February. Pp. 81-84.
- [13] Yokoi, S., Toriwaki, J. and Fukumura, T. - Topological Properties in Digitized Binary Pictures. *Systems Computer Controls*. 1973, Vol. 4. 32-39.
- [14] Zhang, S. and Fu, K. S. - A Thinning Algorithm for Discrete Binary Images. *Proceedings of International Conference on Computers and Applications*. 1984. Beijing, China. 879-886.
- [15] Holt, C. M., Stewart, A., Clint, M. and Perrott. R. H. - An Improved Parallel Thinning Algorithm. 1987. *Communication of the ACM*. Vol. 30. 2: 156-160.
- [16] Chin, R. T., and Dyer, C. R. - Model-Based Recognition in Robot Vision. (1986). *ACM Computing Surveys*, 18(1): 67 – 108, March.
- [17] Besl, P. J. and Jain, R. C. - Three-Dimensional Object Recognition. (1985). *ACM Computing Surveys*, 17(1): 75-145. March.
- [18] Arman, F. and Aggarwal, R. - Model-Based Object Recognition in Dense-Range Images – A Review (1993). *ACM Computing Surveys*, 25(1): 5 – 43, March.

- [19] Chen, J. and Stockman, G. C. - Determining Pose of 3D Objects With Curved Surfaces (1996). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No 1, 52 – 58, January.
- [20] Chiou, R., Hung, K., Guo, J., Chen, C., Fan, T. and Lee, J. Y. - Polyhedron Recognition Using Three-View Analysis (1992). *Pattern Recognition*, 1 (25): 1 – 16.
- [21] Chiou, R., Hung, K., Kweiguo, J., Chen, C., Fan, T. and Lee, J. Y. - Polyhedron Recognition From Multiples 2D Camera Views (1992). *Pattern Recognition*, (1): 1 – 16.
- [22] Borges, D. L. - 3D Recognition by Parts: A Complete Solution Using Parameterized Volumetric Models (1996). *Anais do IX SIBGRAPI – MG*, 111 – 118, Setembro.
- [23] Liu, C. and Tsai, W. - 3D Curved Object Recognition From Multiple 2D Camera Views (1990). *Computer Vision , Graphics and Images Processing*, (50) : 177 – 187.
- [24] Lowe, D. G. - Three-Dimensional Object Recognition From Single Two-Dimensional Images (1987). *Artificial Intelligence*, 31: 355 – 395.
- [25] Nitzan, D. - Three-Dimensional Vision Structure for Robot Applications (1988). *IEEE PAMI*, 10: 291 – 309, May.
- [26] Park, J. H., Chang, T. G. and Choi, J. S. - Three-Dimensional Objects Representation Based on Surface Normal Images (1993). *Pattern Recognition*, 26 (6): 913 – 920.
- [27] Zhang, Z. and Faugeras, O. D. – Estimation of Displacements from two 3D Frames Obtained from Stereo (Juin 1991). *Relatório Técnico INRIA n.º 1440*, France.
- [28] Castro, F. C. C. de, e Amaral, J. N., e Franco, Paulo R. G. – Reconhecimento de Padrões em Imagens Utilizando Redes Neurais Artificiais como Estimador de Correlação Espectral. *II Simpósio Brasileiro de Redes Neurais*. 15 – 20.
- [29] Cortez, P. C. and Carvalho, J. M. – Polygonal Modeling and Shape Recognition (1993). *Relatório Técnico MASI 93.42*, Université Paris.

- [30] Stytz, M. R., Frieder, G., and Frieder O. – Three-Dimensional Medical Imaging: Algorithms and Computer Systems (December 1991). *ACM Computing Surveys*, Vol. 23, no. 4, 421 – 499.
- [31] Brady, M. - Computational Approaches to Image Understanding (March 1982) *Computing Surveys*, Vol. 14, No. 1, 3 – 69.
- [32] Flynn, P. J. and Jain A. K. – CAD – Based Computer Vision: From CAD Models to Relational Graphs (February 1991). *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 2.
- [33] Gonzalez, R. C., *Digital Image Processing*. Addison-Wesley Publishing Company, September 1993.