UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS

TECNOLOGIA PARA O
RECONHECIMENTO DO FORMATO
DE OBJETOS TRI-DIMENSIONAIS

Adilson Gonzaga

Tese apresentada ao Instituto de
Física e Química de São Carlos-USP
para a obtenção do título de
Doutor em Ciências (Física Aplicada) *OK*

Orientador: Prof. Dr. Valentin Obac Roda

DEPARTAMENTO DE FÍSICA E CIÊNCIA DOS MATERIAIS

São Carlos - 1991

SERVIÇO DE BIBLIOTECA E INFORMAÇÃO - IFQSC
FÍSICA

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS

MEMBROS DA COMISSÃO JULGADORA DA TESE DE DOUTORADO DE ADILSON GONZAGA APRESENTADA AO INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS, DA UNIVERSIDADE DE SÃO PAULO, EM 05.07.91

COMISSÃO JULGADORA:

Prof. Dr. Valério Obac Rode

Prof. Dr. Carlos Antonio Ruggiero

Prof. Dr. Alberto Tannus

Prof. Dr. João Antonio Zuffo

Prof. Dr. Boaventura Freire dos Reis

"Toda conquista implica em sacrifícios....

Todo trabalho demanda abnegação..."

à minha esposa Suely e meus filhos
Bruno, Tiago e Rafael pela espera
paciente e pelas regalias que
tivemos que suprimir durante a
execução deste trabalho.

AGRADECIMENTOS :

Para que este trabalho tivesse êxito, várias pessoas contribuíram com sugestões e executaram tarefas de vital importância.

- Como é impossível relembrar o nome de todos aqueles que no transcurso desta atividade geraram idéias que serviram como apoio ao nosso método, o meu agradecimento geral.

- Aos colegas e amigos que dispondo-se a ouvir, contribuíram com a reflexão dos meus próprios pensamentos.

- Em especial, os meus agradecimentos a meu orientador, Prof. Dr. Valentin Obac Roda, que nos momentos difíceis de decisão pode dar a direção correta ao melhor encaminhamento do problema.

- Ao aluno de Engenharia Elétrica e meu orientado em Iniciação Científica, Nelson Corona Jr. pela valiosa colaboração relativa aos programas em TURBO C, sem os quais nosso trabalho não lograria êxito.

- À funcionária do IFQSC, Maria Benedita Souza pelos desenhos caprichados em AUTOCAD.

- Ao técnico da oficina mecânica do Departamento de Engenharia Elétrica da EESC-USP, Rui Bertho pela confecção dos modelos e de todas as partes mecânicas de ajuste, imprescindíveis à operação do sistema.

RESUMO :

Apresentamos neste trabalho o desenvolvimento de um método para o Reconhecimento do Formato de Objetos Tri-dimensionais. Os sistemas tradicionais de Visão Computacional empregam imagens bi-dimensionais obtidas através de câmeras de TV, ricas em detalhes necessários à visão humana. Estes detalhes, em grande parte das aplicações industriais de Robôs, são supérfluos. Os algoritmos tradicionais de classificação consomem, portanto, muito tempo no processamento deste excesso de informação.

Para este trabalho, desenvolvemos um sistema dedicado para reconhecimento que utiliza um feixe de Laser defletido sobre um objeto e a digitalização da Luminância em cada ponto de sua superfície. A intensidade luminosa refletida é proporcional à distância do ponto ao observador. É, portanto, possível determinar parâmetros que classifiquem cada objeto. A inclinação de cada face de um poliedro, o comportamento de suas fronteiras e também a existência de arestas internas, são as características adotadas. Estas características são então rotuladas, permitindo que o programa de classificação busque em um "banco de conhecimento" previamente estabelecido, a descrição dos objetos.

Uma mesa giratória permite a rotação do modelo fornecendo novas vistas ao observador, determinando sua classificação. Todo o sistema é controlado por um Microcomputador cujo programa reconhece em tempo real o objeto em observação.

Para o protótipo construído, utilizamos um Laser de HeNe sendo a recepção do raio refletido realizada por um foto-

transistor.

Os objetos reconhecíveis pelo programa, são poliedros regulares simples, compondo o seguinte conjunto:

- 1 prisma de base triangular.
- 1 cubo.
- 1 pirâmide de base triangular.
- 1 pirâmide de base retangular.

O tratamento matemático empregado visa a comprovação da tecnologia proposta, podendo, na continuação de trabalhos futuros, ser efetivamente extendido a diversos outros objetos como por exemplo os de superfícies curvas.

ABSTRACT :

We present in this work a new method for Three dimensional Shape Recognition.

Traditional Computer Vision systems use bi-dimensional TV camera images. In most of the industrial Robotic applications, the excess of detail obtained by the TV camera, is needless. Traditional classification algorithms spend a lot of time to process the excess of information.

For the present work we developed a dedicated recognition system, which deflects a Laser beam over an object and digitizes the Reflected beam point by point over the surface. The intensity of the reflected beam is proportional to the observer distance. Using this technique it was possible to establish features to classify various objects. These features are the slope of the polyhedral surfaces, the boundary type and the inner edges. For each object the features are labelled and the classification algorithm searches in a "knowledge data base" for the object description.

The recognition system used a He-Ne Laser and the reflected signal was captured by a photo-transistor. The object to be recognized is placed over a rotating table which can be rotated, supplying a new view for the classification. A microcomputer controls the system operation and the object is recognized in real time.

The recognized objects were simple regular polyhedra, just as:

- 1 triangular base prism.
- 1 cube.
- 1 triangular base pyramid.

- 1 rectangular base pyramid.

To check that the proposed technology was correct, we used a dedicated mathematical approach, which can be extended to other surfaces, such as curves, in future works.

ÍNDICE:

CAPÍTULO	1	1
ESTADO ATUAL DA ARTE 1		
1.	INTRODUÇÃO.	1
2.	RECONHECIMENTO DE PADRÕES.	3
3.	CONCEITOS E METODOLOGIAS UTILIZADAS EM RECONHECIMENTO DE PADRÕES.	6
3.1	- Método matemático para Reconhecimento de Padrões.	8
3.2	- Reconhecimento de Padrões Sintático. . .	9
3.3	- Métodos heurísticos para Reconhecimento de Padrões.	11
4.	APLICAÇÕES DE RECONHECIMENTO DE PADRÕES.	12
4.1	- Diferenciação de conceitos e áreas de atuação.	12
4.2	- Técnicas de Processamento de Imagens mais usadas em Visão Artificial.	15
4.2.1	- Técnicas de Segmentação.	17
4.2.2	- Esqueletização.	18
4.2.3	- Suavização ("smoothing").	18
5.	PRINCÍPIOS DA VISÃO DE MÁQUINA.	18
5.1	- Modelo de Visão Bi-dimensional (2D). . .	19
5.2	- Modelo de Visão Tri-dimensional (3D). .	21
5.3	- Técnicas de Iluminação de cenas.	23

5.4	- Aplicações industriais da Visão Artificial.	26
6.	INTELIGÊNCIA ARTIFICIAL.	28
7.	NOVAS TENDÊNCIAS EM VISÃO ARTIFICIAL.	30
8.	COMENTÁRIOS BIBLIOGRÁFICOS.	31
CAPÍTULO 2		35
DETERMINAÇÃO DO FORMATO ATRAVÉS DO SOMBREAMENTO		35
1.	INTRODUÇÃO.	35
2.	REFLEXÃO DA LUZ.	36
2.1	- Fontes de luz pontuais.	36
2.2	- Superfícies Lambertianas.	38
3.	PROPOSTA DO SISTEMA DE RECONHECIMENTO DE OBJETOS.	40
3.1	- O espaço-dual e o espaço-gradiente.	42
3.2	- Arestas convexas e arestas côncavas.	48
3.3	- Proposição do sistema de reconhecimento - Simplificações e restrições.	50
3.4	- Inclinações das superfícies.	52
3.5	- Determinação de características do objeto através da variação da luminância.	56
3.6	Método de rotulagem de linhas em objetos poliédricos.	62
4.	CONCLUSÕES.	66
5.	COMENTÁRIOS BIBLIOGRÁFICOS.	68

CAPÍTULO	3	71
SISTEMA DE RECONHECIMENTO DO FORMATO DE OBJETOS		71
1. INTRODUÇÃO.		71
2. CONFIGURAÇÃO GERAL DO SISTEMA.		73
3. A INTERFACE DIGITAL DE CONTROLE DO SISTEMA.		75
3.1 Deflexão dos espelhos para a varredura do feixe de Laser.		77
3.2 - Rotação da mesa que suporta o objeto.		81
4. A INTERFACE ANALÓGICA PARA AQUISIÇÃO DO SINAL.		83
4.1 - Controle de velocidade do motor de "chopper".		84
4.2 - Detecção e filtragem do sinal refletido sobre o objeto.		85
4.3 - Retificação e integração do sinal.		87
5. AMOSTRAGEM E CONVERSÃO.		88
6. CONCLUSÕES.		91
7. COMENTÁRIOS BIBLIOGRÁFICOS.		92
CAPÍTULO	4	94
PROGRAMAS DE CONTROLE E ALGORÍTMO DE RECONHECIMENTO		94
1. INTRODUÇÃO.		94
2. PROGRAMAS DE CONTROLE DOS DISPOSITIVOS ELETRÔNICOS.		95
2.1 - Exposição do objeto em múltiplas vistas.		96

2.2 - Programa de controle de varredura do feixe de Laser.	99
2.3 - Digitalização do sinal.	101
3. ALGORÍTMO DE RECONHECIMENTO DE OBJETOS.	105
3.1 - Restrição do domínio dos objetos reconhecíveis.	106
3.2 - Extração das características dos objetos.	109
3.3 - Suavização dos dados de entrada.	112
3.4 - Classificação da fronteira e diferenciação das arestas.	113
3.5 - Base de conhecimento. Descrição dos objetos do domínio.	117
3.5.1 - Estratégia de controle "bottom-up".	118
3.5.2 - Estratégia de controle "top-down".	121
3.6 - Programa de reconhecimento.	126
4. CONCLUSÕES.	128
6. COMENTÁRIOS BIBLIOGRÁFICOS.	129
 CAPÍTULO 5	 131
 RESULTADOS E CONCLUSÕES	 131
1. INTRODUÇÃO.	131
2. CURVAS DE SUPERFÍCIE.	132
3. SIMULAÇÃO DE RESULTADOS COM DADOS REAIS.	135
4. RESULTADOS EXPERIMENTAIS.	142
5. CONCLUSÕES FINAIS.	145

REFERÊNCIAS BIBLIOGRÁFICAS.	154
APÊNDICE:	159
APÊNDICE A	159
CIRCUITOS ELETRÔNICOS :	159
APÊNDICE A-1	160
CIRCUITO ELETRÔNICO	
DA INTERFACE DIGITAL	160
APÊNDICE A-2	166
CIRCUITO ELETRÔNICO DE CONTROLE	
DOS GALVANOMETROS.	166
APÊNDICE A-3	168
CIRCUITO ELETRÔNICO DE ACIONAMENTO	
DA MESA GIRATÓRIA.	168
APÊNDICE A-4	171
CIRCUITO ELETRÔNICO	
DA INTERFACE ANALÓGICA.	171
APÊNDICE B	178
PROGRAMAS DE CONTROLE :	178
APÊNDICE B-1	179
PROGRAMA EM ASSEMBLER	
DO CONVERSOR A/D.	179
APÊNDICE B-2	182
PROGRAMA EM LINGUAGEM C PARA	
ESTATÍSTICA NO CONVERSOR A/D.	182

Tecnologia para o Reconhecimento do Formato de Objetos Tri-dimensionais.

APÊNDICE B-3	184
PROGRAMA PRINCIPAL PARA	
O RECONHECIMENTO	184
APÊNDICE B-4	214
PROGRAMAS DE AQUISIÇÃO DE DADOS.	214
 APÊNDICE C	 219
OBJETOS RECONHECÍVEIS :	219
APÊNDICE C-1	220
CURVAS DE SUPERFÍCIE DOS OBJETOS.	220

CAPÍTULO 1

ESTADO ATUAL DA ARTE

1. INTRODUÇÃO.

Desde os primeiros interesses científicos do homem pelo funcionamento de seu próprio corpo que a percepção do mundo exterior desperta especial atenção. Através de um processo dialético de observação a técnica evoluiu para máquinas que tentam imitar não apenas os mecanismos biológicos, mas também seu comportamento psicológico. Com o surgimento dos computadores digitais os maiores esforços tem sido envidados no sentido de transportar para seus circuitos eletrônicos uma representação simbólica do mundo real.

Presentemente, a capacidade das máquinas de perceberem seu ambiente é ainda limitada. Uma grande variedade de transdutores para conversão de parâmetros ambientais, como: luz, som, temperatura, etc., em sinais elétricos, é disponível e estes continuam sendo aperfeiçoados. Os avanços teóricos resultando em modelamentos matemáticos sobre a percepção humana são muito grandes, entretanto o objetivo de uma máquina realmente inteligente ainda não foi alcançado.

Na incessante busca do homem pelo "processo de criação" diversas disciplinas tem surgido tais como a Inteligência Artificial, o Processamento de Imagens, a Robótica e outras que

abrangem grandes áreas e sub-áreas do conhecimento humano.

Os estudos psicológicos e fisiológicos tem fornecido dados importantes sobre a percepção animal, mas não ainda o suficiente para repetir seu desempenho através de computadores. A idéia de associar velocidade de processamento com capacidade de decisão em um domínio abrangente tem fascinado diversos cientistas, entretanto isto encontra uma grande barreira tecnológica: o volume de operações matemáticas necessárias para se representar qualquer manifestação do mundo real em uma máquina é extremamente grande, o que torna lento o processo de decisão. A este respeito a pesquisa científica tem concentrado seus esforços, e novas técnicas, como as de processamento paralelo e redes neurais tem surgido, vislumbrando um futuro promissor.

O nome da disciplina que abriga a teoria da percepção aplicada às diversas áreas da ciência, é Reconhecimento de Padrões e tem suscitado durante várias décadas, discussões, congressos e um grande número de publicações entre artigos, livros e periódicos especializados.

O escopo deste trabalho é o desenvolvimento de uma tecnologia específica para o reconhecimento da forma de objetos tri-dimensionais relevando detalhes no intuito de aumentar a velocidade de decisão. Isto é de fundamental importância em aplicações industriais que utilizem robôs tanto em tarefas repetitivas como em tarefas que necessitem uma contínua adaptação ambiental.

2. RECONHECIMENTO DE PADRÕES.

Todos os problemas específicos de Reconhecimento de Padrões são parte de uma atividade técnica ou científica de âmbito muito maior. Esta área tem sido assunto de investigação de pesquisadores envolvidos com computadores, controle automático, teoria da informação, física aplicada, estatística, psicologia, fisiologia, biologia, medicina e linguística.

Fig.1 - Áreas de atuação de Reconhecimento de Padrões e sua relação com as diversas sub-áreas do conhecimento.

O esquema delineado na Fig.1 [Verhagen 1975] procura relacionar o assunto referente a Reconhecimento de Padrões com outras áreas do conhecimento humano. As ciências exatas tais como: física aplicada, processamento e análise de imagem, engenharia, inteligência artificial e outras, utilizam-se desta ferramenta de maneira bastante ampla. As ciências humanas e as ciências biológicas concentram suas aplicações na tentativa de compreender as funções de percepção do ser humano.

Conforme escreve Watanabe [Watanabe 1969] ... "Reconhecimento de Padrões representa, sem dúvida, uma das mais ambiciosas aventuras científicas deste século, necessitando da colaboração de engenheiros eletrônicos, físicos, fisiólogos, psicólogos, lógicos, matemáticos e filósofos."

No seu sentido mais amplo, Padrões são os meios pelos quais pode-se interpretar o mundo. Alguns possuem sua manifestação física, outros são abstratos como os padrões sociais ou os econômicos. Evidentemente, com esta generalização, se consultarmos autores diversos obteremos um vasto conjunto de definições voltadas para as áreas de especialidades de cada um. Portanto, cumpre adotarmos aqui o conceito aplicado às ciências exatas, mais especificamente à física aplicada e à engenharia de interpretação do mundo através de computadores.

Desta forma restringindo-se o domínio consideramos um Padrão como um objeto ou um conjunto de objetos que podem ser de alguma maneira tratados semelhantemente [Minsky 1961]. Adotamos, assim, o termo Padrão Visual para referirmo-nos aos objetos do mundo real passíveis de serem representados no mundo artificial dos computadores.

Reconhecimento de Padrões é definido aprioristicamente como a descrição e classificação de medidas obtidas de um processo físico ou mental [Fu & Rosenfeld 1976]. De uma forma mais objetiva, é a passagem de um conjunto estruturado de dados para outro conjunto de dados estruturados onde:

- o conjunto inicial de dados representa o resultado de medidas em um determinado domínio e o conjunto final representa a interpretação ou a classificação esperada [Simon 1975].

Um diagrama em blocos simplificado de um sistema que realize o Reconhecimento de Padrões é mostrado na Fig.2.

O conjunto inicial de dados é o padrão de entrada. Este conjunto

Fig.2 - Sistema de Reconhecimento de Padrões.

é submetido ao extrator de características [Fu 1976] gerando medidas características. Estas medidas são então classificadas por um dispositivo classificador, transformando-se no conjunto final de dados que permite a decisão.

No caso específico de Padrões Visuais, o padrão de entrada é o elemento gerador da imagem ou a própria imagem digitalizada por um dispositivo qualquer (sensor ótico, câmera de TV, etc...) e o conjunto de saída é a classificação ou a decisão sobre ela.

A interpretação do conjunto final de dados é diferente para cada tipo de problema. Assim, um observador humano necessita de uma imagem final rica em detalhes para compreender uma cena enquanto na maioria das aplicações em Robótica apenas o formato

de um objeto é necessário para produzir uma ação em resposta à esta interpretação.

3. CONCEITOS E METODOLOGIAS UTILIZADAS EM RECONHECIMENTO DE PADRÕES.

Os conceitos de projeto de sistemas de Reconhecimento de Padrões são motivados pela forma como as classes de padrões são definidas [Tou & Gonzalez 1974]. Quando uma classe de padrões é caracterizada por uma lista de seus membros o sistema de reconhecimento é baseado no conceito de "lista de membros". Quando uma classe de padrões é caracterizada pelas propriedades comuns compartilhadas por todos os seus membros, o sistema utiliza o conceito de "propriedades comuns". Quando uma classe de padrões exhibe propriedades de agrupamento dentro do espaço de padrões, o sistema emprega o conceito de "agrupamento ("clustering")".

A caracterização de uma classe de padrões pelo conceito de lista de membros sugere a técnica de comparação de padrões ("template matching"). Quando um padrão desconhecido é apresentado ao sistema, ele é comparado com os padrões armazenados, um a um. O sistema de reconhecimento classifica o padrão de entrada como um membro da classe de padrões se ele é igual, dentro de certa tolerância, a algum dos padrões armazenados.

A caracterização de uma classe de padrões pelo conceito de propriedades comuns compartilhadas por todos os membros indica a

necessidade de detecção e processamento de características similares. Estas características são armazenadas no sistema de reconhecimento. Quando um padrão desconhecido é observado, suas características são extraídas e então comparadas com as armazenadas no sistema. Este esquema classifica o novo padrão como pertencente à classe de padrões de características similares. Esta técnica é superior à de comparação de padrões pois é tolerante a variações significantes no padrão de entrada.

Uma classe de padrão pode ser caracterizada por suas propriedades de agrupamento ("clustering") dentro do espaço de padrões, quando seus elementos são vetores formados por números reais. Se as classes são definidas por grupos, podem ser empregados esquemas de classificação por distância mínima.

As diferentes técnicas usadas para solucionar problemas de Reconhecimento de Padrões, podem ser agrupadas em três metodologias:

- 1) - abordagem matemática, subdividida em duas categorias:
 - a) determinística
 - b) estatística
- 2) - abordagem sintática, linguística ou estrutural [Fu 1974].
- 3) - métodos heurísticos.

O objetivo de um sistema de Reconhecimento de Padrões é determinar, baseado na informação recebida, a classe de padrão responsável pela geração de um conjunto de medidas similares aos dados observados.

3.1 - Método matemático para Reconhecimento de Padrões.

A abordagem matemática é baseada em regras de classificação e faz uso dos conceitos de propriedades comuns e agrupamento.

O método determinístico não emprega explicitamente as propriedades estatísticas das classes de padrões. Um exemplo típico são os classificadores por treinamento de padrões, iniciando com os "Perceptrons" e evoluindo hoje para as redes neurais.

O diagrama em blocos de um sistema de Reconhecimento de Padrões estatístico é mostrado na Fig.3. O pré-processamento é

Fig.3 - Diagrama em blocos de um sistema de Reconhecimento de Padrões Estatístico.

tratado dentro da área de processamento de sinais ou de análise de imagem. Corresponde ao condicionamento eletrônico do padrão através de filtragens e outras operações matemáticas.

O propósito da extração de características é transformar o espaço original em espaços de dimensões menores para a representação do padrão. As medidas tomadas representam um espaço vetorial N-dimensional cujas coordenadas do vetor traduzem o objeto original [Tou & Gonzalez 1974].

Os métodos mais empregados na abordagem estatística [Mantas 1987] podem ser subdivididos em:

- classificação paramétrica
- classificação não paramétrica
- classificação sequencial
- raciocínio por lógica difusa ("fuzzy").

A classificação paramétrica é baseada na regra de classificação de Bayes. Esta regra tem um ótimo desempenho quando a função densidade de probabilidade de cada população de padrões e a probabilidade de ocorrência de cada classe de padrão são conhecidas.

Funções discriminantes são utilizadas na classificação não paramétrica a fim de separar as diferentes classes de padrões em hiperplanos definidos em um dado hiperespaço.

O raciocínio por lógica difusa emprega o conjunto de elementos difusos ("fuzzy set") que fornecem resultados mais realísticos quando não há conhecimento anterior sobre os padrões e as probabilidades não podem ser calculadas.

3.2 - Reconhecimento de Padrões Sintático.

O diagrama em blocos de um sistema de Reconhecimento de Padrões Sintático, é mostrado na Fig.4. Neste método, um padrão é representado por uma sentença em uma linguagem especificada por uma gramática. Esta linguagem que providencia uma descrição estrutural dos padrões em termo de um conjunto de primitivas e suas relações, é comumente chamada de Linguagem de Descrição de

Fig.4 - Diagrama em blocos de um sistema de Reconhecimento de Padrões Sintático.

Figuras ("Picture Description Language - PDL").

Uma representação alternativa da informação estrutural de um padrão é o conhecido "grafo relacional" no qual os nós designam as primitivas e os ramos mostram a relação entre elas.

O primeiro passo na formulação de um modelo linguístico para a descrição de padrões é a determinação de um conjunto de primitivas em função das quais os padrões de interesse serão descritos. Isto é amplamente influenciado pela natureza dos dados, pela aplicação específica e pela tecnologia empregada na implementação do sistema. Duas regras devem servir de guia para a seleção das primitivas [Fu 1976]:

1 - As primitivas, como elementos básicos do padrão devem providenciar uma descrição compacta mas adequada dos dados em termo das relações estruturais especificadas.

2 - As primitivas devem ser reconhecidas ou extraídas facilmente através de métodos não linguísticos, pois são padrões compactos e simples, e sua informação estrutural não é importante.

Dentro da abordagem estrutural, uma outra técnica destaca-se. É a denominada Rotulagem de Relaxação [Hummel & Zucker 1983] que envolve a consistência de rede via a satisfação de restrições locais aplicada à n-ésima (usualmente binária)

relação entre segmentos vizinhos da imagem. A utilização de gramáticas estocásticas e as técnicas de relaxação probabilística tem sido usadas na tentativa de solucionar problemas de ruído no sinal. Na relaxação as restrições são expressas em valores de compatibilidade os quais indicam a compatibilidade entre dois ou mais segmentos rotulados.

3.3 - Métodos heurísticos para Reconhecimento de Padrões.

A abordagem heurística é baseada na intuição e experiência humana, fazendo uso dos conceitos de lista de membros e de propriedades comuns. Consiste de procedimentos "ad hoc" desenvolvidos para tarefas de reconhecimento especializadas. As técnicas de inteligência artificial são amplamente empregadas com esta abordagem, pois permitem inferências lógicas em bancos de conhecimento estabelecidos previamente.

Apesar das subdivisões apresentadas para as técnicas de Reconhecimento de Padrões, os sistemas mais eficientes tem adotado um método híbrido, empregando parte de cada abordagem conforme necessário.

Este trabalho, em particular, utiliza as regras do reconhecimento sintático através de grafos relacionais para a classificação do padrão, e regras heurísticas na determinação das primitivas que compoem cada padrão de entrada.

4. APLICAÇÕES DE RECONHECIMENTO DE PADRÕES.

Os últimos trinta anos tem se destacado por um interesse crescente e avanços na pesquisa de Reconhecimento de Padrões, Inteligência Artificial e Aprendizagem de Máquina. As aplicações solicitam a cada dia novos investimentos e utilização de recursos cada vez mais aprimorados. A informática tem contribuído de forma contundente para a implementação de soluções para estas necessidades, oferecendo tecnologias cada vez mais poderosas em capacidade de processamento e velocidade de operação. As mais diversas áreas do conhecimento humano tem solicitado continuamente o concurso destas técnicas [Kanal 1974].

As aplicações incluem reconhecimento de caracteres (OCR), sensoriamento remoto, biologia, medicina, reconhecimento de impressões digitais, previsão do tempo, previsão econômica, detecção de padrões de voz, detecção de falhas em linhas de montagens industriais, reconhecimento de objetos (visão), etc... Algumas destas aplicações estão delineadas na Tabela I.

O que podemos observar é que devido à grande extensão do campo de atividade, podemos restringir o Reconhecimento de Padrões Visuais ou Reconhecimento de Padrões de Imagem, a determinadas áreas.

4.1 - Diferenciação de conceitos e áreas de atuação.

Com o intuito de localizarmos o objeto deste trabalho

Tabela 1: Algumas aplicações de Reconhecimento de Padrões.

Aplicação	Padrão de Entrada	Padrão de Saída
Caracteres: -Processamento de Documentos. -Leitura Postal.	-Imagem de um sistema ótico. -Imagem de um sistema ótico.	-Caracteres alfa-numéricos/símbolos -Códigos, enderêços escritos manuais.
Voz:	-Forma de onda.	-Fonemas, palavras.
Previsão do Tempo:	-Dados de sensores de satélites.	-Classes de climas.
Previsão Econômica:	-Indicadores Econômicos.	-Condições da Economia.
Medicina: -Identificação de células. -Raios-X.	-Imagem de Microscópio. -Radiografia.	-Tipos de células. -Presença ou ausência de certas condições.
Sensoreamento Remoto:	-Imagens de Satélites.	-Identificação da geologia.
Visão Artificial:	-Imagens de Objetos/cenas.	-Classes de objetos ou cenas.

dentro do vasto campo de atuação da disciplina de Reconhecimento de Padrões, é de fundamental importância diferenciarmos alguns nomes empregados. Um excelente glossário de termos pode ser visto em [Haralick 1973].

A Fig.5 sugere um quadro onde diversas sub-áreas se interrelacionam definindo alguns campos de atuação:

- Campo I - Define a área de Reconhecimento de Padrões generalizada, situando-se aí o reconhecimento de voz,

reconhecimento de
padrões econômicos,
reconhecimento para
previsão de tempo,
etc...

- Campo II - Define o
Reconhecimento de
Imagens ou de Padrões
Visuais que empregam
técnicas de
processamento de

imagens. Situa-se neste campo, a maioria das aplicações em Visão Artificial, tais como, identificação de células em biologia, reconhecimento de impressões digitais, etc...

- Campo III - Define a análise de imagens, sem reconhecimento. São exemplos deste campo, as aplicações em inspeção visual, melhoras na qualidade de imagem para aplicações médicas ou geológicas, etc...

- Campo IV - Este campo engloba as aplicações de Reconhecimento de Padrões Visuais que dispensam as técnicas de processamento de imagem. Um exemplo típico é o reconhecimento de caracteres escritos formados por imagens binárias pré-processadas por circuitos eletrônicos.

- Campo V - Visão artificial por sensoreamento alternativo (ultra-som, infra-vermelho, laser, etc...).

Apesar das áreas de Reconhecimento de Imagens e de Processamento de Imagens terem se desenvolvido como duas disciplinas separadas, elas tem uma interrelação significativa.

Fig.5 - Interrelacionamento de diferentes sub-áreas e seus diversos campos de atuação.

Técnicamente falando, no Processamento de Imagens a entrada e a saída são ambas imagens, sendo a segunda a versão melhorada da primeira, enquanto que no Reconhecimento de Imagens a entrada é uma imagem mas a saída é uma descrição de seu conteúdo [Rosenfeld 1981]. Uma Imagem é uma representação espacial de um objeto, de uma cena ou de outra imagem.

A Visão Artificial é tratada aqui como uma sub-área de Reconhecimento de Padrões, que engloba o Reconhecimento de Padrões Visuais ou Reconhecimento de Imagens sendo, no entanto, mais abrangente.

Com a utilização cada vez maior de computadores, especialmente de microcomputadores em Robótica, outros termos são também empregados para a Visão Artificial: assim, são comuns referências em Visão Computacional, Visão de Computador, Visão de Máquina, Visão Robótica e Visão de Robôs.

4.2 - Técnicas de Processamento de Imagens mais usadas em Visão Artificial.

A maior parte dos esforços na solução de problemas de Visão de Máquina tem sido através da utilização de imagens obtidas com câmeras de TV. Estas são então codificadas formando uma Imagem Digital que fornecerá subsídios para análise da cena, classificação e decisão necessárias ao processo. A Imagem Digital é um arranjo de números representando valores de intensidade luminosa, ou cor, dos pontos do plano da imagem, ou mais precisamente, os valores médios das vizinhanças destes

pontos [Ballard & Brown 1982]. Os elementos do arranjo são denominados "pixels" (reduzido de "picture elements"). Quando um único valor representa a intensidade luminosa da vizinhança do ponto, a imagem é dita em "escala de cinza". Quando esta intensidade é formada apenas por níveis de branco e preto, a imagem é binária. Nas aplicações que requerem informação sobre cor, vários valores são medidos para cada ponto, representando intensidade, banda espectral, etc...

No entanto, quanto maior é a quantidade de informação a nível de "pixels" da imagem, maior é o tempo necessário para a retirada de características fundamentais para o reconhecimento. Por isso, a maioria das aplicações em Visão de Computador refere-se a imagens binárias ou em escala de cinza sendo comum a utilização do termo "visão de baixo nível". Um Robô, efetivamente não necessita "ver" uma cena ou um objeto com os detalhes exigidos pela visão humana, mas apenas "enxergar" características que permitam uma tomada de decisão confiável.

A disciplina de Processamento de Imagens vem desenvolvendo técnicas que melhoram ou restauram uma cena digitalizada a níveis excelentes para a exigência humana. Um grande inconveniente, no entanto, é o enorme tempo computacional consumido neste processo. Enquanto tentamos imitar uma função humana, nossa compreensão limitada de como a visão de seres humanos funciona torna improvável que possamos solucionar o problema de reconhecimento da mesma forma como as pessoas fazem [Pavlidis 1986], ou que seja esta a melhor maneira para todos os casos.

Entre todas as técnicas investigadas para Processamento de

Imagens, algumas são diretamente aplicáveis ao Reconhecimento de Padrões, mais precisamente à Visão Artificial. Estes procedimentos auxiliam na extração de características que são utilizadas, pela metodologia de reconhecimento implantada, na classificação. Das operações de transformação de imagens, as seguintes se destacam: segmentação, esqueletização e suavização.

4.2.1 - Técnicas de Segmentação.

Segmentação é o processo de discriminar parte de uma imagem do fundo, ou seja, do meio onde ela está. Em outras palavras, é um meio de decidir quais "pixels" pertencem a uma determinada característica e quais não pertencem. Dois métodos se destacam em Visão Computacional:

- 1) Binarização ("Thresholding"): processo de separação de pontos claros de escuros em uma imagem. Cada "pixel" em escala de cinza é comparado com um valor estabelecido como nível de binarização, tornando-se "1" para valores maiores e "0" para valores menores. Este procedimento ressaltar parte da imagem ou esta do seu fundo.
- 2) Detecção de borda : processo que retira informação pertinente à borda de uma imagem em relação ao fundo. Os "pixels" de borda podem ser identificados pela alta taxa de variação da escala de cinza entre a imagem e o fundo. Diversos equacionamentos existem, mas todos produzem melhor ou pior resultado dependendo da aplicação.

4.2.2 - Esqueletização.

Algumas características, tais como estreitamentos em regiões da imagem podem ser detectadas através da geração de um "esqueleto" da região também denominado de eixo médio. Esta construção envolve a computação das menores distâncias de cada "pixel" na região para a borda. Vários algoritmos existem, que calculam eficientemente estas distâncias. A imagem resultante deste processo é "afilada" em seu eixo médio sugerindo a idéia de seu "esqueleto".

4.2.3 - Suavização ("smoothing").

Dentre as diversas técnicas de redução de ruído em imagens digitais, a que mais emprego encontra em Visão Artificial é a de suavização. A idéia básica é substituir o valor da escala de cinza em um ponto pelo valor médio da região vizinha. A inconveniência da suavização é que a média obscurece os detalhes da imagem. Este processo é também conhecido como regularização de funções [Duda & Hart 1973].

5. PRINCÍPIOS DA VISÃO DE MÁQUINA.

Visão de Máquina, Visão Artificial ou Visão Computacional refere-se à extração de informação de uma cena através da análise desta ou de sua imagem. Consideramos aqui apenas imagens

formadas por luz visível ou cenas iluminadas por uma ou mais fontes de luz. A intensidade ou cor da imagem em um dado ponto é fornecida por um sensor, colocado em uma direção conhecida, e corresponde à intensidade ou cor de um ponto sobre a cena ou superfície do objeto. A percepção visual é a relação entre esta entrada visual e modelos pré-estabelecidos para o mundo real. Estes modelos podem ser bi-dimensionais ou tri-dimensionais, dependendo de sua aplicação.

5.1 - Modelo de Visão Bi-dimensional (2D).

Em algumas aplicações tais como, reconhecimento de caracteres, análise da superfície da terra vista de grandes altitudes, imagens diretas de microscópios e outras, a Visão Artificial trata com cenas que são essencialmente bi-dimensionais.

O modelo para o reconhecimento em cenas desta natureza é mostrado na Fig.6. Um objeto pode ser definido como um conjunto de partes cujas propriedades e relações satisfaçam a determinadas condições. O processo de segmentação é usado para detectar partes de uma imagem, ou melhor, para extrair características tais como bordas, linhas ou curvas, pontos de maior intensidade, vértices, etc...O processo de extração de características atribui rótulos a conjunto de "pixels" da imagem, indicando classes especiais às quais estes "pixels" pertençam. Desta forma, a saída de um processo de segmentação pode ser denominada de "imagem simbólica" onde seus elementos

representativos são
rótulos e não "pixels"
em escala de cinza.

Conjuntos de
"pixels" que tenham o
mesmo rótulo podem ser
reagrupados em regiões
conexas e várias partes
destas podem formar, a
nível mais elevado,
conceitos mais
abrangentes. Esta etapa

Fig. 6 - Modelo de Visão Bi-dimensional.

é também conhecida como ressegmentação. Mede-se, então, propriedades geométricas da figura tais como área, comprimento, formato, etc., ou outras que dependam da escala de cinza dos "pixels", tais como a média de intensidades de luz da região.

O reconhecimento é então a tarefa de encontrar subconjuntos da imagem cujas propriedades e relações satisfaçam a condições de similaridade. As partes juntamente com suas propriedades compoem um grafo relacional onde as partes da figura são representadas pelos nós cujos rótulos são as propriedades extraídas, e os rótulos dos ramos são as relações entre elas. O reconhecimento consiste em encontrar sub-grafos que correspondam ao grafo da imagem, ou que satisfaçam às restrições definidas por ele.

5.2 - Modelo de Visão Tri-dimensional (3D).

Quando uma cena contém objetos sólidos, a natureza tri-dimensional não pode ser relevada. A extração de características e a segmentação tornam-se complicadas por causa da variação de iluminação, como por exemplo na geração de sombras, e também porque a intensidade de luz da superfície depende de sua orientação. O reconhecimento é difícil pois na maioria dos casos, um só lado do objeto que compõe a cena é visível, e alguns podem esconder partes de outros.

A estrutura geral adotada para um sistema de reconhecimento

Fig.7 - Modelo de Visão tri-dimensional.

tri-dimensional é apresentado na Fig.7. Os métodos convencionais são baseados na análise de cena 3D como se fosse uma cena bi-dimensional com um ponto de vista fixo. A imagem da cena, portanto, decodificada por um tubo de TV apresenta as mesmas

características tanto sendo retirada de uma cena tri-dimensional real, como de um desenho representativo desta cena, sob o mesmo ângulo de visão. A dificuldade, no entanto, é bem mais elevada. As técnicas convencionais de segmentação sofrem restrições porque as partes visíveis dos objetos variam substancialmente em brilho e intensidade luminosa.

Por melhor motivo que na visão bi-dimensional, o domínio da cena deve ser restrito para cada aplicação sob pena de tornar os sistemas de reconhecimento tri-dimensionais impossíveis de serem projetados. No espaço, além da grande diferenciação de objetos, existem problemas de difíceis soluções. A iluminação da cena e algumas características tais como especularidade e transparência, analisadas em conjunto com objetos opacos, tornam a tarefa de reconhecimento inviável.

Uma grande variedade de técnicas tem sido desenvolvidas para extrair informações sobre as superfícies dos objetos. Uma destas técnicas é a chamada "formato através do sombreamento" ("shape from shading") que opera diretamente sobre a escala de cinza da imagem. Outras são também utilizadas através da detecção de bordas ou do formato das bordas de objetos da imagem. São as denominadas "formato do contorno" ("shape from contour"), "formato da forma" ("shape from shape") e "formato da textura" ("shape from texture"). Todos estes métodos operam sobre um mapa de características derivadas da imagem chamado de "esquema primário" na Fig.7.

O processo de determinar a orientação das superfícies de uma imagem é conhecido como "recobrimento". Alguns autores chamam esta etapa de "esquema 2 1/2 D" e não tri-dimensional,

porque contém informação apenas sobre as superfícies visíveis e não sobre a cena completa. O que este trabalho propõe é o que mostra o ramo delineado mais à direita na Fig.7, ou seja, a extração de características tri-dimensionais diretamente da cena e a representação simbólica sem a necessidade de geração dos esquemas de 2 1/2 D.

É importante salientar que é bastante difícil determinar a orientação de uma superfície, pelos métodos convencionais, usando apenas uma imagem. A situação é no entanto melhorada tomando-se duas ou mais imagens da mesma cena, de diferentes pontos de visão, e usando técnicas de triangulação ou visão estereoscópica. Uma vez conhecida a orientação das superfícies em cada ponto da imagem, é possível, em princípio, segmentá-la em partes correspondentes às superfícies dos objetos da cena. Mesmo que esta operação seja realizada com sucesso, o reconhecimento não é trivial, pois a imagem mostra apenas um lado dos objetos sendo desconhecidas suas orientações.

5.3 - Técnicas de Iluminação de cenas.

Um dos maiores problemas da análise de cenas tri-dimensionais através de computadores é a diferenciação entre fronteiras dos objetos e a sombra produzida devido à localização da fonte de luz.

A visão humana tem uma capacidade fantástica de operar em condições adversas de luminosidade. Um sistema de controle automático na íris e sensores especializados da retina do olho

humano (cones e bastonetes), permitem a adaptação a pequenas variações no nível de iluminação. Na Visão de Máquina, no entanto, parece mais econômico controlar a iluminação sobre a cena visualizada do que emular o funcionamento do olho humano.

A iluminação de uma cena para aplicação em Robótica, tanto para reconhecimento como para inspeção visual, é um fator importante que afeta diretamente a complexidade dos algoritmos utilizados. Um sistema de iluminação bem projetado, adaptado ao esquema de reconhecimento, reduz a quantidade de detalhes em uma imagem como, sombras, reflexões espúrias e detalhes indesejáveis. Em verdade, cada aplicação em particular deve ter seu próprio esquema de iluminação para realçar as características que se queira avaliar. Dessa forma, se levarmos em conta a inspeção visual de uma forma generalizada, podemos observar em [Batchelor & all. 1985] sessenta e três diferentes esquemas de iluminação.

Restringindo no entanto, o domínio para objetos tri-dimensionais opacos, que é o campo de ação da maioria dos robôs industriais, estes esquemas podem ser resumidos nas técnicas delineadas na Fig.8[Gonzalez & Safabakhsh 1982], [West 1983]. A iluminação difusa é empregada para a caracterização de objetos com superfícies regulares, tais como poliedros, cilindros, etc...

Com o objeto localizado entre a fonte de luz e o sensor temos a iluminação em oposição que é útil para a detecção de silhuetas ou a caracterização de partes relevantes como furos ou detalhes internos a corpos translúcidos.

A modulação espacial ou iluminação estruturada, é uma

Fig. 8 - Esquemas básicos de iluminação de cenas tri-dimensionais.

técnica empregada por alguns sistemas, que projeta um padrão de retas, pontos ou grades sobre o objeto. A superfície distorce o padrão de luz projetado e esta distorção é usada para encontrar a curvatura do objeto.

No método direcional a fonte de luz e o sensor de imagem são colocados formando determinados ângulos de incidência e reflexão, diretamente sobre a superfície do objeto. Permite o uso de visão estereoscópica e técnicas de triangulação mas é muito suscetível a sombras e ruídos na imagem complicando os algoritmos de reconhecimento e classificação.

A utilização de um feixe de Laser aparece na literatura especializada de duas maneiras:

- como um método de luz estruturada projetada sobre a cena e captada por uma câmera de TV.
- ou como sistema de telemetria ("range finder") onde é medido o deslocamento de fase ou a largura de pulso entre a fonte de Laser e o retorno ao sensor [Jarvis 1983].

O propósito deste trabalho é um método diferente de iluminação do objeto na cena. Conforme mostra a Fig.9 o Laser é utilizado como fonte de luz e o sensor de imagem nada mais é que um foto-transistor que capta a intensidade do feixe refletido sobre a superfície. Isto elimina o problema de sombras reduzindo a complexidade do algoritmo de reconhecimento.

Fig. 9 - Feixe de Laser como fonte de luz da cena.

5.4 - Aplicações industriais da Visão Artificial.

Sob um ambiente visualmente controlado, ou seja, onde o

sistema de iluminação é conhecido, alguns usos para a Visão de Máquina podem ser classificados da seguinte forma [West 1983]:

- a) Reconhecimento: determinação do formato de objetos e sua orientação apropriada em montagens. É de fundamental importância na automação das linhas de montagens de fábricas.
- b) Identificação: usado para diferenciar entre itens que possuam o mesmo formato mas que tenham alguma característica diferente. Por exemplo, escolher entre duas latas iguais, uma de suco e outra de extrato de tomate. A identificação pode ser feita através dos rótulos ou de códigos previamente gravados.
- c) Acompanhamento ("tracking"): capacidade de localizar objetos e manter o conhecimento de sua posição e orientação em esteiras de montagens industriais.
- d) Inspeção Visual: verificação de defeitos ou manchas em partes, posição dentro de montagens etc... Em geral necessita de técnicas de processamento de imagens, como filtragens e restauração, que melhorem a qualidade da imagem fornecendo a um observador humano condições de inspeção e decisão.
- e) Metrologia Visual: sistemas que realizam medidas de objetos através da imagem sem contato físico com as partes. Podem utilizar feixe de Laser através de telemetria ("range finder") ou por meio de iluminação estruturada.

6. INTELIGÊNCIA ARTIFICIAL.

A abordagem de Inteligência Artificial (IA) em Visão de Máquina envolve conceitos abstratos representados em vários níveis hierárquicos. Em cada nível de abstração, o conhecimento apropriado é usado para identificar componentes de mais alto nível. Diferentes fontes de conhecimento são usadas para modelar os mecanismos que geram os padrões.

As técnicas de IA para extração de informação e reconhecimento é bastante diferente do convencional. Enquanto este utiliza uma grande quantidade de computação numérica em todos os níveis, a abordagem de IA necessita apenas na transformação do sinal digitalizado em uma "string" de símbolos.

As técnicas de IA usam explicitamente heurística codificada para gerar hipóteses sobre a existência de um conceito. A heurística ajuda a reduzir o espaço de busca na direção do objetivo correto. Evidentemente, as metas alcançadas podem não ser as melhores, mas apenas consideradas como "suficientemente boas".

Os sistemas de visão baseados em IA emulam as hipóteses e testam o modelo. Em dada classe de sistemas, a hipótese inicial é gerada pelo usuário e o programa tenta verificá-la. Em outras, o ciclo de geração e verificação é automático.

O projeto de um sistema de Reconhecimento de Padrões Visuais baseado em IA envolve algumas características importantes e interdependentes [Nandhakumar & Aggarwal 1985]:

- A representação do conhecimento.
- Os mecanismos de Inferência.

- A estrutura de controle.

A representação do conhecimento abstrai fatos sobre os objetos e relações entre eles através do uso de redes semânticas, quadros, [Rich 1988] etc... A combinação de conceitos como PARTE DE, SOB, SOBRE, descrevem o estado de cada objeto em uma cena.

As inferências são geradas e avaliadas pela fonte de conhecimento em vários níveis de hierarquia. Nos níveis mais baixos, as fontes de conhecimento que produzem as primeiras transformações, como por exemplo de "pixels" para símbolos, podem ser baseadas em técnicas estatísticas ou descritivas.

A estrutura de controle decide a estratégia para o reconhecimento visando a alocação eficiente dos recursos e providenciando a interação com o usuário.

Se pesquisarmos a origem da Inteligência Artificial e do Reconhecimento de Padrões, poderíamos afirmar que as duas disciplinas são filhas, ou irmãs, da mesma necessidade. Surgidas na década de 50, ambas evoluíram por técnicas específicas a aplicações diversas em sub-áreas do conhecimento humano, e apresentam para o futuro, juntas, a expectativa de solução do velho problema. Com o avanço da tecnologia e dos recursos computacionais em capacidade de memória e velocidade, com o advento das Redes Neurais e com as novas técnicas de Processamento Paralelo descortina-se um futuro promissor para a Visão Artificial.

7. NOVAS TENDÊNCIAS EM VISÃO ARTIFICIAL.

Como referido anteriormente, duas frentes de trabalho apresentam resultados satisfatórios na busca da solução do gargalo encontrado por pesquisas anteriores. O primeiro e grande problema é o relacionado com a grande quantidade de tempo envolvido no processamento da informação. De encontro a isso, vem os estudos de arquiteturas paralelas de microcomputadores. Esta técnica aborda a questão através da paralelização dos algoritmos de processamento de imagens, tendo encontrado respaldo em computação gráfica e nas aplicações que requerem melhoria da qualidade de imagem.

Além do problema relacionado à velocidade de processamento, está o relativo ao aprendizado e à adaptação ao meio, de máquinas inteligentes. O processamento paralelo não é o ideal para todo o tipo de computação [Churchland & Churchland 1990]. No problema de aprendizagem, a arquitetura do cérebro apresenta tecnologia superior. O modelamento neural revivido pela moderna teoria conexionista [Feldman & Ballard 1982] é um outro paradigma em Inteligência Artificial. A idéia é a representação do conhecimento como pesos numéricos de conexões entre unidades de uma rede. Esta abordagem estuda a interconexão de neurônios para sintetizar o modelo do cérebro.

Vários pesquisadores tem desenvolvido modelos de Redes Neurais capazes de reconhecer padrões visuais [Fukushima 1988]. Estes modelos tem a função de auto organização e podem aprender a reconhecer padrões de forma supervisionada ou não supervisionada. Infelizmente, a carência de tecnologia em

circuitos de Redes Neurais, ainda hoje, tem feito com que inúmeros trabalhos de aplicação utilizem simulação em computadores de arquitetura von Neumann. Isto não desmerece de nenhuma forma a expectativa do futuro promissor que esta técnica promete na solução dos problemas de Reconhecimento de Padrões Visuais.

8. COMENTÁRIOS BIBLIOGRÁFICOS.

A área de Reconhecimento de Padrões e suas aplicações vem sendo estudada ao longo dos últimos 40 anos tendo crescido bastante em quantidade de publicações, sociedades dedicadas ao assunto e congressos científicos, a partir da década de 70. Livros como os editados por K.S.Fu [Fu 1974] [Fu 1976] tratam da teoria do Reconhecimento de Padrões Sintático com algumas comparações com o Estatístico, revendo a teoria formal de linguagens e a aplicação de gramáticas. Já o livro de J.T.Tou e R.C.Gonzalez [Tou & Gonzalez 1974] trata o assunto sob ponto de vista diverso, encarando o problema como substancialmente estatístico e reservando apenas um capítulo para a abordagem sintática.

Dentro do contexto de análise de cena, ou seja, Reconhecimento de Padrões Visuais, a obra de R.O.Duda e P.E.Hart [Duda & Hart 1973] é considerada de leitura fundamental. Dividido em duas partes, Classificação de Padrões e Análise de Cena, faz uma clara exposição explorando vários métodos e recursos sem se ater a discussões de eficiência de uma ou de

outra abordagem.

Dedicado exclusivamente à Visão por Computador, a obra de D.H.Ballard e C.M.Brown [Ballard & Brown 1982] cobre de forma clara desde a formação da imagem até a descrição desta através de estruturas relacionais.

O livro de E.Rich sobre Inteligência Artificial traz nos capítulos 10 e 11 os assuntos Percepção e Aprendizado voltados para o problema de visão. É de interessante leitura pois mostra as possíveis colaborações da IA ao assunto.

Voltado exclusivamente para o problema de inspeção visual automática B.G.Batchelor et all [Batchelor et all 1985] faz uma varredura nos métodos, técnicas de iluminação, equipamentos disponíveis e princípios de processamento de imagens. Não discute sobre Visão ou Reconhecimento de Padrões mas fornece uma boa análise dos principais problemas solucionáveis com digitalização de imagens para posterior processamento.

Dentre os artigos de discussão e revisão o escrito por C.J.D.M.Verhagen [Verhagen 1975] procura relacionar a disciplina de Reconhecimento de Padrões com as diversas áreas do conhecimento humano. O escrito por T.Pavlidis [Pavlidis 1986] enfoca, com uma visão crítica, os métodos de análise de imagens existentes. Os dois artigos são de bastante interesse ao pesquisador que desejar localizar-se dentro do assunto. Para uma visão rápida sobre as metodologias empregadas em análise e reconhecimento de imagens, os artigos de [Mantas 1987], [Simon 1975], [Rosenfeld 1981], [Fu & Rosenfeld 1976] são de fácil leitura e mostram o assunto com seus principais conceitos. Um bom glossário de termos empregados pode ser visto em [Haralick

1973] apesar de direcionado ao sensoreamento remoto.

Os artigos de revisão e referências são bastante úteis para uma configuração do estado da arte de qualquer disciplina. Em Reconhecimento de Padrões de forma mais geral, L.Kanal [Kanal 1974] faz uma descrição mostrando os principais jornais, revistas, metodologias e aplicações cobrindo o realizado entre os anos de 1968 a 1974. Já exclusivamente sobre Visão Computacional A.Rosenfeld tem publicado o mesmo estudo revisional anual. Assim temos [Rosenfeld 1988], [Rosenfeld 1989], [Rosenfeld 1990] guias de consulta à artigos, conferências e livros publicados durante os anos de 1987, 1988 e 1989.

Dois artigos de aplicações da Visão Artificial na prática podem ser analisados em [West 1983] e [Gonzalez & Safabakhsh 1982]. Mostram as principais necessidades e as soluções que vem sendo adotadas pelo atual estado da arte.

A contribuição da Inteligência Artificial pode ser observada em [Nandhakumar & Aggarwall 1985] e como leitura complementar adequada os artigos de M.Minsky [Minsky 1961] e P.M.Churchland [Churchland & Churchland 1990] completam o assunto. K.Fukushima [Fukushima 1988] mostra a aplicação de Redes Neurais em Reconhecimento de Padrões Visuais citando o desenvolvimento do Cognitron e do Neocognitron, modelos neurais para aprendizagem.

O artigo de R.A.Jarvis [Jarvis 1983] é o que mais se aproxima da abordagem que será dada neste trabalho. Descreve a técnica da metrologia ("Range finding") aplicada ao problema de Visão Artificial. Apesar desta aproximação o sistema proposto

Tecnologia para o Reconhecimento do Formato de Objetos Tri-dimensionais.

por Jarvis necessita digitalizar completamente uma cena para posterior processamento, empregando os recursos de triangulação para aquisição da imagem através de câmera de TV.

CAPÍTULO 2

DETERMINAÇÃO DO FORMATO ATRAVÉS DO SOMBREAMENTO

1. INTRODUÇÃO.

Objetos representados no plano podem ter aparência tri-dimensional se forem sombreados.

As graduações de intensidade luminosa refletida por um objeto de superfície uniforme, possuem informação suficiente para permitirem o reconhecimento de seu formato, tanto pelo olho biológico como por um computador.

Os receptores presentes na retina do aparelho visual dos vertebrados, denominados de cones e bastonetes [Cavaggioni 1969], recebem e transformam a informação luminosa em mensagem codificada para os centros nervosos mais elevados.

A imagem que vemos de um objeto é correspondente, portanto, à quantidade de luz refletida em cada ponto de sua superfície.

Evidentemente, o formato não pode ser deduzido unicamente através de operações locais isoladas, levando-se em conta que a reflexão é definida em cada ponto, e não para um conjunto de pontos.

Para uma grande quantidade de tipos de superfícies existentes, a fração de luz incidente é espalhada em diversas direções, e é uma função uniforme dos ângulos envolvidos. A orientação destas superfícies pode ser calculada se a fonte de iluminação e as propriedades de reflexão forem conhecidas. Este

cálculo é descrito como a determinação do "formato através do sombreamento" ("shape from shading").[Horn 1975]

2. REFLEXÃO DA LUZ.

A luz é definida como energia radiante visualmente identificável. Está compreendida no espectro eletromagnético entre 380 nm e 780 nm considerando-se que o observador é capaz da percepção de cores. Estes limites possuem uma certa tolerância, pois esta definição está vinculada ao aparelho visual humano e dois destes não são rigorosamente iguais.

A energia radiante, de comprimento de onda adequado, torna visíveis os corpos quando estes a emitem ou refletem em quantidade suficiente para que as células receptoras do olho sejam excitadas, provocando simultaneamente uma sensação de luminosidade e de cor. A quantidade de energia radiante, emitida por unidade de tempo, e avaliada de acordo com a sensação luminosa produzida, é definida como fluxo luminoso (F) de uma fonte.

2.1 - Fontes de luz pontuais.

Embora as fontes luminosas tenham dimensões finitas, quando observadas à distância podem ser consideradas puntiformes. Chama-se de fonte puntiforme, ou pontual, de energia radiante ou de luz, a toda aquela que possa ser reduzida a um ponto, por

abstração, ou seja, cujas dimensões geométricas sejam desprezíveis em relação à distância de observação.

A intensidade luminosa (I) de uma fonte pontual, emitida numa dada direção do espaço, caracteriza o fluxo luminoso (F) no ângulo sólido (Ω) associado a essa direção, conforme mostra a Eq.1.

$$I = \frac{dF}{d\Omega} \quad (1)$$

A iluminância (E) num ponto de uma superfície refere-se ao fluxo recebido por elemento de superfície que o contém.

A Eq.2 mostra esta definição.

$$E = \frac{dF}{dS} \quad (2)$$

Define-se também a luminância (L) de um elemento qualquer de um objeto, como sendo a relação entre a intensidade luminosa (I) em uma dada direção e a área de um elemento da superfície projetada sobre o plano perpendicular à esta direção.

Isto é mostrado pela Eq.3.

$$L = \frac{dI}{dS} \quad (3)$$

A luminância é uma grandeza que caracteriza a visibilidade de superfícies iluminadas.[Cardoso 1979]

A noção de profundidade na análise de um objeto é devido às diferentes luminâncias de suas partes. As regiões de maior ou

menor quantidade de luz refletida, dão a noção de relevo e a posição relativa de partes do objeto.

2.2 - Superfícies Lambertianas.

Todas as superfícies refletoras estão compreendidas entre dois casos limites: as que refletem como um espelho segundo as conhecidas leis da óptica geométrica (especulares), e as que refletem a luz de forma totalmente difusa.

Quando a luminância é constante em uma dada superfície, qualquer que seja a direção de observação, esta satisfaz as leis de Lambert e por isso é dita superfície Lambertiana, ou ainda, um difusor perfeito. Estas leis, também conhecidas como leis da incidência [Guerrini 1979]

estabelecem que:

- a) A iluminância (E) é diretamente proporcional à intensidade luminosa (I).
- b) A iluminância (E) é inversamente proporcional ao quadrado da distância da fonte luminosa.

Fig. 1 - Fonte pontual iluminando uma superfície.

É possível determinar uma única equação para as leis de Lambert. Consideremos uma fonte de luz pontual iluminando a superfície (S) de um objeto, formando um ângulo (i) com a normal e localizada a uma distância (d), conforme mostra a Fig.1. Seja um elemento de superfície

(dS), iluminado e compreendido pelo ângulo sólido elementar (dΩ).

Pela definição de ângulo sólido tem-se:

$$d\Omega = \frac{dS \cdot \cos(i)}{d^2} \quad (4)$$

Assim, da Eq.1 pode-se escrever que:

$$dF = \frac{I \cdot dS \cdot \cos(i)}{d^2} \quad (5)$$

Logo, a iluminância (E) do ponto sobre a superfície (S), descrita na Eq.2 pode ser escrita conforme as leis de Lambert, para um difusor perfeito, conforme mostra a Eq.6.

$$E = \frac{I \cdot \cos(i)}{d^2} \quad (6)$$

A luminância (L) é a característica que fornece a sensação de sombreamento ("shading"), ou seja, a noção de profundidade do objeto iluminado. Deduz-se, então, através das Eq.1, Eq.2 e Eq.3 que:

- dada uma fonte de luz pontual de intensidade luminosa (I), cujo valor dentro de um ângulo sólido (Ω) seja constante, a luminância (L) pode ser escrita na forma da Eq.7, para superfícies Lambertianas.

$$L = \frac{E}{\Omega} \quad (7)$$

Desta forma, conclui-se que:

- a) - se a superfície do objeto for Lambertiana (difusor perfeito),
 - b) - se a fonte luminosa for pontual e de intensidade constante,
 - c) - e o ângulo sólido de observação for sempre o mesmo,
- então, a parcela de luz refletida é diretamente proporcional ao cosseno do ângulo de incidência do feixe luminoso com a normal à superfície e inversamente proporcional ao quadrado da distância da fonte luminosa.

3. PROPOSTA DO SISTEMA DE RECONHECIMENTO DE OBJETOS.

Grande parte de objetos industriais são difusores e possuem forma geométrica regular. O sistema proposto neste trabalho adota a reflexão em superfícies Lambertianas e procura lidar com sólidos de formatos poliédricos para que se possa comprovar a eficiência do processo.

Tradicionalmente, os métodos de Visão de Máquina processam a imagem obtida através de câmeras de TV, segmentando-a em regiões para encontrar bordas e fronteiras. Esta imagem é na verdade um mapeamento das Luminâncias da cena. O que podemos supor é que a maior parte dos trabalhos nesta área são realizados por pesquisadores envolvidos com processamento de imagens. Este estudo preocupa-se com a conversão de imagens em imagens melhoradas, tendo em vista o conforto da visão humana. No entanto, os sistemas computacionais de compreensão de cenas necessitam de descrições simbólicas e não de novas imagens.

É importante salientar que as intensidades luminosas refletidas, ou seja a Luminância codificada em cada ponto, contém por si só uma quantidade de informação suficiente sobre o formato tri-dimensional da cena observada. Esta idéia não é nova, mas segue a linha de pesquisa adotada por Guzman [Guzman 1969], Clowes [Clowes 1971], Huffman [Huffman 1971], Mackworth [Mackworth 1973], Waltz [Waltz 1975], Horn [Horn 1975] e Pentland [Pentland 1988] que procuram estudar como as características do mundo físico atuam sobre tudo o que vemos.

É, desta forma, conveniente pensar que um feixe de luz incidente em uma superfície difusora seja espalhado em todas as direções, de maneira uniforme

segundo ângulos definidos, conforme mostra de maneira simplificada a Fig.2. O ângulo formado entre a normal e o raio luminoso incidente é chamado de ângulo de incidência (i). O ângulo entre a normal e o raio refletido, ou emitido pela superfície, é denominado de ângulo de emissão (e). E o

ângulo entre os dois raios, incidente e refletido, é conhecido como ângulo de fase (g).

O eixo que passa pelo observador e que tem a mesma direção do raio refletido é dito eixo de visão do objeto ou da superfície em observação. Este parâmetro é de fundamental

Fig. 2 - Ângulos de reflexão da luz em uma superfície difusora.

importância na definição e reconhecimento das cenas analisadas.

Dizemos, então, que o formato de um objeto pode ser extraído do "sombreamento", se forem conhecidas as funções de reflexão de sua superfície e as posições da fonte luminosa e do observador.

Evidentemente, de uma forma mais geral, as leis de reflexão possuem parâmetros relacionados à cor e à textura das superfícies. Restringindo, no entanto, nossa aplicação a objetos de uma só cor e difusores perfeitos podemos nos aproximar em grande parte de peças em uma linha de montagem industrial.

3.1 - O espaço-dual e o espaço-gradiente.

Um objeto poliédrico no espaço tri-dimensional pode ser descrito como um conjunto de planos ou semi-planos denominados de superfícies, com intersecções entre si que formam linhas ou arestas cujo ponto comum constituído por três ou mais delas estabelecem um vértice. A representação de um plano no espaço Cartesiano é normalmente dada pela equação:

$$a_x \cdot x + a_y \cdot y + a_z \cdot z + 1 = 0 \quad (8)$$

Esta representação é dita orientada a ponto.

Como no reconhecimento de objetos poliédricos nosso interesse converge para a determinação de suas superfícies formantes, seria mais conveniente uma representação orientada à plano. Tal descrição é apresentada no trabalho de Mackworth

[Mackworth 1973] como espaço-dual e no trabalho de Horn [Horn 1977] como espaço-gradiente.

O método de determinação do formato de objetos através do "sombreamento" utilizado por Horn propõe o uso do "espaço-gradiente" que pode ser entendido como uma projeção do "espaço-dual" empregado por Mackworth.

Fig.3 - Objeto representado no espaço tri-dimensional Cartesiano.

A Fig.3 mostra um objeto representado no espaço Cartesiano. Um ponto sobre uma superfície deste é dado pelas coordenadas (P_x, P_y, P_z) ou simplesmente (x, y, z) , onde um plano que contenha uma de suas superfícies é definido pela Eq.8. A dificuldade portanto, de se trabalhar com a representação

Cartesiana decorre da grande quantidade de valores necessários para se definir um objeto no espaço e sua informação posicional imanente.

O espaço-dual representa o mesmo objeto através de uma redução no domínio. Assim, um plano que no espaço Cartesiano é dado através de uma equação, é representado por um ponto no espaço-dual, especificamente pelos coeficientes da Eq.8, ou seja, (a_x, a_y, a_z) . Segue-se então que o dual de qualquer ponto (x, y, z) localizado sobre um plano descrito no espaço Cartesiano conforme a Eq.8, é um ponto (a_x, a_y, a_z) do espaço-dual.

Se uma linha do espaço tri-dimensional for resultante da intersecção de dois planos, sua representação no espaço-dual será o de uma linha unindo dois pontos conforme mostra a Fig.4. Esta linha é denominada aresta.

Fig.4 - a) Representação no espaço-dual de b) dois planos que possuem uma aresta em comum.

O espaço-gradiente é um subespaço bi-dimensional do espaço-dual onde um ponto (ax, ay, az) corresponde ao ponto $(ax/az, ay/az)$, ou seja,

Fig.5 - Projeção de um ponto do espaço-dual no espaço-gradiente.

sua projeção sobre o plano

$$az = 1.$$

A Fig.5 mostra o subespaço denominado espaço-gradiente dentro do espaço-dual. Esta representação é bastante significativa para o reconhecimento

de objetos através da orientação de suas superfícies.

Considerando que um sólido qualquer será visto por um observador segundo um determinado eixo de visão, adotemos este

eixo como sendo o eixo z Cartesiano. Na Fig.6 podemos analisar esta disposição definindo um plano de imagem como origem do sistema de coordenadas. Este é o plano no qual a imagem do objeto seria projetada de forma ortogonal ou perspectiva dependendo da localização do ponto V, e sobre o qual o observador "veria" a imagem do objeto.

Fig.6 - Eixo de visão e plano de imagem de um objeto no espaço.

Se escrevermos a equação do plano conforme a Eq.9, dizemos que a Eq.10 representa o gradiente do plano, ou seja, as inclinações desta superfície em relação ao eixo x e ao eixo y.

$$-z = \frac{a_x}{a_z} \cdot x + \frac{a_y}{a_z} \cdot y + \frac{1}{a_z} \quad (9)$$

$$(G_x, G_y) = \left(\frac{a_x}{a_z}, \frac{a_y}{a_z} \right) \quad (10)$$

Se derivarmos parcialmente a Eq.9 em relação a x e a y teremos o gradiente do plano escrito em função das derivadas parciais conforme mostra a Eq.11. Esta equação representa então, um ponto

no espaço-gradiente correspondente a um plano no espaço Cartesiano.

Pode-se ver na Fig.6 que $-z$ é a distância de um ponto (x,y,z) da superfície do objeto ao plano de imagem onde $z = 0$. O gradiente representa então a taxa de variação desta distância, relativa a movimentos realizados no plano de imagem, ou melhor, se deslocarmos sobre a superfície do objeto de um ponto P_1 para um ponto P_2 , isto corresponderá a uma variação em x e em y no plano de imagem, logo, pode-se afirmar que:

$$(G_x, G_y) = \left(\frac{\partial(-z)}{\partial x}, \frac{\partial(-z)}{\partial y} \right) \quad (11)$$

Neste método, um plano no espaço tri-dimensional pode ser definido, então, pela sua orientação e é mapeado no "espaço-gradiente" como um ponto (p,q) onde:

$$p = \frac{\partial(-z)}{\partial x} \quad (12)$$

$$q = \frac{\partial(-z)}{\partial y} \quad (13)$$

Dizemos então, que a normal em um ponto sobre o objeto é contruída em $(p,q,-1)$. Como o "espaço-gradiente" descreve o lugar geométrico das inclinações das superfícies, planos paralelos entre si serão mapeados como um único ponto e planos perpendiculares ao eixo de visão serão pontos situados na origem.

Para melhor compreensão desta ferramenta elegante e poderosa de representação das orientações das superfícies no

Fig.7 - Orientação de superfícies e sua representação no espaço-gradiente. a) Rotação em torno do eixo y. b) Rotação em torno do eixo x. c) Representação no espaço-gradiente.

espaço, atentemos à Fig.7. Um plano P1 frontal ao eixo de visão, ou seja, perpendicular ao eixo z possui sua representação no espaço-gradiente como um ponto na origem. Isto significa que qualquer deslocamento sobre este plano tanto em x como em y não causa nenhuma variação em z, logo,

$$p = \frac{\partial(-z)}{\partial x} = q = \frac{\partial(-z)}{\partial y} = 0 \quad (14)$$

Uma rotação do plano P1 em torno do eixo y, como mostrado na Fig.7-a) segundo um ângulo (α) e adotando novas posições no espaço tri-dimensional sucessivamente em P2 e P3, ocasionará, como pode ser visto na Fig.7-c) um deslocamento dos pontos sobre

o eixo p. Isto equivale dizer que um deslocamento sobre este plano causará uma alteração na distância z deste ao plano frontal. Esta variação é dada pela Eq.12 com $q = 0$. O mesmo ocorre se a rotação do plano P1 for em torno do eixo x como mostrado na Fig.7-b). A inclinação de cada plano P4, P5, será representada no espaço-gradiente da Fig.7-c) como pontos P4 e P5 sobre o eixo q. Isto significa que um deslocamento no plano inclinado desta maneira, causará uma variação em z segundo o eixo y, ou seja, a Eq.13 representa esta inclinação com $p = 0$.

Se o plano P1 é girado tanto em torno do eixo x como do eixo y, sua representação no espaço-gradiente é um ponto $P = (p, q)$. A distância da origem a qualquer ponto representativo de um plano é igual à tangente do ângulo (α) de inclinação do mesmo em relação ao plano frontal ou plano de imagem, [Marr 1982] e seu módulo é dado por:

$$\tan(\alpha) = \sqrt{p^2 + q^2} \quad (15)$$

Da mesma forma que no espaço-dual, a projeção de uma aresta no espaço-gradiente será uma linha unindo dois pontos. A concepção geométrica se mantém, pois dois pontos neste espaço definem dois planos e a intersecção entre eles é uma aresta em qualquer dos dois espaços de representação. Esta linha é denominada de linha gradiente da aresta. [Mackworth 1973]

3.2 - Arestas convexas e arestas côncavas.

Se forem sobrepostos o espaço-gradiente e o plano de imagem, o resultado é que as linhas que representam as arestas

Fig.8 - a) Vértice triédrico sobre o plano de imagem. b) Representação convexa. c) Representação côncava.

serão perpendiculares às arestas projetadas ortograficamente neste plano. Exemplificando, se tivermos o espaço-imagem de um vértice triédrico como a projeção ortográfica de um objeto sobre o plano de imagem conforme a Fig.8-a), sua representação no espaço-gradiente poderá ser tanto a mostrada na Fig.8-b) como a da Fig.8-c). Dizemos que se os gradientes são ordenados na mesma direção de como as superfícies correspondentes aparecem em relação às arestas, estas são convexas e se aparecem em direção contrária as arestas são côncavas. Esta duplicidade de interpretação que pode ser descrita como o mapeamento do gradiente no seu modo normal ou no seu modo negado é conhecida na literatura de psicologia como inversão de Necker.

Apesar da transformação do espaço-gradiente permitir um raciocínio geométrico sobre a orientação das superfícies dos

objetos da cena, ela não é capaz de por si só resolver o problema de interpretação da imagem. As medidas das variações de Luminâncias dos planos fornecerão maiores subsídios para a solução deste problema.

3.3 - Proposição do sistema de reconhecimento - Simplificações e restrições.

Retornemos à Fig.2 onde os ângulos relativos à incidência e reflexão da luz são delineados. Se deslocarmos o observador para o mesmo eixo de incidência do feixe luminoso, temos uma simplificação significativa pois, o ângulo de fase torna-se igual a zero ($g = 0$) e o ângulo de incidência (i) igual ao ângulo de emissão (e). Se a fonte de luz for pontual e a superfície do objeto considerado um difusor perfeito, a parcela de luz refletida (Luminância) em cada ponto é diretamente proporcional ao cosseno do ângulo de incidência e inversamente proporcional à distância do observador.

Para se conseguir estas características suponhamos o esquema proposto na Fig.9. Um feixe de Laser, considerado aqui como uma fonte pontual de luz, é dirigido sobre a superfície difusora de um objeto poliédrico situado em um espaço Cartesiano tri-dimensional.

Para que possamos ter o ângulo de incidência (i) igual ao ângulo de emissão (e), um espelho de diâmetro reduzido é colocado formando um ângulo de 45° com o plano da lente da câmera que receberá o feixe refletido (Luminância do ponto P).

Isto permite que tanto o feixe incidente quanto o emitido tenham a mesma direção, a do eixo de visão ou eixo z, formando o mesmo ângulo com a normal à superfície.

O observador é representado por uma câmera construída com um foto-transistor

montado no foco da lente. Este arranjo será descrito com

maior detalhe no Capítulo 3. O sistema possui a capacidade de realizar varreduras tanto no sentido do eixo x como do eixo y.

Consideremos também, apenas objetos poliédricos e difusores perfeitos colocados contra um fundo de alto contraste e assentados sobre sua base em uma superfície plana. Assim, as restrições e simplificações estabelecidas satisfazem nosso modelo, para que possamos delinear um método de determinação das inclinações de suas superfícies, a concavidade ou convexidade das arestas e conseqüentemente reconhecer o formato do objeto.

Fig.9 - Proposição do sistema de reconhecimento. Um Feixe de Laser é convenientemente dirigido sobre um objeto de superfície difusora.

3.4 - Inclinações das superfícies.

Sendo o feixe de Laser incidente capaz de projetar apenas um ponto luminoso sobre a superfície do objeto, consideremos então que o ângulo sólido correspondente é igual à unidade. Isto permite calcular a luminância de cada ponto numericamente igual à iluminância. (Eq.7)

Consideremos a Fig.10, onde a superfície de um objeto é mostrada em corte.

Um feixe de Laser é direcionado segundo o eixo de visão (eixo z) com intensidade constante (I).

Inclinando-se esta superfície de um ângulo (α) medido relativamente ao eixo de visão, pode-se ver que o ângulo de varredura do feixe de Laser (δ) varia de 0, quando o feixe de incidência está sobre o ponto P, até um valor máximo quando o mesmo está sobre o ponto B.

Sejam E_1 , E_2 , e E_P as iluminâncias dos pontos 1, 2, e P escolhidos na superfície e definidos segundo as equações Eq.16, Eq.17 e Eq.18 respectivamente.

$$(16) \quad E_1 = I \cdot \frac{\cos(i_1)}{r_1^2}$$

$$E_2 = I \cdot \frac{\cos(i_2)}{r_2^2} \quad (17)$$

$$E_p = I \cdot \frac{1}{r_0^2} \quad (18)$$

A distância r_0 é considerada entre a fonte de luz e o ponto P paralelamente ao eixo z. O plano de imagem, para simplificação do raciocínio, é adotado passando pelo ponto P da superfície e perpendicular ao eixo de visão. A projeção das distâncias de cada ponto sobre a normal são iguais conforme mostra a Eq.19, e a relação entre os ângulos de incidência e de inclinação da superfície e do feixe de Laser é vista na Eq.20.

Fig.10 - Ângulos de incidência de um feixe de Laser na direção do eixo z, sobre uma superfície inclinada.

$$r_1 \cdot \cos(i_1) = r_2 \cdot \cos(i_2) = r_0 \cdot \sin(a) \quad (19)$$

$$a = 90 - i_1 + \delta_1 = 90 - i_2 + \delta_2 \quad (20)$$

Sendo a distribuição de iluminâncias uniforme, calculemos a variação entre os dois pontos 1 e 2. Subtraindo a Eq.16 da Eq.17 temos $(E_2 - E_1)$ conforme a Eq.21. Isto significa que se o feixe de Laser for deslocado sobre a superfície ao longo do eixo x, a variação de iluminâncias, que representa o "sombreamento" ("shading") segue esta equação.

$$\Delta E = E_2 - E_1 = I \cdot \left[\frac{\cos(i_2)}{r_2^2} - \frac{\cos(i_1)}{r_1^2} \right] \quad (21)$$

Manipulando-se algebricamente, é possível escrever esta variação considerando apenas os ângulos envolvidos. Se expressarmos os valores de r_1 e r_2 em função de r_0 obtido através da Eq.19, e substituirmos este valor na Eq.21 teremos a Eq.22.

$$\Delta E = I \cdot \left[\frac{\cos(i_2)}{(r_0 \cdot \frac{\sin(\alpha)}{\cos(i_2)})^2} - \frac{\cos(i_1)}{(r_0 \cdot \frac{\sin(\alpha)}{\cos(i_1)})^2} \right] \quad (22)$$

Reorganizando os termos convenientemente e substituindo os ângulos de incidência da Eq.22 pelos ângulos de inclinação da superfície e de abertura do feixe de Laser, dados pela Eq.20, podemos reescrever a variação de iluminância conforme a Eq.23.

$$\Delta E = \frac{I}{r_0^2 \cdot \sin^2(\alpha)} \cdot [\cos^3(\delta_2 - \alpha + 90) - \cos^3(\delta_1 - \alpha + 90)] \quad (23)$$

Aplicando-se um pouco mais de trigonometria, chegamos à Eq.24.

$$\Delta E = \frac{I}{r_0^2 \cdot \sin^2(\alpha)} \cdot [\sin^3(\delta_2 - \alpha) - \sin^3(\delta_1 - \alpha)] \quad (24)$$

Finalmente, considerando toda a extensão da superfície, do ponto

$$\Delta E = E_p \cdot \frac{\sin^3(-\alpha) - \sin^3(\delta - \alpha)}{\sin^2(\alpha)} \quad (25)$$

P ao ponto B, o ângulo do feixe de Laser descreve uma abertura máxima igual a (δ) e desta forma, podemos dizer que $(\delta_1 = 0^\circ)$ e $(\delta_2 = \delta)$ resultando na Eq.25. Esta equação descreve a taxa de variação da iluminância em uma superfície inclinada em relação ao eixo x, em função do valor E_p de um ponto perpendicular ao feixe de luz incidente.

Se lançarmos em um gráfico a função de variação escrita na Eq.26, obteremos as curvas mostradas na Fig.11.

$$f(\delta, \alpha) = \frac{\sin^3(-\alpha) - \sin^3(\delta - \alpha)}{\sin^2(\alpha)} \quad (26)$$

Fig.11 - Taxa de variação da iluminância em uma superfície inclinada de um ângulo α , relativamente ao eixo x.

Estas curvas demonstram que para um plano perpendicular ao eixo de visão ($\alpha=90^\circ$) a variação de iluminância ao longo da superfície, dentro dos limites de varredura do feixe de Laser (eixo δ na Fig.11), é desprezível se comparada a planos

inclinados. Por esta razão, a iluminância pode então ser considerada constante, e por isso sua representação no espaço-gradiente é um ponto situado na origem.

Resumindo, podemos afirmar que um deslocamento do feixe de Laser sobre a superfície do objeto, na direção do eixo x e no esquema proposto na Fig.9, produzirá no observador (foto-transistor) localizado no eixo z , variações proporcionais de intensidade de sinal, segundo as curvas da Fig.11. Estas variações convenientemente digitalizadas, fornecerão subsídios para o reconhecimento do objeto através de um Microcomputador.

3.5 - Determinação de características do objeto através da variação da luminância.

Suponhamos o esquema proposto na Fig.9.

Quando o feixe de Laser varre um objeto da esquerda para a direita, os valores digitalizados das luminâncias em cada ponto são armazenados em um "arranjo de intensidades". Este nome identifica-se com "intensidades de nível de cinza", utilizado pelos métodos de reconhecimento tradicionais para designar as graduações dos "pixels" das imagens a serem processadas.[Shirai 1975] Os conceitos são semelhantes apesar das diferenças básicas estruturais.

Analisemos a Fig.12 como sendo o "plano de imagem" de um objeto colocado no espaço Cartesiano, cujo eixo de visão coincide com o eixo z . Seja o plano (x,z) horizontal e o plano (x,y) vertical. O feixe de Laser pode realizar varreduras no

plano (x,y)
da esquerda
para a direita,
ou seja,
varreduras
horizontais,
gerando um
arranjo de
intensidades na
forma da Eq.27.
O sistema pode
realizar também
varreduras no

Fig.12 - "Plano de Imagem" do sistema de reconhecimento. Varredura do feixe de Laser.

$$A_{x_i, y_j} = [I_{x_1, y_j}, I_{x_2, y_j}, \dots, I_{x_n, y_j}] \quad (27)$$

sentido vertical, de baixo para cima no mesmo plano (x,y), sendo o arranjo de intensidades descrito como:

$$A_{x_i, y_j} = [I_{x_i, y_1}, I_{x_i, y_2}, \dots, I_{x_i, y_m}] \quad (28)$$

Cada elemento do arranjo representa a iluminância do ponto medida diretamente sobre o objeto. Um deslocamento realizado pelo feixe de Laser no eixo x ou y, corresponde a uma modificação nesta grandeza, que pode ser entendida como variações proporcionais no eixo z. Logo, podemos inferir daí que o vetor (p,q) que define um ponto no espaço-gradiente correspondente a um plano, pode ser obtido através das equações Eq.29 e Eq.30.

$$P = - \left(\frac{I_{x_1, y_1} - I_{x_{1-1}, y_1}}{x_1 - x_{1-1}} \right) \quad (29)$$

$$Q = - \left(\frac{I_{x_1, y_1} - I_{x_1, y_{1-1}}}{y_1 - y_{1-1}} \right) \quad (30)$$

O sinal de menos é utilizado para se manter a coerência com a Eq.11, pois o plano $z = 0$ coincide com o plano de imagem, e qualquer superfície do objeto apresenta então variações na direção negativa do eixo z .

Podemos tecer agora várias considerações que estabeleçam parâmetros para descrever as inclinações e as arestas de um poliedro. Para uma melhor avaliação destas características observemos a Fig.13; nela, três possíveis inclinações de dois planos que interceptam-se no espaço são mostradas.

A Fig.13-a) apresenta estes dois planos inclinados em relação ao eixo x e ambos paralelos ao eixo y . Quando o "arranjo de intensidades" em x for determinado, duas situações podem ocorrer, analisando-se as componentes (p) para os planos P_1 e P_2 , denominadas aqui de p_1 e p_2 respectivamente:

a) se p_1 é negativo e p_2 positivo implica que a aresta que os une é convexa, independentemente da obtenção do "arranjo de intensidades" em y .

b) se p_1 é positivo e p_2 negativo implica que a aresta que os une é côncava, independentemente da obtenção do "arranjo de intensidades" em y .

Isto equivale a dizer que o "arranjo de intensidades" em x presta-se a determinação de arestas não paralelas ao eixo x .

Na Fig.13-b) temos os dois planos, P_1 e P_2 inclinados

Fig.13 - Análise de três possibilidades de inclinação no espaço de dois planos unidos por uma aresta e projetados em x . a) Inclinação em x . b) Inclinação em y . c) Inclinação em x, y .

relativamnete ao eixo y , com a aresta que os une paralela ao eixo x . Isto determina que a componente p do vetor (p, q) de inclinação das duas superfícies p_1 e p_2 são ambos nulos. A componente q será obtida através do "arranjo de intensidades" em y e, novamente, duas situações podem ser avaliadas:

- a) se q_1 é negativo e q_2 positivo está estabelecida a existência de uma aresta convexa unindo os dois planos.
- b) se q_1 é positivo e q_2 negativo está estabelecida a existência de uma aresta côncava unindo os dois planos.

A Fig.13-c) apresenta o caso geral onde os dois planos P_1 e

Pz possuem sua intersecção inclinada tanto em relação ao eixo x como ao eixo y. Neste caso o vetor de inclinação (p,q) que define cada plano é calculado, cada componente através do "arranjo de intensidades" correspondente, e seu módulo é dado através da Eq.15. A convexidade ou concavidade da aresta pode ser obtida tanto pela componente p como pela componente q, e devem ser coerentes, ou seja, se o "arranjo de intensidades" em x determinar que a aresta é convexa, o realizado em y deverá ter o mesmo resultado. Uma discrepância significará que houve erro no cálculo das componentes do vetor de inclinação.

Considerando, como exemplo, o sólido mostrado na Fig.14, e três varreduras com o feixe de Laser realizadas sobre ele e designadas por $f(x,y_1)$, $f(x,y_2)$ ao longo do eixo x, e $f(x_t,y)$ ao longo do eixo y. Considerando o eixo de visão na direção de z (perpendicular à aresta frontal do desenho), podemos verificar através da Fig.15 o comportamento do "arranjo de intensidades" e das componentes p e q do vetor de inclinação das superfícies. Nesta figura, considerada para a varredura $f(x,y_1)$ mostrada na Fig.14 podemos ver que o perfil de intensidades acompanha o perfil das superfícies do objeto. Da origem ao ponto 1 e do ponto 2 em diante, temos o fundo de alto contraste e perpendicular ao eixo de visão.

Fig.14 - Varredura do feixe de Laser em um sólido.

Fig.15 - Comportamento do "arranjo de intensidades" e das componentes p e q de inclinação das superfícies do sólido da Fig.14.

Consideramos portanto, sua luminância igual a zero. O ponto identificado com a letra (a) corresponde à aresta formada pela duas superfícies observadas. Neste ponto, a componente (p) muda de direção. Assim, tanto a fronteira do objeto (pontos 1 e 2) como a aresta formada pelos dois planos (ponto a) são identificados através da análise do comportamento de apenas uma varredura horizontal. Para se checar a coerência realizamos outras varreduras, como por exemplo a $f(x, y_v)$ e para verificação da inclinação das superfícies com relação ao eixo y , varreduras verticais como a $f(x_t, y)$ são requeridas.

3.6 Método de rotulagem de linhas em objetos poliédricos.

Toda a teoria do processo de se dar rótulos a linhas formantes de objetos tri-dimensionais foi desenvolvida para a representação bi-dimensional destes mesmos objetos. Estes "desenhos" ("line drawings") são o meio de comunicação entre seres humanos, sobre os aspectos quantitativos dos objetos tri-dimensionais. A classe dos poliedros suscitou aos pesquisadores o interesse em descrevê-los através de um meio gráfico de codificação de suas arestas. Este trabalho foi desenvolvido em 1971 independentemente por M.B. Clowes [Clowes 1971] e por D.A. Huffman [Huffman 1971], sendo conhecido hoje pelo nome de método de rotulagem de Huffman-Clowes [Ballard & Brown 1982],[Kirosis 1990].

O "desenho" de um objeto, no entanto, não possui em si a informação tri-dimensional necessária havendo por isso a possibilidade de ambigüidade em sua interpretação e também de objetos impraticáveis fisicamente.

A Fig.16 mostra as convenções adotadas por este método que nos permitirá classificar as arestas que constituem os objetos poliédricos fornecendo subsídios para sua descrição. Assim, uma aresta determinada como convexa através do "arranjo de intensidades" receberá o sinal (+), uma aresta côncava

Fig.16 - Convenção de rotulagem de linhas.

o sinal (-). As linhas que formam as bordas ou fronteiras do objeto serão identificadas com uma seta (--->) cuja direção indica que o plano do objeto está localizado à direita da linha e evidentemente o lado esquerdo é o fundo contra o qual a cena é observada.

Fig.17 - Rotulagem de um cubo sobre uma superfície.

Se examinarmos a Fig.17 veremos o "desenho" de um cubo assentado sobre uma superfície e observado segundo um determinado eixo de visão. As arestas são rotuladas mostrando sua fronteira, um vértice triédrico convexo (formado pela junção das três arestas convexas) e duas arestas côncavas formadas pela intersecção

da superfície que o suporta com suas faces visíveis.

Se reduzirmos nosso domínio para objetos sólidos triédricos, uma investigação sistemática pode mostrar que existem apenas algumas possibilidades de rotulagem dos vértices visíveis destes objetos [Clowes 1971] [Huffman 1971]. Quatro tipos de vértices

Fig.18 - Possibilidades de rotulagem de vértices triédricos de objetos sólidos.

podem ocorrer (T, SETA, GARFO, L) e somente 18 combinações de

Fig.19 - Exemplos de algumas rotulagens de vértices triédricos.

suas arestas são rotuladas conforme mostra a Fig.18, pois uma regra de coerência básica diz que nenhuma linha pode mudar seu rótulo entre dois vértices de um poliedro real. Os vértices GARFO (ou Y) podem ser definidos onde os três ângulos formados são menores do que 180° . Os vértices SETA (ou E) são aqueles onde um dos ângulos é maior que 180° . Os vértices L possuem grau dois e os vértices T são formados por duas de suas arestas colineares.

Como exemplo de algumas dessas rotulagens, atentemos para a

Fig.19 onde tipos de objetos triédricos são mostrados e identificados alguns de seus vértices com as respectivas rotulagens. Os números correspondem aos rótulos de Huffman-Clowes delineados na Fig.18.

É importante salientar que o método de rotulagem de Huffman-Clowes descreve apenas objetos triédricos sólidos de superfícies planares.

Alguns modelos, como por exemplo a caixa mostrada na Fig.20, não podem ser

Fig.20 - Objeto não rotulável pelo método de Huffman-Clowes. Os vértices assinalados podem ser descritos pelo "mundo Origami".

rotulados por este processo. Outros "desenhos" como o da Fig.21

Fig.21 - Objeto impossível de existir fisicamente, mas rotulável pelo método de Huffman-Clowes.

são perfeitamente rotuláveis mas impraticáveis fisicamente. Isto ocorre devido ao fato da técnica ter sido desenvolvida para imagens projetadas de objetos tri-dimensionais ("line drawings"), ou seja, os rótulos na verdade são bi-dimensionais.

Alguns autores, propuseram sistemas expandidos que ajudam a solucionar estas questões como por exemplo o "mundo Origami" [Kanade 1981] onde os objetos são entendidos como se

fossem formados de papel dobrado. Outros, [Chakravarty 1979] apresentaram soluções para a descrição de objetos curvos através da rotulagem de vértices formados com superfícies curvas, e outros ainda [Waltz 1975] preocupados com a iluminação envolvida descreveram a rotulagem de sombras dos objetos. N o s s o trabalho, no entanto, evita o problema dos "objetos impossíveis" pois lida com a informação tri-dimensional da cena de um objeto real presente nela e não com "desenhos" de suas linhas projetadas. O nosso propósito utilizando uma tecnologia baseada na varredura de um feixe de Laser, é de ater-mo-nos ao método de Huffman-Clowes para que possamos estabelecer parâmetros de validação da técnica para outras superfícies.

4. CONCLUSÕES.

A teoria de "determinação do formato através do sombreamento" ("Shape from Shading") é empregada neste trabalho com o intuito de levantar as características básicas de um objeto tri-dimensional.

A utilização do espaço-gradiente proposto por Horn [Horn 1975] permite que poliedros possam ser descritos pela inclinação de suas faces e não por pontos no espaço Cartesiano. Isto reduz sensivelmente a quantidade de informação necessária para a descrição de um objeto e torna-a independente de sua posição na cena.

A utilização de um feixe de Laser como fonte pontual de luz e posicionado na mesma direção do observador determina uma

interpretação mais simplificada das equações de reflexão. Este feixe de Laser, devidamente controlado e dirigido por um Microcomputador, permite levantar um "arranjo de intensidades" luminosas que correspondam às luminâncias das superfícies Lambertianas consideradas. A variação destas intensidades fornece subsídios para a determinação das inclinações dos planos formantes dos objetos e conseqüentemente a convexidade ou concavidade de suas arestas.

Baseado na abordagem de Huffman-Clowes para rotulagem das arestas de triedros sólidos perfeitos, que realiza uma catalogação dos vértices possíveis em uma cena, nosso método classificará os objetos. Portanto, o universo plausível de categorização será constituído por objetos sólidos, triédricos e de superfícies Lambertianas.

Uma primeira vantagem deste nosso sistema de reconhecimento é que ele opera diretamente sobre a cena, em tempo real, e não sobre uma imagem desta, eliminando o problema dos "objetos impossíveis" existentes em outros métodos que empregam rotulagem.

Considerando que através de uma regra de coerência, nenhuma aresta pode modificar seu rótulo entre dois vértices, então, com apenas uma varredura podemos defini-la e conseqüentemente os vértices em sua extremidade. Isto representa outra vantagem do método, pois implica em um aumento significativo de velocidade reduzindo o tempo entre a aquisição da "imagem" e a classificação do objeto.

Portanto, em aplicações industriais, como por exemplo na seleção de peças em linhas de montagem ou no empilhamento de

caixas de diversificados formatos, um Robô guiado através do reconhecimento empregado neste método, rapidamente realizaria sua tarefa. O tempo necessário para o processamento da imagem, considerado nos métodos tradicionais de visão de máquina, foi aqui suprimido.

5. COMENTÁRIOS BIBLIOGRÁFICOS.

O assunto tratado neste capítulo envolve basicamente duas linhas de pesquisa em visão computacional. A primeira é a "determinação do formato através de sombreadimento" ("shape from shading") proposta por Horn [Horn 1975][Horn 1977] de onde utilizamos a técnica de representação de superfícies através do espaço-gradiente. Anteriormente ao trabalho de Horn, Mackworth [Mackworth 1973] propunha a utilização do espaço-dual, com referências ao espaço-gradiente. Recentemente, Horn juntamente com outros pesquisadores tem publicado trabalhos nesta área, mas sempre envolvendo a solução das equações diferenciais determinadas no primeiro trabalho e a utilização do "mapa de reflectâncias", bastante útil quando a fonte de luz é distante.[Ikeuchi & Horn 1981][Horn & Brooks 1986]. No nosso trabalho, no entanto, a fonte de luz está localizada próximo do objeto e o "mapa de reflectâncias" não se aplica.

Outros pesquisadores tem feito vários avanços dentro desta proposta,[Pentland 1984][Lee & Rosenfeld 1985][Pentland 1988][Nayar et al. 1990] dirigindo seus estudos para a solução matemática de várias questões do método.

A segunda linha de pesquisa refere-se à técnica de rotulagem de arestas formantes de objetos triédricos sólidos proposta independentemente por dois trabalhos: o de Clowes [Clowes 1971] e o de Huffman [Huffman 1971] tornando-se conhecida com o nome de rotulagem de Huffman-Clowes. Dois autores antecederam a este método e são dignos de nota: Guzman [Guzman 1969] e Falk [Falk 1972] propuseram soluções que definem os vértices dos sólidos, mas não uma catalogação sistemática dos mesmos.

Preocupado com os objetos formados tanto por superfícies planas como por curvas, Chakravarty [Chakravarty 1979] propôs a rotulagem de vértices com curvas em adendo ao método de Huffman-Clowes. Outro trabalho interessante é o de Kanade [Kanade 1981] que percebendo a inviabilidade de rotulagem de objetos não sólidos perfeitos, estudou a utilização do "Origami World" que supõe que os objetos são formados por papel dobrado. Desta forma uma grande extensão de vértices são acrescentados ao catálogo de Huffman-Clowes.

No artigo publicado por Kirousis [Kirousis 1990], uma formulação formal foi fornecida para regras até então utilizadas de forma intuitiva nos outros estudos sobre rotulagem.

Enfocando aspectos psicológicos da visão, Attneave [Attneave 1954] apresentou este artigo de leitura indispensável a todo pesquisador em visão computacional. Conceitos como o que a informação visual concentra-se ao longo do contorno dos objetos, ou melhor, nos pontos onde um contorno muda de direção abruptamente, é aí amplamente exposto e discutido. Também o livro de Marr [Marr 1982] além de trazer todos os conceitos de

visão computacional, empreende um debate no final do volume, de grande valia para a reflexão sobre os conceitos visuais.

Finalmente, o assunto tratado neste capítulo pode também ser encontrado no trabalho de Barrow e Tenenbaum [Barrow & Tenenbaum 1981] que fez um estudo sobre as diversas técnicas em visão computacional, com especial atenção para a "determinação do formato através do sombreamento" e para a rotulagem de Huffman-Clowes. Também o livro de Ballard [Ballard & Brown 1982] é indispensável para que se possa ter uma idéia comparativa das diversas técnicas existente em visão de computadores.

CAPÍTULO 3

SISTEMA DE RECONHECIMENTO DO FORMATO DE OBJETOS

1. INTRODUÇÃO.

Com a necessidade cada vez maior de automação do processo de produção industrial e também dos sistemas de distribuição física envolvendo manipulação mecânica, os equipamentos de Visão Computacional estão se tornando dia a dia mais importantes.

O assunto tem sido estudado por um grande número de pesquisadores, ao longo de vários anos, e diversos sistemas propostos são implementados quase sempre de modo dedicado.[Jarvis 1977][Yoda et all. 1973]

De uma maneira geral, as soluções são geradas através da aplicação de técnicas definidas em Reconhecimento de Padrões sobre uma imagem processada da cena. Isto requer normalmente uma quantidade de memória elevada disponível para o armazenamento e o processamento dos "pixels", consumindo também um tempo considerável. Por essa razão, são utilizados na maioria destes trabalhos, computadores de grande porte.

Denominamos aqui de sistema alternativo, um método que não empregue imagens digitalizadas para o reconhecimento de objetos. Os mais conhecidos são os medidores de distância ("range finder") que podem utilizar Laser [Jarvis 1983][Rioux 1984][Stuivinaga et all. 1989] ou ultrason [Monchaud 1989]. A literatura sobre Visão Computacional está repleta de propostas

que usando uma câmera de TV captam imagens de cenas tri-dimensionais e a partir de planificações destas categorizam os elementos espaciais.

Acreditamos que os métodos tradicionais empregando uma câmera de TV possuam maior número de adeptos devido ao fato que os pesquisadores em Visão Computacional estão em geral envolvidos com processamento de imagens.

Um fato porém leva-nos a crer que sistemas alternativos possam ser de grande valia em aplicações industriais: uma imagem de TV transporta uma quantidade de informação superior à necessidade da maioria das aplicações de Visão de Máquina na indústria. Em outras palavras, um Robô em grande parte das situações não precisa "ver" com a mesma precisão de detalhes da visão humana. Esta informação "em excesso" é eliminada através do processamento da imagem, consumindo-se grande quantidade de tempo e custo computacional.

O sistema que propomos neste trabalho visa obter dados suficientes para que um manipulador mecânico possa ser acionado no menor tempo possível. Não nos preocupamos aqui com este acionamento, mas somente com os circuitos responsáveis pela aquisição do sinal e do algoritmo necessário ao reconhecimento do formato do objeto considerado na cena.

Dentre as técnicas alternativas possíveis, fizemos opção pela utilização de um feixe de Laser que pode ser convenientemente controlado e dirigido sobre o objeto a ser reconhecido, através de um Microcomputador.

2. CONFIGURAÇÃO GERAL DO SISTEMA.

A Fig.1 mostra um esquema geral de distribuição dos componentes para implementação do sistema de reconhecimento do

Fig.1 - Sistema de reconhecimento do formato de objetos tri-dimensionais, utilizando varredura a Laser.

formato de objetos tri-dimensionais aqui proposto.

Uma mesa giratória suporta um objeto, considerado triédrico e difusor perfeito. Esta mesa permite através de controle do Microcomputador, girar e expor o sólido da cena em 360° a seu observador. Este por sua vez é formado por um foto-transistor, localizado no foco de uma lente, dentro de uma câmera escura.

Este dispositivo transforma a intensidade luminosa, do feixe de Laser refletido sobre o objeto, em um sinal eletrônico condicionado pela interface analógica. O eixo de visão é considerado o eixo Cartesiano z, conforme a análise vista no Capítulo 2.

Fig.2 - Diagrama em blocos do sistema de reconhecimento do formato de objetos tri-dimensionais.

Um feixe de Laser é dirigido sobre o objeto através de dois espelhos ligados a galvanômetros que realizam a varredura no plano (x,y) de acordo com o programa no Microcomputador. O feixe incidente é desviado por um primeiro espelho de reduzido diâmetro, colocado formando um ângulo de 45° com o plano da lente e bem no centro desta. O feixe refletido pelo objeto que carrega a informação de luminância necessária para o reconhecimento, retorna através dos dois espelhos defletores (x,y) e atinge o foto-transistor como um feixe de luz paralela

convergado pela lente.

A interface analógica retira o sinal, filtra-o e adapta-o a nível suficiente de ser convertido pela interface digital do Microcomputador. Esta possui então a capacidade de gerenciamento da posição do objeto, de varredura do feixe de Laser e de conversão analógica/digital do sinal refletido. O sistema é representado em diagrama em blocos conforme a Fig.2.

Descrevemos então, cada bloco fazendo as referências de interligação quando necessárias. Desta forma procuramos dar uma idéia clara da operação do sistema como um todo.

3. A INTERFACE DIGITAL DE CONTROLE DO SISTEMA.

O primeiro bloco discutido pode ser considerado o coração deste sistema de visão. A interface eletrônica que adapta os dados oriundos do programa de controle do Microcomputador, transmite as ordens de varredura aos galvanômetros dos espelhos (x,y), gira a mesa e converte o sinal analógico correspondente ao feixe de Laser refletido, em um número digital coerente. Um diagrama em blocos desta interface é mostrado na Fig.3. (O Apêndice A-1 detalha seu esquema eletrônico completo.)

De uma maneira simplificada, o lado esquerdo da figura mostra a conexão com o Microcomputador (compatível com IBM PC-XT). O duto de dados de 8 bits é acoplado à interface e os dados transferidos através de uma operação de escrita, a um dos quatro registradores internos. Estes, por sua vez, armazenam tanto os valores para os conversores digital/analógico de 12 bits como os

Fig.3 - Interface digital de controle do sistema.

bits de controle de rotação da mesa. Cada conversor D/A aciona um dos galvanômetros dos espelhos de deflexão dos eixos x e y.

O conversor analógico/digital converte o valor de tensão, oriundo da interface analógica e armazenado por um circuito amostrador "sample & hold", em um número digital de 12 bits. Este número é então lido pelo Microcomputador.

Um sensor de posição detecta quando a mesa giratória está em seu ponto de início e transmite esta informação através de 1 bit a um registrador de entrada da interface.

A lógica de controle e mapeamento é responsável pela definição e atribuição dos endereços a cada elemento e também pela ativação destes. A lógica de gatilho permite que os 12 bits necessários ao conversor D/A do eixo x e os 12 bits do conversor D/A do eixo y sejam transferidos simultaneamente para gerar valores concomitantes em suas saídas. Isto evita deflexões indesejáveis causadas por dados anteriores de qualquer uma das duas variáveis.

A Tab.I mostra os endereços das portas que representam a interface digital, bem como a função de cada um dentro do contexto que um programa no Microcomputador necessita gerar.

3.1 - Deflexão dos espelhos para a varredura do feixe de Laser.

Observando-se a Tab.I concluímos que, para se defletir os espelhos dos galvanômetros do eixo x e do eixo y concomitantemente, o programa deve proceder da seguinte forma:

a) os 8 LSB do conversor D/A do eixo x devem ser armazenados na

Tab.I - Endereços das portas da interface digital

PORTA	OPERAÇÃO	FUNÇÃO
210	ESCRITA	Armazena os 8 LSB do D/A do eixo x
211	ESCRITA	Armazena os 4 MSB do D/A do eixo x e o controle da mesa giratória.
212	ESCRITA	Armazena os 8 LSB do D/A do eixo y
213	ESCRITA	Armazena os 4 MSB do D/A do eixo y
214	ESCRITA	Transfere simultaneamente 12 bits do D/A do eixo x e 12 bits do D/A do eixo y
215	-----	Não usada
216	ESCRITA	Início de conversão de 12 bits no A/D
217	ESCRITA	Início de conversão de 8 bits no A/D
216	LEITURA	Leitura dos 8 MSB do A/D
217	LEITURA	Leitura dos 4 LSB do A/D
218	LEITURA	Verificação de fim de conversão no A/D (STS)
219	LEITURA	Amostragem do sinal e armazenamento da informação de início da mesa giratória
21A	LEITURA	Leitura da informação de início da mesa giratória

porta de endereço 210 hexadecimal.

b) os 4 MSB do conversor D/A do eixo x devem ser armazenados na porta de endereço 211 hexadecimal.

c) os 8 LSB do conversor D/A do eixo y devem ser armazenados na porta de endereço 212 hexadecimal.

d) os 4 LSB do conversor D/A do eixo y devem ser armazenados na porta de endereço 213 hexadecimal.

e) a porta de endereço 214 hexadecimal deve ser acionada para escrita transmitindo 24 bits simultaneamente. 12 bits para o conversor D/A do eixo x e 12 bits para o conversor D/A do eixo y.

O programa de deflexão deve tomar o cuidado para não alterar os bits mais significativos do segundo registrador interno da interface pois estes controlam a rotação da mesa. Isto é realizado mantendo-se na memória um registro de valor atualizado e realizando-se um "ou" lógico com cada nova alteração do conversor D/A do eixo x. (detalhes no Capítulo 4)

Na Fig.4 podemos ver o campo definido pela deflexão dos dois galvanômetros, medido no fundo contra o qual é colocado o objeto a ser reconhecido. Este campo é limitado em aproximadamente 30 cm

em cada eixo, para o protótipo implementado neste trabalho.

Evidentemente este tamanho (l_x ou l_y) depende apenas da distância (d) se mantivermos constante o ângulo de deflexão (δ),

conforme mostra a Eq.1. **Fig.4 - Campo de deflexão dos galvanômetros.**

Este parâmetro é

calibrado na placa de interface digital através do ajuste de ganho do amplificador operacional de saída de cada conversor D/A. (ver Apêndice A-1 para detalhes do esquema eletrônico)

$$d = \frac{l_x \cdot \cos(\delta_x / 2)}{2 \cdot \sqrt{1 - \cos^2(\delta_x / 2)}} \quad (1)$$

Estes ganhos são regulados, para este trabalho, tal que os ângulos de deflexão sejam $\delta_x = \delta_y = 20^\circ$. Isto gera um campo de aproximadamente 30 cm X 30 cm a uma distância média de 85 cm dos espelhos.

O circuito de controle dos galvanômetros é formado por uma fonte de tensão constante, ajustável para cada eixo, cujo valor de saída é proporcional ao fornecido pelo conversor D/A correspondente (ver Apêndice A-2). Este circuito fornece a corrente necessária à operação de deflexão com excelente grau de estabilidade e repetibilidade.

Praticamente, para o protótipo implementado podemos ter com o conversor de 12 bits utilizado, 4096 (quatro mil e noventa e seis) pontos no eixo x e 4096 pontos no eixo y. Isto equivale dizer que a variação de um bit na entrada do conversor D/A produz um deslocamento de

$30 \text{ cm} / 4096 = 7,3 \times 10^{-3} \text{ cm}$ em um dos

eixos, medido sobre o fundo da cena. Este valor é evidentemente muito pequeno e uma varredura considerando todos os pontos em qualquer um dos dois eixos seria muito lenta e desnecessária. Se observarmos a Fig.5 podemos ver que o ponto projetado pelo feixe de Laser no fundo da cena é, para esta implementação, de aproximadamente 0,4 cm de diâmetro.

Se fizermos a varredura conforme mostra esta figura, intercalando cada ponto de 0,2 cm em 0,2 cm, no mínimo $0,2 \text{ cm} / 7,3 \times 10^{-3} \text{ cm} = 27$ variações de valores digitais são necessários

Fig.5 - Diâmetro do ponto projetado do feixe de Laser.

para cada ponto. Logo, para completar os 30 cm em cada eixo, $4096/27 = 151$ códigos binários devem ser gerados.

Para esta distância e este campo definido, um conversor D/A de 8 bits seria suficiente. A adoção de um conversor de 12 bits foi realizada porque para campos um pouco maiores (por exemplo 60 cm) o conversor de 8 bits não permitiria este tipo de varredura com a mesma precisão e repetibilidade.

3.2 - Rotação da mesa que suporta o objeto.

Para que o programa de reconhecimento tenha acesso às diversas faces que compõe um objeto, o sistema é dotado de uma mesa giratória que o suporta. Outros autores utilizam também este recurso para a visão múltipla da cena [Connolly & Stenstrom 1986]. Esta mesa é acionada por um motor de passo ligado a ela por uma correia de transmissão dentada, e comandado por um circuito de "driver" (ver Apêndice A-3). Este circuito recebe pulsos da interface digital cujo programa atualiza os 4 bits mais significativos do registrador interno de endereço 211 hexadecimal. Deve ser lembrado, novamente, que os 4 bits menos significativos deste mesmo registrador correspondem aos 4 MSB do conversor D/A do eixo x. Portanto, um pulso no motor de passo não deve alterar o valor do dado de deflexão, devendo por isso ser mantido na memória do Microcomputador, um registro de atualização desta variável.

Outra característica importante da mesa giratória é que ela é indexada. Um foto-acoplador, conforme mostra a Fig.6, registra

a passagem de um traço escuro em um aro transparente. Isto informa ao programa através do bit D0 do registrador interno da interface digital, (endereços 219 e 21A hexadecimal) que a mesa encontra-se na sua posição inicial.

Fig.6 - Indexação da mesa giratória.

Como o acionamento é realizado através de um motor de passo, apenas um índice é suficiente pois o motor trabalha em malha aberta e outras

posições são obtidas enviando-se um número exato de pulsos ao circuito acionador. Desta forma, o objeto é exposto em quatro quadrantes facilitando o reconhecimento de suas superfícies dentro do programa.

Fig.7 - Dois planos e uma aresta convexa. Podem formar um sólido ou não.

Por exemplo, as duas superfícies A e B da Fig.7 tendo sido classificadas como convexas na primeira análise, podem ser formantes de um objeto sólido triédrico ou podem ser apenas dois planos unidos por uma aresta e

apoiados sobre a mesa. Um giro de 180° mostrará outra face deste

objeto. Se o mesmo for um sólido triédrico perfeito suas arestas serão sempre convexas. A concavidade mostrará, por exemplo, que o objeto em questão pode ser apenas dois semi-planos. Como o eixo de visão é paralelo ao eixo Cartesiano z, apenas um lado do objeto poderia causar informação insuficiente para se decidir a respeito de sua classificação.

Por isso, o programa de controle realiza quatro giros de 90° cada, tomando decisão em cada um deles. A coerência destes resultados resulta no reconhecimento do formato do objeto visualizado.

4. A INTERFACE ANALÓGICA PARA AQUISIÇÃO DO SINAL.

A maior dificuldade de recepção da parcela de luz refletida do feixe de Laser incidente sobre o objeto é sua baixa intensidade e conseqüentemente o alto nível do ruído. Quando sinais da ordem de microvolts encontram-se na presença de ruídos na casa de volts, uma solução de medida é a utilização de um amplificador "lock-in". Este tipo de circuito é um voltímetro ac extremamente seletivo usado para medir sinais de frequência única obscurecidos por ruído. A interface analógica do sistema de reconhecimento do formato de objetos tri-dimensionais é basicamente um amplificador "lock-in" conforme mostra o diagrama da Fig.8 (ver Apêndice A-4 para detalhes do circuito eletrônico). Ele rejeita o ruído aleatório, transientes, interferência de frequências discretas e harmônicas. O circuito produz uma saída de proporcional ao valor rms da forma de onda

Fig.8 - Diagrama do amplificador "lock-in" da interface analógica. Vide texto para comentários sobre as formas de onda nos pontos marcados.

senoidal da frequência fundamental do sinal medido.[Fisher 84]

4.1 - Controle de velocidade do motor de "chopper".

Para que possamos ter um sinal de referência em uma frequência constante que faça o chaveamento do amplificador "lock-in" em sincronismo com o sinal proveniente do foto-detector, um motor dc é usado em velocidade constante interrompendo o feixe de Laser. O circuito de controle [Roda 82] detalhado no Apêndice A-4 é mostrado de forma esquemática na Fig.9.

O motor dc que secciona o feixe de Laser é do tipo "cup motor" de baixa inércia possuindo em seu eixo uma pá dupla

Fig.9 - Diagrama do circuito de controle de velocidade do motor de "chopper".

balanceada que também interrompe um foto-acoplador sincronamente. O sinal deste foto-acoplador passa por um conformador que nada mais é do que um disparador saturado que transforma seu sinal em uma onda quadrada de frequência constante igual a 90 Hz. A saída do conformador fornece então a referência para o amplificador "lock-in" sincronizar a recepção.

A onda quadrada na saída do conformador é aplicada em um circuito derivador que por sua vez disparará um monoestável de largura de pulso constante. A integração deste trem de pulsos fornece um valor dc de referência ao circuito de ajuste de velocidade. Sua saída é então amplificada em potência para acionamento do motor. Portanto, a forma de onda no ponto A das figuras 8 e 9 é uma onda quadrada de frequência constante igual a 90 Hz (ver Fig.10).

4.2 - Detecção e filtragem do sinal refletido sobre o objeto.

O feixe incidente de Laser é seccionado a uma frequência

constante de 90 Hz e a parte refletida na direção do eixo de visão é captada por um foto-transistor colocado no foco de uma lente convergente. Como o Laser utilizado é de H e N e , um pequeno filtro de vidro na faixa do vermelho, realiza a

Fig.10 - Formas de onda equivalentes ao circuito da interface analógica.

primeira filtragem ótica para reduzir a interferência de luz ambiental. Este filtro é colocado diretamente sobre o foto-transistor dentro da câmara escura, no caminho ótico do feixe de Laser refletido. Na Fig.8 podemos ver o esquema de captação deste sinal e sua respectiva filtragem sendo que as formas de onda são mostradas na Fig.10. No ponto C o sinal é extremamente ruidoso, mas este ruído está sobreposto a uma onda quadrada na frequência de 90 Hz resultante da detecção da parcela refletida do feixe de laser sobre o objeto de forma "chopperizada". O

primeiro filtro é um passa banda de estado variável [Stout & Kaufman 1976] projetado com amplificadores operacionais e que reforça a frequência fundamental do sinal ruidoso captado, reduzindo enormemente as restantes. O segundo filtro, também ativo, é um rejeita banda duplo-T [Stout & Kaufman 1976] com a frequência de corte em 60 Hz.

O resultado da detecção e da filtragem do sinal, mostrado pelas formas de onda da Fig.10 nos pontos C e D, é a geração de uma senóide na frequência de 90 Hz cuja amplitude é proporcional à parte refletida do feixe de Laser sobre o objeto. Podemos afirmar então que a amplitude desta senóide equivale à Luminância do ponto sobre a superfície Lambertiana visualizada segundo o eixo z.

4.3 - Retificação e integração do sinal.

Após ter sido filtrado, o sinal agora é retificado sincronamente com a referência de 90 Hz gerada pelo circuito de controle do motor de "chopper". Através do esquema da Fig.11 podemos ver que isto é realizado através do chaveamento das partes positivas e negativas do sinal, alternadamente na entrada de dois amplificadores inversores. Observando também a Fig.12 podemos ver as diversas formas de onda em cada ponto dentro do retificador/misturador. Para detalhes a respeito do circuito eletrônico que realiza esta tarefa, consultar o Apêndice A-4.

Assim, a senóide aplicada na entrada do circuito, no ponto D, é retificada sob controle da referência de 90 Hz. Esta é

Fig.11 - Esquema do retificador/misturador do circuito do amplificador "lock-in".

ajustada na sua fase para que coincida o pico positivo do sinal com o seu patamar positivo. O resultado da comparação dos dois semi-ciclos obtidos é igual à soma com um dos semi-ciclos invertido.

A saída do comparador é então aplicada em um circuito integrador que produz um nível de tensão contínua igual ao valor rms da forma de onda de entrada. Este valor dc equivale proporcionalmente ao valor da Luminância do ponto sobre o objeto de superfícies difusoras observado segundo um eixo de visão paralelo ao eixo z Cartesiano.

5. AMOSTRAGEM E CONVERSÃO.

Apesar do sinal dc ser praticamente isento de ruído, podem ocorrer pequenas oscilações oriundas principalmente da fonte

Laser que não mantém constante a intensidade do feixe, e também das fontes de alimentação do Microcomputador e dos circuitos em geral.

Para melhorar-mos o desempenho das leituras digitais, um circuito amostrador-

retertor ("Sample &

Fig.12 - Retificação do sinal controlado pela referência de 90 Hz.

Hold") foi incluído mantendo constante o valor dc na entrada do conversor A/D durante o tempo de conversão. Seu funcionamento é bastante simples pois é fundamentado no chaveamento de um capacitor que retém sua carga durante um tempo determinado. Seu circuito pode ser visto no Apêndice A-1. Foi implementado junto à placa de interface digital dado que o mesmo é acionado pelo programa do Microcomputador através de uma operação de leitura no registrador interno desta interface de endereço 219 hexadecimal.(ver Tab.I)

Uma vantagem deste circuito é permitir a calibração dos níveis dirigidos à entrada do conversor A/D podendo então estabelecer códigos coerentes com a reflexão dos objetos. Por exemplo, o nível lógico zero é ajustado, através deste circuito, para representar o fundo contra o qual os objetos são vistos e o nível de fundo de escala do conversor é adotado para uma superfície difusora, perpendicular ao eixo de visão, mais próxima possível do observador.

O conversor A/D de 12 bits representa tres endereços internos da interface digital. Para que possamos ter um código em sua saída, equivalente à tensão em sua entrada, o programa no Microcomputador deve acessá-los da seguinte forma: (ver Tab.1)

- a) uma operação de escrita na porta 216 hexadecimal inicia o processo de conversão em 12 bits.
- b) o programa espera então que o dado esteja pronto, lendo a porta 218 hexadecimal (STS).
- c) ao se confirmar que o A/D tenha realizado a conversão, o código do dado é lido de endereços da interface. Os 8 MSB são lidos do endereço hexadecimal 216 e os 4 LSB do endereço hexadecimal 217.

O "arranjo de intensidades" referenciado no capítulo 2, é constituído então por um conjunto de códigos inteiros retirados pelo conversor A/D diretamente da parte refletida do feixe de Laser sobre o objeto. Cada valor dentro do arranjo é um código obtido do conversor A/D e sua posição dentro do arranjo, equivale a um código enviado ao conversor D/A para o eixo x e um código enviado ao conversor D/A para o eixo y.

6. CONCLUSÕES.

Apresentamos neste capítulo, o esquema dos circuitos de controle e aquisição de sinal para um sistema de reconhecimento do formato de objetos tri-dimensionais alternativo. Entendemos como sistema alternativo um método de Visão Computacional que não seja baseado exclusivamente em processamento de uma imagem obtida através de uma câmera de TV.

Sugerimos então, a utilização de um feixe de Laser defletido sobre o objeto da cena, cujas superfícies são Lambertianas e a aquisição da Luminância, ou seja, da parte refletida deste feixe através do uso de um foto-transistor. Este é colocado dentro de uma câmera escura e no foco de uma lente.

O alinhamento do observador e da fonte luminosa é facilmente implementado colocando-se um espelho de reduzido diâmetro formando um ângulo de 45° com o plano da lente que reflete o feixe incidente e não oferece obstáculo ao feixe refletido.

O objeto é girado de até 360° , em quatro quadrantes, para que o observador possa "ver" quatro vistas o que facilita a sua classificação e diferenciação de similaridade se analisado de um só ângulo.

Com a sugestão deste método entendemos que em grande parte das aplicações industriais de Visão Computacional, uma imagem obtida com uma câmera de TV possui em si informação excessiva. Esta é então expurgada através de processos de segmentação, detecção de bordas, esqueletização e outros que consomem tempo elevado a um razoável custo de processamento.

Nosso método, de baixo custo de implementação, pretende também contribuir para aplicações em tempo real que impliquem na separação de peças, ou caixas, em linhas de montagens automatizadas.

O reconhecimento do formato dos objetos é tarefa para o programa de categorização a ser analisado no capítulo 4, operando em um Microcomputador compatível com PC-XT e aplicando a teoria detalhada no capítulo 2.

7. COMENTÁRIOS BIBLIOGRÁFICOS.

Diversos trabalhos tem sido executados ao longo dos anos, em Visão de Máquina aplicados à área industrial. Para cada aplicação dedicada temos a utilização de diferentes tipos de sensores [Stuivinaga et all. 1989] [Sachs 1985].

Murray [Murray 1988] em seu trabalho afirma que "os olhos de um sistema de controle podem ser simples como sensores foto-elétricos, ou complexos como os sofisticados sistemas de visão."

O trabalho de Yoda [Yoda et all. 1973] utiliza diretamente uma câmera de TV e retira de uma imagem binária apenas a informação de posição do objeto em uma esteira industrial. Isto reduz o tempo de tomada de decisão, mas além do objeto não ser reconhecido, o sistema está super-dimensionado para a tarefa simples que realiza.

O processamento e a reconstrução da imagem é a tônica do trabalho de Jarvis [Jarvis 1977] na proposta de um laboratório de experiências em Visão de Robôs. O mesmo autor apresenta

depois [Jarvis 1983] um trabalho cuja preocupação visível é o que chamamos de um método alternativo. Um feixe de Laser é defletido sobre o objeto da cena mas seu processo de varredura forma novamente uma imagem para posterior processamento.

O trabalho que realmente podemos classificar como um método alternativo no reconhecimento de objetos tri-dimensionais é o apresentado por Ishii [Ishii & Nagata 1975]. Neste trabalho, a captação do feixe refletido do Laser é realizada por um dispositivo ótico bidimensional que por triangulação estabelece parâmetros iniciais. O sistema realiza então, o seguimento da borda dos objetos com o feixe de Laser, classificando-os sem a necessidade de digitalização da imagem.

A idéia de se utilizar de uma mesa giratória para que várias faces de um sólido possam ser observadas e conclusões tiradas para a sua classificação é usada por Connolly [Connolly & Stenstrom 1986]. Neste trabalho, os segmentos de linha são extraídos de cada imagem e transformados em um sistema de coordenadas na determinação do poliedro observado.

O emprego de medidores de distância ("range finder") é discutido por Rioux [Rioux 1984], Stuivinaga [Stuivinaga et al. 1989] e Monchaud [Monchaud 1989], como método de sensoriamento para Robôs em sistemas automatizados, operando em distâncias médias de 0 a 10 metros.

Concluindo, podemos dizer que nosso método de reconhecimento do formato de objetos tri-dimensionais utiliza a técnica dos medidores de distâncias, mas reconhece o sólido em observação sem perder tempo com a digitalização da imagem da cena.

CAPÍTULO 4

PROGRAMAS DE CONTROLE E ALGORÍTMO DE RECONHECIMENTO

1. INTRODUÇÃO.

O formato é a propriedade de um objeto que é invariante à translação, escala e rotação do mesmo. O formato é independente também do material que o constitui, da textura, da cor e das propriedades de reflexão de suas superfícies. [Tanimoto 1985] No entanto, estas propriedades podem ser utilizadas como elementos de determinação do formato tri-dimensional de um objeto. O formato de um quadrado azul, de matéria plástica e de 5 cm de lado é o de um quadrado.

O formato é um aspecto visual de grande importância para o reconhecimento de um objeto. O fato de seres humanos reconhecerem figuras em desenhos com somente algumas linhas de contorno, sugere o poder de informação contido na representação da forma de objetos. De uma maneira um pouco mais formal, podemos definir a característica do formato S de um objeto, por um conjunto de pontos no plano ou no espaço cuja função $S: P(E^n) \rightarrow R$ atribui um número real a cada conjunto de pontos no espaço Euclidiano n -dimensional.

Com $n = 2$, S é uma característica de formato bi-dimensional e com $n = 3$, S é uma característica de formato tri-dimensional.

Esta definição não traduz de maneira efetiva propriedades que classifiquem qualquer conjunto de pontos, mas estabelece o domínio no qual estas podem ser inferidas.

Existem em Visão Computacional várias propostas de utilização de características de formato [Ballard & Brown 1982], a maioria delas obtidas de esquemas bi-dimensionais representativos de estruturas tri-dimensionais. A tônica de nosso trabalho, no entanto, reside em captar a informação tri-dimensional inerente ao processo de digitalização da intensidade luminosa refletida, (luminância) através do circuito proposto no Capítulo 3. Limitando o universo do conjunto de pontos $S: P(E^3)$ a objetos triédricos, convexos e formados por superfícies Lambertianas, o algoritmo de classificação de um subconjunto destes é aqui descrito. Os programas de controle dos dispositivos eletrônicos de deflexão, rotação do objeto e aquisição do sinal, necessários para a interação com este algoritmo, são detalhados para permitir uma melhor compreensão do sistema proposto.

2. PROGRAMAS DE CONTROLE DOS DISPOSITIVOS ELETRÔNICOS.

A interface eletrônica digital esquematizada no Capítulo 3 e detalhada no Apêndice A1, é utilizada em um Microcomputador compatível com IBM PC-XT e controlada por programas desenvolvidos em linguagem C (mais especificamente Turbo C). Sua função característica pode ser subdividida em três atividades principais: a deflexão dos espelhos que fazem a varredura do

feixe de Laser, a rotação da mesa que suporta o objeto e a aquisição do sinal. Estes programas coordenados pelo algoritmo de reconhecimento, permitem respectivamente a determinação de um ponto sobre a superfície do objeto, a exposição de diferentes vistas ao observador e a montagem do arranjo de intensidades. A perfeita sincronia destas partes torna possível a diferenciação de um sólido triédrico de outro pertencente ao domínio do problema, na tarefa de reconhecimento do formato tri-dimensional.

2.1 - Exposição do objeto em múltiplas vistas.

Nenhum ser humano quando olha para um plano perpendicular ao seu eixo de visão, com a forma de um quadrado, pode afirmar com certeza se observa um objeto sólido, por exemplo um cubo, ou apenas uma superfície bi-dimensional. Se aplicarmos uma transformação real no objeto, como por exemplo uma translação dentro de determinados limites, podemos realizar variações em sua escala ou gerar diferenças de posição, sem obtermos, no entanto, nenhuma informação adicional sobre a forma definitiva do mesmo. Mas se rotacionarmos o mesmo objeto à frente do observador, outras vistas que permaneciam ocultas trarão subsídios suficientes para sua compreensão. Alguns trabalhos em Visão Computacional enfocam esta característica, girando o objeto para procurar a identificação com o modelo [Underwood & Coats Jr. 1975] ou transformando o modelo tal que a melhor vista se enquadre com a cena observada [Dhome & Kasvand 1987].

A mesa rotatória que suporta o objeto a ser reconhecido em nosso trabalho, pode ser girada em até 360° no sentido horário ou anti-horário. Um exemplo de um programa que posiciona-a no início do sistema de coordenadas é mostrado na Fig.1. Isto é necessário para que o algoritmo de reconhecimento atribua à primeira vista do objeto, um ângulo de visão igual a 0°, possibilitando a determinação de novos ângulos a partir deste.

```
void inic(void)
{
    int i ;
    unsigned char ch ;
    ch = inportb(0x219) ;
    ch = inportb(0x21A) ;
    ch = ch & 1 ;
    while( ch == 1 )
    {
        outportb(0x211,0x5F) ;
        outportb(0x211,0x4F) ;
        outportb(0x211,0x5F) ;
        for( i = 0 ; i < 1000 ; i++ ) ;
        ch = inportb(0x219) ;
        ch = inportb(0x21A) ;
        ch = ch & 1 ;
    }
}
```

A operação deste programa é extremamente simples e vinculada diretamente ao circuito eletrônico da interface digital

Fig.1 - Exemplo de programa de posicionamento da mesa giratória no início.

mostrada no Apêndice A1.

O endereço 219 hexadecimal desta, corresponde ao pino de gatilho (G) do circuito integrado 74LS373 da interface. Uma leitura nesta posição faz com que o sinal do sensor de início da mesa seja carregado no bit menos significativo do registrador de entrada cujo endereço é 21A hexadecimal. O programa envia então pulsos ao motor de passo, no sentido horário, monitorando continuamente o sensor, até que um valor lógico zero seja encontrado no bit menos significativo do endereço hexadecimal 21A.

Fig.2 - "Bytes" de controle para acionamento do motor de passo da mesa giratória.

A Fig.2 mostra a forma como os pulsos são gerados para controle do motor de passo da mesa giratória. Os quatro bits menos significativos do endereço 211 hexadecimal são

Fig.3 - Geração de pulso no motor de passo da mesa giratória.

enviados para os quatro bits mais significativos do conversor D/A do eixo x. Portanto, apesar de mostrado neste exemplo que o valor destes bits assumem nível lógico um, convém salientar que

os mesmos devem participar de uma operação lógica com o valor atual do conversor, durante a execução do algoritmo principal.

O bit D6 da palavra de controle permite a operação normal do motor de passo se for colocado em nível lógico um e desabilita as bobinas para evitar super aquecimento quando assumir valor zero. O bit D5 permite acionar o motor tanto no sentido horário quando seu valor for zero, como no sentido anti-horário quando seu valor for um. O bit D4 é o responsável pela geração do pulso no motor de passo, conforme mostra a Fig.3. Sua alternância entre os níveis lógicos um e zero permite a movimentação da mesa a uma velocidade determinada pelo "loop de for" mostrado no programa de controle da Fig.1. Cada pulso no motor gera um deslocamento angular de $7,5^\circ$ e corresponde a aproximadamente $0,6^\circ$ na mesa giratória.

Assim, quando o algoritmo de reconhecimento necessitar girar o objeto de um determinado ângulo, bastará acionar o motor de passo a uma velocidade constante e em um só sentido, de um número fixo de pulsos enviados à interface digital conforme a descrição acima. Várias vistas do objeto podem então ser analisadas de acordo com a necessidade do programa de reconhecimento.

2.2 - Programa de controle de varredura do feixe de Laser.

Para que possamos ter o feixe de Laser incidindo sobre a superfície do objeto e um controle da posição de cada ponto de incidência, um programa que aciona os galvanômetros de deflexão

horizontal e vertical é necessário. Conforme descrito no Capítulo 3 e detalhado no Apêndice A1, o circuito que comanda esta operação é realizado por dois conversores D/A de 12 bits. São denominados de DAX e DAY pois comandam a deflexão nos eixos x e y respectivamente, do "plano de imagem" do objeto em $z = -1$.

O campo de visão é determinado pelos limites de deflexão mínimo e máximo dos dois conversores. Cada varredura realizada pelo algoritmo de reconhecimento gera um "arranjo de intensidades" que possui como endereço de cada ponto os valores absolutos dos conversores DAX e DAY.

```
void main()
{
while (bioskey(1) == 0)
{
/* Início do sistema de coordenadas */
outportb(0x210,0) ;
outportb(0x211,8) ;
outportb(0x212,0) ;
outportb(0x213,8) ;
outportb(0x214,0) ;
/* Limite máximo no eixo x */
/* Máxima largura de campo */
outportb(0x210,0xFF) ;
outportb(0x211,0x0F) ;
outportb(0x212,0) ;
outportb(0x213,8) ;
outportb(0x214,0) ;
/* Limite máximo no eixo y */
/* Máxima altura de campo */
outportb(0x210,0) ;
outportb(0x211,8) ;
outportb(0x212,0xFF) ;
outportb(0x213,0x0F) ;
outportb(0x214,0) ;
}
}
```

O exemplo de programa mostrado na Fig.4 refere-se à deflexão entre os limites máximo e mínimo de cada conversor, definindo portanto o campo de visão. A posição hexadecimal 214 define o gatilho dos

Fig.4 - Exemplo de programa de controle dos conversores D/A para deflexão do feixe de Laser.

conversores.

Uma operação de escrita nesta porta carrega simultaneamente os 12 bits do conversor D/A do eixo x e os 12 bits do conversor D/A do eixo y. Isto faz com que o feixe de Laser não seja dirigido a nenhum estado intermediário de deflexão formado pelo estado anterior de um dos conversores com o estado atual do outro.

Os quatro bits mais significativos do conversor DAX são supridos pelos quatro bits menos significativos da porta 211 hexadecimal, cujos bits mais significativos controlam o motor de passo da mesa giratória. Por esta razão, estes devem ser atualizados através de uma operação lógica com a palavra de controle do motor de passo, conforme descrito na secção anterior.

Se observarmos novamente o arranjo de intensidades A_{x_1, y_j} formado por uma varredura do feixe de Laser mostrado pelas equações Eq.27 e Eq.28 do Capítulo 2, veremos que os índices x_1 e y_j são respectivamente os valores absolutos dos conversores DAX e DAY responsáveis pela deflexão do feixe de Laser.

Portanto, a intensidade luminosa refletida (luminância no ponto) definida por I_{x_1, y_j} possui como coordenadas os valores dos conversores DAX e DAY em cada elemento do arranjo.

2.3 - Digitalização do sinal.

O conversor A/D é o dispositivo eletrônico responsável pela transformação do sinal contínuo correspondente à intensidade luminosa refletida sobre o objeto, em um número digital. Este

dispositivo foi esquematizado no Capítulo 3 e detalhado no circuito da interface digital apresentado no Apêndice A1.

Para que o conversor possa operar convenientemente e gerar cada valor $I_{x1,y1}$ do arranjo de intensidades, um programa de controle é necessário tanto a nível de operação do conversor como na redução de ruídos do sinal. Na entrada do conversor A/D um circuito de amostragem e retenção é utilizado para que pequenas variações no sinal não interfiram na geração do código digital durante uma conversão. O diagrama de tempos da Fig.5 apresenta a operação do conjunto formado pelo conversor A/D e

Fig.5 - Diagrama de tempos para digitalização do sinal.

circuito amostrador, cujo programa básico de controle é exemplificado na Fig.6. Uma operação de leitura realizada na

porta 219 hexadecimal gera um pulso de amostragem no circuito amostrador o que faz com que o capacitor de retenção carregue-se com o valor da tensão a ser convertida. A autorização para início de conversão é fornecida através de uma operação de escrita na porta 216 hexadecimal.

```
void main()
{
    int ch;
    unsigned char cc,cb;
    for(;;)
    {
        ch=getch();
        if (ch=='a') break;
        cc=inportb(0x219); /* Amostragem */
        outportb(0x216,0); /* Início */
        cc=inportb(0x218); /* Dado pronto? */
        cb=cc & 01;
        while (cb==01) /* Espera fim de */
        { /* conversão */
            cc=inportb(0x218);
            cb=cc & 01;
        }
        cc=inportb(0x216); /* Lê 8 MSB */
        cb=inportb(0x217); /* Lê 4 LSB */
    }
}
```

O programa fica então à espera do fim de conversão lendo o bit menos significativo da porta 218 hexadecimal.

Depois de um tempo aproximado de

Fig.6 - Exemplo de programa para controle de operação do conversor A/D.

35µs (tc = tempo de

conversão) para o conversor AD574 utilizado, o dado está pronto. O programa acessa a porta 216 hexadecimal e realiza a leitura dos 8 bits mais significativos do conversor A/D e em seguida a porta 217 hexadecimal fornece os quatro bits menos significativos formantes do código de 12 bits.

A implementação prática, no entanto, mostrou que flutuações na intensidade da fonte Laser e interferências ambientais ocasionam ruídos no sinal captado. Uma solução utilizada foi a de reduzirmos o ruído aleatório através da média de leituras sucessivas em um mesmo ponto. Para que o tempo de conversão

total não fosse substancialmente prejudicado, desenvolvemos um programa em linguagem Assembler que tira a média de 16 (dezesseis) leituras consecutivas de um mesmo ponto. Este programa encontra-se no Apêndice B1. O ruído é assim minimizado permitindo leituras um pouco mais confiáveis.

Uma outra solução empregada envolve a estatística dos valores digitalizados em um mesmo ponto, substituindo aqueles que ultrapassarem um desvio médio estabelecido. O programa que realiza esta operação é mostrado no Apêndice B2. A utilização destes recursos melhoram sensivelmente o resultado da digitalização do sinal refletido, mas o compromisso com a velocidade total de reconhecimento deve ser levado em conta. Desse modo, se o domínio de objetos classificáveis pelo algoritmo é tal que poucas varreduras sejam necessárias, a média pode ser aumentada e o desvio máximo reduzido. Caso objetos de maior complexidade exijam um maior número de varreduras para sua identificação completa, a velocidade de digitalização deve ser aumentada em detrimento de uma melhor qualidade do sinal digitalizado.

Para o algoritmo de reconhecimento que aqui desenvolvemos, o sinal é digitalizado 16 (dezesseis) vezes em um mesmo ponto por um programa em Assembler e seu valor médio calculado. Um programa em Turbo C recalcula uma nova média utilizando 32 (trinta e duas) destas médias em Assembler e verifica se o desvio de cada uma ultrapassa 50 (cinquenta) códigos digitais do conversor. Em caso positivo, a leitura que ultrapassar é substituída por outra e a média recalculada. O número de iterações não ultrapassa 128 (cento e vinte e oito).

Como os objetos que avaliamos são de construção regular e simples, o número de varreduras necessárias é o mínimo possível. O tempo consumido com a purificação do sinal digitalizado não compromete, portanto, de maneira contundente a velocidade de reconhecimento global.

3. ALGORÍTMO DE RECONHECIMENTO DE OBJETOS.

Tradicionalmente os sistemas de Visão Computacional são subdivididos em duas fases seqüenciais: - a visão de baixo nível e a visão de alto nível [Rao & Jain 1988]. O processo de visão em baixo nível refere-se à extração de características diretamente ligadas ao sistema de sensoreamento, descrevendo-as de forma mais abstratas. A transformação dos códigos digitais do arranjo de intensidades em rótulos representa este processo. A visão em alto nível corresponde à descrição do objeto através dos recursos obtidos pela visão em baixo nível e utilizando-se de conhecimento anterior previamente armazenado sobre o domínio da cena. Conseqüentemente, os sistemas de Visão Computacional possuem uma estreita relação com a área de Inteligência Artificial denominada de Sistemas Baseados em Conhecimento [Rich 1988].

O programa do Microcomputador é então alimentado com informação oriunda do sensor de imagem (foto-transistor) e deve usar o conhecimento sobre os objetos armazenados para tomar a decisão de aceitá-lo ou não como um elemento pertencente ao universo interpretável.

3.1 - Restrição do domínio dos objetos reconhecíveis.

O reconhecimento de objetos tri-dimensionais e a determinação de suas posições e orientações no espaço são os problemas principais dos sistemas de Visão de Robôs.

A maioria dos métodos existentes empregam alguma forma de detecção de bordas através de segmentação de regiões [Haralick & Shapiro 1985] sendo que o reconhecimento é baseado na comparação destas bordas detectadas com as fornecidas pelo modelo tri-dimensional [Besl & Jain 1985].

Bordas (arestas), vértices e fronteiras são características significantes na identificação de um objeto no espaço 3D.

Considerando o conjunto dos poliedros convexos e suas generalizações conforme mostra a Fig.7, retiramos os quatro mais simples para comprovação do método aqui proposto. Poliedro convexo é todo aquele que se tomarmos o plano que contenha uma de suas faces, todo o objeto estará contido em apenas um dos dois semi-espacos dividido pelo plano. Estes quatro objetos são então mostrados na Fig.8 com as diferentes variações posicionais em relação ao eixo de visão, supondo-se que os mesmos possam ficar apoiados em uma de suas faces sobre a mesa giratória. A nomenclatura adotada será utilizada pelo algoritmo para facilidade de referência. Assim teremos:

- objeto 1P = prisma de base triangular assentado sobre sua base.
- objeto 1PD = prisma de base triangular assentado sobre uma de suas faces.
- objeto 2P = prisma de base retangular (cubo).

Fig.7 - Conjunto dos poliedros convexos.

Fig.8 - Conjunto de poliedros e suas posições reconhecidas pelo algoritmo de reconhecimento.

- objeto 1C = pirâmide de base triangular.
- objeto 2C = pirâmide de base retangular assentada sobre sua base.
- objeto 2CD = pirâmide de base retangular assentada sobre uma de suas faces.

O domínio dos objetos reconhecíveis é então modelado na base de conhecimento do algoritmo de reconhecimento, considerando os seis modelos da Fig.8 em todas as possíveis vistas quando estes são girados em torno do eixo y Cartesiano.

3.2 - Extração das características dos objetos.

Dentre todas as características que podem ser utilizadas para o reconhecimento de um objeto poliédrico, algumas são imediatas e tem sido usadas exaustivamente pelos métodos tradicionais. São elas as arestas e os vértices que devidamente rotulados fornecem a descrição necessária à sua classificação conforme mostrado no Capítulo 2.

O arranjo de intensidades obtido pela varredura do feixe de Laser contém a informação necessária para a extração destas características. Seja, então, o arranjo obtido por uma varredura horizontal para $y = y_1$:

$$A_{x_1, y_1} = [I_{x_1, y_1}, I_{x_2, y_1}, \dots, I_{x_n, y_1}]$$

A primeira derivada deste arranjo em relação a x fornece, como já visto, as inclinações das superfícies em relação ao plano (x,y), ou seja:

$$p = - \left(\frac{I_{x_i, y_1} - I_{x_{i-1}, y_1}}{x_i - x_{i-1}} \right)$$

Isto significa que se calcular-mos a inclinação da superfície em cada ponto (x_1, y_1) obtemos o arranjo de inclinações, escrito da seguinte forma:

$$p_{x_1, y_1} = [p_{x_1, y_1}, p_{x_2, y_1}, \dots, p_{x_n, y_1}]$$

A derivada deste arranjo de inclinações, ou seja, a segunda derivada do arranjo de intensidades em relação a x , fornece a informação da existência ou não de arestas nesta varredura e sua eventual localização. Dessa maneira, temos:

$$\frac{\partial p}{\partial x} = \frac{p_{x_i, y_1} - p_{x_{i-1}, y_1}}{x_i - x_{i-1}}$$

A Fig.9 mostra o conceito geométrico empregado. O arranjo de intensidades transporta a informação sobre o perfil do objeto durante a varredura do feixe de Laser. A primeira derivada deste arranjo gera o arranjo de inclinações definindo o comportamento das inclinações das superfícies. A segunda derivada é responsável pela geração do arranjo de arestas. Isto é interpretado pelo algoritmo de reconhecimento, levando-se em conta as restrições estabelecidas, da seguinte forma:

- o fundo contra o qual o objeto é visto é calibrado para apresentar valor zero no arranjo de intensidades, logo, a primeira e a segunda derivadas são também zero.
- A varredura horizontal é realizada sempre da esquerda para a direita.
- O primeiro valor não zero encontrado ou no arranjo de

intensidades,

ou no arranjo

de inclinações

ou no arranjo

de arestas,

define a

posição da

fronteira do

objeto à

esquerda

(x_e, y_1).

Retrocedendo-se

do último

elemento do

arranjo, no

sentido da

direita para a

esquerda,

determina-se a

posição da

fronteira

direita do

objeto (x_a, y_1),

através da

localização do primeiro elemento não zero. As fronteiras são então rotuladas (--->).

- se entre as duas arestas representativas da fronteira do objeto o valor dos elementos do arranjo de arestas for zero (ou

Fig.9 - Determinação das arestas e das inclinações das superfícies.

próximo de zero) não existe aresta interna e o arranjo de inclinações deve ser analisado.

- caso um valor diferente de zero e de módulo elevado seja encontrado no arranjo de arestas, o local no arranjo onde ele se encontra (x_a, y_1) representa uma aresta interna detectada pela varredura horizontal. A aresta é então rotulada como convexa(+).
- se o valor dos elementos do arranjo de inclinações for constante, dentro de uma certa tolerância, entre as fronteiras do objeto, então não existe aresta interna. A varredura detectou um plano inclinado e sua inclinação em relação ao plano (x,y) é dada pelo sinal positivo ou negativo dos elementos do arranjo ($p=+$ ou $p=-$).

Resumindo, o arranjo de intensidades obtido com a primeira varredura horizontal é utilizado para extrair as seguintes características do objeto:

- fronteira do objeto: posição dentro do arranjo onde começa e termina o objeto em observação.
- aresta interna: posição dentro do arranjo onde duas faces mudam de inclinação.
- inclinação da superfície: quando não existe aresta interna na vista do objeto em observação.

3.3 - Suavização dos dados de entrada.

Apesar de todos os recursos utilizados para redução de ruído na aquisição do sinal, o arranjo de intensidades vem associado a ruídos muito difíceis de serem eliminados na entrada

dos dados, causando variações indesejáveis. Para que uma função unidimensional como o arranjo de intensidades, ou bidimensional como uma imagem possa aproximar suas características originais, uma filtragem é necessária. Esta filtragem realiza uma suavização da função e é conseguida através da convolução entre a entrada e um pulso retangular [Castleman 1979]. Este pulso move-se da esquerda para a direita, sobre a função, produzindo um resultado que é em cada ponto, uma média local da função de entrada.

Podemos então, suavizar uma imagem ou uma função unidimensional através da substituição de seu valor em cada ponto pela média de valores em torno da vizinhança do ponto [Rosenfeld 1969]. A média local tem o efeito de suprimir as variações de alta frequência enquanto preserva o formato básico da função de entrada.

Se tomarmos o arranjo de intensidade e o convoluirmos com um pulso de largura de tres pontos vizinhos podemos escrever a equação da média local obtida, como:

$$\tilde{f}_{x_i, y_j} = \frac{1}{3} \cdot \sum_{k=i-1}^{i+1} I_{x_k, y_j}$$

p/ $i=2, \dots, n-1$

e, ainda,

$$\begin{aligned} f_{x_1, y_j} &= \tilde{f}_{x_2, y_j} \\ f_{x_n, y_j} &= \tilde{f}_{x_{n-1}, y_j} \end{aligned}$$

O novo arranjo de intensidade suavizado é formado então, pelos pontos $\tilde{x}_{i,y}$ sendo que o primeiro elemento é estabelecido igual ao segundo e o último igual ao penúltimo. Isto porque a vizinhança é escolhida à esquerda e à direita do ponto para o cálculo da média, e dessa forma o primeiro e último dos elementos ficam prejudicados. Como a tentativa é eliminar saltos na continuidade da função de entrada, a repetição dos valores médios calculados próximos aos pontos terminais contribui nesse sentido.

3.4 - Classificação da fronteira e diferenciação das arestas.

Conhecida a fronteira do objeto, através da primeira varredura horizontal, o programa calcula o meio do objeto no eixo x, em x_t

$$x_t = \frac{x_d + x_e}{2}$$

e executa a primeira varredura vertical conforme mostra a Fig.10. Esta detecta então a altura do objeto quando os valores do arranjo de intensidade atingirem zero, isto é, correspondendo ao fundo da cena.

A regra de coerência estabelecida pelos métodos de rotulagem, diz que uma aresta não pode alterar seu rótulo entre dois vértices. Baseado nisso, o algoritmo de reconhecimento realiza uma segunda varredura horizontal a uma altura y_v (a

aproximadamente 10 pontos abaixo da fronteira superior do objeto). Esta varredura fornece então subsídios para a classificação do tipo de fronteira e para diferenciação da aresta detectada. A Fig.11 mostra os parâmetros utilizados para esta tarefa. Definimos então, para a varredura

Fig.10 - Varreduras para extração das características.

Fig.11 - Parâmetros utilizados para diferenciação das arestas.

horizontal realizada em y_1 :

$$X_1 = X_a - X_e$$

$$X_2 = X_d - X_a$$

$$X_{h1} = X_d - X_e$$

e para a varredura horizontal realizada em y_v :

$$X_3 = X_a - X_e$$

$$X_4 = X_d - X_a$$

$$X_{h2} = X_d - X_e$$

Com estes parâmetros calculados, o programa classifica o objeto de acordo com sua fronteira em:

- a) Objeto de fronteira paralela (FP): caso x_{h1} seja aproximadamente igual a x_{h2} .
- b) Objeto de fronteira convergente (FC): caso x_{h2} seja menor (ou bem menor que) x_{h1} .

Esta classificação reduz o domínio dos objetos analisados e ao mesmo tempo rotula suas fronteiras.

Fig.12 - Tipos de arestas possíveis de ocorrer considerando os objetos pertencentes ao domínio.

Observando a Fig.12 podemos ver que as arestas podem ser diferenciadas em sete tipos possíveis de ocorrências dado o universo de objetos classificáveis pelo algoritmo de reconhecimento. Assim, podemos escrever que:

- se a fronteira do objeto é convergente (FC) e,
 - se $x_1 = x_3$ e $x_2 <> x_4$ então a aresta é do tipo a_{1c} .
 - se $x_2 = x_4$ e $x_1 <> x_3$ então a aresta é do tipo a_{2c} .
 - se $x_1 > x_3$ e $x_2 <> x_4$ então a aresta é do tipo a_{4c} .
 - se $x_1 < x_3$ e $x_2 <> x_4$ então a aresta é do tipo a_{5c} .
- se a fronteira do objeto é paralela (FP) e,
 - se $x_1 = x_3$ e $x_2 = x_4$ então a aresta é do tipo a_{3p} .
 - se $x_1 > x_3$ e $x_2 <> x_4$ então a aresta é do tipo a_{4p} .
 - se $x_1 < x_3$ e $x_2 <> x_4$ então a aresta é do tipo a_{5p} .

Dessa maneira, considerando-se os objetos definidos pela Fig.8 e todas as suas possíveis posições em relação ao eixo de visão; considerando também o objeto isolado na cena e sempre assentado sobre uma de suas faces: as características definidas acima são suficientes para a tarefa de reconhecimento.

3.5 - Base de conhecimento. Descrição dos objetos do domínio.

Um tema constante em pesquisas de Visão de Máquina tem sido a necessidade de uma representação adequada do conhecimento visual e a utilização de uma estratégia de controle que otimize a tarefa de percepção [Havens & Mackworth 1983].

O que deve disparar o processo de reconhecimento: uma entrada de baixo nível tal como uma aresta, ou a expectativa de se ter um objeto, como por exemplo um cubo? Os dois extremos tem sido utilizado em Computação Visual e quase sempre empregados conjuntamente [Ballard & Brown 1982]. Quando a estratégia de controle caminha da imagem para sua estrutura segmentada e depois para sua descrição, temos o chamado controle "bottom-up", ou dirigido a dados. Quando modelos de alto nível presentes na base de conhecimento geram a expectativa de previsão e o sistema faz a verificação destas previsões, o processo é conhecido como controle "top-down", ou dirigido a modelo [Barry et al 1988].

3.5.1 - Estratégia de controle "bottom-up".

Este processo está ligado diretamente à visão de baixo nível e pode ser entendido através da execução de três passos:

- 1) - Pré-processamento: controle do sistema e transformação dos dados binários oriundos do conversor A/D em um arranjo de intensidades. Esta fase é independente do domínio.
- 2) - Processamento (segmentação): determinação do arranjo de inclinações e do arranjo de arestas com as respectivas rotulagens.
- 3) - Compreensão: relacionamento dos dados obtidos com o domínio dos objetos reconhecíveis e determinação de classes.

Este procedimento cria através de dados binários, descrições de mais alto nível que serão empregadas no reconhecimento do objeto visualizado. A árvore hierárquica de classificação mostrada na Fig.13 estabelece os parâmetros para esta descrição. Assim, para cada vista do objeto o algoritmo de reconhecimento atravessa a árvore para gerar um nó representativo. Os objetos do domínio conforme apresentados na Fig.8 são classificáveis em cada um dos nós terminais da árvore conforme a possibilidade de ocorrência em qualquer uma de suas vistas, como a seguir:

- nó 1 - objetos 1P, 1PD, 2P e 2CD , apresentam fronteira paralela (FP), e a superfície localizada entre os limites da fronteira não possui arestas mas sim inclinação igual a zero em relação ao eixo de visão, em alguma de suas possíveis vistas.
- nó 2 - objetos 1P e 2CD, apresentam fronteira paralela (FP), e a superfície localizada entre os limites da fronteira não possui

Fig. 13 - Árvore de classificação dos objetos do domínio.

arestas mas sim inclinação positiva em relação ao eixo de visão, em alguma de suas possíveis vistas.

- nó 3 - objetos 1P, 2P e 2CD, apresentam fronteira paralela (FP) e uma aresta entre os limites da fronteira. Esta aresta é classificável de acordo com a descrição feita no item anterior e apresentada na Fig.12. Este nó abre-se então, em outros conforme mostra a Fig.14.

- nó 4 - objetos 1P e 2CD, apresentam fronteira paralela (FP), e a superfície localizada entre os limites da fronteira não possui arestas mas sim inclinação negativa em relação ao eixo de visão,

em alguma de suas possíveis vistas.

- nó 5 - objetos 1C, 2C, 1PD e 2CD, apresentam fronteira convergente (FC), e a superfície localizada entre os limites da fronteira não possui arestas mas sim inclinação igual a zero em relação ao eixo de visão, em alguma de suas possíveis vistas.

- nó 6 - objetos 1C, 2C e 2CD, apresentam fronteira convergente (FC), e a superfície localizada entre os limites da fronteira não possui arestas mas sim inclinação positiva em relação ao eixo de visão, em alguma de suas possíveis vistas.

- nó 7 - objetos 1C, 2C, 1PD e 2CD, apresentam fronteira convergente (FC) e uma aresta entre os limites da fronteira. Esta aresta é classificável de acordo com a descrição feita no item anterior e apresentada na Fig.12. Este nó abre-se então, em outros conforme mostra a Fig.14.

- nó 8 - objetos 1C, 2C e 2CD, apresentam fronteira convergente (FC), e a superfície localizada entre os limites da fronteira não possui arestas mas sim inclinação negativa em relação ao eixo de visão, em alguma de suas possíveis vistas.

Dessa forma, consideramos que um objeto possa ser descrito através de quatro vistas, giradas entre si de 90°. Cada vez que o feixe de Laser produz os arranjos de intensidades através das duas varreduras horizontais e da varredura vertical, o processo "bottom-up" é empregado para determinar um elemento do vetor de descrição do objeto. Cada elemento deste vetor corresponde a uma busca na árvore com o objeto girado de 90° em relação ao ponto anterior.

Fig. 14 - Definição das arestas na árvore de classificação.

3.5.2 - Estratégia de controle "top-down".

A técnica mais comum de controle dirigido a modelo ("top-down") é a de hipótese-e-verificação ou também, gerar-e-testar [Rich 1988]. Se a geração de soluções possíveis for feita sistematicamente, então o procedimento poderá encontrar uma solução, se ela existir.

Para formulação da base de conhecimento que descreve os objetos do domínio, consideramos os nós terminais da árvore das figuras Fig.13 e Fig.14 como geradora dos elementos dos "vetores de descrição". O estabelecimento destes vetores segue um processo completamente heurístico levando-se em conta o tipo de objeto que o sistema é capaz de reconhecer. Cada "vetor de descrição" em particular é então constituído de quatro elementos sendo respectivamente equivalentes ao nó terminal da árvore em cada busca para a vista do objeto em 0°, 90°, 180° e 270°. O

domínio dos objetos reconhecíveis é então gerado formando a base de conhecimento a ser testada conforme o nó visitado na árvore.

Os objetos são então descritos na base através dos seguintes "vetores de descrição":

- objeto 1P = (1, 4, a_{3p}, 2) ou
(4, a_{3p}, 2, 1) ou
(a_{3p}, 2, 1, 4) ou
(2, 1, 4, a_{3p})
- objeto 2P = (1, 1, 1, 1) ou
(a_{3p}, a_{3p}, a_{3p}, a_{3p})
- objeto 1PD = (1, 5, 1, 5) ou
(5, 1, 5, 1) ou
(a_{1c}, a_{2c}, a_{1c}, a_{2c}) ou
(a_{2c}, a_{1c}, a_{2c}, a_{1c})
- objeto 1C = (5, 8, 7, 6) ou
(8, 7, 6, 5) ou
(7, 6, 5, 8) ou
(6, 5, 8, 7)
- objeto 2C = (5, 5, 5, 5) ou
(a_{4c}, a_{4c}, a_{4c}, a_{4c})

- objeto 2CD = (1, 8, a_{4p}, 6) ou
(8, a_{4p}, 6, 1) ou
(a_{4p}, 6, 1, 8) ou
(6, 1, 8, a_{4p}) ou
(2, 5, a_{5p}, a_{2c}) ou
(5, a_{5p}, a_{2c}, 2) ou
(a_{5p}, a_{2c}, 2, 5) ou
(a_{2c}, 2, 5, a_{5p}) ou
(4, a_{1c}, a_{4p}, 5) ou
(a_{1c}, a_{4p}, 5, 4) ou
(a_{4p}, 5, 4, a_{1c}) ou
(5, 4, a_{1c}, a_{4p}) ou
(a_{1c}, a_{4c}, a_{5c}, a_{2c}) ou
(a_{4c}, a_{5c}, a_{2c}, a_{1c}) ou
(a_{5c}, a_{2c}, a_{1c}, a_{4c}) ou
(a_{2c}, a_{1c}, a_{4c}, a_{5c}) ou
(a_{2c}, 2, a_{4c}, a_{5p}) ou
(2, a_{4c}, a_{5p}, a_{2c}) ou
(a_{4c}, a_{5p}, a_{2c}, 2) ou
(a_{5p}, a_{2c}, 2, a_{4c}) ou
(8, a_{4p}, 5, 4) ou
(a_{4p}, 5, 4, 8) ou
(5, 4, 8, a_{4p}) ou
(4, 8, a_{4p}, 5)

Como pode ser visto, quanto mais regular é o objeto mais simples é sua descrição. Tomemos como exemplo o objeto 2P que é um cubo. Se a vista sob ângulo 0° da mesa giratória, isto é, o

primeiro elemento do "vetor de descrição" for 1 (equivalente ao nó 1 da árvore) significa que uma face de fronteira paralela e perpendicular ao eixo de visão está sendo observada. Como suas faces possuem ângulo reto entre elas, uma rotação na mesa de um ângulo de 90° fará com que o próximo elemento do vetor seja novamente 1, ou seja, uma nova face de inclinação zero será vista, e assim sucessivamente para os quatro elementos do vetor.

Da mesma forma, se para o primeiro elemento do vetor foi detectado uma aresta entre os limites de uma fronteira paralela (aresta do tipo asp , ver Fig.12) o raciocínio heurístico nos leva a concluir que nas próximas três vistas com giro de 90° serão também detectadas arestas do mesmo tipo.

Evidentemente a base de conhecimento pode ser ampliada para cada objeto levando-se em conta outras possibilidades de visualização, gerando-se portanto novos "vetores de descrição".

Quando o objeto possui várias possibilidades de posição sobre a mesa giratória e conseqüentemente diversas vistas podem representá-lo (exemplo da pirâmide de base quadrada apoiada sobre uma de suas faces - objeto 2CD) o número de "vetores" que o descrevem é elevado.

Uma otimização que pode ser realizada na base de conhecimento é através da observação dos dois primeiros elementos de cada "vetor de descrição". Para os objetos do domínio definido, não há repetição de valores, exceto para uma vista do objeto 1PD e 2CD. Isto nos leva a concluir que é possível reduzir a base de conhecimento e conseqüentemente o número de testes necessários ao controle "top-down" de 4 para 2 em cada "vetor". Ou seja, cada objeto pode ser representado por

apenas duas de suas vistas giradas de 90° entre si. Assim, o banco de conhecimento reduzido pode ser gerado da seguinte maneira:

- objeto 1P = (1,4) ou (4,a_{3p}) ou (a_{3p},2) ou (2,1)
- objeto 2P = (1,1) ou (a_{3p},a_{3p})
- objeto 1PD = (1,5) ou (a_{1c},a_{2c}) ou (a_{2c},a_{1c})* ou (5,1)
- objeto 1C = (5,8) ou (8,7) ou (7,6) ou (6,5)
- objeto 2C = (5,5) ou (a_{4c},a_{4c})
- objeto 2CD = (1,8) ou (8,a_{4p}) ou (a_{4p},6) ou (6,1) ou
(2,5) ou (5,a_{5p}) ou (a_{5p},a_{2c}) ou (a_{2c},2) ou
(4,a_{1c}) ou (a_{1c},a_{4p}) ou (a_{4p},5) ou (5,4) ou
(a_{1c},a_{4c}) ou (a_{4c},a_{5c}) ou (a_{5c},a_{2c}) ou (a_{2c},a_{1c})*
ou
(a_{2c},2) ou (2,a_{4c}) ou (a_{4c},a_{5p}) ou (a_{5p},2) ou
(8,a_{4p}) ou (a_{4p},5) ou (5,4) ou (4,8)

Com isso cada objeto só necessita ser girado uma vez: de 0° para 90°, fornecendo duas vistas para interpretação. Os vetores marcados com (*) na base de conhecimento mostrada acima possuem os mesmos elementos, não permitindo portanto uma diferenciação entre os dois objetos (1PD e 2CD) com apenas duas vistas. Caso

isto ocorra, o programa de reconhecimento avança mais 90° na mesa giratória obtendo assim mais um elemento do "vetor" permitindo a caracterização completa do objeto.

3.6 - Programa de reconhecimento.

O procedimento global para o reconhecimento dos objetos utilizando a tecnologia proposta é descrito pelo diagrama em blocos da Fig.15 e desenvolvido em linguagem C conforme a listagem apresentada no Apêndice B-3. Resumidamente podemos descrevê-lo através dos seguintes passos:

- 1) Parâmetros iniciais: parte do programa que coloca a mesa giratória em sua posição de referência e os conversores D/A dos espelhos (x,y) no início do sistema de coordenadas.
- 2) Classificação das fronteiras do objeto: parte do programa que; realiza a varredura horizontal em y_1 da esquerda para a direita; determina a fronteira esquerda e direita do objeto; calcula a metade entre as duas fronteiras (x_t); realiza uma varredura vertical em x_t ; determina a altura do objeto; realiza a segunda varredura horizontal em y_v ; determina a fronteira esquerda e direita do objeto; aplica a regra de coerência e rotula as fronteiras.
- 3) Determinação das inclinações ou arestas: parte do programa que calcula o arranjo de inclinações e o arranjo de arestas

determinando os parâmetros necessários para busca na árvore de montagem do vetor de descrição do objeto.

4) Montagem do vetor de descrição: parte do programa que para cada vista do objeto determina um elemento do vetor de descrição de acordo com a árvore que representa os parâmetros de classificação.

5) Rotação da mesa de 90°: Parte do programa que altera a vista do objeto para que novo elemento do vetor de descrição possa ser determinado. Enquanto o número de vistas não for suficiente para o reconhecimento do objeto, o programa deve voltar ao passo 2.

6) Comparar vetor de descrição com a base de conhecimento:

parte do programa que realiza o controle "top-down" onde o vetor de descrição montado pelo controle "bottom-up" é comparado com a

Fig. 15 - Diagrama em blocos do programa de reconhecimento.

base de conhecimento programada.

7) Reconhecimento do objeto: parte final do programa de reconhecimento onde se o objeto faz parte do domínio reconhecível uma saída é providenciada (por exemplo, um desenho do objeto em sua posição detectada é colocado no vídeo). Caso contrário uma mensagem de erro ocorre permitindo ao sistema uma nova tentativa.

4. CONCLUSÕES.

Apresentamos neste capítulo um algoritmo para o reconhecimento do formato de objetos tri-dimensionais utilizando o sistema proposto no Capítulo 3. Os objetos pertencentes ao domínio do problema são supostos poliedros regulares e deste universo são retirados quatro sólidos que apoiados sobre a base ou sobre uma de suas faces fornecem um conjunto de seis possibilidades a serem analisadas.

Cada um desses objetos quando colocados sobre a mesa giratória é varrido pelo feixe de Laser horizontalmente, uma vez, da esquerda para a direita próximo à sua base; outra vez próximo ao topo. Seu perfil, captado através da Luminância em cada ponto da varredura, constitui um arranjo de intensidades que analisado fornece subsídios para o reconhecimento do objeto. A primeira derivada deste arranjo fornece as inclinações das superfícies e a segunda derivada fornece o local da aresta interna, caso houver. A fronteira determina o primeiro parâmetro

de classificação permitindo a busca em uma árvore, através de uma estratégia de controle "bottom-up", de elementos para a construção de um vetor de descrição do objeto.

Tanto a fronteira como as arestas internas detectadas são rotuladas pelo método de Huffman-Clowes e admitida a regra de coerência onde uma aresta não pode alterar seu rótulo entre dois vértices, apenas duas varreduras horizontais são realizadas.

Uma base de conhecimento é armazenada constituída de vetores de descrição para todos os objetos do domínio em suas diversas possibilidades de apresentação ao observador. O programa através de um processo "top-down" compara o vetor de descrição obtido com os da base de conhecimento e reconhece o objeto.

É interessante salientar que outros algoritmos podem ser implementados com a mesma tecnologia. Um exemplo seria o de descrever os objetos através de seus vértices rotulados pelo método de Huffman-Clowes. O cálculo dos vértices, no entanto, exigiria outras transformações a partir das arestas rotuladas. Para objetos simples como os poliedros regulares, nosso método economiza tempo e custo de processamento pois admitida esta hipótese não faz inferências desnecessárias e nem coleta excesso de informação para realizar a tarefa de reconhecimento.

6. COMENTÁRIOS BIBLIOGRÁFICOS.

A bibliografia utilizada neste capítulo pode ser avaliada em segmentos. As que tratam de técnicas de Inteligência

Artificial fornecem os recursos para a implementação do algoritmo de reconhecimento [Rich 1988] [Havens & Mackworth 1983] [Rao & Jain 1988] [Tanimoto 1985]. Estas técnicas são de grande importância em Visão Computacional pois a percepção humana simulada por algoritmos estabelece parâmetros valiosos para a implementação de máquinas mais versáteis.

O livro de Ballard [Ballard & Brown 1982] discute os métodos de Visão de Máquina explorando o assunto desde a formação da imagem até a utilização dos recursos de IA. É portanto uma das mais completas obras no setor.

As técnicas de processamento de imagens são apresentadas por Castleman [Castleman 1979] e por Rosenfeld [Rosenfeld 1969] de onde retiramos as informações sobre os métodos de suavização de funções. Dois trabalhos utilizam o recurso de interpretação de múltiplas vistas de objetos. Dhome [Dhome & Kasvand 1987] aplica transformações em seus modelos tais como rotação, escala e translação, até que as vistas reais dos objetos da cena interpretada coincidam ou estejam próximas deles. Underwood [Underwood & Coats Jr. 1975] desenvolve um trabalho de reconhecimento dos sólidos através de várias imagens de diferentes vistas quando o objeto é girado no espaço.

Dois artigos de interesse geral apresentam um levantamento sobre diversas técnicas em segmentação de imagens [Haralick & Shapiro 1985] e sobre reconhecimento de objetos tri-dimensionais [Besl & Jain 1985] este último citando inclusive a utilização de Laser em Visão Computacional.

CAPÍTULO 5

RESULTADOS E CONCLUSÕES.

1. INTRODUÇÃO.

A Tecnologia para o Reconhecimento de Objetos Tri-dimensionais proposta neste trabalho apresentou resultados satisfatórios referentes à sua meta básica.

Os métodos tradicionais de Visão Computacional ocupam-se primordialmente com o processamento de imagens ponto a ponto ("pixel" a "pixel"), como descrevemos no Capítulo 1. Nosso sistema, no entanto, requer apenas um número pequeno de varreduras que é proporcional à complexidade do domínio dos objetos analisados. Para objetos simples, tais como poliedros regulares, a técnica de processar uma imagem obtida através de câmeras de TV é essencialmente dispendiosa. A quantidade de informação é excessiva e os programas de reconhecimento necessitam eliminar esta redundância, consumindo por isso grande quantidade de tempo.

A tecnologia desenvolvida em nosso trabalho prevê um universo formado por um número limitado de objetos poliédricos, como mostramos no Capítulo 4, e o banco de conhecimento é gerado de acordo com esta restrição. O sistema eletrônico consiste de um feixe de Laser convenientemente dirigido por um Microcomputador sobre estes objetos, esquema este que apresentamos no Capítulo 3.

Empregamos, então, a técnica de se obter o "formato através do sombreamento" e o modelo de rotulagem de arestas e vértices desenvolvido por Huffman-Clowes como discutimos no Capítulo 2.

Considerando estes parâmetros, apresentamos alguns resultados neste último Capítulo, para uma melhor análise da potencialidade de todo o sistema implementado.

2. CURVAS DE SUPERFÍCIE.

Nosso sistema de aquisição de dados é baseado na quantidade de luz refletida do feixe de Laser, ou seja, na Luminância de cada ponto sobre a cena. Se gerarmos uma varredura completa de $n \times m$ pontos teremos uma imagem digital correspondente à esta variação dando-nos a idéia do comportamento das superfícies envolvidas. O programa apresentado no Apêndice B-4 realiza esta tarefa gerando um arquivo correspondente a uma imagem de 50×50 pontos, que é posteriormente tratado pelo programa ENERGRAPH [Enertronics 1986]. Este levanta as curvas de superfícies do objeto, conforme a visão do observador colocado no eixo z . O resultado pode ser visto nas figuras seguintes.

A Fig.1 mostra um cubo com uma de suas faces voltada para o observador colocado no eixo z . A varredura do feixe de Laser foi realizada horizontalmente da esquerda para a direita e subindo gradativamente no eixo y . O método de aquisição do sinal leva em conta a média e a estatística dos valores de conversão, conforme apresentado no Capítulo 4.

O fundo contra o qual os objetos são vistos apresentam

Luminância zero, ou melhor, calibrada no sistema para corresponder ao valor zero. Isto representa a base formada pelo plano (x,y) onde a varredura do feixe de Laser é realizada. A cota de cada ponto (x,y) equivale ao valor refletido e é colocada no eixo z . Desta forma, podemos observar o comportamento de

Fig.1 - Curvas de superfície de um cubo com uma face voltada para o observador.

cada varredura e comparar o resultado com o da Fig.2 onde a aquisição do sinal foi realizada de modo simples, sem nenhuma técnica de redução de ruído, ou seja, quando apenas um valor do código digital do conversor A/D é levado em conta.

Fig.2 - Curvas de superfície de um cubo. Aquisição sem eliminação de ruído.

Podemos observar o índice elevado de ruído na digitalização do sinal. As bordas do objeto que devem fornecer informação para sua classificação são arredondadas e a superfície apresenta variações significativas. Isto pode não causar dificuldades ao algoritmo de reconhecimento quando a face do objeto for

perpendicular ao eixo de visão, mas no caso de determinação de uma aresta interna, este ruído é extremamente inconveniente

podendo produzir falsas arestas e falsas inclinações.

A Fig.3 mostra as curvas de superfície de um objeto com uma aresta interna visível ao observador. O algoritmo de aquisição de dados utilizado possui redução de ruído como mostrado no Capítulo 4. Podemos observar através da comparação da Fig.3 com a Fig.4 que a aresta é bem mais visível. As curvas de superfície quando não filtradas no processo de conversão apresentam bordas arredondadas, dificultando a extração de características.

Fig.3 - Cubo com aresta interna visível ao observador. Algoritmo com redução de ruído.

Fig.4 - Cubo com aresta interna visível ao observador. Algoritmo sem redução de ruído.

Como pode ser observado, mesmo com o método estatístico para melhoria do sinal, a superfície do objeto não se apresenta de forma regular, sendo necessário portanto o emprego de suavização. Isto será avaliado no próximo item.

O Apêndice C-1 apresenta as curvas de superfície dos objetos pertencente ao universo reconhecível pelo algoritmo proposto. Apesar de nosso método não tratar de superfícies curvas, um cilindro e um cone assentados

respectivamente sobre suas bases foram incluídos neste Apêndice com o intuito de mostrar a possibilidade de se desenvolver outros programas que reconheçam também este tipo de superfície.

3. SIMULAÇÃO DE RESULTADOS COM DADOS REAIS.

O processo que utilizamos para simulação dos resultados foi o de adquirirmos os dados referentes ao "arranjo de intensidades" diretamente através de uma varredura horizontal de cada objeto do domínio e submetê-los ao programa MATHCAD [Mathcad 1988]. Esta tarefa é realizada pelo programa VLOTUS.C apresentado no Apêndice B-4. Este programa dirige o feixe de Laser sobre o objeto colocado no campo visual e digitaliza, com redução de ruídos, a Luminância de cada ponto formando um arquivo correspondente ao "arranjo de intensidades".

Graficamente, podemos denominar o "arranjo de intensidades" como o perfil do objeto na varredura. A Fig.5 mostra os dados referentes a um cubo (objeto 2P) colocado no campo de visão com uma aresta interna visível ao observador e localizada aproximadamente no meio da varredura. Comparando-se a Fig.5 com a Fig.6 podemos observar que o nível de ruído proveniente da aquisição

Fig.5 - Perfil da superfície de um cubo com aresta visível.

simples, com apenas uma leitura do conversor A/D, por ponto do arranjo, fornece praticamente uma indefinição. Torna-se, assim, bastante difícil

Fig.6 - Perfil de um cubo com aresta visível adquirido sem redução de ruído.

para o programa principal de reconhecimento recuperar a aresta visível na Fig.6.

Considerando portanto, como exemplo, o perfil obtido pelo arranjo de intensidades mostrado na Fig.5, o programa procede a uma suavização para a eliminação de eventuais discontinuidades, conforme foi discutido no Capítulo 4. Esta

suavização pode ser obtida discretamente através da média calculada a cada número fixo de pontos, por exemplo três (3), da seguinte forma:

$$SN_I = \frac{1}{3} \cdot (N_{I-1,0} + N_{I,0} + N_{I+1,0})$$

onde $I=1, \dots, 48$ para o conjunto de cinquenta (50) pontos. O resultado é a curva mostrada na Fig.7 onde as características do perfil são acentuadas e a redução da inclinação da aresta não é significativa, a ponto de não permitir sua identificação.

Derivando-se este perfil em relação ao eixo x (eixo I nas figuras), obtemos a curva da Fig.8. De forma discreta, a derivada pode ser obtida através da subtração de dois pontos onde o deslocamento no eixo x seja unitário. Assim temos,

$$DN_I = SN_{I+1} - SN_I$$

Os pontos de máximo desta função determinam as bordas do objeto e a aresta pode ser observada pela mudança de sinal que ocorre no ponto equivalente (aproximadamente no centro do gráfico).

Fig. 7 - Perfil suavizado do cubo com aresta visível da Fig. 5.

que é a curva mostrada na Fig. 9. Podemos ver então, que uma pequena deformação ocorrida no perfil mostrado na Fig. 5, gera uma falsa aresta próximo à verdadeira.

O conhecimento prévio dos objetos reconhecíveis, pode no entanto, fornecer subsídios para um

A segunda derivada do perfil, ou seja, do arranjo de intensidades, pode ser obtida derivando-se a curva DN da Fig. 8. Desta forma temos:

$$DDN_I = DN_{I+1} - DN_I$$

Fig. 8 - Primeira derivada do perfil mostrado na Fig. 5

tratamento matemático mais dedicado. Assim, em nosso caso, como

estamos lidando com superfícies planas de sólidos poliédricos. é de se supor que teóricamente os perfis possam ser aproximados por linhas retas.

Se tomarmos a primeira derivada mostrada na Fig.8,

Fig.9 - Segunda derivada. Detecção de aresta interna.

Fig.10 - Perfil do objeto eliminando-se o fundo.

determinamos as bordas do objeto pelo máximo e mínimo da função. Assim, a fronteira do objeto mais à esquerda é denominada de

$$y=I$$

$$p/ \quad DN_I = \max(DN)$$

e a fronteira mais à direita é

calculada como

$$yy=I$$

$$p/ \quad DN_I = \min(DN)$$

Colocando em um gráfico, apenas o perfil do objeto, sem o fundo contra qual o

Fig.11 - Linearização do perfil do objeto.

mesmo é observado, eliminamos na função da Fig.5 os valores antes da fronteira esquerda e depois da fronteira direita. O resultado é visto na Fig.10. Calculando então, o ponto de máximo do perfil suavizado do objeto como,

$$ym = I$$

$$p/ \quad SN_I = \max(SN)$$

podemos interpolar linearmente [Davis 1975] cada parte do perfil entre cada borda do objeto (y e yy) e este ponto de máximo (ym). O resultado é a aproximação da superfície por uma reta (se $SN_{ym} = SN_y = SN_{yy}$) ou por dois segmentos de reta (se $SN_{ym} > SN_y$ e $SN_{ym} > SN_{yy}$) conforme mostra a Fig.11.

A primeira derivada deste perfil linearizado é isenta de ruídos e pode ser vista na Fig.12. A segunda derivada, mostrada na

Fig.12 - Primeira derivada do perfil linearizado.

Fig.13, localiza o ponto onde ocorre a inversão de sentido na inclinação das duas superfícies, ou seja, o local da aresta interna. É

Fig.13 - Segunda derivada. Detecção de aresta interna.

Fig.14 Perfil de uma superfície perpendicular ao eixo de visão.

Fig.16 - Segunda derivada do perfil perpendicular ao eixo de visão.

VLOTUS.C por uma varredura sobre a superfície de um cubo, perpendicular ao eixo de visão, gerando o gráfico de perfil da Fig.14.

A primeira e a segunda derivadas desta função, aparecem

importante salientar que sua amplitude é proporcional ao ângulo entre elas. Logo, uma verificação do valor máximo da segunda derivada permite concluir sobre a existência ou não de aresta interna na superfície em observação.

Fig.15 - Primeira derivada do perfil perpendicular ao observador.

Seja, por exemplo, o mesmo procedimento matemático aplicado a um "arranjo de intensidades"

obtido através do programa

Fig.17 - Linearização da função da Fig.14.

calculando-se seu máximo valor, podemos aproximar o perfil por uma reta ou por dois segmentos de reta. A Fig.17 mostra esta linearização para o caso do perfil da superfície perpendicular ao eixo de visão.

A primeira e a segunda derivadas, mostradas nas Fig.18 e Fig.19 respectivamente vem confirmar o cálculo apresentado no Capítulo 2 (ver Fig.10 e Fig.11 do Capítulo 2) onde a variação da Luminância em uma superfície perpendicular ao eixo de visão não é constante em toda a sua extensão, mas apresenta uma pequena variação. Esta variação não é significativa para o programa de reconhecimento que procura por amplitudes da segunda derivada superiores a 80 pontos digitais, valor mínimo estimado para representação de uma aresta interna visível. O que podemos concluir deste processo é que apesar do sinal captado pelo sistema, apresentar ruídos inerentes ao meio de aquisição, o

nas figuras Fig.15 e Fig.16 respectivamente.

Utilizando-se do mesmo procedimento, ou seja, eliminando-se no perfil o que estiver fora de suas bordas (fronteira do objeto), e

Fig.18 - Primeira derivada da função linearizada.

pequeno tratamento matemático realizado fornece subsídios suficientes para a determinação das características necessárias à classificação do objeto.

Fig.19 - Segunda derivada da função.

4. RESULTADOS EXPERIMENTAIS.

O algoritmo implementado em TURBO C, que opera sobre o protótipo desenvolvido, para o reconhecimento do formato de objetos tri-dimensionais, apresentou o seguinte desempenho:

- Considerando o universo de objetos reconhecíveis formados pelos poliedros descritos no Capítulo 4, optamos pelo número mínimo de varreduras. Isto significa que o sistema realiza para cada vista, uma varredura horizontal inferior, uma varredura vertical no meio do objeto e uma varredura horizontal superior. Apenas duas vistas são consideradas para a classificação dos objetos.

Para poliedros de fronteira paralela tais como os objetos 1P (prisma de base triangular assentado sobre sua base) e 2P

(cubo) o índice de "acertos" é de aproximadamente 80%, conforme mostra a Tab.I. Este índice equivale a um ensaio de 10 (dez) medidas em 5 (cinco) posições diferentes, para cada objeto.

A principal fonte de "erros" é relativa a variações de medidas em uma varredura,

oriundas por exemplo de *Tab.I - Porcentagem de reconhecimento correto no sistema com apenas uma varredura localizada.* flutuações na intensidade do feixe de Laser, impedindo sua

correção. Outro fator prejudicial está relacionado às superfícies não perfeitamente difusoras dos modelos utilizados (poliedros construídos com chapas de PVC lixadas). No entanto, a

Objeto 1P	80%
Objeto 2P	80%
Objeto 1C	60%
Objeto 2C	60%
Objeto 1PD	30%
Objeto 2CD	30%

convivência com estes fatores adversos permite a verificação das deficiências e da versatilidade do método.

A maior dificuldade reside em se adaptar os parâmetros de classificação (x_1, x_2, x_3, x_4) (ver Fig.11 do Capítulo 4) para que uma aresta possa ser definida corretamente dentro do ramo da árvore. O parâmetro que define se a superfície é plana, inclinada negativamente ou positivamente em relação ao observador, é bastante crítico sendo responsável pelo reconhecimento de um objeto como outro de característica similar. Por exemplo, o Objeto 1C (pirâmide de base triangular) com o Objeto 2C (pirâmide de base quadrada) possuem uma pequena diferença entre as inclinações de suas superfícies (90° e 60°). Isto pode fazer o programa confundí-los, em determinadas

posições destes objetos.

A solução adotada para reduzir o número de erros, foi a de estabelecer a técnica de "votação" entre três varreduras horizontais, deslocadas de 5 pontos no eixo y. Desta forma o programa, em detrimento da velocidade de reconhecimento, realiza duas varreduras horizontais, gerando os parâmetros de classificação e checando a coerência entre eles. Caso discordem, uma terceira varredura é gerada para realizar a "votação" entre elas e decidir por duas iguais. Isto aumenta significativamente o índice de "acertos" dos objetos visualizados, conforme mostra a Tab.II.

É importante salientar que os índices mostrados não são absolutos, mas aproximados pelos resultados das medidas realizadas.

Objeto 1P	100%
Objeto 2P	100%
Objeto 1C	70%
Objeto 2C	70%
Objeto 1PD	60%
Objeto 2C	60%

Durante os testes, foi possível checar a eficiência do "banco de conhecimento" *Tab.II - Índice de reconhecimento para o sistema com "votação".*

préviamente estabelecido. Os parâmetros de classificação foram listados, durante cada reconhecimento, e a geração de opções não previstas, tais como:

- face inclinada positiva (código 2) e aresta vertical (código 15) para o objeto 2P (cubo);
- face plana (código 5) e face inclinada positiva (código 2) para o objeto 1PD (prisma apoiado sobre uma de suas faces);
- e outras mostradas no Apêndice B-3 sob o comentário "opções de treinamento", foram incluídas para um melhor desempenho do

programa. Isto é uma demonstração que qualquer objeto que possa ser classificado através dos parâmetros estabelecidos neste trabalho, pode ter seu "banco de conhecimento" estabelecido empiricamente pelo próprio programa de reconhecimento.

Concluindo, os resultados poderão ser melhorados significativamente através do avanço tecnológico dos circuitos e equipamentos empregados. A pequena diferença entre os valores das inclinações de superfícies de objetos similares, é extremamente susceptível a ruídos inerentes ao sistema de aquisição. No entanto, a segurança de um equipamento tecnologicamente superior, permitirá a melhoria do índice de "acertos" e também a diversificação do universo de objetos reconhecíveis.

5. CONCLUSÕES FINAIS.

A Tecnologia para o Reconhecimento do Formato de Objetos Tri-dimensionais proposta neste trabalho, cumpre seus objetivos dentro das restrições estabelecidas. Através de uma abordagem alternativa para o problema de Visão Computacional demonstramos que conhecido o domínio dos objetos e sendo estes de estrutura visual simples, é possível determinar parâmetros que os classifiquem.

A generalização da função de entrada nos métodos tradicionais através de uma imagem obtida por câmera de TV e digitalizada, tem aplicabilidade em grande parte dos casos onde objetos complexos compõem cenas com sombras e superposições. No

entanto, determinadas situações onde objetos poliédricos simples e solitários estabeleçam o universo a ser reconhecido, nosso método pode ser empregado com redução de custo e tempo de decisão. Enquanto uma imagem transporta uma grande quantidade de informações sobre a cena, tendo que ser drasticamente reduzida pelo processamento matemático, nosso método apenas retira parâmetros de partes significantes do objeto que são determinantes para sua classificação.

A utilização de um mesmo caminho ótico para o feixe de Laser incidente e para o feixe de retorno refletido sobre a superfície, otimiza sensivelmente o equacionamento e a geometria do sistema. Um mesmo bloco contém o canhão de Laser, os espelhos defletores e o sistema de recepção do sinal.

O emprego da mesa giratória é eficiente quando definimos a posição do observador de forma a permitir variações coerentes entre as diferentes vistas que formam o banco de conhecimento do sistema. Para os objetos definidos, a descrição pelas inclinações de suas superfícies e pela posição das arestas internas, mostrou que o eixo e o plano de rotação adotados geraram um banco de dados bastante simplificado. No entanto, outros eixos de visão podem ser utilizados e novas descrições levantadas em pesquisas futuras.

O tratamento de superfícies curvas não foi aventado, neste trabalho, dado que estabelecemos para o desenvolvimento do programa de reconhecimento a determinação de parâmetros referentes à arestas e inclinações de superfícies planas. No entanto, a inclusão no Apêndice C-1 de curvas de superfícies obtidas pelo sistema, de um cilindro e de um cone, demonstram a

viabilidade de se desenvolver outros programas que procedam à extração destas características.

Tecnologicamente o protótipo implementado pode ser otimizado em busca de um melhor desempenho do sistema. A qualidade do sinal recebido pelo foto-transistor, a ser digitalizado, é fundamental para a minimização de erros de reconhecimento. Esta qualidade está intimamente relacionada com o equipamento de Laser utilizado. A intensidade luminosa do feixe incidente deve ser constante para que as medidas de Luminância sejam coerentes. Variações nesta intensidade entre um ponto e outro da superfície causam erros de interpretação na localização das arestas internas.

A escolha da frequência de "chopper" do feixe de Laser em 90 Hz com o intuito de facilitar a implementação dos filtros do "lock-in", apresentou a inconveniência de estar muito próximo da frequência de 120 Hz oriunda de lâmpadas fluorescentes. O aumento do número de pás ou ranhuras que interceptam o feixe, de duas (2) para vinte (20) ou trinta (30) aumentará a frequência de trabalho para aproximadamente 1 KHz o que deverá melhorar a operação do sistema em luz ambiente. Evidentemente os filtros deverão ser reprojatados para a nova frequência de trabalho.

Finalmente, a velocidade do protótipo foi comprometida pela utilização de galvanômetros mecânicos. Estes, apesar de serem de excelente qualidade, apresentam uma inércia inerente. Este fato não permite que no deslocamento de um ponto para outro ao longo de um eixo, a leitura do conversor A/D possa ser realizada sem a inclusão de atrasos entre duas leituras consecutivas. Acreditamos que com o avanço da tecnologia, defletores eletro-

Tecnologia para o Reconhecimento do Formato de Objetos Tri-dimensionais.

óticos possam ter seus custos reduzidos para utilização em um sistema como o por nós proposto.

A utilização de um Microcomputador de melhor desempenho e a otimização dos parâmetros tecnológicos, como referido acima, podem efetivamente reduzir o tempo de classificação de um objeto, que no protótipo implementado é de aproximadamente 3 minutos, considerando a varredura com "votação". No entanto, é importante salientar que este tempo é também função da complexidade do objeto que pode gerar uma base de conhecimento maior, forçando mais varreduras e mais giros na mesa, para que mais vistas estejam disponíveis para sua classificação.

Mas, a visão de um Robô em tarefa dedicada de separar peças mecânicas ou caixas de papelão em uma linha de montagem industrial, nem sempre necessita da mesma riqueza de detalhes da visão de um ser humano. É portanto plausível afirmar que a Tecnologia para o Reconhecimento do Formato de Objetos Tri-dimensionais desenvolvida neste trabalho apresenta um campo promissor de pesquisas futuras em Visão Computacional.

REFERÊNCIAS BIBLIOGRÁFICAS.

[Attneave 1954]

Attneave, F. - "Some Informational Aspects of Visual Perception.", *Psychological Review*, Vol.61, Nº 3, 1954, pp.183-193.

[Ballard & Brown 1982]

Ballard, D.H. and Brown, C.M. - "Computer Vision.", Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982.

[Barrow & Tenenbaum 1981]

Barrow, H.G. and Tenenbaum, J.M. - "Computational Vision.", *Proceedings of the IEEE*, Vol.69, Nº 5, May 1981, pp.572-595.

[Barry et al 1988]

Barry, M. - Cyrluk, D. - Kapur, D. - Mundy, J. and Nguyen, V.D. - "A Multi-Level Geometric Reasoning System for Vision.", *Artificial Intelligence*, Vol.37, Nº 1-3, December 1988, pp.291-332.

[Batchelor & all. 1985]

Batchelor, B.G. - Hill, D.A. and Hodgson, D.C. - "Automated Visual Inspection.", IFS (Publications) Ltd, UK, North-Holland, 1985.

[Besl & Jain 1985]

Besl, P.J. and Jain, R.C. - "Three-dimensional Object Recognition.", *Comput. Surveys*, Vol.17, Nº 1, March 1985, pp.75-145.

[Bradley 1984]

Bradley, D.J. - "Assembly Language Programming for the IBM Personal Computer.", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

[Brunet 1987]

Brunet, M. - "Laser-Based 3D Digitizing.", *Computer Graphics World*, November 1987, pp.103-104.

[Cardoso 1979]

Cardoso, R.P. - "Luminotécnica Fundamental", Dinalivro, Lisboa, Portugal, 1979.

[Castleman 1979]

Castleman, K.R. - "Digital Image Processing.", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1979.

[Cavaggioni 1969]

Cavaggioni, A. - "Physiology of the Retina.", (in Automatic Interpretation and Classification of Images, A. Grasselli Ed.), Academic Press, New York, 1969, pp.175-185.

[Chakravarty 1979]

Chakravarty, I. - "A Generalized Line Junction Labeling Scheme with Applications to Scene Analysis.", IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol.PAMI-1, No 2, April 1979, pp.202-205.

[Churchland & Churchland 1990]

Churchland, P.M. and Churchland, P.S. - "Could a Machine Think?", Scientific American, Vol.262, No1, January 1990, pp.32-37.

[Clowes 1971]

Clowes, M.B. - "On Seeing Things.", Artificial Intelligence, Vol.2, 1971, pp.79-116.

[Connolly & Stenstrom 1986]

Connolly, C.I. and Stenstrom, J.R. - "Construction of Polyedral Models from Multiple Range Views.", Eighth International Conference on Pattern Recognition, Paris-France, October 27-31, 1986, pp.85-87.

[Davis 1975]

Davis, P.J. - "Interpolation and Approximation.", Dover Publications, Inc., New York, 1975.

[Dhome & Kasvand 1987]

Dhome, M. and Kasvand, T. - "Polyhedra Recognition by Hypothesis Accumulation.", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.PAMI-9, No 3, May 1987, pp.429-438.

[Duda & Hart 1973]

Duda, R.O. and Hart, P.E. - "Pattern Classification and Scene Analysis.", John Wiley & Sons, New York, 1973.

[Enertronics 1986]

Enertronics - "A Quick Tip to Energraphics 2.0.", Enertronics Research, Inc., St. Louis, 1986.

[Falk 1972]

Falk, G. - "Interpretation of Imperfect Line Data as a Three-Dimensional Scene.", *Artificial Intelligence*, Vol.3, No 2, Spring 1972, pp.101-144.

[Feldman & Ballard 1982]

Feldman, J.A. and Ballard, D.H. - "Connectionist Models and Their Properties.", *Cognitive Science*, Vol.6, 1982, pp.205-254.

[Fisher 84]

Fisher, E.H. - "The Evolution of the Modern Lock-in Amplifier.", *Application Note IAN 35*, Ithaco, 1984, pp.1-12.

[Fu 1974]

Fu, K.S. - "Syntactic Methods in Pattern Recognition.", *Academic Press*, New York, 1974.

[Fu 1976]

Fu, K.S. - "Digital Pattern Recognition.", *Springer-Verlag*, Berlin Heidelberg, 1976.

[Fu & Rosenfeld 1976]

Fu, K.S. and Rosenfeld, A. - "Pattern Recognition and Image Processing.", *IEEE Transactions on Computers*, Vol.C-25, No 12, December 1976, pp.1336-1345.

[Fukushima 1988]

Fukushima, K. - "A Neural Network for Visual Pattern Recognition.", *IEEE Computer*, Vol.21, No3, March 1988, pp.65-75.

[Gonzalez & Safabakhsh 1982]

Gonzalez, R.C. and Safabakhsh, R. - "Computer Techniques for Industrial Applications and Robot Control.", *IEEE Computer*, Vol.15, No 12, December 1982, pp.17-32.

[Grasselli 1969]

Grasselli, A. - "Automatic Interpretation and Classification of Images.", *Academic Press*, New York, 1969.

[Guerrini 1979]

Guerrini, D.P. - "Introdução à Iluminação.", *EESC-USP, Departamento de Eletricidade, Publicação 007/85*, São Carlos, 1979.

[Guzman 1969]

Guzman, A. - "Decomposition of a Visual Scene into Three-Dimensional Bodies.", (in Automatic Interpretation and Classification of Images, Ed. A. Grasselli), Academic Press, 1969, pp.243-276.

[Haralick 1973]

Haralick, R.M. - "Glossary and Index to Remotely Sensed Image Pattern Recognition Concepts.", Pattern Recognition, Vol.5, 1973, pp.391-403.

[Haralick & Shapiro 1985]

Haralick, R.M. and Shapiro, L.G. - "Survey: Image Segmentation Techniques.", Computer Vision, Graphics, and Image Processing, Vol.29, 1985, pp.100-132.

[Havens & Mackworth 1983]

Havens, W. and Mackworth, A. - "Representing Knowledge of the Visual World.", IEEE Computer, Vol.16, No 10, 1983, pp.90-96.

[Horn 1975]

Horn, B.K.P. - "Obtaining Shape from Shading Information.", (in The Psychology of Computer Vision, P.H. Winston Ed.), McGraw-Hill Book Co., New York, 1975, pp.115-155.

[Horn 1977]

Horn, B.K.P. - "Understanding Image Intensities.", Artificial Intelligence, Vol.8, 1977, pp.201-231.

[Horn & Brooks 1986]

Horn, B.K.P. and Brooks, M.J. - "The Variational Approach to Shape from Shading.", Computer Vision, Graphics, and Image Processing, Vol.33, 1986, pp.174-208.

[Huffman 1971]

Huffman, D.A. - "Impossible Objects as Nonsense Sentences.", Machine Intelligence 6, B. Meltzer and D. Michie (eds.), Edinburgh University Press, Edinburgh, 1971, pp.295-323.

[Hummel & Zucker 1983]

Hummel, R.A. and Zucker, S.W. - "On the Foundations of Relaxation Labeling Process.", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.PAMI-5, No 3, May 1983, pp.267-286.

[Ikeuchi & Horn 1981]

Ikeuchi, K. and Horn, B.K.P. - "Numerical Shape from Shading and Occluding Boundaries.", *Artificial Intelligence*, Vol.17, 1981, pp.141-184.

[Ishii & Nagata 1976]

Ishii, M. and Nagata, T. - "Feature Extraction of Three-Dimensional Objects and Visual Processing in a Hand-Eye System using a Laser Tracker.", *Pattern Recognition*, Vol.8, 1976, pp.229-237.

[Jarvis 1977]

Jarvis, R.A. - "A Simple Hand/Eye Experimental Set-up for Computer Vision Research.", *IEEE Computer*, Vol.10, No 9, September 1977, pp.65-69.

[Jarvis 1983]

Jarvis, R.A. - "A Perspective on Range Finding Techniques for Computer Vision.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-5, No 2, March 1983, pp.122-139.

[Jarvis 1983]

Jarvis, R.A. - "A Laser Time-of-Flight Range Scanner for Robotic Vision.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-5, No 5, September 1983, pp.505-512.

[Kanade 1981]

Kanade, T. - "Recovery of the Three-Dimensional Shape of an Object from a Single View.", *Artificial Intelligence*, Vol 17, 1981, pp.409-460.

[Kanal 1974]

Kanal, L. - "Patterns in Pattern Recognition: 1968-1974.", *IEEE Transactions on Information Theory*, Vol.IT-20, No 6, November 1974, pp.697-722.

[Kirosis 1990]

Kirosis, L.M. - "Effectively Labeling Planar Projections of Polyedra.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.12, No 2, February 1990, pp.123-130.

[Lee & Rosenfeld 1985]

Lee, C.H. and Rosenfeld, A. - "Improved Methods of Estimating Shape from Shading Using the Light Source Coordinate System.", *Artificial Intelligence*, Vol.26, 1985, pp.125-143.

[Mackworth 1973]

Mackworth, A.K. - "Interpreting Pictures of Polyedral Scenes.", *Artificial Intelligence*, Vol.4, No 2, 1973, pp.121-137.

[Mantas 1987]

Mantas, J. - "Methodologies in Pattern Recognition and Image Analysis - A brief Survey.", *Pattern Recognition*, Vol.20, No 1, 1987, pp.1-6.

[Marr 1982]

Marr, D. - "Vision - A Computational Investigation into the Human Representation and Processing of Visual Information.", W.H. Freeman and Co., New York, 1982.

[Mathcad 1988]

Mathcad - "Mathcad 2.0 User's Guide.", MathSoft Inc., Cambridge, Mass., 1988.

[Minsky 1961]

Minsky, M. - "Steps toward artificial intelligence.", *Proceedings of IRE*, Vol.49, 1961, pp.8-30.

[Monchaud 1989]

Monchaud, S. - "3-D Vision and Range Finding Techniques.", *Optics and Lasers in Engineering*, Vol.10, No 3-4, 1989, pp.161-178.

[Murray 1988]

Murray, J. - "Object ID Sensors output On-Line Results.", *Control Engineering*, May 1988, pp.98-101.

[Nandhakumar & Aggarwal 1985]

Nandhakumar, N. and Aggarwal, J.K. - "The Artificial Intelligence Approach to Pattern Recognition - A Perspective and an Overview.", *Pattern Recognition*, Vol.18, No6, 1985, pp.383-389.

[Nayar et. all. 1990]

Nayar, S.K. - Ikeuchi, K. and Kanade, T. - "Determining Shape and Reflectance of Hybrid Surfaces by Photometric Sampling.", IEEE Transactions on Robotics and Automation, Vol.6, No 4, August 1990, pp.418-430.

[Pavlidis 1986]

Pavlidis, T. - "A Critical Survey of Image Analysis Methods.", Eighth Int. Conf. on Pattern Recognition, Paris, France, October 27-31, 1986, pp.502-511.

[Pentland 1984]

Pentland, A. - "Local Shading Analysis.", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.PAMI-6, No 2, March 1984, pp.170-187.

[Pentland 1988]

Pentland, A. - "On the Extraction of Shape Information from Shading.", Saint Paul, Minnesota Proc. AAAI88 The Seventh Nat. Conf. on Artif. Intell., Vol.2, 21-26 August 1988, pp.826-830.

[Rao & Jain 1988]

Rao, A.R. and Jain, R. - "Knowledge Representation and Control in Computer Vision Systems.", IEEE Expert, Spring 1988, pp.64-79.

[Rice & Knight 1973]

Rice, H.S. and Knight, R.M. - "Technical Mathematics.", McGraw-Hill Book, Co., New York, 1973.

[Rich 1988]

Rich, E. - "Inteligência Artificial", Editora McGraw-Hill Ltda., São Paulo, SP, 1988.

[Rioux 1984]

Rioux, M. - "Laser Range Finder based on Synchronized Scanners.", Applied Optics, Vol.23, No 21, 1 November 1984, pp.3837-3844.

[Roda 82]

Roda, V.O. - "Speed Controller and Indicator.", Electronic Product Design, September 1982, pp.33.

[Rosenfeld 1969]

Rosenfeld, A. - "Picture Processing by Computer.", Academic Press, New York, N.Y., 1969.

[Rosenfeld 1981]

Rosenfeld, A. - "Image Pattern Recognition.", Proceedings of the IEEE, Vol.69, No 5, May 1981, pp.596-605.

[Rosenfeld 1988]

Rosenfeld, A. - "Image Analysis and Computer Vision: 1987.", Computer Vision, Graphics, and Image Processing, Vol. 42, No 2, 1988, pp.234-293.

[Rosenfeld 1989]

Rosenfeld, A. - "Image Analysis and Computer Vision: 1988.", Computer Vision, Graphics, and Image Processing, Vol. 46, No 2, May 1989, pp.196-264.

[Rosenfeld 1990]

Rosenfeld, A. - "Image Analysis and Computer Vision: 1989.", Computer Vision, Graphics, and Image Processing, Vol. 50, No 2, May 1990, pp.188-240.

[Sachs 1985]

Sachs, F. - "Sensors and Cameras for Machine Vision.", Laser Focus/Electro-Optics, July 1985, pp.82-89.

[Schildt 1990]

Schildt, H. - "Turbo C Avançado - Guia do Usuário.", McGraw-Hill, São Paulo, SP, 1990.

[Shirai 1975]

Shirai, Y. - "Analyzing Intensity Arrays using Knowledge about Scenes.", (in the Psychology of Computer Vision, Ed. P.H. Winston), McGraw-Hill Book Co., New York, 1975, pp.93-113.

[Simon 1975]

Simon, J.C. - "Recent progress to formal approach of Pattern Recognition and scene analysis.", Pattern Recognition, Vol.7, 1975, pp.117-124.

[Stout & Kaufman 1976]

Stout, D.F. and Kaufman, M. - "Handbook of Operational Amplifier Circuit Design.", McGraw-Hill Book Co., New York, 1976.

[Stuivinga et all. 1989]

Stuivinga, M. - Nobel, J. - Verbeek, P.W. - Steenvoorden, G.K. and Joon, M.M. - "Range-Finding Camera based on a Position-Sensitive Device Array.", *Sensors and Actuators*, Vol.17, 1989, pp.255-258.

[Tanimoto 1985]

Tanimoto, S. - "TUTORIAL: Artificial Intelligence.", Eighth International Conference on Pattern Recognition, Paris, France, October 27, 1985.

[Tou & Gonzalez 1974]

Tou, J.T. and Gonzalez, R.C. - "Pattern Recognition Principles.", Addison-Wesley Publ. Co., Reading, Massachusetts, 1974.

[Underwood & Coats Jr. 1975]

Underwood, S.A. and Coats Jr., C.L. - "Visual Learning from Multiple Views.", *IEEE Transactions on Computers*, Vol.C-24, No 6, June 1975, pp.651-661.

[Verhagen 1975]

Verhagen, C.J.D.M. - "Some general remarks about Pattern Recognition; its definition; its relation with other disciplines; a literature survey.", *Pattern Recognition*, Vol.7, 1975, pp.109-116.

[Waltz 1975]

Waltz, D. - "Understanding Line Drawings of Scenes with Shadows.", (in *Psychology of Computer Vision*, Ed. P.H. Winston), McGraw-Hill Book Co., New York, 1975, pp.19-91.

[Watanabe 1969]

Watanabe, S. - "Methodologies of Pattern Recognition.", Academic Press, New York, 1969.

[West 1983]

West, P.C. - "Machine Vision in Practice.", *IEEE Transactions on Industry Applications*, Vol.IA-19, No 5, September/October 1983, pp.794-801.

[Willen & Krantz 1983]

Willen, D.C. and Krantz, J.I. - "8088 Assembler Language Programming: The IBM PC.", Howard W. Sams & Co., Inc., Indiana, 1983.

[Winston 1975]

Winston, P.H. - "The Psychology of Computer Vision.", McGraw-Hill Book Co., New York, 1975.

[Yoda et al. 1973]

Yoda, H. - Ikeda, S. and Ejiri, M. - "A New Attempt of Selecting Objects Using a Hand-eye System.", Hitachi Review, Vol.22, No 9, 1973, pp.362-365.

APÊNDICE A

CIRCUITOS ELETRÔNICOS :

A-1 ----- INTERFACE DIGITAL

A-2 ----- CONTROLE DOS GALVANÔMETROS

A-3 ----- ACIONAMENTO DA MESA GIRATÓRIA

A-4 ----- INTERFACE ANALÓGICA

APÊNDICE A-1

CIRCUITO ELETRÔNICO DA INTERFACE DIGITAL

Folha 1	-----	CIRCUITO DOS REGISTRADORES INTERNOS.
Folha 2	-----	MAPEAMENTO DA INTERFACE.
Folha 3	-----	CIRCUITO DOS CONVERSORES D/A.
Folha 4	-----	CIRCUITO DO CONVERSOR A/D.
Folha 5	-----	CIRCUITO AMOSTRADOR/RETENTOR.

Tecnologia para o Reconhecimento do Formato de Objetos Tri-dimensionais.

Tecnologia para o Reconhecimento de Formato de Objetos Tri-dimensionais.

Tecnologia para o reconhecimento de formato de objetos 171-dimensionais.

APÊNDICE A-2

CIRCUITO ELETRÔNICO DE CONTROLE DOS GALVANÔMETROS.

Folha 1 ----- CIRCUITO DOS CONTROLADORES (X,Y).

Nome: _____		
CIRCUITO DE CONTROLE DOS GALVANOMETROS		
Tem: Num	do Doc.	REV
A	APENDICE A-2	1.0
Data: _____		1 de 1

APÊNDICE A-3

CIRCUITO ELETRÔNICO DE ACIONAMENTO DA MESA GIRATÓRIA.

Folha 1 ---- CIRCUITO DE CONTROLE DO MOTOR DE
PASSO.

Folha 2 ---- SENSOR DE POSICIONAMENTO DA MESA.

Tecnologia para o reconhecimento do formato de Objetos Tri-dimensionais.

MESA GIRATORIA			
Nome:			
CIRCUITO DE REFERENCIA DE POSICIONAMENTO			
Tam:	Num do Doc.	REV	
A	APENDICE A-3	1.1	
Data:		Abri 18, 1991	Folha 2 de 2

Tecnologia para o Reconhecimento do formato de objetos tri-dimensionais.

APÊNDICE A-4

CIRCUITO ELETRÔNICO DA INTERFACE ANALÓGICA.

Folha 1	-----	CIRCUITO DE CONTROLE DE VELOCIDADE DO MOTOR DE "CHOPPER".
Folha 2	-----	CIRCUITO DE DEFASAGEM DO SINAL DE REFERÊNCIA.
Folha 3	-----	CIRCUITO DE DETECÇÃO DO SINAL REFLETIDO.
Folha 4	-----	CIRCUITO DE FILTRAGEM DO SINAL.
Folha 5	-----	CIRCUITO DE RETIFICAÇÃO. MISTURADOR.
Folha 6	-----	CIRCUITO DE INTEGRAÇÃO DO SINAL.

INTERFACE ANALOGICA			
Nome: _____			
CIRCUITO DE DEFASAGEM DA REFERENCIA			
Tem:	Num. do Doc.		REV
A	APENDICE A-4		1.1
Data:	Abril 18, 1991		Folha 2 de 8

INTERFACE ANALOGICA			
Nome:			
AMPLIFICACAO DO SINAL			
Tem	Num do Doc.		REV
A	APENDICE A-4		1.1
Data:	Abril 18, 1981		Folha 2 de

FILTRO PASSA BANDA (80 Hz)

FILTRO REJEITA BANDA (80 Hz)

Tecnologia para o Reconhecimento do Formato de Objetos Tri-dimensionais.

INTERFACE ANALOGICA			
Nome: FILTRAGEM DO SINAL			
Tam: A	Num do Doc: A	APENDICE A-4	REV 1.1
Data: Abril 18, 1991/Folha 4 de 8

Tecnologia para o Reconhecimento do formato de objetos tri-dimensionais.

Tecnologia para o reconhecimento do formato de objetos 171-dimensionais.

APÊNDICE B

PROGRAMAS DE CONTROLE :

B-1 ----- ASSEMBLER DO A/D

B-2 ----- ESTATÍSTICA NO A/D

B-3 ----- RECONHECIMENTO DE OBJETOS

B-4 ----- AQUISIÇÃO DE DADOS

APÊNDICE B-1

PROGRAMA EM ASSEMBLER DO CONVERSOR A/D.

Este programa escrito em linguagem Assembler do Microprocessador Intel 8088, realiza o controle e a leitura do conversor A/D. Dispara o início de conversão e aguarda os dados quando estes estiverem disponíveis para leitura. Desta forma o tempo limitante da velocidade de operação fica restrito ao conversor e não ao programa.

Para que se possa reduzir ruídos aleatórios nas medidas, o programa realiza a média de dezesseis (16) leituras consecutivas.

```

-----
;      Tecnologia para o reconhecimento do formato de objetos
;      tridimensionais.
-----
;      _LeAD16 : Rotina que lê o conversor A/D 16 vezes e retorna a
;                  média aritmética das mesmas.
-----
;
;
_TEXT      segment      byte public 'CODE'
DGROUP    group        _DATA,_BSS
          assume        cs:_TEXT,ds:DGROUP
_TEXT      ends
_DATA      segment word public 'DATA'
d@         label        byte
d@w        label        word
_DATA      ends
_BSS       segment word public 'BSS'
b@         label        byte
b@w        label        word
_BSS       ends
_TEXT      segment      byte public 'CODE'
_Le_AD16   proc near

          xor          cx,cx          ; ch soma msb, cl contador
          xor          bx,bx          ; soma lsw

adlo:      mov          dx,537          ; amostragem e retenção
          in           al,dx
          in           al,dx
          in           al,dx
          mov          dx,534          ; 0x216 inicia conversão
          out          dx,al
          mov          dx,536          ; 0x218 STS status
loxx2:     in           al,dx
          and          al,1          ; só interessa o bit 0
          cmp          al,1
          je           loxx2
          mov          dx,534          ; msb A/D
          in           al,dx
          mov          ah,al
          inc          dx              ; lsb A/D
          in           al,dx          ; leitura completa
          add          bx,ax          ; lsw
          adc          ch,0          ; msw
          inc          cl
          cmp          cl,16
          jl           adlo
;          neste ponto, temos a média de 16
;          leituras nos regs. ch (msw) e bx (lsb)
          mov          ah,ch
          mov          al,bh
          ret

```

```
_Le_AD16    endp
_TEXT       ends
_BSS        segment word public 'BSS'
_BSS        ends
_DATA       segment word public 'DATA'
s@          label      byte
_DATA       ends
_TEXT       segment    byte public 'CODE'
_TEXT       ends
            public     _Le_AD16
            end
```

APÊNDICE B-2

PROGRAMA EM LINGUAGEM C PARA ESTATÍSTICA NO CONVERSOR A/D.

Este programa escrito em linguagem C (TURBO C) realiza uma estatística de valores lidos do conversor A/D, através da rotina mostrada no APÊNDICE A-1. Lê trinta e dois (32) dados convertidos, calcula a média e o desvio padrão de cada leitura. Caso algum dos dados ultrapassem o valor estabelecido de cinquenta pontos (50), este é descartado e substituído por um novo. O processo suporta até cento e vinte e oito (128) erros, retornando ao programa principal a média das leituras.

Isto é realizado para que se possa garantir cada ponto dentro de uma dada tolerância (50 valores digitais) e reduzir assim o ruído gerado por todo o sistema de aquisição.

```
-----  
; Rotina de estatística do conversor A/D  
-----  
;  
  
#include <stdlib.h>  
#include <dos.h>  
  
#define NROLTR 32  
#define DESVIOMAX 50  
#define NROT 128  
#define TEMPOENTRELEITURAS 10  
  
int Le_AD16(void);  
  
/* */  
/* */  
/* Aquisição: realiza uma coleta junto ao conversor */  
/* A/D. Verifica se os dados são coerentes. */  
/* */  
/* */  
/* */  
  
int Aquisicao(void){  
    static int DadosColetados[NROLTR];  
    static long Soma;  
    static long Media;  
    int erro = 0;  
    int i;  
    int cont = 0;  
    for(i = 0 ; i < NROLTR ; i ++ ){  
        DadosColetados[i] = Le_AD16();  
        delay(1);  
    }  
    do{  
        cont++;  
        erro = 0;  
        Soma = 0;  
        for(i = 0 ; i < NROLTR ; i ++ ){  
            Soma += DadosColetados[i];  
        }  
        Media = Soma/NROLTR;  
        for(i = 0 ; i < NROLTR ; i ++ ){  
            if( ( labs(Media - DadosColetados[i] )) > DESVIOMAX){  
                erro = 1;  
                DadosColetados[i] = Le_AD16();  
                delay(TEMPOENTRELEITURAS);  
            }  
        }  
    }while( (erro) && (cont < NROT));  
    return((int)Media);  
}
```

APÊNDICE B-3

**PROGRAMA PRINCIPAL PARA O RECONHECIMENTO
DE OBJETOS TRI-DIMENSIONAIS.**

Este é o programa escrito em linguagem C (TURBO C) que reconhece qualquer dos objetos referidos no Capítulo 4, em qualquer das posições possíveis.

IMG5.H

```
int Deflete(int X, int Y);
int DefleteI(int X, int Y);

#define ES      0x219
#define SENSOR  0x21A
#define LSBDAx  0x210
#define MSBDAx  0x211
#define LSBDAy  0x212
#define MSBDAy  0x213
#define GATILHO 0x214
#define MSBAD   0x216
#define LSBAD   0x217
#define STS     0x218
#define NUMEROPONTOS 157
#define PASSOVARREDURA 13      /* para 157 pontos */
#define MINLUM (4095/5)

/*-----*/
/*
/*
/*
/*  Observação importante :
/*  O port 0x211 contem:
/*  nibble inf.      : 4 bits mais sig do da1 (x).
/*  nibble superior : controle do motor de passo.
/*
/*
/*
/*-----*/

struct conversor_t{
    unsigned lsb    : 8;
    unsigned msb    : 8;
};

union coordenada_t{
    struct {
        unsigned da    : 12;
        unsigned mesa : 4;
    }valor;
    struct conversor_t byte;
};

struct Class {
    int Tamanho[2];
    int X1[2];
    int X2[2];
    int Plan[2];
    int Fronteira;
    int Aresta;
    int VetorDescr[4];
    int OBJETO;
};
```

```

struct Varr{
    int Luminancia[NUMEROPONTOS];
    int LumSuav[NUMEROPONTOS];
    int FronteiraEsquerda;
    int FronteiraDireita;
    int MaxLum;
    int PosMaxLum;
    int MinLum;
    int LumFrontEsq;
    int LumFrontDir;
};

/* Banco de Conhecimento */
int Objeto_1P[4][4] = { { 1,4,15,2},
                        { 4,15,2,1},
                        { 15,2,1,4},
                        { 2,1,4,15} };

int Objeto_2P[5][4]={1,1,1,1},
                    {15,15,15,15},
                    {2,2,0,0}, /* opções de treinamento */
                    {2,15,0,0}};

int Objeto_1PD[10][4]={1,5,1,5},
                      {5,1,5,1},
                      {11,12,11,12},
                      {12,11,12,11},
                      {11,2,0,0}, /* opções de treinamento */
                      {5,2,0,0},
                      {11,11,0,0},
                      {13,11,0,0},
                      {13,11,0,0},
                      {5,15,0,0}};

int Objeto_1C[8][4]={5,8,13,6},
                    {8,13,6,5},
                    {13,6,5,8},
                    {6,5,8,13},
                    {6,8,0,0}, /* opções de treinamento */
                    {8,8,0,0},
                    {6,12,0,0},
                    {8,5,0,0}};

int Objeto_2C[2][4]={5,5,5,5},
                    {13,13,13,13}};

int Objeto_2CD[24][4]={1,8,16,6},
                      {8,16,6,1},
                      {16,6,1,8},
                      {6,1,8,16},
                      {2,5,17,12},
                      {5,17,12,2},
                      {17,12,2,5},
                      {12,2,5,17},
                      {4,11,16,5},
                      {11,16,5,4},

```


Tecnologia para o Reconhecimento do Formato de Objetos Tri-dimensionais.

{16,5,4,11},
{5,4,11,16},
{11,13,14,12},
{13,14,12,11},
{14,12,11,13},
{12,11,13,14},
{12,2,13,17},
{2,13,17,12},
{13,17,12,2},
{17,12,2,13},
{8,16,5,4},
{16,5,4,8},
{5,4,8,16},
{4,8,16,5}};

IMG5.C

```
#include <math.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <process.h>
#include "img5.h"

union coordenada_t da_x;
union coordenada_t da_y;
int posicaomesa;

/*                                     */
/*                                     */
/* Prototypes                         */
/*                                     */
/*                                     */

void Suaviza(struct Varr *Varre);
int Le_AD16(void);
void InicializaEspelhos(void);
void InicializaMesa(void);
void NPassos(int np, int dir, int tpms);
int Varredura( int PassoX, int PassoY,
               int InicioX, int FimX,
               int InicioY, int FimY,
               int *Vetor, int MaxLeit);
int VarreY( int PassoY,
            int InicioX,
            int *Vetor, int MinL);
int AchaLimitesVetor(int TamanhoVetor, struct Varr *Varre);
int Aquisicao(void);
void DetPosMaxMinLum(struct Varr *Varre);
void DetParamClass(struct Varr *Varre, struct Class *Classif, int
NroVarr);
void ClassificaFronteira(struct Class *Classif);
void AchaLumFront(struct Varr *Varre);
void AjustaParam(struct Class *Classif);
void ClassificaAresta(struct Class *Classif);
void MontaVetorDescr(int Vista, struct Class *Classif);
void RecObj_1P(struct Class *Classif);
void RecObj_2P(struct Class *Classif);
void RecObj_1PD(struct Class *Classif);
void RecObj_1C(struct Class *Classif);
void RecObj_2C(struct Class *Classif);
void RecObj_2CD(struct Class *Classif);
void OBJETO_1P(void);
void OBJETO_1PD(void);
void OBJETO_2P(void);
void OBJETO_1C(void);
void OBJETO_2C(void);
void OBJETO_2CD(void);
```

```
void main(void)
{

    static struct Varr Varre;
    static struct Class Classif;
    int NroVarr;
    int MeioObjeto;
    int Vista = 0;

    int I;
    int Vota;
    int Inclinaçao[3];
    int PosY;
    int contvezes;

    /*
    *
    *   Inicialização dos conversores DA
    *
    */

    InicializaEspelhos();

    /*
    *
    *   Inicialização da mesa giratoria
    *
    */

    printf("\nIniciando a mesa giratoria");
    InicializaMesa() ;
    printf("\nMesa na posicao zero") ;

                                /* Processo "Bottom-up" */

    for(Vista=0;Vista<2;Vista++){

        if(Vista==1)
            NPassos(148,0,10);    /* Gira a mesa de 90 graus
                                   para a segunda vista do
                                   objeto.    */

    /*=====*/
    /*
    /*   Realização do processo de Votação:
    /*
    /*   Realização de duas varreduras horizontais e teste dos
    /*   parâmetros de classificação:
    /*   Se iguais --- ok
    /*   Se diferentes --- outra varredura
    /*
    /*=====*/
```

```
NroVarr = 0;
PosY=0x7ff/10;
contvezes = 0;

for(Vota=0;Vota<2;Vota++){          /* Realiza duas Varreduras
                                      Inferiores */

/*
* Varredura Horizontal Inferior
*/

Varredura( PASSOVARREDURA,0,
           0, 0x7ff,
           PosY, 0x7ff,
           Varre.Luminancia, NUMEROPONTOS);
AchaLimitesVetor(NUMEROPONTOS, &Varre);
Suaviza(&Varre);
DetPosMaxMinLum(&Varre);
AchaLumFront(&Varre);
DetParamClass(&Varre,&Classif,NroVarr);
Inclinacao[Vota]=Classif.Plan[NroVarr];
PosY += 5*PASSOVARREDURA;
    }

if(Inclinacao[0] != Inclinacao[1]){    /* Realiza a terceira
                                      varredura */
    Vota = 2;
    PosY += 5*PASSOVARREDURA;
    Varredura( PASSOVARREDURA,0,
               0, 0x7ff,
               PosY, 0x7ff,
               Varre.Luminancia, NUMEROPONTOS);
    AchaLimitesVetor(NUMEROPONTOS, &Varre);
    Suaviza(&Varre);
    DetPosMaxMinLum(&Varre);
    AchaLumFront(&Varre);
    DetParamClass(&Varre,&Classif,NroVarr);
    Inclinacao[Vota]=Classif.Plan[NroVarr];
    for(Vota=0;Vota<2;Vota++){
        if(Inclinacao[2] == Inclinacao[Vota])

Classif.Plan[NroVarr]=Inclinacao[Vota];
        else contvezes++;
    }
    }

    if(contvezes == 2)
        printf("\nErro na varredura : %d",NroVarr);

/*
* Varredura Vertical
*/

MeioObjeto = (Varre.FronteiraDireita +
               Varre.FronteiraEsquerda)/2;
```

```
Varre.FronteiraDireita = VarreY( PASSOVARREDURA,
                                   MeioObjeto*PASSOVARREDURA,
                                   Varre.Luminancia, MINLUM);

/*
 * Varredura Horizontal Superior
 */

NroVarr = 1;
PosY = (Varre.FronteiraDireita-20)*PASSOVARREDURA;
contvezes = 0;

for(Vota=0;Vota<2;Vota++){          /* Realiza duas Varreduras
                                     Superiores */

Varredura( PASSOVARREDURA, 0,
           0, 0x7ff,
           PosY, 0x7ff,
           Varre.Luminancia, NUMEROPONTOS);
AchaLimitesVetor(NUMEROPONTOS, &Varre);
Suaviza(&Varre);
DetPosMaxMinLum(&Varre);
AchaLumFront(&Varre);
DetParamClass(&Varre,&Classif,NroVarr);
Inclinacao[Vota]=Classif.Plan[NroVarr];
PosY -= 5*PASSOVARREDURA;
    }
if(Inclinacao[0] != Inclinacao[1]){    /* Realiza a terceira
                                     varredura */
    Vota = 2;
    PosY -= 5*PASSOVARREDURA;
    Varredura( PASSOVARREDURA,0,
              0, 0x7ff,
              PosY, 0x7ff,
              Varre.Luminancia, NUMEROPONTOS);
    AchaLimitesVetor(NUMEROPONTOS, &Varre);
    Suaviza(&Varre);
    DetPosMaxMinLum(&Varre);
    AchaLumFront(&Varre);
    DetParamClass(&Varre,&Classif,NroVarr);
    Inclinacao[Vota]=Classif.Plan[NroVarr];
    for(Vota=0;Vota<2;Vota++){
        if(Inclinacao[2] == Inclinacao[Vota])

Classif.Plan[NroVarr]=Inclinacao[Vota];
        else contvezes++;
    }
}

if(contvezes == 2)
    printf("\nErro na varredura : %d",NroVarr);
```

```
/*
 * Classificação da Fronteira e Aresta do objeto
 */
ClassificaFronteira(&Classif);
if(Classif.Plan[0]==0)
    ClassificaAresta(&Classif);
/*
 * Descrição do objeto
 */
MontaVetorDescr(Vista,&Classif);
    }    /* Fim do loop de "for" para duas
          vistas */

/*
 * Terceira Vista do objeto. Caso único (objeto 1PD ou 2CD)
 */
if((Classif.VetorDescr[0]==12) && (Classif.VetorDescr[1]==11))
{
    Vista++;
    NPassos(148,0,10);    /* Gira a mesa de 90 graus para a
                           terceira vista do objeto.    */

/*=====*/
/*
/*      Realização do processo de Votação:
/*
/*      Realização de duas varreduras horizontais e teste dos
/*      parâmetros de classificação:
/*      Se iguais --- ok
/*      Se diferentes --- outra varredura
/*
/*=====*/

NroVarr = 0;
PosY=0x7ff/10;
contvezes = 0;

for(Vota=0;Vota<2;Vota++){    /* Realiza duas Varreduras
                               Inferiores */

/*
 * Varredura Horizontal Inferior
 */
```

```

Varredura( PASSOVARREDURA,0,
           0, 0X7ff,
           PosY, 0x7ff,
           Varre.Luminancia, NUMEROPONTOS);
AchaLimitesVetor(NUMEROPONTOS, &Varre);
Suaviza(&Varre);
DetPosMaxMinLum(&Varre);
AchaLumFront(&Varre);
DetParamClass(&Varre,&Classif,NroVarr);
Inclinacao[Vota]=Classif.Plan[NroVarr];
PosY += 5*PASSOVARREDURA;
    }
if(Inclinacao[0] != Inclinacao[1]){      /* Realiza a terceira
                                           varredura */
    Vota = 2;
    PosY += 5*PASSOVARREDURA;
    Varredura( PASSOVARREDURA,0,
               0, 0X7ff,
               PosY, 0x7ff,
               Varre.Luminancia, NUMEROPONTOS);
    AchaLimitesVetor(NUMEROPONTOS, &Varre);
    Suaviza(&Varre);
    DetPosMaxMinLum(&Varre);
    AchaLumFront(&Varre);
    DetParamClass(&Varre,&Classif,NroVarr);
    Inclinacao[Vota]=Classif.Plan[NroVarr];
    for(Vota=0;Vota<2;Vota++){
        if(Inclinacao[2] == Inclinacao[Vota])
Classif.Plan[NroVarr]=Inclinacao[Vota];
        else contvezes++;
    }
    }
    if(contvezes == 2)
        printf("\nErro na varredura : %d",NroVarr);

/*
 * Varredura Vertical
 */

MeioObjeto = (Varre.FronteiraDireita +
              Varre.FronteiraEsquerda)/2;

Varre.FronteiraDireita = VarreY( PASSOVARREDURA,
                                MeioObjeto*PASSOVARREDURA,
                                Varre.Luminancia, MINLUM);

/*
 * Varredura Horizontal Superior
 */

```

```
NroVarr = 1;
PosY = (Varre.FronteiraDireita-20)*PASSOVARREDURA;
contvezes = 0;

for(Vota=0;Vota<2;Vota++){          /* Realiza duas Varreduras
                                     Superiores */

Varredura( PASSOVARREDURA, 0,
           0, 0x7ff,
           PosY, 0x7ff,
           Varre.Luminancia, NUMEROPONTOS);
AchaLimitesVetor(NUMEROPONTOS, &Varre);
Suaviza(&Varre);
DetPosMaxMinLum(&Varre);
AchaLumFront(&Varre);
DetParamClass(&Varre,&Classif,NroVarr);
Inclinacao[Vota]=Classif.Plan[NroVarr];
PosY -= 5*PASSOVARREDURA;
    }
if(Inclinacao[0] != Inclinacao[1]){    /* Realiza a terceira
                                     varredura */
    Vota = 2;
    PosY -= 5*PASSOVARREDURA;
    Varredura( PASSOVARREDURA,0,
               0, 0x7ff,
               PosY, 0x7ff,
               Varre.Luminancia, NUMEROPONTOS);
    AchaLimitesVetor(NUMEROPONTOS, &Varre);
    Suaviza(&Varre);
    DetPosMaxMinLum(&Varre);
    AchaLumFront(&Varre);
    DetParamClass(&Varre,&Classif,NroVarr);
    Inclinacao[Vota]=Classif.Plan[NroVarr];
    for(Vota=0;Vota<2;Vota++){
        if(Inclinacao[2] == Inclinacao[Vota])
Classif.Plan[NroVarr]=Inclinacao[Vota];
        else contvezes++;
    }
    }

if(contvezes == 2)
    printf("\nErro na varredura : %d",NroVarr);

/*
 * Classificação da Fronteira e Aresta do objeto
 */

ClassificaFronteira(&Classif);

if(Classif.Plan[0]==0)
    ClassificaAresta(&Classif);
```



```
/*
 * Descrição do objeto
 */

MontaVetorDescr(Vista,&Classif);

/*Fim da terceira vista */

}

/* Processo "Top-down" */

switch(Classif.VetorDescr[0]){
    case 1:
        RecObj_1P(&Classif);
        RecObj_2P(&Classif);
        RecObj_1PD(&Classif);
        RecObj_2CD(&Classif);
        break;

    case 2:
        RecObj_1P(&Classif);
        RecObj_2CD(&Classif);
        RecObj_2P(&Classif);
        break;

    case 4:
        RecObj_1P(&Classif);
        RecObj_2CD(&Classif);
        break;

    case 5:
        RecObj_1PD(&Classif);
        RecObj_2C(&Classif);
        RecObj_2CD(&Classif);
        RecObj_1C(&Classif);
        break;

    case 6:
    case 8:
        RecObj_1C(&Classif);
        RecObj_2CD(&Classif);
        break;

    case 11:
    case 12:
        RecObj_1PD(&Classif);
        RecObj_2CD(&Classif);
        break;
```

```
        case 13:
            RecObj_1C(&Classif);
            RecObj_2C(&Classif);
            RecObj_2CD(&Classif);
            RecObj_1PD(&Classif);
            break;

        case 14:
        case 16:
        case 17:
            RecObj_2CD(&Classif);
            break;

        case 15:
            RecObj_1P(&Classif);
            RecObj_2P(&Classif);
            break;

    }

switch(Classif.OBJETO){
    case 1:
        OBJETO_1P();
        break;

    case 2:
        OBJETO_2P();
        break;

    case 3:
        OBJETO_1PD();
        break;

    case 4:
        OBJETO_1C();
        break;

    case 5:
        OBJETO_2C();
        break;

    case 6:
        OBJETO_2CD();
        break;

    case 0:
        printf("\nObjeto nao classificado!...");
        break;

}

}
```

```
/*-----*/
/*
/*  RecObj_1P: Reconhece Objeto 1P.
/*  Classif->OBJETO = 1 ---- Código para Objeto 1P.
/*
/*-----*/

void RecObj_1P(struct Class *Classif)
{
    int l;
    for(l=0;l<4;l++){
        if((Classif->VetorDescr[0]==Objeto_1P[l][0]) &&
            (Classif->VetorDescr[1]==Objeto_1P[l][1]))
            Classif->OBJETO = 1;    /* Código para
                                    Objeto 1P */
    }
    return;
}

/*-----*/
/*
/*  RecObj_2P: Reconhece Objeto 2P.
/*  Classif->OBJETO = 2 ---- Código para Objeto 2P.
/*
/*-----*/

void RecObj_2P(struct Class *Classif)
{
    int l;
    for(l=0;l<5;l++){
        if((Classif->VetorDescr[0]==Objeto_2P[l][0]) &&
            (Classif->VetorDescr[1]==Objeto_2P[l][1]))
            Classif->OBJETO = 2;    /* Código para
                                    Objeto 2P */
    }
    return;
}

/*-----*/
/*
/*  RecObj_1PD: Reconhece Objeto 1PD.
/*  Classif->OBJETO = 3 ---- Código para Objeto 1PD.
/*
/*-----*/

void RecObj_1PD(struct Class *Classif)
{
    int l;
    for (l=0;l<10;l++){
        if((Classif->VetorDescr[0]==Objeto_1PD[l][0]) &&
            (Classif->VetorDescr[1]==Objeto_1PD[l][1]))
            Classif->OBJETO = 3;    /* Código para
                                    Objeto 1PD */
    }
    return;
}
```

```
/*-----*/
/*
/*  RecObj_1C: Reconhece Objeto 1C.
/*  Classif->OBJETO = 4 ---- Código para Objeto 1C.
/*
/*-----*/

void RecObj_1C(struct Class *Classif)
{
    int l;
    for(l=0;l<8;l++){
        if((Classif->VetorDescr[0]==Objeto_1C[l][0]) &&
            (Classif->VetorDescr[1]==Objeto_1C[l][1]))
            Classif->OBJETO = 4;    /* Código para
                                    Objeto 1C */
    }

    return;
}

/*-----*/
/*
/*  RecObj_2C: Reconhece Objeto 2C.
/*  Classif->OBJETO = 5 ---- Código para Objeto 2C.
/*
/*-----*/

void RecObj_2C(struct Class *Classif)
{
    int l;
    for(l=0;l<2;l++){
        if((Classif->VetorDescr[0]==Objeto_2C[l][0]) &&
            (Classif->VetorDescr[1]==Objeto_2C[l][1]))
            Classif->OBJETO = 5;    /* Código para
                                    Objeto 2C */
    }

    return;
}

/*-----*/
/*
/*  RecObj_2CD: Reconhece Objeto 2CD.
/*  Classif->OBJETO = 6 ---- Código para Objeto 2CD.
/*
/*-----*/

void RecObj_2CD(struct Class *Classif)
{
    int l;
    for(l=0;l<24;l++){
        if((Classif->VetorDescr[0]==Objeto_2CD[l][0]) &&
            (Classif->VetorDescr[1]==Objeto_2CD[l][1]))
            Classif->OBJETO = 6;    /* Código para
                                    Objeto 2CD */
    }

    return;
}
```

```
/*-----*/
/*
/* MontaVetorDescr: Verifica os parâmetros de
/* classificação e monta o Vetor de Descrição do Objeto
/*
/*-----*/
```

```
void MontaVetorDescr(int Vista, struct Class *Classif)
```

```
{
if(Classif->Frenteira==0){
    if(Classif->Plan[0]==1)
        Classif->VetorDescr[Vista]=1;    /* Nó 1 */

    else if(Classif->Plan[0]==2)
        Classif->VetorDescr[Vista]=2;    /* Nó 2 */

    else if(Classif->Plan[0]==3)
        Classif->VetorDescr[Vista]=4;    /* Nó 4 */

    else
        Classif->VetorDescr[Vista]=Classif->Aresta;
                                           /* Nó 3 */
    }
else if(Classif->Frenteira==1){
    if(Classif->Plan[0]==1)
        Classif->VetorDescr[Vista]=5;    /* Nó 5 */

    else if(Classif->Plan[0]==2)
        Classif->VetorDescr[Vista]=6;    /* Nó 6 */

    else if(Classif->Plan[0]==3)
        Classif->VetorDescr[Vista]=8;    /* Nó 8 */

    else
        Classif->VetorDescr[Vista]=Classif->Aresta;
                                           /* Nó 7 */
    }
return;
}
```

```
/*-----*/
/*
/* AjustaParam: Ajusta os parâmetros X1, X2, X3, X4 de
/* classificação das arestas para erro inferior a 5 pontos.
/*
/*-----*/
```

```
void AjustaParam(struct Class *Classif)
```

```
{
if(abs(Classif->X1[0] - Classif->X1[1]) < 20 )
    Classif->X1[1] = Classif->X1[0];
else return;
}
```

```
/*-----*/
/*
/* ClassificaAresta: Verifica os parâmetros de
/* classificação e guarda em Aresta o código para a vista
/* alc --- código 11      a3p --- código 15
/* a2c --- código 12      a4p --- código 16
/* a4c --- código 13      a5p --- código 17
/* a5c --- código 14      código 10 --- erro
/*
/*-----*/
```

```
void ClassificaAresta(struct Class *Classif)
{
if(Classif->Fronteira == 1){
    if((abs(Classif->X1[0] - Classif->X1[1]) < 10))
        Classif->Aresta = 11;

    else if((abs(Classif->X2[0] -
        Classif->X2[1])<10))
        Classif->Aresta = 12;

    else if((Classif->X1[0] > Classif->X1[1]))
        Classif->Aresta = 13;

    else if((Classif->X1[0] < Classif->X1[1]) &&
        (Classif->X2[0] > Classif->X2[1]))
        Classif->Aresta = 14;
    }
else if(Classif->Fronteira == 0){
    if((Classif->X1[0] < 10) &&
        (Classif->X2[1] < 10))
        Classif->Aresta = 17;

    else if((Classif->X1[1] < 10) &&
        (Classif->X2[0] < 10))
        Classif->Aresta = 16;

    else Classif->Aresta = 15;
}

return;
}
```

```
/*-----*/
/*
/* DetParamClass : Calcula os parâmetros de cada vista do
/* objeto.
/* Determina:
/* Plan[NroVarr] = 0 ---- Existe aresta interna
/* Plan[NroVarr] = 1 ---- Face Plana
/* Plan[NroVarr] = 2 ---- Inclinação Positiva
/* Plan[NroVarr] = 3 ---- Inclinação Negativa
/*
/*-----*/
```

```
void DetParamClass(struct Varr *Varre, struct Class *Classif,
int NroVarr)
{
    int media;
    int desviomedio = 0;

    Classif->Tamanho[NroVarr] = Varre->FronteiraDireita -
                                Varre->FronteiraEsquerda;
    Classif->X1[NroVarr] = Varre->PosMaxLum -
                           Varre->FronteiraEsquerda;
    Classif->X2[NroVarr] = Varre->FronteiraDireita -
                           Varre->PosMaxLum ;
    media = (Varre->LumFrontEsq + Varre->LumFrontDir +
              Varre->MaxLum)/3;
    desviomedio += (abs(media-Varre->LumFrontEsq) +
                    abs(media-Varre->LumFrontDir)
                    + abs(media - Varre->MaxLum))/3;

    if(desviomedio < 100)
        Classif->Plan[NroVarr] = 1;          /* Face Plana */

        else if(Classif->X1[NroVarr] < 15)
            Classif->Plan[NroVarr] = 3; /* Inclinação Negativa */

        else if(Classif->X2[NroVarr] < 15)
            Classif->Plan[NroVarr] = 2; /* Inclinação Positiva */

        else Classif->Plan[NroVarr] = 0;
                                           /* Existe aresta */

    return;
}

/*-----*/
/*
/* ClassificaFronteira: Determina o Formato da Fronteira
/* do objeto depois de duas varreduras.
/* Classif->Fronteira = 0 --- Fronteira Paralela
/* Classif->Fronteira = 1 --- Fronteira Convergente
/*
/*-----*/

void ClassificaFronteira(struct Class *Classif)
{
    if(Classif->Tamanho[0] > Classif->Tamanho[1]+15){
        Classif->Fronteira = 1; /* Fronteira Convergente FC */
    }
    else{
        Classif->Fronteira = 0; /* Fronteira Paralela FP */
    }
    return;
}
```

```
/*-----*/
/*
/*  DetPosMaxMinLum:  Varre o objeto entre as bordas,
/*  e encontra o ponto em que a luminância é máxima e
/*  mínima.
/*
/*
/*-----*/
```

```
void DetPosMaxMinLum(struct Varr *Varre)
{
    int i;
    int Max = 0, Min = 0xffff;
    int PosMax;

    for(i = Varre->FronteiraEsquerda;
        i < Varre->FronteiraDireita; i++){
        if( Varre->LumSuav[i] > Max+5 ){
            Max = Varre->LumSuav[i] ;
            PosMax = i ;
        }
        if( Varre->LumSuav[i] < Min ){
            Min = Varre->LumSuav[i];
        }
    }
    Varre->PosMaxLum = PosMax;
    Varre->MaxLum = Max;
    Varre->MinLum = Min;
    return;
}
```

```
/*-----*/
/*
/*  AchaLumFront: calcula a luminancia das bordas do
/*  objeto
/*
/*
/*-----*/
```

```
void AchaLumFront(struct Varr *Varre)
{
    Varre->LumFrontEsq =
        Varre->LumSuav[Varre->FronteiraEsquerda];
    Varre->LumFrontDir =
        Varre->LumSuav[Varre->FronteiraDireita];
    return;
}
```



```
/*-----*/
/*
/*  Suaviza: Aplicar somente depois de encontradas as
/*  fronteiras do objeto.
/*
/*-----*/
```

```
void Suaviza(struct Varr *Varre)
{
```

```
    int i;

    for( i = 0; i < NUMEROPONTOS ; i++){
        Varre->LumSuav[i] = 0;
    }
    Varre->LumSuav[Varre->FronteiraEsquerda] =
        (Varre->Luminancia[Varre->FronteiraEsquerda]+
         Varre->Luminancia[Varre->FronteiraEsquerda+1])/2;
    for(i = Varre->FronteiraEsquerda+1; i <
Varre->FronteiraDireita ; i++){
        Varre->LumSuav[i] =
        (Varre->Luminancia[i-1]+Varre->Luminancia[i]+
         Varre->Luminancia[i+1] )/3;
    }
    Varre->LumSuav[Varre->FronteiraDireita] =
        (Varre->Luminancia[Varre->FronteiraDireita]+
         Varre->Luminancia[Varre->FronteiraDireita-1])/2;
}
```

```
/*-----*/
/*
/*  AchaLimitesVetor:
/*
/*  Esta rotina analisa um vetor contendo os dados
/*  da luminância e encontra limites do objeto.
/*
/*
/*  Argumentos de entrada:
/*  - ponteiro para o vetor.
/*  - tamanho do vetor.
/*  - ponteiro para estrutura que conterà os limites.
/*  Valor de retorno:
/*  - diferente de zero se os limites foram encontrados.
/*
/*-----*/
```

```
int AchaLimitesVetor(int TamanhoVetor, struct Varr *Varre)
{
```

```
    int situacao = 1;
    int contador = 0;
```

```
    do{
```

```
    if( (Varre->Luminancia[contador] > MINLUM) &&
        ( Varre->Luminancia[contador+1] >MINLUM) ){
        situacao = 0;
    }
    contador++;
}while( (situacao) && (contador < TamanhoVetor)) ;
if(situacao)
    return(situacao);

Varre->FronteiraEsquerda = contador+3;
situacao = 1;
contador = TamanhoVetor;
do{

    if( (Varre->Luminancia[contador] > MINLUM) &&
        ( Varre->Luminancia[contador-1] > MINLUM)){
        situacao = 0;
    }
    contador--;
}while( (situacao) && (contador >0));
Varre->FronteiraDireita = contador - 3;
return(1);
}
```

```
/*-----*/
/*
/*      Varredura:
/*
/*      Esta rotina realiza uma varredura horizontal e/ou
/*      vertical, armazenando o valor do conversor ad no
/*      buffer Vetor.
/*      Argumentos de entrada:
/*      - valor do passo na direcao x e y;
/*      - limites da varredura (de 0 a 0x7ff).
/*      Valor de retorno:
/*      - o numero de dados coletados.
/*
/*
/*-----*/
```

```
int Varredura( int PassoX, int PassoY,
               int InicioX, int FimX,
               int InicioY, int FimY,
               int *Vetor, int MaxLeit)
{
    int NumDados = 0;

    while( (InicioX < FimX) && (InicioY <
        FimY) && (NumDados<MaxLeit)){
        Deflete(InicioX, InicioY);
```

```
    *Vetor = Aquisicao();
    InicioX+= PassoX;
    InicioY+= PassoY;
    NumDados++;
    Vetor++;
}
return(NumDados);
}
```

```
/*-----*/
/*
/*      VarreY:
/*
/*      Esta rotina realiza uma varredura
/*      vertical, armazenando o valor do conversor ad no
/*      buffer Vetor.
/*      Argumentos de entrada:
/*      - valor do passo na direcao      y;
/*      Valor de retorno:
/*      - o numero de dados coletados.
/*
/*
/*-----*/
```

```
int VarreY( int PassoY,
            int InicioX,
            int *Vetor, int MinL)
{
    int NumDados = 0;
    int InicioY = 0;
    int FimVarredura;

    Deflete(InicioX, InicioY);
    FimVarredura = Le_AD16();

    while((InicioY < 0x7ff) && (FimVarredura > MinL)){
        Deflete(InicioX, InicioY);
        FimVarredura = Le_AD16();
        *Vetor = FimVarredura;
        InicioY += PassoY;
        NumDados++;
        Vetor++;
    }
    return(NumDados);
}
```

```
/*-----*/
/*
/*  NPassos: Faz com que a mesa gire de np passos na      */
/*  direção horária ou antihorária (dir = 0,1)            */
/*  com um tempo de tpms milisegundos entre um passo     */
/*  e outro.                                              */
/*-----*/
```

```
void NPassos(int np, int dir, int tpms)
{
    char controle[2][2] = { 5,4,7,6};
    int i;

    da_x.valor.mesa = controle[dir][0];
    outportb(MSBDAX, da_x.byte.msb);
    delay(100);
    for( i = 0 ; i < np ; ++i ){
        da_x.valor.mesa = controle[dir][0];
        outportb(MSBDAX, da_x.byte.msb);
        delay(1);
        da_x.valor.mesa = controle[dir][1];
        outportb(MSBDAX, da_x.byte.msb);
        delay(1);
        da_x.valor.mesa = controle[dir][0];
        outportb(MSBDAX, da_x.byte.msb);
        delay(tpms);
    }
    if(dir){
        posicaomesa = posicaomesa - np;
    }
    else{
        posicaomesa = posicaomesa + np;
    }
    da_x.valor.mesa = da_x.valor.mesa & 0x3;
    outportb(MSBDAX, da_x.byte.msb);
}
```

```
/*-----*/
/*
/*  InicializaMesa: Coloca a mesa giratória na marca zero. */
/*-----*/
```

```
void InicializaMesa(void)
{
    char ch ;
    char controle[2] = { 5,4};

    inportb(ES);
}
```

```
ch = inportb(SENSOR);
while( ch & 1 ){
    da_x.valor.mesa = controle[0];
    outportb(MSBDAX, da_x.byte.msb);
    delay(1);
    da_x.valor.mesa = controle[1];
    outportb(MSBDAX, da_x.byte.msb);
    delay(1);
    da_x.valor.mesa = controle[0];
    outportb(MSBDAX, da_x.byte.msb);
    delay(10);
    inportb(ES);
    ch = inportb(SENSOR);
}
da_x.valor.mesa = 3;
outportb(MSBDAX, da_x.byte.msb);
posicaomesa = 0;
}

/*-----*/
/*
/*
/* Deflete : desloca o feixe para as coordenadas x e y.
/* arg. entrada: x e y, de 0 a 0x7ff.
/* retorna 0 se coordenadas fora da faixa.
/*
/*
/*-----*/

int Deflete(int X, int Y)
{
    if( (X<0) && (X>0x7ff) && (Y<0) && (Y>0x7ff) )
        return(0);
    if( ! ((X<0) || (X>0x7ff)) ){
        da_x.valor.da = X | 0x800;
    }
    if( ! ((Y<0) || (Y>0x7ff)) ){
        da_y.valor.da = Y | 0x800;
    }

    outportb(LSBDAX, da_x.byte.lsb);
    outportb(MSBDAX, da_x.byte.msb);
    outportb(LSBDAY, da_y.byte.lsb);
    outportb(MSBDAY, da_y.byte.msb);
    outportb(GATILHO, 0);
    delay(100);
    return(1);
}
```

```
/*
 *
 *  Inicialização dos espelhos defletores
 *
 */

void InicializaEspelhos(void)
{
    da_x.byte.lsb = 0x00;
    da_x.byte.msb = 0x38;
    da_y.byte.lsb = 0x00;
    da_y.byte.msb = 0x08;

    outportb(LSBDAX, da_x.byte.lsb);
    outportb(MSBDAX, da_x.byte.msb);
    outportb(LSBDAY, da_y.byte.lsb);
    outportb(MSBDAY, da_y.byte.msb);

    outportb(GATILHO, 0);
    return;
}
```

AQUIS15.C

```
#include <stdlib.h>
#include <dos.h>

#define NROLTR 8
#define DESVIOMAX 50
#define NUMEROTENTATIVAS 16

int Le_AD16(void);

/*
/*
/*      Aquisição: realiza uma coleta junto ao conversor
/*      AD. Verifica se os dados sao coerentes.
/*      Realiza estatística no conversor.
/*
/*
/*
/*

int Aquisicao(void)
{
    static int DadosColetados[NROLTR];
    static long Soma;
    static long Media;
    int erro = 0;
    int i;
    int cont = 0;

    for(i = 0 ; i < NROLTR ; i ++ ){
        DadosColetados[i] = Le_AD16();
        delay(1);
    }
    do{
        cont++;
        erro = 0;
        Soma = 0;
        for(i = 0 ; i < NROLTR ; i ++ ){
            Soma += DadosColetados[i];
        }
        Media = Soma/NROLTR;
        for(i = 0 ; i < NROLTR ; i ++ ){
            if( ( labs(Media - DadosColetados[i] )) > DESVIOMAX){
                erro = 1;
                DadosColetados[i] = Le_AD16();
            }
        }
    }while( (erro) && (cont < NUMEROTENTATIVAS));

    return((int)Media);
}
```

SAÍDAS GRÁFICAS NO TERMINAL DE VÍDEO.

```
#include <conio.h>
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>

void OBJETO_1P(void)
{
    int graphmode, graphdriver;
    int cubo1[]={200,180,200,70,450,20,
                450,100,400,180,200,180,200,70,
                400,70,450,20,400,70,400,180};
    int cubo2[]={200,180,450,100};

    directvideo=0;

    detectgraph(&graphdriver, &graphmode);
    clrscr();

    initgraph(&graphdriver, &graphmode, "C:\\\\TC\\\\");

    setlinestyle(SOLID_LINE,0,NORM_WIDTH);

    drawpoly(sizeof(cubo1)/(2*sizeof(int)),cubo1);

    setlinestyle(DOTTED_LINE,0,NORM_WIDTH);

    drawpoly(sizeof(cubo2)/(2*sizeof(int)),cubo2);

    getch();
    closegraph();
}

#include <conio.h>
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>

void OBJETO_1PD(void)
{
    int graphmode, graphdriver;
    int cubo1[]={250,180,120,130,220,30,250,180,
                220,30,470,30,500,180,250,180};
    int cubo2[]={120,130,370,130,470,30,370,130,500,180};

    directvideo=0;

    detectgraph(&graphdriver, &graphmode);
    clrscr();

    initgraph(&graphdriver, &graphmode, "C:\\\\TC\\\\");
```



```
setlinestyle(SOLID_LINE,0,NORM_WIDTH);  
drawpoly(sizeof(cubo1)/(2*sizeof(int)),cubo1);  
setlinestyle(DOTTED_LINE,0,NORM_WIDTH);  
drawpoly(sizeof(cubo2)/(2*sizeof(int)),cubo2);  
    getch();  
    closegraph();  
}
```

```
#include <conio.h>  
#include <graphics.h>  
#include <stdlib.h>  
#include <stdio.h>
```

```
void OBJETO_2P(void)  
{  
    int graphmode, graphdriver;  
    int cubo1[]={200,180,200,70,300,20,500,20,  
                500,100,400,180,200,180,200,70,  
                400,70,500,20,400,70,400,180};  
    int cubo2[]={200,180,300,100,300,20,300,100,500,100};  
  
    directvideo=0;  
  
    detectgraph(&graphdriver, &graphmode);  
    clrscr();  
  
    initgraph(&graphdriver, &graphmode, "C:\\\\TC\\\\");  
  
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);  
  
    drawpoly(sizeof(cubo1)/(2*sizeof(int)),cubo1);  
  
    setlinestyle(DOTTED_LINE,0,NORM_WIDTH);  
  
    drawpoly(sizeof(cubo2)/(2*sizeof(int)),cubo2);  
  
    getch();  
    closegraph();  
}
```

```
#include <conio.h>
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>

void OBJETO_1C(void)
{
    int graphmode, graphdriver;
    int cubo1[]={350,180,200,130,350,20,500,130,
                350,180,350,20,350,180};
    int cubo2[]={200,130,500,130};

    directvideo=0;

    detectgraph(&graphdriver, &graphmode);
    clrscr();

    initgraph(&graphdriver, &graphmode, "C:\\\\TC\\\\");

    setlinestyle(SOLID_LINE,0,NORM_WIDTH);

    drawpoly(sizeof(cubo1)/(2*sizeof(int)),cubo1);

    setlinestyle(DOTTED_LINE,0,NORM_WIDTH);

    drawpoly(sizeof(cubo2)/(2*sizeof(int)),cubo2);

    getch();
    closegraph();
}
```

```
#include <conio.h>
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>

void OBJETO_2C(void)
{
    int graphmode, graphdriver;
    int cubo1[]={350,180,200,130,350,20,500,130,
                350,180,350,20,350,180};
    int cubo2[]={200,130,360,90,350,20,360,90,500,130};

    directvideo=0;

    detectgraph(&graphdriver, &graphmode);
    clrscr();

    initgraph(&graphdriver, &graphmode, "C:\\\\TC\\\\");

    setlinestyle(SOLID_LINE,0,NORM_WIDTH);

    drawpoly(sizeof(cubo1)/(2*sizeof(int)),cubo1);
```

```
setlinestyle(DOTTED_LINE,0,NORM_WIDTH);  
  
drawpoly(sizeof(cubo2)/(2*sizeof(int)),cubo2);  
  
    getch();  
    closegraph();  
}
```

```
#include <conio.h>  
#include <graphics.h>  
#include <stdlib.h>  
#include <stdio.h>  
  
void OBJETO_2CD(void)  
{  
    int graphmode, graphdriver;  
    int cubo1[]={400,190,200,150,200,45,400,70,  
400,190,500,20,400,70,200,45,500,20,400,190,500,20};  
    int cubo2[]={200,150,500,20};  
  
    directvideo=0;  
  
    detectgraph(&graphdriver, &graphmode);  
    clrscr();  
  
    initgraph(&graphdriver, &graphmode, "C:\\\\TC\\\\");  
  
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);  
  
    drawpoly(sizeof(cubo1)/(2*sizeof(int)),cubo1);  
  
    setlinestyle(DOTTED_LINE,0,NORM_WIDTH);  
  
    drawpoly(sizeof(cubo2)/(2*sizeof(int)),cubo2);  
  
    getch();  
    closegraph();  
}
```

APÊNDICE B-4

PROGRAMAS DE AQUISIÇÃO DE DADOS.

Estes programas escritos em linguagem C (TURBO C) realizam a aquisição de dados, isto é, digitalizam o sinal recebido pelo foto-transistor e armazenam o resultado em um arquivo apropriado.

O primeiro programa denominado CNIVEL.C faz uma varredura completa no objeto horizontalmente da esquerda para a direita e verticalmente de baixo para cima. O arquivo é gerado no padrão do programa ENERGRAPH que faz o desenho das curvas de superfície mostradas no Apêndice C-1.

O segundo programa denominado VLOTUS.C faz apenas uma varredura horizontal gerando o "arranjo de intensidades" no padrão do programa LOTUS 123 que é aceito também pelo programa MATHCAD. Através deste último, é possível então, a partir de dados reais adquiridos pelo VLOTUS, traçar as curvas resultantes que aparecem no Capítulo 5. O tratamento matemático dado pelo MATHCAD é o mesmo utilizado pelo programa principal de reconhecimento.

PROGRAMA CNIVEL.C

```
#include <conio.h>
#include <sys\stat.h>
#include <io.h>
#include <fcntl.h>
#include <stdio.h>
#include <dos.h>
#include "img.h"
#include <process.h>

union coordenada_t da_x;
union coordenada_t da_y;

int Varredura( int PassoX, int PassoY,
               int InicioX, int FimX,
               int InicioY, int FimY,
               int *Vetor, int MaxLeit);

void InicializaEspelhos(void);

void main()
{
    char nome_arq[30];
    int i,j,k;
    int descritor;
    FILE *stream;
    int Varred[50];

    InicializaEspelhos();
    cprintf("Entre com o nome do arquivo: ");
    i = -1;
    do{
        i++;
        nome_arq[i] = getche();
    }while(nome_arq[i] != 0xD);
    nome_arq[i] = 0;
    descritor = open(nome_arq,O_CREAT,S_IWRITE);
    stream = fdopen(descritor,"w");
    stream->flags = _F_RDWR;
    if(stream == NULL)
        cprintf("\nFalha na abertura de arquivo");
    else{
        fprintf(stream,"teste\n01-01-90\nx\ny\nz\n50,50,0,1,0,1\n4,4,4,1
06,66\n");
        fprintf(stream,"45,45,3,0\n");
        while(kbhit())
            getch();
        for( j =0 ; j < 50 ; j++){
            if(kbhit()){
                fclose(stream);
                exit(0);
            }
        }
    }
}
```

```

printf("\nVarred nro: %d",j);
Varredura( PASSOVARREDURA,0,
          0, 0X7ff,
          j * PASSOVARREDURA, 0x7ff,
          Varred, 50);
for(k = 0; k < 50 ; k++){
    fprintf(stream,"%4f\n", (float)Varred[k]/(float)0xffff);
}
}
fclose(stream);
}
}
/*-----*/
/*
/*      Varredura:
/*
/*      Esta rotina realiza uma varredura horizontal e/ou
/*      vertical, armazenando o valor do conversor a/d no
/*      " buffer " Vetor.
/*      Argumentos de entrada:
/*      - valor do passo na direção x e y;
/*      - limites da varredura (de 0 a 0xffff).
/*      Valor de retorno:
/*      - o numero de dados coletados.
/*-----*/

int Varredura( int PassoX, int PassoY,
              int InicioX, int FimX,
              int InicioY, int FimY,
              int *Vetor, int MaxLeit)
{
    int NumDados = 0;

    while( (InicioX < FimX) && (InicioY <
FimY)&&(NumDados<MaxLeit)){
        Deflete(InicioX, InicioY);
        *Vetor = Aquisicao();
        InicioX+= PassoX;
        InicioY+= PassoY;
        NumDados++;
        Vetor++;
    }
    return(NumDados);
}

```

PROGRAMA VLOTUS.C

```
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <dos.h>
#include "img.h"

union coordenada_t da_x;
union coordenada_t da_y;

int Varredura( int PassoX, int PassoY,
               int InicioX, int FimX,
               int InicioY, int FimY,
               int *Vetor, int MaxLeit);

void InicializaEspelhos(void);

void main(void)
{
    char nome_arq[30];
    int k;
    FILE *stream;
    int Varred[50];
    InicializaEspelhos();
    cprintf("Entre com o nome do arquivo: ");
    cscanf("%s", nome_arq);
    stream = fopen(nome_arq, "w+");
    if(stream == NULL)
        cprintf("\nFalha na abertura de arquivo");
    else{
        Varredura( PASSOVARREDURA, 0,
                   0, 0x7fff,
                   0x7fff/2, 0x7fff,
                   Varred, 50);
        for(k = 0; k < 50; k++){
            fprintf(stream, "%5d %3d\n", Varred[k], k+1);
        }
        fclose(stream);
    }
}

/*-----*/
/*
/*      Varredura:
/*
/*      Esta rotina realiza uma varredura horizontal e/ou
/*      vertical, armazenando o valor do conversor A/D no
/*      " buffer " Vetor.
/*      Argumentos de entrada:
/*      - valor do passo na direção x e y;
/*      - limites da varredura (de 0 a 0xffff).
/*      Valor de retorno:
/*      - o numero de dados coletados.
/*-----*/
```

```
int Varredura( int PassoX, int PassoY,  
               int InicioX, int FimX,  
               int InicioY, int FimY,  
               int *Vetor, int MaxLeit)  
{  
    int NumDados = 0;  
    while( (InicioX < FimX) && (InicioY <  
FimY)&&(NumDados<MaxLeit)){  
        Deflete(InicioX, InicioY);  
        *Vetor = Aquisicao();  
        InicioX+= PassoX;  
        InicioY+= PassoY;  
        NumDados++;  
        Vetor++;  
    }  
    return(NumDados);  
},
```


APÊNDICE C

OBJETOS RECONHECÍVEIS :

C-1 ----- CURVAS DE SUPERFÍCIES

APÊNDICE C-1

CURVAS DE SUPERFÍCIE DOS OBJETOS.

Folha 1	-----	OBJETOS 1P E 1PD (PRISMA DE BASE TRIANGULAR).
Folha 2	-----	OBJETO 2P (CUBO).
Folha 3	-----	OBJETO 1C (PIRÂMIDE DE BASE TRIANGULAR).
Folha 4	-----	OBJETO 2C (PIRÂMIDE DE BASE RETANGULAR).
Folha 5	-----	OBJETO 2CD (PIRÂMIDE DE BASE RETANGULAR).
Folha 6	-----	OBJETOS DE SUPERFÍCIE CURVA. (CILINDRO E CONE)

OBJETO 1P - Prisma de base triangular com aresta visível ao observador.

OBJETO 1PD - Prisma de base triangular apoiado em uma das faces e perpendicular ao observador.

OBJETO 2P - Cubo com face plana voltada para o observador.

OBJETO 2P - Cubo com uma aresta visível ao observador.

OBJETO 1C - Pirâmide de base triangular com face plana voltada para o observador.

OBJETO 1C - Pirâmide de base triangular com aresta visível ao observador.

OBJETO 2C - Pirâmide de base retangular assentada sobre a base e com face perpendicular ao eixo de visão.

OBJETO 2C - Pirâmide de base retangular assentada sobre a base e com aresta visível ao observador.

OBJETO 2CD - Pirâmide de base retangular apoiada sobre uma de suas faces.

OBJETO 2CD - Pirâmide de base retangular apoiada sobre uma de suas faces.

OBJETO DE SUPERFÍCIE CURVA - CONE - apoiado sobre sua base.

OBJETO DE SUPERFÍCIE CURVA - CILINDRO - apoiado sobre sua base.